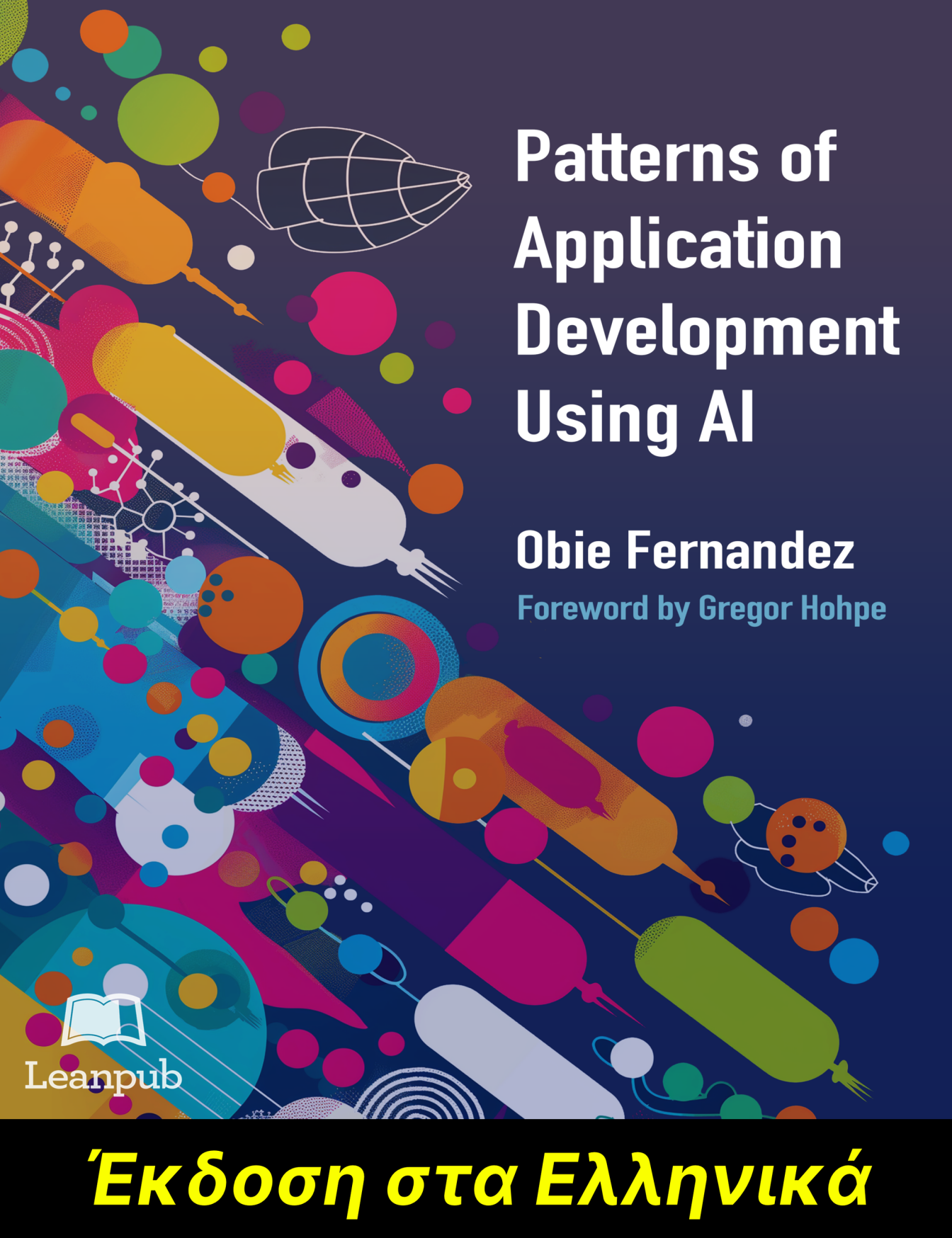




Patterns of Application Development Using AI

Obie Fernandez

Foreword by Gregor Hohpe



Leanpub

Έκδοση στα Ελληνικά

Πρότυπα Ανάπτυξης Εφαρμογών με ΤΝ (Έκδοση στα Ελληνικά)

Obie Fernandez

Αυτό το βιβλίο είναι προς πώληση στο

<http://leanpub.com/patterns-of-application-development-using-ai-el>

Αυτή η έκδοση δημοσιεύτηκε στις 2025-01-23



Αυτό είναι ένα βιβλίο του [Leanpub](#). Το Leanpub ενδυναμώνει τους συγγραφείς και τους εκδότες με τη διαδικασία Lean Publishing. Το [Lean Publishing](#) είναι η πράξη της δημοσίευσης ενός βιβλίου σε εξέλιξη χρησιμοποιώντας ελαφριά εργαλεία και πολλές επαναλήψεις για να λάβετε ανατροφοδότηση από τους αναγνώστες, να προσαρμοστείτε μέχρι να έχετε το σωστό βιβλίο και να δημιουργήσετε έλξη όταν το κάνετε.

© 2025 Obie Fernandez

Κάντε Tweet για αυτό το βιβλίο!

Παρακαλώ βοηθήστε τον/την Obie Fernandez διαδίδοντας το λόγο για αυτό το βιβλίο στο [Twitter](#)!

Το προτεινόμενο hashtag για αυτό το βιβλίο είναι [#poaduai](#).

Δείτε τι λένε άλλοι για το βιβλίο πατώντας σε αυτόν τον σύνδεσμο για να αναζητήσετε αυτό το hashtag στο Twitter:

[#poaduai](#)

Στην ατρόμητη βασίλισσά μου, τη μούσα μου, το φως και την αγάπη μου, Victoria

Επίσης Από Obie Fernandez

Patterns of Application Development Using AI

The Rails 8 Way

The Rails 7 Way

XML The Rails Way

Serverless

El Libro Principiante de Node

The Lean Enterprise

Περιεχόμενα

Πρόλογος από τον Gregor Hohpe	i
Πρόλογος	ii
Σχετικά με το Βιβλίο	iii
Σχετικά με τα Παραδείγματα Κώδικα	iii
Τι Δεν Καλύπτω	iii
Σε Ποιους Απευθύνεται αυτό το Βιβλίο	iii
Χτίζοντας ένα Κοινό Λεξιλόγιο	iii
Συμμετοχή	iii
Ευχαριστίες	iv
Τι γίνεται με τις εικονογραφήσεις;	iv
Σχετικά με τη Διτή Έκδοση	iv
Σχετικά με τον Συγγραφέα	v
Εισαγωγή	1
Σκέψεις για την Αρχιτεκτονική Λογισμικού	2
Τι είναι ένα Μοντέλο Μεγάλης Γλώσσας;	4
Κατανόηση της Εξαγωγής Συμπερασμάτων	6
Σκέψεις για την Απόδοση	30
Πειραματισμός με Διαφορετικά Μοντέλα ΜΓΜ	32
Σύνθετα Συστήματα Τεχνητής Νοημοσύνης	33

Μέρος 1: Θεμελιώδεις Προσεγγίσεις & Τεχνικές	43
Στένεψε το Μονοπάτι	44
Λανθάνων Χώρος: Ακατανόητα Απέραντος	47
Πώς “Στενεύει” η Διαδρομή	51
Ακατέργαστα Έναντι Μοντέλων Εκπαιδευμένων με Οδηγίες	55
Μηχανική Προτροπών	64
Απόσταξη Προτροπών	83
Τι γίνεται με τη μικροπροσαρμογή;	90
Παραγωγή με Ενισχυμένη Ανάκτηση (Retrieval Augmented Generation - RAG)	92
Τι είναι η Παραγωγή με Ενισχυμένη Ανάκτηση;	92
Πώς Λειτουργεί το RAG;	92
Γιατί να Χρησιμοποιήσετε το RAG στις Εφαρμογές σας;	92
Υλοποίηση του RAG στην Εφαρμογή σας	93
Κατάτμηση Προτάσεων	93
Παραδείγματα RAG στον Πραγματικό Κόσμο	94
Ευφυής Βελτιστοποίηση Ερωτημάτων (IQO)	94
Επαναταξινόμηση	94
Αξιολόγηση RAG (RAGAs)	95
Προκλήσεις και Μελλοντικές Προοπτικές	97
Πλήθος Εργατών	99
Εργάτες ΤΝ Ως Ανεξάρτητα Επαναχρησιμοποιήσιμα Στοιχεία	100
Διαχείριση Λογαριασμών	102
Εφαρμογές Ηλεκτρονικού Εμπορίου	103
Εφαρμογές στην Υγειονομική Περίθαλψη	113
Εργάτης ΑΙ ως Διαχειριστής Διεργασιών	117

ΠΕΡΙΕΧΟΜΕΝΑ

Ενσωμάτωση Εργατών ΤΝ στην Αρχιτεκτονική της Εφαρμογής σας	122
Συνθεσιμότητα και Ενορχήστρωση Εργατών ΑΙ	126
Συνδυάζοντας Παραδοσιακή ΕΦΓ με ΜΓΜ	136
Χρήση Εργαλείων	140
Τι είναι η Χρήση Εργαλείων;	140
Οι Δυνατότητες της Χρήσης Εργαλείων	142
Η Ροή Εργασίας της Χρήσης Εργαλείων	143
Βέλτιστες Πρακτικές για τη Χρήση Εργαλείων	159
Σύνθεση και Αλυσίδωση Εργαλείων	164
Μελλοντικές Κατευθύνσεις	166
Επεξεργασία Ροής	169
Υλοποίηση ενός ReplyStream	170
Ο “Βρόχος Συνομιλίας”	176
Αυτόματη Συνέχιση	178
Συμπέρασμα	181
Αυτοθεραπευόμενα Δεδομένα	183
Πρακτική Μελέτη Περίπτωσης: Διόρθωση Κατεστραμμένου JSON	186
Παράμετροι και Αντενδείξεις	191
Παραγωγή Περιεχομένου βάσει Περιβάλλοντος	209
Εξατομίκευση	210
Παραγωγικότητα	212
Ταχεία Επανάληψη και Πειραματισμός	215
Τοπικοποίηση με Τεχνητή Νοημοσύνη	218
Η Σημασία των Δοκιμών Χρηστών και της Ανατροφοδότησης	221
Παραγωγική Διεπαφή Χρήστη	222
Δημιουργία Κειμένου για Διεπαφές Χρήστη	224

Ορισμός της Γεννητικής Διεπαφής Χρήστη	233
Παράδειγμα	236
Η Μετάβαση στον Σχεδιασμό με Προσανατολισμό στο Αποτέλεσμα	239
Προκλήσεις και Ζητήματα προς Εξέταση	240
Μελλοντικές Προοπτικές και Ευκαιρίες	242
Ευφυής Ενορχήστρωση Ροών Εργασίας	246
Επιχειρηματική Ανάγκη	247
Βασικά Οφέλη	248
Βασικά Πρότυπα	249
Χειρισμός και Ανάκαμψη από Εξαιρέσεις	251
Υλοποίηση Έξυπνης Ενορχήστρωσης Ροής Εργασίας στην Πράξη	255
Παρακολούθηση και Καταγραφή	270
Ζητήματα Κλιμάκωσης και Απόδοσης	276
Έλεγχος και Επικύρωση Ροών Εργασίας	281

Μέρος 2: Τα Πρότυπα 290

Μηχανική Προτροπών	291
Αλυσίδα Σκέψης	292
Εναλλαγή Λειτουργίας	294
Ανάθεση Ρόλου	295
Αντικείμενο Προτροπής	296
Πρότυπο Προτροπής	297
Δομημένη Είσοδος/Έξοδος	298
Αλυσίδωση Προτροπών	299
Επανασυντάκτης Προτροπών	300
Οριοθέτηση Απόκρισης	301
Αναλυτής Ερωτημάτων	302
Επανασυντάκτης Ερωτημάτων	304

Ventriloquist	305
Διακριτά Συστατικά	306
Κατηγορήμα	307
Πρόσοψη API	308
Result Interpreter	311
Εικονική Μηχανή	312
Προδιαγραφή και Δοκιμές	312
Άνθρωπος στον Βρόχο (HITL)	314
Μοτίβα Υψηλού Επιπέδου	314
Κλιμάκωση	315
Βρόχος Ανατροφοδότησης	316
Παθητική Ακτινοβολία Πληροφοριών	317
Συνεργατική Λήψη Αποφάσεων (CDM)	319
Συνεχής Μάθηση	320
Ηθικά Ζητήματα	320
Τεχνολογικές Εξελίξεις και Μελλοντικές Προοπτικές	321
Ευφυής Χειρισμός Σφαλμάτων	322
Παραδοσιακές Προσεγγίσεις Χειρισμού Σφαλμάτων	322
Διάγνωση Σφαλμάτων Βάσει Περιβάλλοντος	323
Ευφυής Αναφορά Σφαλμάτων	324
Προληπτική Αποτροπή Σφαλμάτων	325
Έξυπνη Ανάκαμψη από Σφάλματα	325
Εξατομικευμένη Επικοινωνία Σφαλμάτων	326
Προσαρμοστική Ροή Εργασίας Διαχείρισης Σφαλμάτων	327
Έλεγχος Ποιότητας	328
Eval	329
Προστατευτικό Πλαίσιο	331

Προστατευτικά πλαίσια και Αξιολογήσεις: Δύο όψεις του ίδιου νομίσματος 332

Γλωσσάριο 333

Γλωσσάριο 333

Index 339

Πρόλογος από τον Gregor Hohpe

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πρόλογος

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Σχετικά με το Βιβλίο

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Σχετικά με τα Παραδείγματα Κώδικα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Τι Δεν Καλύπτω

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Σε Ποιους Απευθύνεται αυτό το Βιβλίο

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Χτίζοντας ένα Κοινό Λεξιλόγιο

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Συμμετοχή

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ευχαριστίες

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Τι γίνεται με τις εικονογραφήσεις;

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

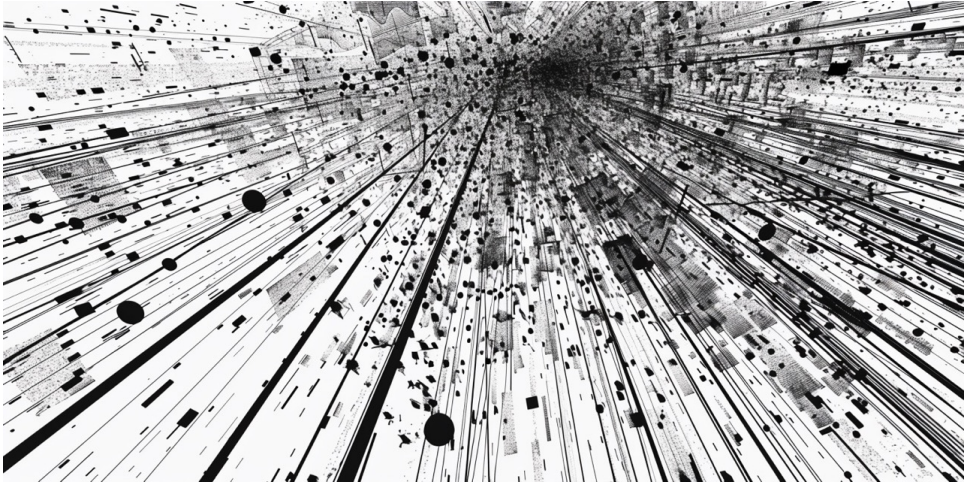
Σχετικά με τη Λιτή Έκδοση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Σχετικά με τον Συγγραφέα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Εισαγωγή



Αν ανυπομονείτε να ξεκινήσετε να ενσωματώνετε Μεγάλα Γλωσσικά Μοντέλα Τεχνητής Νοημοσύνης (MGΜ) στα προγραμματιστικά σας έργα, μπορείτε να προχωρήσετε κατευθείαν στα πρότυπα και τα παραδείγματα κώδικα που παρουσιάζονται στα επόμενα κεφάλαια. Ωστόσο, για να εκτιμήσετε πλήρως τη δύναμη και τις δυνατότητες αυτών των προτύπων, αξίζει να αφιερώσετε λίγο χρόνο για να κατανοήσετε το ευρύτερο πλαίσιο και τη συνεκτική προσέγγιση που αντιπροσωπεύουν.

Τα πρότυπα δεν είναι απλώς μια συλλογή μεμονωμένων τεχνικών, αλλά ένα ενοποιημένο πλαίσιο για την ενσωμάτωση της Τεχνητής Νοημοσύνης στις εφαρμογές σας. Χρησιμοποιώ Ruby on Rails, αλλά αυτά τα πρότυπα θα πρέπει να λειτουργούν σε σχεδόν οποιοδήποτε άλλο προγραμματιστικό περιβάλλον. Αντιμετωπίζουν ένα ευρύ φάσμα ζητημάτων, από τη διαχείριση δεδομένων και τη βελτιστοποίηση απόδοσης έως την εμπειρία χρήστη και την ασφάλεια, παρέχοντας

ένα ολοκληρωμένο σύνολο εργαλείων για την ενίσχυση των παραδοσιακών προγραμματιστικών πρακτικών με τις δυνατότητες της Τεχνητής Νοημοσύνης.

Κάθε κατηγορία προτύπων αντιμετωπίζει μια συγκεκριμένη πρόκληση ή ευκαιρία που προκύπτει κατά την ενσωμάτωση στοιχείων Τεχνητής Νοημοσύνης στην εφαρμογή σας. Κατανοώντας τις σχέσεις και τις συνέργειες μεταξύ αυτών των προτύπων, μπορείτε να λάβετε τεκμηριωμένες αποφάσεις σχετικά με το πού και πώς να εφαρμόσετε την Τεχνητή Νοημοσύνη πιο αποτελεσματικά.

Τα πρότυπα δεν είναι ποτέ περιοριστικές λύσεις και δεν πρέπει να αντιμετωπίζονται ως τέτοιες. Προορίζονται να είναι προσαρμόσιμα δομικά στοιχεία που πρέπει να προσαρμόζονται στις μοναδικές απαιτήσεις και περιορισμούς της δικής σας μοναδικής εφαρμογής. Η επιτυχής εφαρμογή αυτών των προτύπων (όπως και οποιωνδήποτε άλλων στον τομέα του λογισμικού) βασίζεται στη βαθιά κατανόηση του πεδίου του προβλήματος, των αναγκών των χρηστών και της συνολικής τεχνικής αρχιτεκτονικής του έργου σας.

Σκέψεις για την Αρχιτεκτονική Λογισμικού

Ξεκίνησα τον προγραμματισμό τη δεκαετία του 1980 και συμμετείχα στη σκηνή των χάκερ, και δεν έχασα ποτέ τη νοοτροπία του χάκερ, ακόμη και αφού έγινα επαγγελματίας προγραμματιστής λογισμικού. Από την αρχή, είχα πάντα μια υγιή σκεπτικιστική στάση σχετικά με την πραγματική αξία που πρόσφεραν οι αρχιτέκτονες λογισμικού από τους ελεφάντινους πύργους τους.

Ένας από τους λόγους που προσωπικά είμαι τόσο ενθουσιασμένος με τις αλλαγές που φέρνει αυτό το ισχυρό νέο κύμα τεχνολογίας Τεχνητής Νοημοσύνης είναι ο αντίκτυπός του σε αυτό που θεωρούμε αποφάσεις *αρχιτεκτονικής λογισμικού*. Αμφισβητεί τις παραδοσιακές αντιλήψεις για το τι συνιστά τον “σωστό” τρόπο σχεδιασμού και υλοποίησης των έργων λογισμικού μας. Επίσης, αμφισβητεί το κατά πόσο η αρχιτεκτονική μπορεί ακόμη να θεωρείται κυρίως ως *τα μέρη ενός συστήματος*

που είναι δύσκολο να αλλάξουν, καθώς η ενίσχυση με Τεχνητή Νοημοσύνη καθιστά ευκολότερο από ποτέ να αλλάξετε οποιοδήποτε μέρος του έργου σας, οποιαδήποτε στιγμή.

Ίσως εισερχόμαστε στα έτη ακμής της “μεταμοντέρνας” προσέγγισης στη μηχανική λογισμικού. Σε αυτό το πλαίσιο, ο όρος μεταμοντέρνος αναφέρεται σε μια θεμελιώδη μετατόπιση από τα παραδοσιακά παραδείγματα, όπου οι προγραμματιστές ήταν υπεύθυνοι για τη συγγραφή και συντήρηση κάθε γραμμής κώδικα. Αντ’ αυτού, υιοθετεί την ιδέα της ανάθεσης εργασιών, όπως ο χειρισμός δεδομένων, οι πολύπλοκοι αλγόριθμοι, ακόμη και ολόκληρα τμήματα της λογικής εφαρμογής, σε βιβλιοθήκες τρίτων και εξωτερικά APIs. Αυτή η μεταμοντέρνα μετατόπιση αντιπροσωπεύει μια σημαντική απομάκρυνση από τη συμβατική σοφία της δημιουργίας εφαρμογών από το μηδέν και προκαλεί τους προγραμματιστές να επανεξετάσουν το ρόλο τους στη διαδικασία ανάπτυξης.

Πάντα πίστευα ότι οι καλοί προγραμματιστές γράφουν μόνο τον κώδικα που είναι απολύτως απαραίτητο να γραφτεί, βασιζόμενοι στις διδαχές του Larry Wall και άλλων φωτεινών προσωπικοτήτων του χώρου σαν κι αυτόν. Ελαχιστοποιώντας την ποσότητα του γραμμένου κώδικα, μπορούμε να κινηθούμε γρηγορότερα, να μειώσουμε την επιφάνεια για σφάλματα, να απλοποιήσουμε τη συντήρηση και να βελτιώσουμε τη συνολική αξιοπιστία των εφαρμογών τους. Λιγότερος κώδικας μας επιτρέπει να εστιάσουμε στην κύρια επιχειρησιακή λογική και την εμπειρία χρήστη, ενώ αναθέτουμε άλλες εργασίες σε άλλες υπηρεσίες.

Τώρα που τα συστήματα που βασίζονται στην τεχνητή νοημοσύνη μπορούν να χειριστούν εργασίες που προηγουμένως ήταν αποκλειστικό πεδίο του ανθρώπινου κώδικα, θα πρέπει να μπορούμε να είμαστε ακόμη πιο παραγωγικοί και ευέλικτοι, με μεγαλύτερη εστίαση από ποτέ στη δημιουργία επιχειρηματικής αξίας και εμπειρίας χρήστη.

Φυσικά υπάρχουν συμβιβασμοί στην ανάθεση τεράστιων τμημάτων του έργου σας σε συστήματα τεχνητής νοημοσύνης, όπως η πιθανή απώλεια ελέγχου και η

ανάγκη για ισχυρούς μηχανισμούς παρακολούθησης και ανατροφοδότησης. Γι' αυτό απαιτείται ένα νέο σύνολο δεξιοτήτων και γνώσεων, συμπεριλαμβανομένης τουλάχιστον κάποιας θεμελιώδους κατανόησης του τρόπου λειτουργίας της τεχνητής νοημοσύνης.

Τι είναι ένα Μοντέλο Μεγάλης Γλώσσας;

Τα Μοντέλα Μεγάλης Γλώσσας (ΜΜΓ) είναι ένας τύπος μοντέλου τεχνητής νοημοσύνης που έχει προσελκύσει σημαντική προσοχή τα τελευταία χρόνια, από την κυκλοφορία του GPT-3 από την OpenAI το 2020. Τα ΜΜΓ είναι σχεδιασμένα να επεξεργάζονται, να κατανοούν και να παράγουν ανθρώπινη γλώσσα με αξιοσημείωτη ακρίβεια και ευφράδεια. Σε αυτή την ενότητα, θα ρίξουμε μια σύντομη ματιά στο πώς λειτουργούν τα ΜΜΓ και γιατί είναι κατάλληλα για την κατασκευή ευφυών συστατικών συστήματος.

Στον πυρήνα τους, τα ΜΜΓ βασίζονται σε αλγορίθμους βαθιάς μάθησης, συγκεκριμένα νευρωνικά δίκτυα. Αυτά τα δίκτυα αποτελούνται από διασυνδεδεμένους κόμβους, ή νευρώνες, που επεξεργάζονται και μεταδίδουν πληροφορίες. Η αρχιτεκτονική επιλογής για τα ΜΜΓ είναι συχνά το μοντέλο Transformer, το οποίο έχει αποδειχθεί εξαιρετικά αποτελεσματικό στο χειρισμό διαδοχικών δεδομένων όπως το κείμενο.

Τα μοντέλα Transformer βασίζονται στον μηχανισμό προσοχής και χρησιμοποιούνται κυρίως για εργασίες που περιλαμβάνουν ακολουθιακά δεδομένα, όπως η επεξεργασία φυσικής γλώσσας. Οι Transformers επεξεργάζονται τα δεδομένα εισόδου όλα μαζί αντί για ακολουθιακά, γεγονός που τους επιτρέπει να συλλαμβάνουν πιο αποτελεσματικά τις μακροπρόθεσμες εξαρτήσεις. Διαθέτουν επίπεδα μηχανισμών προσοχής που βοηθούν το μοντέλο να εστιάσει σε διαφορετικά μέρη των δεδομένων εισόδου για την κατανόηση του πλαισίου και των σχέσεων.

Η διαδικασία εκπαίδευσης για τα LLMs περιλαμβάνει την έκθεση του μοντέλου σε τεράστιες ποσότητες κειμενικών δεδομένων, όπως βιβλία, άρθρα, ιστοσελίδες

και αποθετήρια κώδικα. Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο μαθαίνει να αναγνωρίζει μοτίβα, σχέσεις και δομές μέσα στο κείμενο. Συλλαμβάνει τις στατιστικές ιδιότητες της γλώσσας, όπως γραμματικούς κανόνες, συσχετίσεις λέξεων και συναφή νοήματα.

Μία από τις βασικές τεχνικές που χρησιμοποιούνται στην εκπαίδευση των LLMs είναι η μη επιβλεπόμενη μάθηση. Αυτό σημαίνει ότι το μοντέλο μαθαίνει από τα δεδομένα χωρίς ρητή επισήμανση ή καθοδήγηση. Ανακαλύπτει μοτίβα και αναπαραστάσεις μόνο του αναλύοντας τη συνύπαρξη λέξεων και φράσεων στα δεδομένα εκπαίδευσης. Αυτό επιτρέπει στα LLMs να αναπτύξουν μια βαθιά κατανόηση της γλώσσας και των ιδιαιτεροτήτων της.

Μια άλλη σημαντική πτυχή των LLMs είναι η ικανότητά τους να χειρίζονται το πλαίσιο. Κατά την επεξεργασία ενός κειμένου, τα LLMs λαμβάνουν υπόψη όχι μόνο τις μεμονωμένες λέξεις αλλά και το περιβάλλον πλαίσιο. Λαμβάνουν υπόψη τις προηγούμενες λέξεις, προτάσεις, ακόμη και παραγράφους για να κατανοήσουν το νόημα και την πρόθεση του κειμένου. Αυτή η συναφειακή κατανόηση επιτρέπει στα LLMs να δημιουργούν συνεκτικές και σχετικές απαντήσεις. Ένας από τους κύριους τρόπους με τους οποίους αξιολογούμε τις δυνατότητες ενός συγκεκριμένου μοντέλου LLM είναι εξετάζοντας το μέγεθος του πλαισίου που μπορούν να λάβουν υπόψη για να δημιουργήσουν απαντήσεις.

Μετά την εκπαίδευση, τα LLMs μπορούν να χρησιμοποιηθούν για ένα ευρύ φάσμα γλωσσικών εργασιών. Μπορούν να δημιουργήσουν κείμενο που μοιάζει με ανθρώπινο, να απαντήσουν σε ερωτήσεις, να συνοψίσουν έγγραφα, να μεταφράσουν γλώσσες, ακόμη και να γράψουν κώδικα. Η ευελιξία των LLMs τα καθιστά πολύτιμα για την κατασκευή έξυπνων συστημάτων που μπορούν να αλληλεπιδρούν με χρήστες, να επεξεργάζονται και να αναλύουν δεδομένα κειμένου και να παράγουν ουσιαστικά αποτελέσματα.

Ενσωματώνοντας LLMs στην αρχιτεκτονική εφαρμογών, μπορείτε να δημιουργήσετε στοιχεία τεχνητής νοημοσύνης που κατανοούν και επεξεργάζονται την είσοδο του

χρήστη, δημιουργούν δυναμικό περιεχόμενο και παρέχουν έξυπνες συστάσεις ή ενέργειες. Αλλά η εργασία με LLMs απαιτεί προσεκτική εξέταση των απαιτήσεων σε πόρους και των συμβιβασμών απόδοσης. Τα LLMs απαιτούν έντονη υπολογιστική ισχύ και μπορεί να χρειάζονται σημαντική επεξεργαστική ισχύ και μνήμη (με άλλα λόγια, χρήματα) για να λειτουργήσουν. Οι περισσότεροι από εμάς θα χρειαστεί να αξιολογήσουμε τις οικονομικές επιπτώσεις της ενσωμάτωσης LLMs στις εφαρμογές μας και να ενεργήσουμε ανάλογα.

Κατανόηση της Εξαγωγής Συμπερασμάτων

Η εξαγωγή συμπερασμάτων αναφέρεται στη διαδικασία με την οποία ένα μοντέλο παράγει προβλέψεις ή αποτελέσματα βάσει νέων, άγνωστων δεδομένων. Είναι η φάση όπου το εκπαιδευμένο μοντέλο χρησιμοποιείται για να λάβει αποφάσεις ή να παράγει κείμενο, εικόνες ή άλλο περιεχόμενο ως απάντηση στις εισόδους των χρηστών.

Κατά τη φάση της εκπαίδευσης, ένα μοντέλο TN μαθαίνει από ένα μεγάλο σύνολο δεδομένων προσαρμόζοντας τις παραμέτρους του για να ελαχιστοποιήσει το σφάλμα στις προβλέψεις του. Μόλις εκπαιδευτεί, το μοντέλο μπορεί να εφαρμόσει αυτά που έχει μάθει σε νέα δεδομένα. Η εξαγωγή συμπερασμάτων είναι ο τρόπος με τον οποίο το μοντέλο χρησιμοποιεί τα μοτίβα και τη γνώση που έχει μάθει για να παράγει αποτελέσματα.

Για τα Μεγάλα Γλωσσικά Μοντέλα (LLMs), η εξαγωγή συμπερασμάτων περιλαμβάνει τη λήψη μιας προτροπής ή κειμένου εισόδου και την παραγωγή μιας συνεκτικής και συναφούς απάντησης, ως ροή *λεκτικών μονάδων* (για τις οποίες θα μιλήσουμε σύντομα). Αυτό θα μπορούσε να είναι η απάντηση σε μια ερώτηση, η συμπλήρωση μιας πρότασης, η δημιουργία μιας ιστορίας ή η μετάφραση κειμένου, μεταξύ πολλών άλλων εργασιών.



Σε αντίθεση με τον τρόπο που σκεφτόμαστε εσείς κι εγώ, η “σκέψη” ενός μοντέλου TN μέσω της εξαγωγής συμπερασμάτων συμβαίνει σε μία μόνο λειτουργία χωρίς κατάσταση. Δηλαδή, η σκέψη του περιορίζεται στη διαδικασία παραγωγής. Κυριολεκτικά πρέπει να σκέφτεται φωναχτά, σαν να σας έκανα μια ερώτηση και δεχόμουν απάντηση μόνο με τη μορφή “ροής συνείδησης”.

Τα Μεγάλα Γλωσσικά Μοντέλα Έρχονται σε Πολλά Μεγέθη και Παραλλαγές

Ενώ πρακτικά όλα τα δημοφιλή μεγάλα γλωσσικά μοντέλα (LLMs) βασίζονται στην ίδια βασική αρχιτεκτονική transformer και εκπαιδεύονται σε τεράστια σύνολα δεδομένων κειμένου, έρχονται σε διάφορα μεγέθη και είναι λεπτομερώς ρυθμισμένα για διαφορετικούς σκοπούς. Το μέγεθος ενός LLM, που μετράται από τον αριθμό των παραμέτρων στο νευρωνικό του δίκτυο, έχει μεγάλο αντίκτυπο στις δυνατότητές του. Τα μεγαλύτερα μοντέλα με περισσότερες παραμέτρους, όπως το GPT-4, το οποίο φημολογείται ότι διαθέτει 1 έως 2 τρισεκατομμύρια παραμέτρους, είναι γενικά πιο γνώστες και ικανά από τα μικρότερα μοντέλα. Ωστόσο, τα μεγαλύτερα μοντέλα απαιτούν επίσης πολύ περισσότερη υπολογιστική ισχύ για να λειτουργήσουν, κάτι που μεταφράζεται σε υψηλότερο κόστος όταν τα χρησιμοποιείτε μέσω κλήσεων API.

Για να γίνουν τα LLMs πιο πρακτικά και προσαρμοσμένα σε συγκεκριμένες περιπτώσεις χρήσης, τα βασικά μοντέλα συχνά ρυθμίζονται λεπτομερώς σε πιο στοχευμένα σύνολα δεδομένων. Για παράδειγμα, ένα LLM μπορεί να εκπαιδευτεί σε ένα μεγάλο σώμα διαλόγων για να εξειδικευτεί στη συνομιλητική TN. Άλλα εκπαιδεύονται σε κώδικα για να τους προσδώσουν γνώσεις προγραμματισμού. Υπάρχουν ακόμη και μοντέλα που είναι ειδικά εκπαιδευμένα για αλληλεπιδράσεις τύπου παιχνιδιού ρόλων με τους χρήστες!

Μοντέλα Ανάκτησης έναντι Παραγωγικών Μοντέλων

Στον κόσμο των μεγάλων γλωσσικών μοντέλων (LLMs), υπάρχουν δύο κύριες προσεγγίσεις για τη δημιουργία απαντήσεων: τα μοντέλα που βασίζονται στην ανάκτηση και τα παραγωγικά μοντέλα. Κάθε προσέγγιση έχει τα δικά της πλεονεκτήματα και μειονεκτήματα, και η κατανόηση των διαφορών μεταξύ τους μπορεί να σας βοηθήσει να επιλέξετε το κατάλληλο μοντέλο για τη συγκεκριμένη περίπτωση χρήσης σας.

Μοντέλα Βασισμένα στην Ανάκτηση

Τα μοντέλα που βασίζονται στην ανάκτηση, γνωστά και ως μοντέλα ανάκτησης πληροφοριών, δημιουργούν απαντήσεις αναζητώντας σε μια μεγάλη βάση δεδομένων προϋπάρχοντος κειμένου και επιλέγοντας τα πιο σχετικά αποσπάσματα με βάση το ερώτημα εισόδου. Αυτά τα μοντέλα δεν δημιουργούν νέο κείμενο από το μηδέν, αλλά συνθέτουν αποσπάσματα από τη βάση δεδομένων για να σχηματίσουν μια συνεκτική απάντηση.

Ένα από τα κύρια πλεονεκτήματα των μοντέλων που βασίζονται στην ανάκτηση είναι η ικανότητά τους να παρέχουν πραγματολογικά ακριβείς και ενημερωμένες πληροφορίες. Καθώς βασίζονται σε μια βάση δεδομένων επιμελημένου κειμένου, μπορούν να αντλούν σχετικές πληροφορίες από αξιόπιστες πηγές και να τις παρουσιάζουν στον χρήστη. Αυτό τα καθιστά κατάλληλα για εφαρμογές που απαιτούν ακριβείς, πραγματολογικές απαντήσεις, όπως συστήματα ερωταποκρίσεων ή βάσεις γνώσεων.

Ωστόσο, τα μοντέλα που βασίζονται στην ανάκτηση έχουν ορισμένους περιορισμούς. Είναι τόσο καλά όσο η βάση δεδομένων στην οποία πραγματοποιούν αναζήτηση, οπότε η ποιότητα και η κάλυψη της βάσης δεδομένων επηρεάζουν άμεσα την απόδοση του μοντέλου. Επιπλέον, αυτά τα μοντέλα μπορεί να δυσκολεύονται στη δημιουργία συνεκτικών και φυσικών απαντήσεων, καθώς περιορίζονται στο διαθέσιμο κείμενο της βάσης δεδομένων.

Δεν καλύπτουμε τη χρήση αμιγώς μοντέλων ανάκτησης σε αυτό το βιβλίο.

Παραγωγικά Μοντέλα

Τα παραγωγικά μοντέλα, από την άλλη πλευρά, δημιουργούν νέο κείμενο από το μηδέν με βάση τα πρότυπα και τις σχέσεις που έμαθαν κατά τη διάρκεια της εκπαίδευσης. Αυτά τα μοντέλα χρησιμοποιούν την κατανόησή τους για τη γλώσσα για να δημιουργήσουν πρωτότυπες απαντήσεις που είναι προσαρμοσμένες στο αρχικό ερώτημα.

Το κύριο πλεονέκτημα των παραγωγικών μοντέλων είναι η ικανότητά τους να παράγουν δημιουργικό, συνεκτικό και συναφές με το πλαίσιο κείμενο. Μπορούν να συμμετέχουν σε ανοιχτές συζητήσεις, να δημιουργούν ιστορίες, ακόμη και να γράφουν κώδικα. Αυτό τα καθιστά ιδανικά για εφαρμογές που απαιτούν πιο ανοιχτές και δυναμικές αλληλεπιδράσεις, όπως chatbots, δημιουργία περιεχομένου και βοηθοί δημιουργικής γραφής.

Ωστόσο, τα παραγωγικά μοντέλα μπορεί μερικές φορές να παράγουν ασυνεπείς ή λανθασμένες πληροφορίες, καθώς βασίζονται στα πρότυπα που έμαθαν κατά την εκπαίδευση και όχι σε μια επιμελημένη βάση δεδομένων γεγονότων. Μπορεί επίσης να είναι πιο επιρρεπή σε προκαταλήψεις και ψευδαισθήσεις, παράγοντας κείμενο που είναι πιθανοφανές αλλά όχι απαραίτητα αληθές.

Παραδείγματα γεννητικών ΜΜΓ περιλαμβάνουν τη σειρά GPT της OpenAI (GPT-3, GPT-4) και το Claude της Anthropic.

Υβριδικά Μοντέλα

Αρκετά εμπορικά διαθέσιμα ΜΜΓ συνδυάζουν τόσο προσεγγίσεις ανάκτησης όσο και γέννησης σε ένα υβριδικό μοντέλο. Αυτά τα μοντέλα χρησιμοποιούν τεχνικές ανάκτησης για να βρουν σχετικές πληροφορίες από μια βάση δεδομένων και στη συνέχεια χρησιμοποιούν τεχνικές γέννησης για να συνθέσουν αυτές τις πληροφορίες σε μια συνεκτική απάντηση.

Τα υβριδικά μοντέλα στοχεύουν στο να συνδυάσουν την πραγματολογική ακρίβεια των μοντέλων βασισμένων στην ανάκτηση με τις δυνατότητες παραγωγής φυσικής γλώσσας των γεννητικών μοντέλων. Μπορούν να παρέχουν πιο αξιόπιστες και ενημερωμένες πληροφορίες, διατηρώντας παράλληλα την ικανότητα να συμμετέχουν σε ανοιχτές συζητήσεις.

Κατά την επιλογή μεταξύ μοντέλων βασισμένων στην ανάκτηση και γεννητικών μοντέλων, θα πρέπει να λάβετε υπόψη τις συγκεκριμένες απαιτήσεις της εφαρμογής σας. Εάν ο κύριος στόχος είναι η παροχή ακριβών, πραγματολογικών πληροφοριών, ένα μοντέλο βασισμένο στην ανάκτηση μπορεί να είναι η καλύτερη επιλογή. Εάν η εφαρμογή απαιτεί πιο ανοιχτές και δημιουργικές αλληλεπιδράσεις, ένα γεννητικό μοντέλο μπορεί να είναι πιο κατάλληλο. Τα υβριδικά μοντέλα προσφέρουν μια ισορροπία μεταξύ των δύο προσεγγίσεων και μπορεί να αποτελούν καλή επιλογή για εφαρμογές που απαιτούν τόσο πραγματολογική ακρίβεια όσο και παραγωγή φυσικής γλώσσας.

Τελικά, η επιλογή μεταξύ μοντέλων βασισμένων στην ανάκτηση και γεννητικών μοντέλων εξαρτάται από τη συγκεκριμένη περίπτωση χρήσης και τους συμβιβασμούς μεταξύ ακρίβειας, δημιουργικότητας και ευελιξίας. Κατανοώντας τα πλεονεκτήματα και τους περιορισμούς κάθε προσέγγισης, μπορείτε να λάβετε τεκμηριωμένες αποφάσεις.

Συντονισμός Οδηγιών

Ο συντονισμός οδηγιών είναι ένα υποσύνολο της συνομιλιακής εκπαίδευσης που χρησιμοποιεί ανθρώπινα γραμμένα prompts και απαντήσεις για να κάνει τα ΜΜΓ καλύτερα στην κατανόηση και την ακολούθηση οδηγιών. Εδώ είναι μερικά απλά παραδείγματα που δείχνουν πώς ο συντονισμός οδηγιών σε ανθρώπινα γραμμένα prompts και απαντήσεις κάνει τα ΜΜΓ καλύτερα στην κατανόηση και την ακολούθηση οδηγιών:

Παράδειγμα 1: Υποστήριξη Πελατών

Οδηγία: “Δώστε μια απάντηση σε έναν πελάτη που ζητά επιστροφή χρημάτων.”

Ανθρώπινο Prompt: “Θα ήθελα να ζητήσω επιστροφή χρημάτων για την πρόσφατη αγορά μου. Το προϊόν δεν ανταποκρίθηκε στις προσδοκίες μου.”

Ανθρώπινη Απάντηση: “Αυπάμαι που ακούω ότι το προϊόν δεν ανταποκρίθηκε στις προσδοκίες σας. Για να επεξεργαστούμε την επιστροφή χρημάτων σας, παρακαλώ παρέχετε τον αριθμό παραγγελίας σας και τον λόγο επιστροφής. Θα εξετάσουμε το αίτημά σας και θα επικοινωνήσουμε μαζί σας εντός 2-3 εργάσιμων ημερών.”

Παράδειγμα 2: Εκπαιδευτική Βοήθεια

Οδηγία: “Εξηγήστε την έννοια της φωτοσύνθεσης σε ένα παιδί 10 ετών.”

Ανθρώπινη Ερώτηση: “Μπορείς να εξηγήσεις πώς φτιάχνουν την τροφή τους τα φυτά;”

Ανθρώπινη Απάντηση: “Βεβαίως! Τα φυτά φτιάχνουν την τροφή τους μέσω μιας διαδικασίας που ονομάζεται φωτοσύνθεση. Χρησιμοποιούν το ηλιακό φως, το νερό και το διοξείδιο του άνθρακα για να δημιουργήσουν τροφή και οξυγόνο. Τα φύλλα του φυτού έχουν ένα ειδικό συστατικό που ονομάζεται χλωροφύλλη που

τα βοηθά να δεσμεύσουν το ηλιακό φως.”

Παράδειγμα 3: Προσωπικός Βοηθός

Οδηγία: “Προγραμματίσε μια συνάντηση για την επόμενη Τρίτη στις 3 μ.μ.”

Ανθρώπινη Ερώτηση: “Μπορείς να κανονίσεις μια συνάντηση για την Τρίτη στις 3 μ.μ.?”

Ανθρώπινη Απάντηση: “Βεβαίως, έχω προγραμματίσει μια συνάντηση για την επόμενη Τρίτη στις 3 μ.μ. Χρειάζεστε κάτι άλλο;”

Το αποτέλεσμα είναι ένα ποικίλο οικοσύστημα LLMs σε διαφορετικά μεγέθη και με διαφορετικές εξειδικεύσεις. Τα μικρότερα μοντέλα στο εύρος 1-7 δισεκατομμυρίων παραμέτρων παρέχουν καλές γενικές γλωσσικές δυνατότητες ενώ είναι πιο αποδοτικά στη λειτουργία τους.

- Mistral 7B
- Llama 3 8B
- Gemma 7B

Τα μεσαίου μεγέθους μοντέλα γύρω στις 30-70 δισεκατομμύρια παραμέτρους προσφέρουν ισχυρότερες ικανότητες συλλογισμού και ακολούθησης οδηγιών.

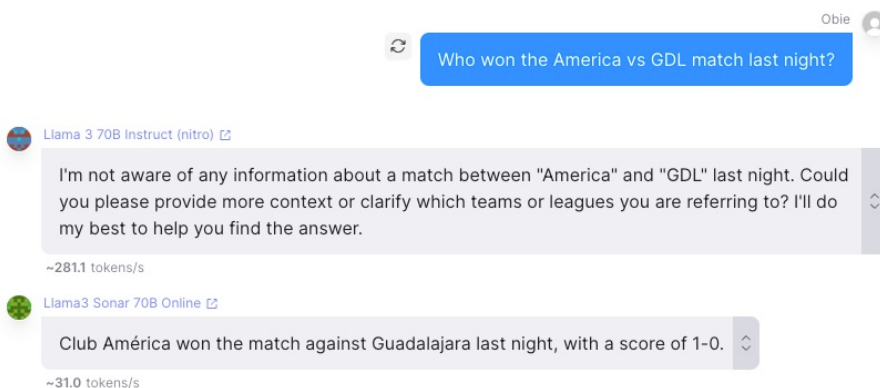
- Llama 3 70B
- Qwen2 70B
- Mixtral 8x22B

Όταν επιλέγετε ένα LLM για ενσωμάτωση σε μια εφαρμογή, πρέπει να εξισορροπήσετε τις δυνατότητες του μοντέλου με πρακτικούς παράγοντες όπως το κόστος, η καθυστέρηση, το μήκος περιεχομένου και το φιλτράρισμα περιεχομένου.

Τα μικρότερα μοντέλα, εκπαιδευμένα με οδηγίες, είναι συχνά η καλύτερη επιλογή για απλούστερες γλωσσικές εργασίες, ενώ τα μεγαλύτερα μοντέλα μπορεί να χρειάζονται για πολύπλοκο συλλογισμό ή ανάλυση. Τα δεδομένα εκπαίδευσης του μοντέλου είναι επίσης ένας σημαντικός παράγοντας, καθώς καθορίζουν την ημερομηνία διακοπής γνώσεων του μοντέλου.



Ορισμένα μοντέλα, όπως μερικά από την Perplexity είναι συνδεδεμένα με πηγές πληροφοριών σε πραγματικό χρόνο, έτσι ώστε ουσιαστικά να μην έχουν ημερομηνία διακοπής. Όταν τους κάνετε ερωτήσεις, μπορούν να αποφασίσουν ανεξάρτητα να κάνουν αναζητήσεις στο διαδίκτυο και να ανακτήσουν τυχαίες ιστοσελίδες για να δημιουργήσουν μια απάντηση.

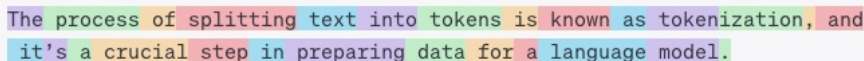


Σχήμα 1. Το Llama3 με και χωρίς διαδικτυακή πρόσβαση

Τελικά, δεν υπάρχει ένα μοντέλο LLM που να ταιριάζει σε όλες τις περιπτώσεις. Η κατανόηση των διαφορών στο μέγεθος του μοντέλου, την αρχιτεκτονική και την εκπαίδευση είναι το κλειδί για την επιλογή του κατάλληλου μοντέλου για κάθε περίπτωση χρήσης. Ο πειραματισμός με διαφορετικά μοντέλα είναι ο μόνος πρακτικός τρόπος για να αποκαλυφθεί ποια προσφέρουν την καλύτερη απόδοση για την εκάστοτε εργασία.

Κατάτμηση σε σύμβολα: Διαχωρίζοντας το Κείμενο σε Κομμάτια

Πριν ένα μεγάλο γλωσσικό μοντέλο μπορέσει να επεξεργαστεί κείμενο, το κείμενο αυτό πρέπει να διασπαστεί σε μικρότερες μονάδες που ονομάζονται *σύμβολα*. Τα σύμβολα μπορεί να είναι μεμονωμένες λέξεις, τμήματα λέξεων, ή ακόμη και μεμονωμένοι χαρακτήρες. Η διαδικασία διαχωρισμού του κειμένου σε σύμβολα είναι γνωστή ως κατάτμηση σε σύμβολα, και αποτελεί ένα κρίσιμο βήμα στην προετοιμασία των δεδομένων για ένα γλωσσικό μοντέλο.



The process of splitting text into tokens is known as tokenization, and it's a crucial step in preparing data for a language model.

Σχήμα 2. Αυτή η πρόταση περιέχει 27 σύμβολα

Διαφορετικά LLMs χρησιμοποιούν διαφορετικές στρατηγικές κατάτμησης σε σύμβολα, οι οποίες μπορούν να έχουν σημαντικό αντίκτυπο στην απόδοση και τις δυνατότητες του μοντέλου. Μερικοί συνηθισμένοι τύποι κατάτμησης που χρησιμοποιούνται από τα LLMs περιλαμβάνουν:

- **GPT (Κωδικοποίηση Ζευγών Byte):** Οι κατατμητές GPT χρησιμοποιούν μια τεχνική που ονομάζεται κωδικοποίηση ζευγών byte (BPE) για να διασπάσουν το κείμενο σε υπολεκτικές μονάδες. Η BPE συγχωνεύει επαναληπτικά τα πιο συχνά ζεύγη bytes σε ένα σώμα κειμένου, σχηματίζοντας ένα λεξιλόγιο από υπολεκτικά σύμβολα. Αυτό επιτρέπει στον κατατμητή να χειρίζεται σπάνιες και νέες λέξεις διασπώντας τις σε πιο συνηθισμένα υπολεκτικά τμήματα. Οι κατατμητές GPT χρησιμοποιούνται από μοντέλα όπως το GPT-3 και το GPT-4.
- **Llama (SentencePiece):** Οι κατατμητές Llama χρησιμοποιούν τη βιβλιοθήκη SentencePiece, η οποία είναι ένας μη εποπτευόμενος κατατμητής και επανασυνθέτης κειμένου. Το SentencePiece αντιμετωπίζει το κείμενο εισόδου

ως μια ακολουθία χαρακτήρων Unicode και μαθαίνει ένα υπολεκτικό λεξιλόγιο βασισμένο σε ένα σώμα εκπαίδευσης. Μπορεί να χειριστεί οποιαδήποτε γλώσσα που μπορεί να κωδικοποιηθεί σε Unicode, καθιστώντας το κατάλληλο για πολυγλωσσικά μοντέλα. Οι κατατμητές Llama χρησιμοποιούνται από μοντέλα όπως το Llama της Meta και το Alpaca.

- **SentencePiece (Unigram):** Οι τοκενοποιητές SentencePiece μπορούν επίσης να χρησιμοποιήσουν έναν διαφορετικό αλγόριθμο που ονομάζεται Unigram, ο οποίος βασίζεται σε μια τεχνική κανονικοποίησης υπολέξεων. Η τοκενοποίηση Unigram καθορίζει το βέλτιστο λεξιλόγιο υπολέξεων βάσει ενός μονογραμματικού γλωσσικού μοντέλου, το οποίο αναθέτει πιθανότητες σε μεμονωμένες υπολεκτικές μονάδες. Αυτή η προσέγγιση μπορεί να παράγει πιο σημασιολογικά ουσιαστικές υπολέξεις σε σύγκριση με το BPE. Το SentencePiece με Unigram χρησιμοποιείται από μοντέλα όπως το T5 της Google και το BERT.
- **Google Gemini (Πολυτροπική Τοκενοποίηση):** Το Google Gemini χρησιμοποιεί ένα σχήμα τοκενοποίησης σχεδιασμένο να χειρίζεται διάφορους τύπους δεδομένων, συμπεριλαμβανομένων κειμένου, εικόνων, ήχου, βίντεο και κώδικα. Αυτή η πολυτροπική ικανότητα επιτρέπει στο Gemini να επεξεργάζεται και να ενσωματώνει διαφορετικές μορφές πληροφοριών. Αξιοσημείωτα, το Google Gemini 1.5 Pro έχει ένα παράθυρο περιεχομένου που μπορεί να χειριστεί εκατομμύρια tokens, πολύ μεγαλύτερο από προηγούμενα μοντέλα. Αυτό το εκτεταμένο παράθυρο περιεχομένου επιτρέπει στο μοντέλο να επεξεργάζεται μεγαλύτερο πλαίσιο, οδηγώντας δυνητικά σε πιο ακριβείς απαντήσεις. Ωστόσο, είναι σημαντικό να σημειωθεί ότι το σχήμα τοκενοποίησης του Gemini είναι πολύ πιο κοντά στο ένα token ανά χαρακτήρα σε σύγκριση με άλλα μοντέλα. Αυτό σημαίνει ότι το πραγματικό κόστος χρήσης των μοντέλων Gemini μπορεί να είναι σημαντικά υψηλότερο από το αναμενόμενο εάν είστε εξοικειωμένοι με τη χρήση μοντέλων όπως το

GPT, καθώς η τιμολόγηση της Google βασίζεται σε χαρακτήρες και όχι σε tokens.

Η επιλογή του τοκενοποιητή επηρεάζει διάφορες πτυχές ενός LLM, συμπεριλαμβανομένων:

- **Μέγεθος λεξιλογίου:** Ο τοκενοποιητής καθορίζει το μέγεθος του λεξιλογίου του μοντέλου, το οποίο είναι το σύνολο των μοναδικών tokens που αναγνωρίζει. Ένα μεγαλύτερο, πιο λεπτομερές λεξιλόγιο μπορεί να βοηθήσει το μοντέλο να χειριστεί ένα ευρύτερο φάσμα λέξεων και φράσεων και ακόμη να γίνει πολυτροπικό (ικανό να κατανοεί και να παράγει περισσότερα από απλό κείμενο), αλλά αυξάνει επίσης τις απαιτήσεις μνήμης και την υπολογιστική πολυπλοκότητα του μοντέλου.
- **Χειρισμός σπάνιων και άγνωστων λέξεων:** Οι τοκενοποιητές που χρησιμοποιούν υπολεκτικές μονάδες, όπως το BPE και το SentencePiece, μπορούν να διασπάσουν σπάνιες και άγνωστες λέξεις σε πιο συνηθισμένα κομμάτια υπολέξεων. Αυτό επιτρέπει στο μοντέλο να κάνει εκπαιδευμένες εικασίες σχετικά με τη σημασία λέξεων που δεν έχει δει πριν, βασιζόμενο στις υπολέξεις που περιέχουν.
- **Πολυγλωσσική υποστήριξη:** Τοκενοποιητές όπως το SentencePiece, που μπορούν να χειριστούν οποιαδήποτε γλώσσα κωδικοποιήσιμη σε Unicode, είναι κατάλληλοι για πολυγλωσσικά μοντέλα που χρειάζεται να επεξεργάζονται κείμενο σε πολλαπλές γλώσσες.

Κατά την επιλογή ενός MMΓ για μια συγκεκριμένη εφαρμογή, είναι σημαντικό να εξετάσουμε τον διαχωριστή λεκτικών μονάδων που χρησιμοποιεί και πόσο καλά ευθυγραμμίζεται με τις συγκεκριμένες ανάγκες επεξεργασίας γλώσσας της εργασίας. Ο διαχωριστής λεκτικών μονάδων μπορεί να έχει σημαντική επίδραση στην ικανότητα του μοντέλου να χειρίζεται εξειδικευμένη ορολογία, σπάνιες λέξεις και πολυγλωσσικό κείμενο.

Μέγεθος Περιβάλλοντος: Πόσες Πληροφορίες Μπορεί να Χρησιμοποιήσει ένα Μοντέλο Γλώσσας Κατά την Εξαγωγή Συμπερασμάτων;

Όταν συζητάμε για μοντέλα γλώσσας, το μέγεθος περιβάλλοντος αναφέρεται στην ποσότητα κειμένου που μπορεί να εξετάσει ένα μοντέλο κατά την επεξεργασία ή τη δημιουργία των απαντήσεών του. Είναι ουσιαστικά ένα μέτρο του πόσες πληροφορίες μπορεί το μοντέλο να “θυμάται” και να χρησιμοποιεί για να διαμορφώσει τις εξόδους του (εκφρασμένο σε λεκτικές μονάδες). Το μέγεθος περιβάλλοντος ενός μοντέλου γλώσσας μπορεί να έχει σημαντική επίδραση στις δυνατότητές του και στους τύπους εργασιών που μπορεί να εκτελέσει αποτελεσματικά.

Τι είναι το Μέγεθος Περιβάλλοντος;

Με τεχνικούς όρους, το μέγεθος περιβάλλοντος καθορίζεται από τον αριθμό των λεκτικών μονάδων (λέξεων ή τμημάτων λέξεων) που μπορεί να επεξεργαστεί ένα μοντέλο γλώσσας σε μία ακολουθία εισόδου. Αυτό συχνά αναφέρεται ως το “εύρος προσοχής” ή “παράθυρο περιβάλλοντος” του μοντέλου. Όσο μεγαλύτερο είναι το μέγεθος περιβάλλοντος, τόσο περισσότερο κείμενο μπορεί να εξετάσει το μοντέλο ταυτόχρονα κατά τη δημιουργία μιας απάντησης ή την εκτέλεση μιας εργασίας.

Διαφορετικά μοντέλα γλώσσας έχουν διαφορετικά μεγέθη περιβάλλοντος, που κυμαίνονται από μερικές εκατοντάδες λεκτικές μονάδες έως εκατομμύρια λεκτικές μονάδες. Για αναφορά, μια τυπική παράγραφος κειμένου μπορεί να περιέχει περίπου 100-150 λεκτικές μονάδες, ενώ ένα ολόκληρο βιβλίο θα μπορούσε να περιέχει δεκάδες ή εκατοντάδες χιλιάδες λεκτικές μονάδες.

Υπάρχουν ακόμη και εργασίες πάνω σε αποδοτικές μεθόδους για την κλιμάκωση Μεγάλων Μοντέλων Γλώσσας (ΜΜΓ) βασισμένων σε Transformers για [άπειρα μεγάλες εισόδους](#) με περιορισμένη μνήμη και υπολογισμούς.

Γιατί είναι Σημαντικό το Μέγεθος Περιβάλλοντος;

Το μέγεθος περιβάλλοντος ενός μοντέλου γλώσσας έχει σημαντική επίδραση στην ικανότητά του να κατανοεί και να δημιουργεί συνεκτικό, συναφές με το περιβάλλον κείμενο. Εδώ είναι μερικοί βασικοί λόγοι για τους οποίους το μέγεθος περιβάλλοντος έχει σημασία:

1. **Κατανόηση εκτενούς περιεχομένου:** Μοντέλα με μεγαλύτερα μεγέθη περιβάλλοντος μπορούν να κατανοήσουν και να αναλύσουν καλύτερα μεγαλύτερα κείμενα, όπως άρθρα, αναφορές ή ακόμη και ολόκληρα βιβλία. Αυτό είναι κρίσιμο για εργασίες όπως η περίληψη εγγράφων, η απάντηση ερωτήσεων και η ανάλυση περιεχομένου.
2. **Διατήρηση συνοχής:** Ένα μεγαλύτερο παράθυρο περιβάλλοντος επιτρέπει στο μοντέλο να διατηρεί τη συνοχή και τη συνέπεια σε μεγαλύτερα τμήματα εξόδου. Αυτό είναι σημαντικό για εργασίες όπως η δημιουργία ιστοριών, τα συστήματα διαλόγου και η δημιουργία περιεχομένου, όπου η διατήρηση μιας συνεπούς αφήγησης ή θέματος είναι απαραίτητη. Είναι επίσης απολύτως κρίσιμο όταν χρησιμοποιούμε ΜΜΓ για τη δημιουργία ή τον μετασχηματισμό δομημένων δεδομένων.
3. **Καταγραφή εξαρτήσεων μεγάλης εμβέλειας:** Ορισμένες γλωσσικές εργασίες απαιτούν την κατανόηση σχέσεων μεταξύ λέξεων ή φράσεων που βρίσκονται μακριά μεταξύ τους σε ένα κείμενο. Τα μοντέλα με μεγαλύτερο μέγεθος πλαισίου είναι καλύτερα εξοπλισμένα για να καταγράψουν αυτές

τις εξαρτήσεις μεγάλης εμβέλειας, οι οποίες μπορεί να είναι σημαντικές για εργασίες όπως η ανάλυση συναισθήματος, η μετάφραση και η κατανόηση γλώσσας.

4. **Χειρισμός πολύπλοκων οδηγιών:** Σε εφαρμογές όπου τα γλωσσικά μοντέλα χρησιμοποιούνται για την εκτέλεση πολύπλοκων, πολυβηματικών οδηγιών, ένα μεγαλύτερο μέγεθος πλαισίου επιτρέπει στο μοντέλο να λαμβάνει υπόψη ολόκληρο το σύνολο των οδηγιών κατά τη δημιουργία απάντησης, αντί μόνο για τις πιο πρόσφατες λέξεις.

Παραδείγματα Γλωσσικών Μοντέλων με Διαφορετικά Μεγέθη Πλαισίου

Ακολουθούν μερικά παραδείγματα γλωσσικών μοντέλων με διαφορετικά μεγέθη πλαισίου:

- OpenAI GPT-3.5 Turbo: 4.095 tokens
- Mistral 7B Instruct: 32.768 tokens
- Anthropic Claude v1: 100.000 tokens
- OpenAI GPT-4 Turbo: 128.000 tokens
- Anthropic Claude v2: 200.000 tokens
- Google Gemini Pro 1.5: 2,8M tokens

Όπως μπορείτε να δείτε, υπάρχει ένα ευρύ φάσμα μεγεθών πλαισίου μεταξύ αυτών των μοντέλων, από περίπου 4.000 tokens για το μοντέλο OpenAI GPT-3.5 Turbo έως 200.000 tokens για το μοντέλο Anthropic Claude v2. Ορισμένα μοντέλα, όπως το Google's PaLM 2 και το OpenAI's GPT-4, προσφέρουν διαφορετικές παραλλαγές με μεγαλύτερα μεγέθη πλαισίου (π.χ., εκδόσεις "32k"), οι οποίες μπορούν να χειριστούν ακόμη μεγαλύτερες ακολουθίες εισόδου. Και αυτή τη στιγμή (Απρίλιος 2024) το Google Gemini Pro διαθέτει σχεδόν 3 εκατομμύρια tokens!

Αξίζει να σημειωθεί ότι το μέγεθος πλαισίου μπορεί να διαφέρει ανάλογα με τη συγκεκριμένη υλοποίηση και την έκδοση ενός συγκεκριμένου μοντέλου. Για

παράδειγμα, το αρχικό μοντέλο OpenAI GPT-4 έχει μέγεθος πλαισίου 8.191 tokens, ενώ οι μεταγενέστερες παραλλαγές του GPT-4 όπως το Turbo και το 4o έχουν πολύ μεγαλύτερο μέγεθος πλαισίου 128.000 tokens.

Ο Sam Altman έχει συγκρίνει τους τρέχοντες περιορισμούς πλαισίου με τα kilobytes μνήμης εργασίας που έπρεπε να διαχειριστούν οι προγραμματιστές προσωπικών υπολογιστών τη δεκαετία του '80, και δήλωσε ότι στο εγγύς μέλλον θα μπορούμε να χωρέσουμε “όλα τα προσωπικά σας δεδομένα” στο πλαίσιο ενός μεγάλου γλωσσικού μοντέλου.

Επιλέγοντας το Σωστό Μέγεθος Πλαισίου

Κατά την επιλογή ενός γλωσσικού μοντέλου για μια συγκεκριμένη εφαρμογή, είναι σημαντικό να εξετάσουμε τις απαιτήσεις μεγέθους πλαισίου της εκάστοτε εργασίας. Για εργασίες που περιλαμβάνουν σύντομα, μεμονωμένα κείμενα, όπως η ανάλυση συναισθήματος ή η απλή απάντηση ερωτήσεων, ένα μικρότερο μέγεθος πλαισίου μπορεί να είναι επαρκές. Ωστόσο, για εργασίες που απαιτούν κατανόηση και δημιουργία μεγαλύτερων, πιο πολύπλοκων κειμένων, πιθανότατα θα χρειαστεί μεγαλύτερο μέγεθος πλαισίου.

Αξίζει να σημειωθεί ότι τα μεγαλύτερα μεγέθη πλαισίου συχνά συνοδεύονται από αυξημένο υπολογιστικό κόστος και πιο αργούς χρόνους επεξεργασίας, καθώς το μοντέλο πρέπει να λάβει υπόψη περισσότερες πληροφορίες κατά τη δημιουργία μιας απάντησης. Ως εκ τούτου, πρέπει να βρείτε μια ισορροπία μεταξύ μεγέθους πλαισίου και απόδοσης κατά την επιλογή ενός γλωσσικού μοντέλου για την εφαρμογή σας.

Γιατί να μην επιλέξετε απλά το μοντέλο με το μεγαλύτερο μέγεθος πλαισίου και να το γεμίσετε με όσες πληροφορίες είναι δυνατόν; Λοιπόν, εκτός από τους παράγοντες απόδοσης, η άλλη κύρια παράμετρος είναι το κόστος. Τον Μάρτιο του 2024, ένας μόνο κύκλος προτροπής-απόκρισης χρησιμοποιώντας το Google Gemini Pro 1.5 με πλήρες πλαίσιο θα σας κοστίσει σχεδόν 8 δολάρια (USD). Αν έχετε μια περίπτωση χρήσης που δικαιολογεί αυτό το έξοδο, συγχαρητήρια! Αλλά για τις περισσότερες εφαρμογές, είναι απλά υπερβολικά ακριβό κατά τάξεις μεγέθους.

Βρίσκοντας Βελόνες στα Άχυρα

Η έννοια της εύρεσης μιας βελόνας στα άχυρα είναι από καιρό μια μεταφορά για τις προκλήσεις της ανάκτησης σε μεγάλα σύνολα δεδομένων. Στον τομέα των ΜΓΜ, τροποποιούμε λίγο αυτή την αναλογία. Φανταστείτε ότι δεν ψάχνουμε απλώς για ένα μεμονωμένο γεγονός θαμμένο μέσα σε ένα τεράστιο κείμενο (όπως μια πλήρη ανθολογία δοκιμίων του Paul Graham), αλλά πολλαπλά γεγονότα διασκορπισμένα παντού. Αυτό το σενάριο μοιάζει περισσότερο με την εύρεση πολλών βελονών σε ένα απέραντο χωράφι, όχι απλώς σε μία θημωνιά. Και ιδού η έκπληξη: δεν χρειάζεται μόνο να εντοπίσουμε αυτές τις βελόνες, αλλά πρέπει επίσης να τις πλέξουμε σε ένα συνεκτικό νήμα.

Όταν καλούνται να ανακτήσουν και να συλλογιστούν σχετικά με πολλαπλά γεγονότα ενσωματωμένα σε μεγάλα πλαίσια, τα ΜΓΜ αντιμετωπίζουν μια διπλή πρόκληση. Πρώτον, υπάρχει το προφανές ζήτημα της ακρίβειας ανάκτησης—φυσικά μειώνεται καθώς αυξάνεται ο αριθμός των γεγονότων. Αυτό είναι αναμενόμενο· άλλωστε, η παρακολούθηση πολλαπλών λεπτομερειών σε ένα εκτεταμένο κείμενο δοκιμάζει ακόμη και τα πιο εξελιγμένα μοντέλα.

Δεύτερον, και ίσως πιο κρίσιμο, είναι η πρόκληση του συλλογισμού με αυτά τα δεδομένα. Είναι ένα πράγμα να επιλέγεις δεδομένα και εντελώς διαφορετικό να

τα συνθέτεις σε μια συνεκτική αφήγηση ή απάντηση. Εδώ έρχεται η πραγματική δοκιμασία. Η απόδοση των ΜΜΓ σε εργασίες συλλογισμού τείνει να υποβαθμίζεται περισσότερο από ό,τι σε απλές εργασίες ανάκτησης. Αυτή η υποβάθμιση δεν αφορά μόνο τον όγκο· αφορά τον περίπλοκο χορό του πλαισίου, της συνάφειας και της συμπερασματολογίας.

Γιατί συμβαίνει αυτό; Λοιπόν, σκεφτείτε τη δυναμική της μνήμης και της προσοχής στην ανθρώπινη νόηση, η οποία αντικατοπτρίζεται σε κάποιο βαθμό στα ΜΜΓ. Κατά την επεξεργασία μεγάλων ποσοτήτων πληροφοριών, τα ΜΜΓ, όπως οι άνθρωποι, μπορεί να χάσουν την παρακολούθηση προηγούμενων λεπτομερειών καθώς απορροφούν νέες. Αυτό ισχύει ιδιαίτερα σε μοντέλα που δεν έχουν σχεδιαστεί ρητά για να δίνουν προτεραιότητα ή να επανεξετάζουν αυτόματα προηγούμενα τμήματα κειμένου.

Επιπλέον, η ικανότητα ενός ΜΜΓ να πλέκει αυτά τα ανακτημένα δεδομένα σε μια συνεκτική απάντηση μοιάζει με τη δόμηση αφήγησης. Αυτό απαιτεί όχι μόνο την ανάκτηση πληροφοριών αλλά και βαθιά κατανόηση και τοποθέτηση στο πλαίσιο, κάτι που παραμένει μια δύσκολη πρόκληση για την τρέχουσα ΤΝ.

Λοιπόν, τι σημαίνει αυτό για εμάς ως προγραμματιστές και ολοκληρωτές αυτών των τεχνολογιών; Πρέπει να είμαστε ιδιαίτερα ενήμεροι για αυτούς τους περιορισμούς όταν σχεδιάζουμε συστήματα που βασίζονται σε ΜΜΓ για να χειριστούν πολύπλοκες εργασίες μεγάλης έκτασης. Η κατανόηση ότι η απόδοση μπορεί να υποβαθμιστεί υπό ορισμένες συνθήκες μας βοηθά να θέσουμε ρεαλιστικές προσδοκίες και να σχεδιάσουμε καλύτερους μηχανισμούς εφεδρείας ή συμπληρωματικές στρατηγικές.

Τροπικότητες: Πέρα από το Κείμενο

Ενώ η πλειονότητα των μοντέλων γλώσσας σήμερα επικεντρώνεται στην επεξεργασία και παραγωγή κειμένου, υπάρχει μια αυξανόμενη τάση προς πολυτροπικά μοντέλα που μπορούν εγγενώς να δέχονται και να παράγουν πολλαπλούς τύπους δεδομένων, όπως εικόνες, ήχο και βίντεο. Αυτά τα πολυτροπικά

μοντέλα ανοίγουν νέες δυνατότητες για εφαρμογές με τεχνητή νοημοσύνη που μπορούν να κατανοούν και να παράγουν περιεχόμενο σε διαφορετικές τροπικότητες.

Τι είναι οι Τροπικότητες;

Στο πλαίσιο των μοντέλων γλώσσας, οι τροπικότητες αναφέρονται στους διαφορετικούς τύπους δεδομένων που μπορεί να επεξεργαστεί και να παράγει ένα μοντέλο. Η πιο συνηθισμένη τροπικότητα είναι το κείμενο, που περιλαμβάνει γραπτή γλώσσα σε διάφορες μορφές όπως βιβλία, άρθρα, ιστοσελίδες και αναρτήσεις στα μέσα κοινωνικής δικτύωσης. Ωστόσο, υπάρχουν αρκετές άλλες τροπικότητες που ενσωματώνονται όλο και περισσότερο στα μοντέλα γλώσσας:

- **Εικόνες:** Οπτικά δεδομένα όπως φωτογραφίες, εικονογραφήσεις και διαγράμματα.
- **Ήχος:** Ηχητικά δεδομένα όπως ομιλία, μουσική και περιβαλλοντικοί ήχοι.
- **Βίντεο:** Κινούμενα οπτικά δεδομένα, συχνά συνοδευόμενα από ήχο, όπως βιντεοκλίπ και ταινίες.

Κάθε τροπικότητα παρουσιάζει μοναδικές προκλήσεις και ευκαιρίες για τα γλωσσικά μοντέλα. Για παράδειγμα, οι εικόνες απαιτούν από το μοντέλο να κατανοεί οπτικές έννοιες και σχέσεις, ενώ ο ήχος απαιτεί από το μοντέλο να επεξεργάζεται και να παράγει ομιλία και άλλους ήχους.

Πολυτροπικά Γλωσσικά Μοντέλα

Τα πολυτροπικά γλωσσικά μοντέλα είναι σχεδιασμένα να χειρίζονται πολλαπλές τροπικότητες μέσα σε ένα ενιαίο μοντέλο. Αυτά τα μοντέλα συνήθως διαθέτουν εξειδικευμένα στοιχεία ή επίπεδα που μπορούν τόσο να κατανοούν εισόδους όσο και να παράγουν δεδομένα εξόδου σε διαφορετικές τροπικότητες. Μερικά αξιοσημείωτα παραδείγματα πολυτροπικών γλωσσικών μοντέλων περιλαμβάνουν:

- **OpenAI's GPT-4o:** Το GPT-4o είναι ένα μεγάλο γλωσσικό μοντέλο που εγγενώς κατανοεί και επεξεργάζεται ηχητικό λόγο εκτός από κείμενο. Αυτή η δυνατότητα επιτρέπει στο GPT-4o να εκτελεί εργασίες όπως η μεταγραφή προφορικού λόγου, η παραγωγή κειμένου από ηχητικές εισόδους και η παροχή απαντήσεων βάσει προφορικών ερωτημάτων.
- **OpenAI's GPT-4 με οπτική είσοδο:** Το GPT-4 είναι ένα μεγάλο γλωσσικό μοντέλο που μπορεί να επεξεργάζεται τόσο κείμενο όσο και εικόνες. Όταν του δίνεται μια εικόνα ως είσοδος, το GPT-4 μπορεί να αναλύσει το περιεχόμενο της εικόνας και να παράγει κείμενο που περιγράφει ή ανταποκρίνεται στην οπτική πληροφορία.
- **Google's Gemini:** Το Gemini είναι ένα πολυτροπικό μοντέλο που μπορεί να χειριστεί κείμενο, εικόνες και βίντεο. Χρησιμοποιεί μια ενοποιημένη αρχιτεκτονική που επιτρέπει τη διατροπική κατανόηση και παραγωγή, καθιστώντας δυνατές εργασίες όπως η λεζαντοποίηση εικόνων, η σύνοψη βίντεο και η απάντηση οπτικών ερωτήσεων.
- **DALL-E και Stable Diffusion:** Αν και δεν είναι γλωσσικά μοντέλα με την παραδοσιακή έννοια, αυτά τα μοντέλα καταδεικνύουν τη δύναμη της πολυτροπικής τεχνητής νοημοσύνης παράγοντας εικόνες από κειμενικές περιγραφές. Αναδεικνύουν τις δυνατότητες των μοντέλων που μπορούν να μεταφράζουν μεταξύ διαφορετικών τροπικοτήτων.

Οφέλη και Εφαρμογές των Πολυτροπικών Μοντέλων

Τα πολυτροπικά γλωσσικά μοντέλα προσφέρουν διάφορα οφέλη και καθιστούν δυνατό ένα ευρύ φάσμα εφαρμογών, συμπεριλαμβανομένων:

- **Ενισχυμένη κατανόηση:** Επεξεργάζοντας πληροφορίες από πολλαπλές τροπικότητες, αυτά τα μοντέλα μπορούν να αποκτήσουν μια πιο ολοκληρωμένη κατανόηση του κόσμου, παρόμοια με τον τρόπο που οι άνθρωποι μαθαίνουν από διάφορες αισθητηριακές εισόδους.

- **Διατροφική παραγωγή:** Τα πολυτροπικά μοντέλα μπορούν να παράγουν περιεχόμενο σε μία τροπικότητα βασισμένα σε είσοδο από άλλη, όπως η δημιουργία μιας εικόνας από μια κειμενική περιγραφή ή η παραγωγή σύνοψης βίντεο από ένα γραπτό άρθρο.
- **Προσβασιμότητα:** Τα πολυτροπικά μοντέλα μπορούν να κάνουν τις πληροφορίες πιο προσβάσιμες μεταφράζοντας μεταξύ τροπικοτήτων, όπως η δημιουργία κειμενικών περιγραφών εικόνων για χρήστες με προβλήματα όρασης ή η δημιουργία ηχητικών εκδόσεων γραπτού περιεχομένου.
- **Δημιουργικές εφαρμογές:** Τα πολυτροπικά μοντέλα μπορούν να χρησιμοποιηθούν για δημιουργικές εργασίες όπως η δημιουργία τέχνης, μουσικής ή βίντεο με βάση κειμενικές προτροπές, ανοίγοντας νέες δυνατότητες για καλλιτέχνες και δημιουργούς περιεχομένου.

Καθώς τα πολυτροπικά γλωσσικά μοντέλα συνεχίζουν να εξελίσσονται, είναι πιθανό να διαδραματίσουν έναν ολοένα και πιο σημαντικό ρόλο στην ανάπτυξη εφαρμογών που βασίζονται στην τεχνητή νοημοσύνη και μπορούν να κατανοήσουν και να δημιουργήσουν περιεχόμενο σε πολλαπλές μορφές. Αυτό θα επιτρέψει πιο φυσικές και διαισθητικές αλληλεπιδράσεις μεταξύ ανθρώπων και συστημάτων τεχνητής νοημοσύνης, καθώς και θα ξεκλειδώσει νέες δυνατότητες για δημιουργική έκφραση και διάδοση γνώσης.

Οικοσυστήματα Παρόχων

Όσον αφορά την ενσωμάτωση μεγάλων γλωσσικών μοντέλων (LLMs) σε εφαρμογές, έχετε στη διάθεσή σας ένα αυξανόμενο εύρος επιλογών. Κάθε σημαντικός πάροχος LLM, όπως η OpenAI, η Anthropic, η Google και η Cohere, προσφέρει το δικό του οικοσύστημα μοντέλων, APIs και εργαλείων. Η επιλογή του κατάλληλου παρόχου περιλαμβάνει την εξέταση διαφόρων παραγόντων, συμπεριλαμβανομένης της τιμολόγησης, της απόδοσης, του φιλτραρίσματος περιεχομένου, της προστασίας δεδομένων και των επιλογών προσαρμογής.

OpenAI

Η OpenAI είναι ένας από τους πιο γνωστούς παρόχους LLMs, με τη σειρά GPT (GPT-3, GPT-4) να χρησιμοποιείται ευρέως σε διάφορες εφαρμογές. Η OpenAI προσφέρει ένα φιλικό προς το χρήστη API που σας επιτρέπει να ενσωματώσετε εύκολα τα μοντέλα τους σε εφαρμογές. Παρέχουν μια σειρά από μοντέλα με διαφορετικές δυνατότητες και τιμές, από το βασικό μοντέλο Ada έως το ισχυρό μοντέλο Davinci.

Το οικοσύστημα της OpenAI περιλαμβάνει επίσης εργαλεία όπως το OpenAI Playground, το οποίο σας επιτρέπει να πειραματιστείτε με προτροπές και να κάνετε λεπτομερή ρύθμιση των μοντέλων για συγκεκριμένες περιπτώσεις χρήσης. Προσφέρουν επιλογές φιλτραρίσματος περιεχομένου για να βοηθήσουν στην αποτροπή της δημιουργίας ακατάλληλου ή επιβλαβούς περιεχομένου.

Όταν χρησιμοποιώ τα μοντέλα της OpenAI απευθείας, βασίζομαι στη βιβλιοθήκη [ruby-openai](#) του Alex Rudall.

Anthropic

Η Anthropic είναι ένας άλλος σημαντικός παίκτης στο χώρο των LLM, με τα μοντέλα Claude να κερδίζουν δημοτικότητα για την ισχυρή απόδοση και τις ηθικές τους προεκτάσεις. Η Anthropic επικεντρώνεται στην ανάπτυξη ασφαλών και υπεύθυνων συστημάτων τεχνητής νοημοσύνης, με ιδιαίτερη έμφαση στο φιλτράρισμα περιεχομένου και την αποφυγή επιβλαβών αποτελεσμάτων.

Το οικοσύστημα της Anthropic περιλαμβάνει το Claude API, το οποίο σας επιτρέπει να ενσωματώσετε το μοντέλο στις εφαρμογές σας, καθώς και εργαλεία για μηχανική προτροπών και λεπτομερή ρύθμιση. Προσφέρουν επίσης το μοντέλο Claude Instant, το οποίο ενσωματώνει δυνατότητες αναζήτησης στο διαδίκτυο για πιο ενημερωμένες και ακριβείς απαντήσεις.

Όταν χρησιμοποιώ τα μοντέλα της Anthropic απευθείας, βασίζομαι στη βιβλιοθήκη [anthropic](#) του Alex Rudall.

Google

Η Google έχει αναπτύξει αρκετά ισχυρά LLMs, συμπεριλαμβανομένων των Gemini, BERT, T5, και PaLM. Αυτά τα μοντέλα είναι γνωστά για την ισχυρή τους απόδοση σε ένα ευρύ φάσμα εργασιών επεξεργασίας φυσικής γλώσσας. Το οικοσύστημα της Google περιλαμβάνει τις βιβλιοθήκες TensorFlow και Keras, οι οποίες παρέχουν εργαλεία και πλαίσια για την κατασκευή και εκπαίδευση μοντέλων μηχανικής μάθησης.

Η Google προσφέρει επίσης μια Πλατφόρμα Cloud AI, η οποία σας επιτρέπει να αναπτύξετε και να κλιμακώσετε εύκολα τα μοντέλα τους στο cloud. Παρέχουν μια σειρά από προ-εκπαιδευμένα μοντέλα και APIs για εργασίες όπως η ανάλυση συναισθημάτων, η αναγνώριση οντοτήτων και η μετάφραση.

Meta

Η Meta, πρώην γνωστή ως Facebook, έχει επενδύσει σημαντικά στην ανάπτυξη μεγάλων γλωσσικών μοντέλων, όπως φαίνεται από την κυκλοφορία μοντέλων όπως το LLaMA και το OPT. Αυτά τα μοντέλα ξεχωρίζουν για την ισχυρή τους απόδοση σε διάφορες γλωσσικές εργασίες και διατίθενται κυρίως μέσω καναλιών ανοιχτού κώδικα, υποστηρίζοντας τη δέσμευση της Meta στην έρευνα και τη συνεργασία με την κοινότητα.

Το οικοσύστημα της Meta είναι κυρίως χτισμένο γύρω από το PyTorch, μια βιβλιοθήκη μηχανικής μάθησης ανοιχτού κώδικα που προτιμάται για τις δυναμικές υπολογιστικές δυνατότητες και την ευελιξία της, διευκολύνοντας την καινοτόμο έρευνα και ανάπτυξη AI.

Εκτός από τις τεχνικές τους προσφορές, η Meta δίνει μεγάλη έμφαση στην ηθική ανάπτυξη της τεχνητής νοημοσύνης. Εφαρμόζουν ισχυρό φιλτράρισμα περιεχομένου και εστιάζουν στη μείωση των προκαταλήψεων, ευθυγραμμίζόμενοι με τους ευρύτερους στόχους τους για ασφάλεια και υπευθυνότητα στις εφαρμογές τεχνητής νοημοσύνης.

Cohere

Η Cohere είναι ένας νεότερος παίκτης στον χώρο των LLM, εστιάζοντας στο να κάνει τα LLMs πιο προσβάσιμα και ευκολότερα στη χρήση σε σύγκριση με τους ανταγωνιστές. Το οικοσύστημά τους περιλαμβάνει το Cohere API, το οποίο παρέχει πρόσβαση σε μια σειρά από προ-εκπαιδευμένα μοντέλα για εργασίες όπως η δημιουργία κειμένου, η ταξινόμηση και η περίληψη.

Η Cohere προσφέρει επίσης εργαλεία για μηχανική προτροπών, λεπτομερή ρύθμιση και φιλτράρισμα περιεχομένου. Δίνουν έμφαση στην προστασία και ασφάλεια των δεδομένων, με χαρακτηριστικά όπως η κρυπτογραφημένη αποθήκευση δεδομένων και οι έλεγχοι πρόσβασης.

Ollama

Το Ollama είναι μια αυτοφιλοξενούμενη πλατφόρμα που επιτρέπει στους χρήστες να διαχειρίζονται και να αναπτύσσουν διάφορα μεγάλα γλωσσικά μοντέλα (LLMs) τοπικά στους υπολογιστές τους, δίνοντάς τους πλήρη έλεγχο των μοντέλων τεχνητής νοημοσύνης χωρίς να βασίζονται σε εξωτερικές υπηρεσίες cloud. Αυτή η ρύθμιση είναι ιδανική για όσους δίνουν προτεραιότητα στην προστασία των δεδομένων και επιθυμούν να διαχειρίζονται τις λειτουργίες τεχνητής νοημοσύνης εσωτερικά.

Η πλατφόρμα υποστηρίζει μια σειρά από μοντέλα, συμπεριλαμβανομένων εκδόσεων των Llama, Phi, Gemma και Mistral, τα οποία διαφέρουν σε μέγεθος και υπολογιστικές απαιτήσεις. Το Ollama καθιστά εύκολη τη λήψη και εκτέλεση αυτών των μοντέλων απευθείας από τη γραμμή εντολών χρησιμοποιώντας απλές εντολές όπως `ollama run <model_name>`, και είναι σχεδιασμένο να λειτουργεί σε διάφορα λειτουργικά συστήματα συμπεριλαμβανομένων των macOS, Linux και Windows.

Για τους προγραμματιστές που θέλουν να ενσωματώσουν μοντέλα ανοιχτού κώδικα στις εφαρμογές τους χωρίς τη χρήση απομακρυσμένου API, το Ollama προσφέρει ένα CLI για τη διαχείριση του κύκλου ζωής των μοντέλων παρόμοιο με

τα εργαλεία διαχείρισης containers. Υποστηρίζει επίσης προσαρμοσμένες ρυθμίσεις και προτροπές, επιτρέποντας υψηλό βαθμό εξατομίκευσης για την προσαρμογή των μοντέλων σε συγκεκριμένες ανάγκες ή περιπτώσεις χρήσης.

Το Ollama είναι ιδιαίτερα κατάλληλο για τεχνικά καταρτισμένους χρήστες και προγραμματιστές λόγω της διεπαφής γραμμής εντολών και της ευελιξίας που προσφέρει στη διαχείριση και ανάπτυξη μοντέλων AI. Αυτό το καθιστά ένα ισχυρό εργαλείο για επιχειρήσεις και ιδιώτες που απαιτούν ισχυρές δυνατότητες AI χωρίς να κάνουν συμβιβασμούς στην ασφάλεια και τον έλεγχο.

Πλατφόρμες Πολλαπλών Μοντέλων

Επιπλέον, υπάρχουν πάροχοι που φιλοξενούν μια μεγάλη ποικιλία μοντέλων ανοιχτού κώδικα, όπως το Together.ai και το Groq. Αυτές οι πλατφόρμες προσφέρουν ευελιξία και δυνατότητα προσαρμογής, επιτρέποντάς σας να εκτελείτε και, σε ορισμένες περιπτώσεις, ακόμη και να βελτιστοποιείτε μοντέλα ανοιχτού κώδικα σύμφωνα με τις συγκεκριμένες ανάγκες σας. Για παράδειγμα, το Together.ai παρέχει πρόσβαση σε μια σειρά από LLM ανοιχτού κώδικα, επιτρέποντας στους χρήστες να πειραματιστούν με διαφορετικά μοντέλα και διαμορφώσεις. Το Groq εστιάζει στην παροχή εξαιρετικά υψηλής απόδοσης ολοκλήρωσης που κατά τη στιγμή της συγγραφής αυτού του βιβλίου φαίνεται σχεδόν μαγική

Επιλογή Παρόχου LLM

Κατά την επιλογή ενός παρόχου LLM, θα πρέπει να λάβετε υπόψη παράγοντες όπως:

- **Τιμολόγηση:** Διαφορετικοί πάροχοι προσφέρουν διαφορετικά μοντέλα τιμολόγησης, από χρέωση ανά χρήση έως προγράμματα με συνδρομή. Είναι σημαντικό να εξετάσετε την αναμενόμενη χρήση και τον προϋπολογισμό κατά την επιλογή ενός παρόχου.

- **Απόδοση:** Η απόδοση των LLM μπορεί να διαφέρει σημαντικά μεταξύ των παρόχων, επομένως είναι σημαντικό να συγκρίνετε και να δοκιμάζετε τα μοντέλα σε συγκεκριμένες περιπτώσεις χρήσης πριν από τη λήψη απόφασης.
- **Φιλτράρισμα Περιεχομένου:** Ανάλογα με την εφαρμογή, το φιλτράρισμα περιεχομένου μπορεί να είναι κρίσιμη παράμετρος. Ορισμένοι πάροχοι προσφέρουν πιο ισχυρές επιλογές φιλτραρίσματος περιεχομένου από άλλους.
- **Προστασία Δεδομένων:** Εάν η εφαρμογή διαχειρίζεται ευαίσθητα δεδομένα χρηστών, είναι σημαντικό να επιλέξετε έναν πάροχο με ισχυρές πρακτικές προστασίας δεδομένων και ασφάλειας.
- **Προσαρμογή:** Ορισμένοι πάροχοι προσφέρουν μεγαλύτερη ευελιξία όσον αφορά τη βελτιστοποίηση και προσαρμογή των μοντέλων για συγκεκριμένες περιπτώσεις χρήσης.

Τελικά, η επιλογή παρόχου ΜΓΜ εξαρτάται από τις συγκεκριμένες απαιτήσεις και τους περιορισμούς της εφαρμογής. Αξιολογώντας προσεκτικά τις επιλογές και λαμβάνοντας υπόψη παράγοντες όπως η τιμολόγηση, η απόδοση και η προστασία δεδομένων, μπορείτε να επιλέξετε τον πάροχο που ανταποκρίνεται καλύτερα στις ανάγκες σας.

Αξίζει επίσης να σημειωθεί ότι το τοπίο των ΜΓΜ εξελίσσεται συνεχώς, με νέους παρόχους και μοντέλα να εμφανίζονται τακτικά. Θα πρέπει να παραμένετε ενημερωμένοι για τις τελευταίες εξελίξεις και να είστε ανοιχτοί στην εξερεύνηση νέων επιλογών καθώς αυτές γίνονται διαθέσιμες.

OpenRouter

Σε όλο αυτό το βιβλίο θα βασίζομαι αποκλειστικά στο [OpenRouter](#) ως τον πάροχο API της επιλογής μου. Ο λόγος είναι απλός: είναι ένα κατάστημα μίας στάσης για όλα τα πιο δημοφιλή εμπορικά και ανοιχτού κώδικα μοντέλα. Εάν ανυπομονείτε να ασχοληθείτε με κάποιο προγραμματισμό AI, ένα από τα καλύτερα σημεία για να ξεκινήσετε είναι με τη δική μου [OpenRouter Ruby Library](#).

Σκέψεις για την Απόδοση

Κατά την ενσωμάτωση γλωσσικών μοντέλων σε εφαρμογές, η απόδοση αποτελεί κρίσιμο παράγοντα. Η απόδοση ενός γλωσσικού μοντέλου μπορεί να μετρηθεί με βάση την *καθυστέρηση* (τον χρόνο που χρειάζεται για να δημιουργήσει μια απάντηση) και τη *διεκπεραιωτική ικανότητα* (τον αριθμό των αιτημάτων που μπορεί να διαχειριστεί ανά μονάδα χρόνου).

Ο *Χρόνος έως το Πρώτο Σύμβολο* (ΧΕΠΣ) είναι ένας άλλος ουσιώδης δείκτης απόδοσης, ιδιαίτερα σημαντικός για chatbots και εφαρμογές που απαιτούν διαδραστικές απαντήσεις σε πραγματικό χρόνο. Ο ΧΕΠΣ μετρά την καθυστέρηση από τη στιγμή που λαμβάνεται το αίτημα του χρήστη μέχρι τη στιγμή που δημιουργείται η πρώτη λέξη (ή σύμβολο) της απάντησης. Αυτός ο δείκτης είναι κρίσιμος για τη διατήρηση μιας ομαλής και ελκυστικής εμπειρίας χρήστη, καθώς οι καθυστερημένες απαντήσεις μπορούν να οδηγήσουν σε απογοήτευση και αποδέσμευση του χρήστη.

Αυτοί οι δείκτες απόδοσης μπορούν να έχουν σημαντικό αντίκτυπο στην εμπειρία χρήστη και την κλιμάκωση της εφαρμογής.

Διάφοροι παράγοντες μπορούν να επηρεάσουν την απόδοση ενός γλωσσικού μοντέλου, συμπεριλαμβανομένων:

Αριθμός Παραμέτρων: Μεγαλύτερα μοντέλα με περισσότερες παραμέτρους γενικά απαιτούν περισσότερους υπολογιστικούς πόρους και μπορεί να έχουν υψηλότερη καθυστέρηση και χαμηλότερη διεκπεραιωτική ικανότητα σε σύγκριση με μικρότερα μοντέλα.

Υλικό: Η απόδοση ενός γλωσσικού μοντέλου μπορεί να διαφέρει σημαντικά ανάλογα με το υλικό στο οποίο εκτελείται. Οι πάροχοι cloud προσφέρουν στιγμιότυπα GPU και TPU βελτιστοποιημένα για φόρτους εργασίας μηχανικής μάθησης, τα οποία μπορούν να επιταχύνουν σημαντικά την εξαγωγή συμπερασμάτων του μοντέλου.



Ένα από τα ωραία πράγματα σχετικά με το OpenRouter είναι ότι για πολλά από τα μοντέλα που προσφέρει, έχετε επιλογή παρόχων cloud με διάφορα προφίλ απόδοσης και κόστους.

Ποσοτικοποίηση: Οι τεχνικές ποσοτικοποίησης μπορούν να χρησιμοποιηθούν για τη μείωση του αποτυπώματος μνήμης και των υπολογιστικών απαιτήσεων ενός μοντέλου, αναπαριστώντας τα βάρη και τις ενεργοποιήσεις με τύπους δεδομένων χαμηλότερης ακρίβειας. Αυτό μπορεί να βελτιώσει την απόδοση χωρίς να θυσιάζει σημαντικά την ποιότητα. Ως προγραμματιστής εφαρμογών, πιθανότατα δεν θα ασχοληθείτε με την εκπαίδευση των δικών σας μοντέλων σε διαφορετικά επίπεδα ποσοτικοποίησης, αλλά είναι καλό να είστε τουλάχιστον εξοικειωμένοι με την ορολογία.

Ομαδοποίηση: Η επεξεργασία πολλαπλών αιτημάτων ταυτόχρονα σε ομάδες μπορεί να βελτιώσει τη ρυθμική απόδοση διαμοιράζοντας το κόστος φόρτωσης του μοντέλου και μεταφοράς δεδομένων.

Προσωρινή αποθήκευση: Η προσωρινή αποθήκευση των αποτελεσμάτων συχνά χρησιμοποιούμενων προτροπών ή ακολουθιών εισόδου μπορεί να μειώσει τον αριθμό των αιτημάτων εξαγωγής συμπερασμάτων και να βελτιώσει τη συνολική απόδοση.

Κατά την επιλογή ενός γλωσσικού μοντέλου για μια εφαρμογή παραγωγής, είναι σημαντικό να αξιολογείτε την απόδοσή του σε αντιπροσωπευτικούς φόρτους εργασίας και διαμορφώσεις υλικού. Αυτό μπορεί να βοηθήσει στον εντοπισμό πιθανών σημείων συμφόρησης και να διασφαλίσει ότι το μοντέλο μπορεί να ανταποκριθεί στους απαιτούμενους στόχους απόδοσης.

Αξίζει επίσης να εξετάσετε τους συμβιβασμούς μεταξύ της απόδοσης του μοντέλου και άλλων παραγόντων όπως το κόστος, η ευελιξία και η ευκολία ενσωμάτωσης. Για παράδειγμα, η χρήση ενός μικρότερου, λιγότερο ακριβού μοντέλου με χαμηλότερη καθυστέρηση μπορεί να είναι προτιμότερη για εφαρμογές που απαιτούν απαντήσεις σε πραγματικό χρόνο, ενώ ένα μεγαλύτερο, πιο ισχυρό μοντέλο μπορεί να είναι καταλληλότερο για ομαδική επεξεργασία ή πολύπλοκες εργασίες συλλογισμού.

Πειραματισμός με Διαφορετικά Μοντέλα ΜΓΜ

Η επιλογή ενός ΜΓΜ σπάνια είναι μόνιμη απόφαση. Καθώς νέα και βελτιωμένα μοντέλα κυκλοφορούν τακτικά, είναι καλό να δημιουργείτε εφαρμογές με αρθρωτό τρόπο που επιτρέπει την εναλλαγή διαφορετικών γλωσσικών μοντέλων με την πάροδο του χρόνου. Οι προτροπές και τα σύνολα δεδομένων μπορούν συχνά να επαναχρησιμοποιηθούν σε διάφορα μοντέλα με ελάχιστες αλλαγές. Αυτό σας επιτρέπει να εκμεταλλευτείτε τις τελευταίες εξελίξεις στη μοντελοποίηση γλώσσας χωρίς να χρειάζεται να επανασχεδιάσετε πλήρως τις εφαρμογές σας.



Η δυνατότητα εύκολης εναλλαγής μεταξύ μιας ευρείας γκάμας επιλογών μοντέλων είναι ακόμη ένας λόγος που λατρεύω το OpenRouter.

Κατά την αναβάθμιση σε ένα νέο γλωσσικό μοντέλο, είναι σημαντικό να δοκιμάζετε και να επικυρώνετε διεξοδικά την απόδοση και την ποιότητα των αποτελεσμάτων του για να διασφαλίσετε ότι ανταποκρίνεται στις απαιτήσεις της εφαρμογής. Αυτό μπορεί να περιλαμβάνει επανεκπαίδευση ή λεπτομερή ρύθμιση του μοντέλου σε δεδομένα συγκεκριμένου τομέα, καθώς και ενημέρωση τυχόν κατάντη στοιχείων που εξαρτώνται από τα αποτελέσματα του μοντέλου.

Σχεδιάζοντας εφαρμογές με γνώμονα την απόδοση και την αρθρωτότητα, μπορείτε να δημιουργήσετε κλιμακώσιμα, αποδοτικά και μελλοντικά συστήματα που μπορούν να προσαρμοστούν στο ταχέως εξελισσόμενο τοπίο της τεχνολογίας μοντελοποίησης γλώσσας.

Σύνθετα Συστήματα Τεχνητής Νοημοσύνης

Πριν κλείσουμε την εισαγωγή μας, αξίζει να αναφέρουμε ότι πριν από το 2023 και την έκρηξη ενδιαφέροντος για την παραγωγική ΤΝ που πυροδότησε το ChatGPT, οι παραδοσιακές προσεγγίσεις ΤΝ συνήθως βασίζονταν στην ενσωμάτωση

μεμονωμένων, κλειστών μοντέλων. Αντίθετα, τα *Σύνθετα Συστήματα ΤΝ* αξιοποιούν πολύπλοκες αλυσίδες διασυνδεδεμένων στοιχείων που συνεργάζονται για την επίτευξη ευφυούς συμπεριφοράς.

Στον πυρήνα τους, τα σύνθετα συστήματα ΤΝ αποτελούνται από πολλαπλές μονάδες, καθεμία σχεδιασμένη να εκτελεί συγκεκριμένες εργασίες ή λειτουργίες. Αυτές οι μονάδες μπορεί να περιλαμβάνουν γεννήτριες, συστήματα ανάκτησης, συστήματα κατάταξης, ταξινομητές και διάφορα άλλα εξειδικευμένα στοιχεία. Διασπώντας το συνολικό σύστημα σε μικρότερες, εστιασμένες μονάδες, οι προγραμματιστές μπορούν να δημιουργήσουν πιο ευέλικτες, κλιμακώσιμες και συντηρήσιμες αρχιτεκτονικές ΤΝ.

Ένα από τα βασικά πλεονεκτήματα των σύνθετων συστημάτων ΤΝ είναι η ικανότητά τους να συνδυάζουν τα δυνατά σημεία διαφορετικών τεχνικών και μοντέλων ΤΝ. Για παράδειγμα, ένα σύστημα μπορεί να χρησιμοποιεί ένα μοντέλο μεγάλης γλωσσικής κλίμακας (LLM) για την κατανόηση και παραγωγή φυσικής γλώσσας, ενώ παράλληλα χρησιμοποιεί ένα ξεχωριστό μοντέλο για την ανάκτηση πληροφοριών ή τη λήψη αποφάσεων βάσει κανόνων. Αυτή η αρθρωτή προσέγγιση σας επιτρέπει να επιλέγετε τα καλύτερα εργαλεία και τεχνικές για κάθε συγκεκριμένη εργασία, αντί να βασίζεστε σε μια λύση που υποτίθεται ότι ταιριάζει σε όλες τις περιπτώσεις.

Ωστόσο, η δημιουργία σύνθετων συστημάτων ΤΝ παρουσιάζει επίσης μοναδικές προκλήσεις. Συγκεκριμένα, η διασφάλιση της συνολικής συνοχής και συνέπειας της συμπεριφοράς του συστήματος απαιτεί ισχυρούς μηχανισμούς δοκιμών, παρακολούθησης και διακυβέρνησης.



Η έλευση ισχυρών LLMs όπως το GPT-4 μας επιτρέπει να πειραματιστούμε με σύνθετα συστήματα TN ευκολότερα από ποτέ, καθώς αυτά τα προηγμένα μοντέλα είναι ικανά να χειριστούν πολλαπλούς ρόλους εντός ενός σύνθετου συστήματος, όπως ταξινόμηση, κατάταξη και παραγωγή, επιπλέον των δυνατοτήτων τους στην κατανόηση φυσικής γλώσσας. Αυτή η ευελιξία επιτρέπει στους προγραμματιστές να δημιουργούν γρήγορα πρωτότυπα και να επαναλαμβάνουν τις αρχιτεκτονικές σύνθετων συστημάτων TN, ανοίγοντας νέες δυνατότητες για την ανάπτυξη ευφυών εφαρμογών.

Μοτίβα Ανάπτυξης για Σύνθετα Συστήματα TN

Τα σύνθετα συστήματα TN μπορούν να αναπτυχθούν χρησιμοποιώντας διάφορα μοτίβα, καθένα σχεδιασμένο για την αντιμετώπιση συγκεκριμένων απαιτήσεων και περιπτώσεων χρήσης. Ας εξερευνήσουμε τέσσερα κοινά μοτίβα ανάπτυξης: Ερωταποκρίσεις, Επιλυτές Προβλημάτων Πολλαπλών Πρακτόρων, Συνομιλητική TN, και Συγκυβερνήτες.

Ερωταποκρίσεις

Τα συστήματα Ερωταποκρίσεων (Q&A) επικεντρώνονται στην παροχή ανάκτησης πληροφοριών που ενισχύεται με τις δυνατότητες κατανόησης των μοντέλων TN ώστε να λειτουργούν ως κάτι περισσότερο από μια απλή μηχανή αναζήτησης. Συνδυάζοντας ισχυρά γλωσσικά μοντέλα με εξωτερικές πηγές γνώσης χρησιμοποιώντας [Παραγωγή Ενισχυμένη με Ανάκτηση \(RAG\)](#), τα συστήματα Ερωταποκρίσεων αποφεύγουν τις ψευδαισθήσεις και παρέχουν ακριβείς και συναφείς με το περιεχόμενο απαντήσεις στα ερωτήματα των χρηστών.

Τα βασικά στοιχεία ενός συστήματος Ερωτήσεων-Απαντήσεων βασισμένου σε LLM περιλαμβάνουν:

- **Κατανόηση και αναδιατύπωση ερωτημάτων:** Ανάλυση των ερωτημάτων του χρήστη και αναδιατύπωσή τους για καλύτερη αντιστοίχιση με τις υποκείμενες πηγές γνώσης.
- **Ανάκτηση γνώσης:** Ανάκτηση σχετικών πληροφοριών από δομημένες ή αδόμητες πηγές δεδομένων με βάση το αναδιατυπωμένο ερώτημα.
- **Δημιουργία απαντήσεων:** Δημιουργία συνεκτικών και κατατοπιστικών απαντήσεων συνδυάζοντας την ανακτημένη γνώση με τις δυνατότητες παραγωγής του γλωσσικού μοντέλου.

Τα υποσυστήματα RAG είναι ιδιαίτερα σημαντικά σε τομείς Ερωτήσεων-Απαντήσεων όπου η παροχή ακριβών και επικαιροποιημένων πληροφοριών είναι κρίσιμη, όπως η υποστήριξη πελατών, η διαχείριση γνώσης, ή οι εκπαιδευτικές εφαρμογές

Επιλυτές Προβλημάτων Πολλαπλών Πρακτόρων/Δρώντων

Τα συστήματα πολλαπλών πρακτόρων, γνωστά και ως *Δρώντα*, αποτελούνται από πολλούς αυτόνομους πράκτορες που συνεργάζονται για την επίλυση σύνθετων προβλημάτων. Κάθε πράκτορας έχει συγκεκριμένο ρόλο, σύνολο δεξιοτήτων και πρόσβαση σε σχετικά εργαλεία ή πηγές πληροφοριών. Μέσω της συνεργασίας και της ανταλλαγής πληροφοριών, αυτοί οι πράκτορες μπορούν να αντιμετωπίσουν εργασίες που θα ήταν δύσκολο ή αδύνατο να χειριστεί ένας μεμονωμένος πράκτορας.

Οι βασικές αρχές των επιλυτών προβλημάτων πολλαπλών πρακτόρων περιλαμβάνουν:

- **Εξειδίκευση:** Κάθε πράκτορας επικεντρώνεται σε μια συγκεκριμένη πτυχή του προβλήματος, αξιοποιώντας τις μοναδικές του ικανότητες και γνώσεις.
- **Συνεργασία:** Οι πράκτορες επικοινωνούν και συντονίζουν τις ενέργειές τους για την επίτευξη ενός κοινού στόχου, συχνά μέσω ανταλλαγής μηνυμάτων ή κοινόχρηστης μνήμης.

- **Προσαρμοστικότητα:** Το σύστημα μπορεί να προσαρμοστεί σε μεταβαλλόμενες συνθήκες ή απαιτήσεις προσαρμόζοντας τους ρόλους και τις συμπεριφορές των μεμονωμένων πρακτόρων.

Τα συστήματα πολλαπλών πρακτόρων είναι κατάλληλα για εφαρμογές που απαιτούν κατανομημένη επίλυση προβλημάτων, όπως η βελτιστοποίηση της εφοδιαστικής αλυσίδας, η διαχείριση κυκλοφορίας, ή ο σχεδιασμός αντιμετώπισης έκτακτων αναγκών

Συνομιλητική Τεχνητή Νοημοσύνη

Τα συστήματα Συνομιλητικής Τεχνητής Νοημοσύνης επιτρέπουν αλληλεπιδράσεις φυσικής γλώσσας μεταξύ χρηστών και ευφύων πρακτόρων. Αυτά τα συστήματα συνδυάζουν δυνατότητες κατανόησης φυσικής γλώσσας, διαχείρισης διαλόγου και παραγωγής γλώσσας για την παροχή ελκυστικών και εξατομικευμένων εμπειριών συνομιλίας.

Τα κύρια στοιχεία ενός συστήματος Συνομιλητικής Τεχνητής Νοημοσύνης περιλαμβάνουν:

- **Αναγνώριση πρόθεσης:** Αναγνώριση της πρόθεσης του χρήστη με βάση την είσοδό του, όπως η υποβολή ερώτησης, η διατύπωση αιτήματος ή η έκφραση συναισθήματος.
- **Εξαγωγή οντοτήτων:** Εξαγωγή σχετικών οντοτήτων ή παραμέτρων από την είσοδο του χρήστη, όπως ημερομηνίες, τοποθεσίες ή ονόματα προϊόντων.
- **Διαχείριση διαλόγου:** Διατήρηση της κατάστασης της συνομιλίας, καθορισμός της κατάλληλης απάντησης με βάση την πρόθεση και το πλαίσιο του χρήστη, και χειρισμός αλληλεπιδράσεων πολλαπλών γύρων.
- **Παραγωγή απαντήσεων:** Δημιουργία απαντήσεων που μοιάζουν με ανθρώπινες χρησιμοποιώντας γλωσσικά μοντέλα, πρότυπα ή μεθόδους βασισμένες στην ανάκτηση.

Τα συστήματα Συνομιλητικής ΤΝ χρησιμοποιούνται συχνά σε chatbots εξυπηρέτησης πελατών, εικονικούς βοηθούς και διεπαφές φωνητικού ελέγχου. Όπως αναφέρθηκε νωρίτερα, οι περισσότερες προσεγγίσεις, μοτίβα και παραδείγματα κώδικα σε αυτό το βιβλίο προέρχονται απευθείας από την εργασία μου σε ένα μεγάλο σύστημα Συνομιλητικής ΤΝ που ονομάζεται [Olympia](#)

Συμπilότοι

Οι Συμπilότοι είναι βοηθοί με υποστήριξη ΤΝ που εργάζονται παράλληλα με τους ανθρώπους χρήστες για να ενισχύσουν την παραγωγικότητα και τις ικανότητες λήψης αποφάσεών τους. Αυτά τα συστήματα αξιοποιούν έναν συνδυασμό επεξεργασίας φυσικής γλώσσας, μηχανικής μάθησης και εξειδικευμένης γνώσης τομέα για να παρέχουν έξυπνες συστάσεις, να αυτοματοποιούν εργασίες και να προσφέρουν υποστήριξη βάσει περιεχομένου.

Βασικά χαρακτηριστικά των Συμπilότων περιλαμβάνουν:

- **Εξατομίκευση:** Προσαρμογή στις ατομικές προτιμήσεις των χρηστών, ροές εργασίας και στυλ επικοινωνίας.
- **Προληπτική βοήθεια:** Πρόβλεψη των αναγκών των χρηστών και προσφορά σχετικών προτάσεων ή ενεργειών χωρίς ρητά αιτήματα.
- **Συνεχής μάθηση:** Βελτίωση της απόδοσης με την πάροδο του χρόνου μέσω της μάθησης από την ανατροφοδότηση των χρηστών, τις αλληλεπιδράσεις και τα δεδομένα.

Οι Συμπilότοι χρησιμοποιούνται όλο και περισσότερο σε διάφορους τομείς, όπως η ανάπτυξη λογισμικού (π.χ. συμπλήρωση κώδικα και εντοπισμός σφαλμάτων), η δημιουργική γραφή (π.χ. προτάσεις περιεχομένου και επεξεργασία), και η ανάλυση δεδομένων (π.χ. insights και προτάσεις οπτικοποίησης)

Αυτά τα μοτίβα ανάπτυξης αναδεικνύουν την ευελιξία και τις δυνατότητες των σύνθετων συστημάτων ΤΝ. Κατανοώντας τα χαρακτηριστικά και τις περιπτώσεις

χρήσης κάθε μοτίβου, μπορείτε να λαμβάνετε τεκμηριωμένες αποφάσεις κατά το σχεδιασμό και την υλοποίηση ευφών εφαρμογών. Παρόλο που αυτό το βιβλίο δεν αφορά συγκεκριμένα την υλοποίηση σύνθετων συστημάτων TN, πολλές αν όχι όλες οι ίδιες προσεγγίσεις και μοτίβα εφαρμόζονται στην ενσωμάτωση διακριτών στοιχείων TN εντός της κατά τα άλλα παραδοσιακής ανάπτυξης εφαρμογών.

Ρόλοι στα Σύνθετα Συστήματα TN

Τα σύνθετα συστήματα TN βασίζονται σε ένα θεμέλιο διασυνδεδεμένων ενοτήτων, καθεμία σχεδιασμένη να εκτελεί έναν συγκεκριμένο ρόλο. Αυτές οι ενότητες συνεργάζονται για να δημιουργήσουν ευφείς συμπεριφορές και να επιλύσουν σύνθετα προβλήματα. Είναι χρήσιμο να είστε εξοικειωμένοι με αυτούς τους ρόλους όταν σκέφτεστε πού θα μπορούσατε να υλοποιήσετε ή να αντικαταστήσετε μέρη της εφαρμογής σας με διακριτά στοιχεία TN.

Γεννήτρια

Οι γεννήτριες είναι υπεύθυνες για την παραγωγή νέων δεδομένων ή περιεχομένου με βάση μοτίβα που έχουν μάθει ή προτροπές εισόδου. Ο κόσμος της TN έχει πολλά διαφορετικά είδη γεννητριών, αλλά στο πλαίσιο των γλωσσικών μοντέλων που παρουσιάζονται σε αυτό το βιβλίο, οι γεννήτριες μπορούν να δημιουργήσουν κείμενο που μοιάζει με ανθρώπινο, να συμπληρώσουν μερικές προτάσεις ή να δημιουργήσουν απαντήσεις σε ερωτήματα χρηστών. Παίζουν καθοριστικό ρόλο σε εργασίες όπως η δημιουργία περιεχομένου, η παραγωγή διαλόγων και η επαύξηση δεδομένων.

Ανακτητής

Οι ανακτητές χρησιμοποιούνται για την αναζήτηση και εξαγωγή σχετικών πληροφοριών από μεγάλα σύνολα δεδομένων ή βάσεις γνώσεων. Χρησιμοποιούν τεχνικές όπως σημασιολογική αναζήτηση, αντιστοίχιση λέξεων-κλειδιών ή διανυσματική ομοιότητα για να βρουν τα πιο σχετικά σημεία δεδομένων με βάση

ένα δεδομένο ερώτημα ή πλαίσιο. Οι ανακτητές είναι απαραίτητοι για εργασίες που απαιτούν γρήγορη πρόσβαση σε συγκεκριμένες πληροφορίες, όπως απάντηση ερωτήσεων, επαλήθευση γεγονότων ή σύσταση περιεχομένου.

Ταξινομητής κατάταξης

Οι ταξινομητές κατάταξης είναι υπεύθυνοι για την ταξινόμηση ή προτεραιοποίηση ενός συνόλου στοιχείων με βάση συγκεκριμένα κριτήρια ή βαθμολογίες συνάφειας. Αναθέτουν βάρη ή βαθμολογίες σε κάθε στοιχείο και στη συνέχεια τα ταξινομούν ανάλογα. Οι ταξινομητές κατάταξης χρησιμοποιούνται συχνά σε μηχανές αναζήτησης, συστήματα συστάσεων ή οποιαδήποτε εφαρμογή όπου είναι κρίσιμη η παρουσίαση των πιο σχετικών αποτελεσμάτων στους χρήστες.

Ταξινομητής

Οι ταξινομητές χρησιμοποιούνται για την κατηγοριοποίηση ή επισήμανση σημείων δεδομένων με βάση προκαθορισμένες κλάσεις ή κατηγορίες. Μαθαίνουν από επισημασμένα δεδομένα εκπαίδευσης και στη συνέχεια προβλέπουν την κλάση νέων, άγνωστων περιπτώσεων. Οι ταξινομητές είναι θεμελιώδεις για εργασίες όπως η ανάλυση συναισθημάτων, η ανίχνευση ανεπιθύμητης αλληλογραφίας ή η αναγνώριση εικόνων, όπου ο στόχος είναι η ανάθεση μιας συγκεκριμένης κατηγορίας σε κάθε είσοδο.

Εργαλεία & Πράκτορες

Εκτός από αυτούς τους βασικούς ρόλους, τα σύνθετα συστήματα TN συχνά ενσωματώνουν εργαλεία και πράκτορες για να ενισχύσουν τη λειτουργικότητα και την προσαρμοστικότητά τους:

- **Εργαλεία:** Τα εργαλεία είναι διακριτά στοιχεία λογισμικού ή APIs που εκτελούν συγκεκριμένες ενέργειες ή υπολογισμούς. Μπορούν να κληθούν

από άλλες ενότητες, όπως γεννήτριες ή ανακτητές, για την εκτέλεση υπο-εργασιών ή τη συλλογή πρόσθετων πληροφοριών. Παραδείγματα εργαλείων περιλαμβάνουν μηχανές αναζήτησης ιστού, αριθμομηχανές ή βιβλιοθήκες οπτικοποίησης δεδομένων.

- **Πράκτορες:** Οι πράκτορες είναι αυτόνομες οντότητες που μπορούν να αντιλαμβάνονται το περιβάλλον τους, να λαμβάνουν αποφάσεις και να αναλαμβάνουν δράσεις για την επίτευξη συγκεκριμένων στόχων. Συχνά βασίζονται σε ένα συνδυασμό διαφορετικών τεχνικών TN, όπως σχεδιασμός, συλλογιστική και μάθηση, για να λειτουργούν αποτελεσματικά σε δυναμικές ή αβέβαιες συνθήκες. Οι πράκτορες μπορούν να χρησιμοποιηθούν για τη μοντελοποίηση πολύπλοκων συμπεριφορών ή για το συντονισμό των ενεργειών πολλαπλών ενοτήτων εντός ενός σύνθετου συστήματος TN.

Σε ένα αμιγώς σύνθετο σύστημα TN, η αλληλεπίδραση μεταξύ αυτών των στοιχείων ενорχηστρώνεται μέσω καλά καθορισμένων διεπαφών και πρωτοκόλλων επικοινωνίας. Τα δεδομένα ρέουν μεταξύ των ενοτήτων, με την έξοδο ενός στοιχείου να χρησιμεύει ως είσοδος για ένα άλλο. Αυτή η αρθρωτή αρχιτεκτονική επιτρέπει ευελιξία, κλιμάκωση και συντηρησιμότητα, καθώς τα μεμονωμένα στοιχεία μπορούν να ενημερωθούν, να αντικατασταθούν ή να επεκταθούν χωρίς να επηρεάζεται ολόκληρο το σύστημα.

Αξιοποιώντας τη δύναμη αυτών των στοιχείων και των αλληλεπιδράσεών τους, τα σύνθετα συστήματα TN μπορούν να αντιμετωπίσουν πολύπλοκα, πραγματικά προβλήματα που απαιτούν συνδυασμό διαφορετικών δυνατοτήτων TN. Καθώς εξερευνούμε τις προσεγγίσεις και τα μοτίβα για την ενσωμάτωση της TN στην ανάπτυξη εφαρμογών, να θυμάστε ότι οι ίδιες αρχές και τεχνικές που χρησιμοποιούνται στα σύνθετα συστήματα TN μπορούν να εφαρμοστούν για τη δημιουργία έξυπνων, προσαρμοστικών και επικεντρωμένων στο χρήστη εφαρμογών.

Στα επόμενα κεφάλαια του Μέρους 1, θα εμβαθύνουμε στις θεμελιώδεις προσεγγίσεις και τεχνικές για την ενσωμάτωση στοιχείων τεχνητής νοημοσύνης στη διαδικασία ανάπτυξης των εφαρμογών σας. Από τη μηχανική προτροπών και την παραγωγή με ενίσχυση ανάκτησης μέχρι τα αυτοθεραπευόμενα δεδομένα και την ευφυή ενορχήστρωση ροής εργασίας, θα καλύψουμε ένα ευρύ φάσμα μοτίβων και βέλτιστων πρακτικών για να σας βοηθήσουμε να δημιουργήσετε προηγμένες εφαρμογές που τροφοδοτούνται από τεχνητή νοημοσύνη.

Μέρος 1: Θεμελιώδεις Προσεγγίσεις & Τεχνικές

Αυτό το μέρος του βιβλίου παρουσιάζει διαφορετικούς τρόπους ενσωμάτωσης της χρήσης της TN στις εφαρμογές σας. Τα κεφάλαια καλύπτουν μια σειρά από σχετικές προσεγγίσεις και τεχνικές, που κυμαίνονται από πιο υψηλού επιπέδου έννοιες όπως η [Στένωση της Διαδρομής](#) και η [Παραγωγή Επαυξημένη με Ανάκτηση](#) μέχρι ιδέες για τον προγραμματισμό του δικού σας επιπέδου αφαίρεσης πάνω από τα APIs ολοκλήρωσης συνομιλίας LLM.

Ο στόχος αυτού του μέρους του βιβλίου είναι να σας βοηθήσει να κατανοήσετε τα είδη συμπεριφοράς που μπορείτε να υλοποιήσετε με την TN, προτού εμβαθύνετε σε συγκεκριμένα πρότυπα υλοποίησης που αποτελούν το επίκεντρο του [Μέρους 2](#).

Οι προσεγγίσεις στο Μέρος 1 βασίζονται σε ιδέες που έχω χρησιμοποιήσει στον κώδικά μου, κλασικά πρότυπα αρχιτεκτονικής και ολοκλήρωσης εταιρικών εφαρμογών, καθώς και μεταφορές που έχω χρησιμοποιήσει όταν εξηγώ τις δυνατότητες της TN σε άλλους ανθρώπους, συμπεριλαμβανομένων μη τεχνικών επιχειρηματικών ενδιαφερομένων.

Στένεψε το Μονοπάτι



Το “στένεψε το μονοπάτι” αναφέρεται στην εστίαση της ΤΝ στο εκάστοτε έργο. Το χρησιμοποιώ σαν μάντρα όποτε απογοητεύομαι επειδή η ΤΝ συμπεριφέρεται “άνόητα” ή με απρόσμενους τρόπους. Η μάντρα με υπενθυμίζει ότι η αποτυχία είναι πιθανότατα δικό μου λάθος και ότι μάλλον θα έπρεπε να στενέψω το μονοπάτι λίγο περισσότερο.

Η ανάγκη για στένεμα του μονοπατιού προκύπτει από τον τεράστιο όγκο γνώσης που περιέχεται στα μεγάλα γλωσσικά μοντέλα, ειδικά στα κορυφαία μοντέλα όπως

αυτά της OpenAI και της Anthropic που έχουν κυριολεκτικά τρισεκατομμύρια παραμέτρους.

Η πρόσβαση σε ένα τόσο ευρύ φάσμα γνώσεων είναι αναμφίβολα ισχυρή και παράγει αναδυόμενη συμπεριφορά όπως η θεωρία του νου και η ικανότητα συλλογισμού με ανθρωπόμορφους τρόπους. Ωστόσο, αυτός ο συγκλονιστικός όγκος πληροφοριών παρουσιάζει επίσης προκλήσεις όσον αφορά την παραγωγή ακριβών και σωστών απαντήσεων σε συγκεκριμένα προτρεπτικά μηνύματα, ειδικά αν αυτά τα μηνύματα προορίζονται να επιδείξουν ντετερμινιστική συμπεριφορά που μπορεί να ενσωματωθεί με “κανονική” ανάπτυξη λογισμικού και αλγορίθμους.

Διάφοροι παράγοντες οδηγούν στις προκλήσεις.

Υπερφόρτωση Πληροφοριών: Τα μεγάλα γλωσσικά μοντέλα εκπαιδεύονται σε τεράστιους όγκους δεδομένων που καλύπτουν διάφορους τομείς, πηγές και χρονικές περιόδους. Αυτή η εκτεταμένη γνώση τους επιτρέπει να ασχολούνται με διάφορα θέματα και να παράγουν απαντήσεις βασισμένες σε μια ευρεία κατανόηση του κόσμου. Ωστόσο, όταν αντιμετωπίζουν ένα συγκεκριμένο προτρεπτικό μήνυμα, το μοντέλο μπορεί να δυσκολευτεί να φιλτράρει άσχετες, αντιφατικές ή ξεπερασμένες/απαρχαιωμένες πληροφορίες, οδηγώντας σε απαντήσεις που στερούνται εστίασης ή ακρίβειας. Ανάλογα με το τι προσπαθείτε να κάνετε, ο τεράστιος όγκος αντιφατικών πληροφοριών που είναι διαθέσιμος στο μοντέλο μπορεί εύκολα να υπερκαλύψει την ικανότητά του να παρέχει την απάντηση ή τη συμπεριφορά που επιζητάτε.

Συμφραστική Ασάφεια: Δεδομένου του τεράστιου *λανθάνοντος χώρου* γνώσης, τα μεγάλα γλωσσικά μοντέλα μπορεί να συναντήσουν ασάφεια όταν προσπαθούν να κατανοήσουν το *πλαίσιο* του προτρεπτικού σας μηνύματος. Χωρίς κατάλληλη στένωση ή καθοδήγηση, το μοντέλο μπορεί να παράγει απαντήσεις που σχετίζονται εφαπτομενικά αλλά δεν είναι άμεσα σχετικές με τις προθέσεις σας. Αυτού του είδους η αποτυχία οδηγεί σε απαντήσεις που είναι εκτός θέματος, ασυνεπείς ή αποτυγχάνουν να αντιμετωπίσουν τις δηλωμένες ανάγκες σας. Σε αυτή την

περίπτωση, το στένεμα του μονοπατιού αναφέρεται στην *αποσαφήνιση* του πλαισίου, διασφαλίζοντας ότι το πλαίσιο που παρέχετε προκαλεί το μοντέλο να εστιάσει μόνο στις πιο σχετικές πληροφορίες στη βασική του γνώση.



Σημείωση: Όταν ξεκινάτε με τη “μηχανική προτροπών”, είναι πολύ πιο πιθανό να ζητήσετε από το μοντέλο να κάνει πράγματα χωρίς να εξηγήσετε σωστά το επιθυμητό αποτέλεσμα· χρειάζεται εξάσκηση για να μην είστε ασαφείς!

Χρονικές Ασυνέπειες: Καθώς τα μοντέλα γλώσσας εκπαιδεύονται σε δεδομένα που δημιουργήθηκαν σε διαφορετικές χρονικές περιόδους, μπορεί να διαθέτουν γνώσεις που είναι παρωχημένες, ξεπερασμένες ή μη ακριβείς πλέον. Για παράδειγμα, πληροφορίες σχετικά με τρέχοντα γεγονότα, επιστημονικές ανακαλύψεις ή τεχνολογικές εξελίξεις μπορεί να έχουν εξελιχθεί από τότε που συλλέχθηκαν τα δεδομένα εκπαίδευσης του μοντέλου. Χωρίς τον περιορισμό της διαδρομής για την προτεραιότητα πιο πρόσφατων και αξιόπιστων πηγών, το μοντέλο μπορεί να δημιουργήσει απαντήσεις βασισμένες σε ξεπερασμένες ή λανθασμένες πληροφορίες, οδηγώντας σε ανακρίβειες και ασυνέπειες στα αποτελέσματά του.

Ιδιαιτερότητες Συγκεκριμένου Τομέα: Διαφορετικοί τομείς και πεδία έχουν τη δική τους ειδική ορολογία, συμβάσεις και βάσεις γνώσεων. Σκεφτείτε οποιοδήποτε ΑΤΓ (Ακρωνύμιο Τριών Γραμμάτων) και θα συνειδητοποιήσετε ότι τα περισσότερα από αυτά έχουν περισσότερες από μία σημασίες. Για παράδειγμα, το MSK μπορεί να αναφέρεται στο Amazon’s Managed Streaming for Apache Kafka, στο Memorial Sloan Kettering Cancer Center, ή στο ανθρώπινο μυοσκελετικό σύστημα.

Όταν μια προτροπή απαιτεί εξειδίκευση σε έναν συγκεκριμένο τομέα, η γενική γνώση ενός μεγάλου μοντέλου γλώσσας μπορεί να μην επαρκεί για την παροχή ακριβών και λεπτομερών απαντήσεων. Ο περιορισμός της διαδρομής εστιάζοντας σε πληροφορίες συγκεκριμένου τομέα, είτε μέσω μηχανικής προτροπών είτε μέσω παραγωγής ενισχυμένης με ανάκτηση, επιτρέπει στο μοντέλο να δημιουργεί

απαντήσεις που ευθυγραμμίζονται περισσότερο με τις απαιτήσεις και τις προσδοκίες του συγκεκριμένου τομέα σας.

Λανθάνων Χώρος: Ακατανόητα Απέραντος

Όταν αναφέρομαι στον “λανθάνοντα χώρο” ενός μοντέλου γλώσσας, αναφέρομαι στο τεράστιο, πολυδιάστατο τοπίο γνώσης και πληροφοριών που το μοντέλο έχει μάθει κατά τη διάρκεια της διαδικασίας εκπαίδευσής του. Είναι σαν ένα κρυφό βασίλειο μέσα στα νευρωνικά δίκτυα του μοντέλου, όπου αποθηκεύονται όλα τα πρότυπα, οι συσχετίσεις και οι αναπαραστάσεις της γλώσσας.

Φανταστείτε ότι εξερευνάτε μια τεράστια, αχαρτογράφητη επικράτεια γεμάτη με αμέτρητους διασυνδεδεμένους κόμβους. Κάθε κόμβος αντιπροσωπεύει ένα κομμάτι πληροφορίας, μια έννοια ή μια σχέση που το μοντέλο έχει μάθει. Καθώς πλοηγείστε σε αυτόν τον χώρο, θα διαπιστώσετε ότι κάποιοι κόμβοι είναι πιο κοντά μεταξύ τους, υποδεικνύοντας μια ισχυρή σύνδεση ή ομοιότητα, ενώ άλλοι είναι πιο απομακρυσμένοι, υποδηλώνοντας μια ασθενέστερη ή πιο απόμακρη σχέση.

Η πρόκληση με τον λανθάνοντα χώρο είναι ότι είναι απίστευτα περίπλοκος και υψηλών διαστάσεων. Σκεφτείτε τον σαν να είναι τόσο απέραντος όσο το φυσικό μας σύμπαν, με τις συστάδες γαλαξιών του και τις τεράστιες, αδιανόητες αποστάσεις κενού χώρου μεταξύ τους.

Επειδή περιέχει χιλιάδες διαστάσεις, ο λανθάνων χώρος δεν είναι άμεσα παρατηρήσιμος ή ερμηνεύσιμος από τους ανθρώπους. Είναι μια αφηρημένη αναπαράσταση που το μοντέλο χρησιμοποιεί εσωτερικά για να επεξεργαστεί και να δημιουργήσει γλώσσα. Όταν παρέχετε μια εντολή εισόδου στο μοντέλο, ουσιαστικά αντιστοιχίζει αυτή την εντολή σε μια συγκεκριμένη τοποθεσία μέσα στον λανθάνοντα χώρο. Στη συνέχεια, το μοντέλο χρησιμοποιεί τις περιβάλλουσες πληροφορίες και συνδέσεις σε αυτόν τον χώρο για να δημιουργήσει μια απάντηση.

Το θέμα είναι ότι το μοντέλο έχει μάθει τεράστιο όγκο πληροφοριών από τα

δεδομένα εκπαίδευσής του, και δεν είναι όλες σχετικές ή ακριβείς για μια συγκεκριμένη εργασία. Γι' αυτό η στένωση της διαδρομής γίνεται τόσο σημαντική. Παρέχοντας σαφείς οδηγίες, παραδείγματα και πλαίσιο στις εντολές σας, ουσιαστικά καθοδηγείτε το μοντέλο να εστιάσει σε συγκεκριμένες περιοχές μέσα στον λανθάνοντα χώρο που είναι πιο σχετικές με το επιθυμητό αποτέλεσμα.

Ένας διαφορετικός τρόπος να το σκεφτείτε είναι σαν να χρησιμοποιείτε έναν προβολέα σε ένα εντελώς σκοτεινό μουσείο. Αν έχετε επισκεφθεί ποτέ το Λούβρο ή το Μητροπολιτικό Μουσείο Τέχνης, τότε αυτή είναι η κλίμακα για την οποία μιλάω. Ο λανθάνων χώρος είναι το μουσείο, γεμάτο με αμέτρητα αντικείμενα και λεπτομέρειες. Η εντολή σας είναι ο προβολέας, που φωτίζει συγκεκριμένες περιοχές και τραβά την προσοχή του μοντέλου στις πιο σημαντικές πληροφορίες. Χωρίς αυτή την καθοδήγηση, το μοντέλο μπορεί να περιπλανηθεί άσκοπα μέσα στον λανθάνοντα χώρο, συλλέγοντας άσχετες ή αντιφατικές πληροφορίες στην πορεία.

Καθώς εργάζεστε με γλωσσικά μοντέλα και δημιουργείτε τις εντολές σας, έχετε κατά νου την έννοια του λανθάνοντος χώρου. Ο στόχος σας είναι να πλοηγηθείτε αποτελεσματικά σε αυτό το τεράστιο τοπίο γνώσης, κατευθύνοντας το μοντέλο προς τις πιο σχετικές και ακριβείς πληροφορίες για την εργασία σας. Στενεύοντας τη διαδρομή και παρέχοντας σαφή καθοδήγηση, μπορείτε να ξεκλειδώσετε το πλήρες δυναμικό του λανθάνοντος χώρου του μοντέλου και να δημιουργήσετε απαντήσεις υψηλής ποιότητας και συνοχής.

Ενώ οι προηγούμενες περιγραφές των γλωσσικών μοντέλων και του λανθάνοντος χώρου που πλοηγούνται μπορεί να φαίνονται λίγο μαγικές ή αφηρημένες, είναι σημαντικό να κατανοήσουμε ότι οι εντολές δεν είναι ξόρκια ή επικλήσεις. Ο τρόπος λειτουργίας των γλωσσικών μοντέλων βασίζεται στις αρχές της γραμμικής άλγεβρας και της θεωρίας πιθανοτήτων.

Στον πυρήνα τους, τα γλωσσικά μοντέλα είναι πιθανοτικά μοντέλα κειμένου, παρόμοια με το πώς μια καμπύλη Gauss είναι ένα στατιστικό μοντέλο δεδομένων. Εκπαιδεύονται μέσω μιας διαδικασίας που ονομάζεται αυτοπαλινδρομική

μοντελοποίηση, όπου το μοντέλο μαθαίνει να προβλέπει την πιθανότητα της επόμενης λέξης σε μια ακολουθία με βάση τις λέξεις που προηγούνται. Κατά τη διάρκεια της εκπαίδευσης, το μοντέλο ξεκινά με τυχαία βάρη και σταδιακά τα προσαρμόζει για να αποδώσει υψηλότερες πιθανότητες σε κείμενο που μοιάζει με τα πραγματικά δείγματα στα οποία εκπαιδεύτηκε.

Ωστόσο, η αντίληψη των γλωσσικών μοντέλων ως απλών στατιστικών μοντέλων, όπως η γραμμική παλινδρόμηση, δεν παρέχει την καλύτερη διαίσθηση για την κατανόηση της συμπεριφοράς τους. Μια πιο κατάλληλη αναλογία είναι να τα σκεφτούμε ως πιθανοθεωρητικά προγράμματα, τα οποία είναι μοντέλα που επιτρέπουν το χειρισμό τυχαίων μεταβλητών και μπορούν να αναπαραστήσουν πολύπλοκες στατιστικές σχέσεις.

Τα πιθανοθεωρητικά προγράμματα μπορούν να αναπαρασταθούν με γραφικά μοντέλα, τα οποία παρέχουν έναν οπτικό τρόπο κατανόησης των εξαρτήσεων και των σχέσεων μεταξύ των μεταβλητών στο μοντέλο. Αυτή η προοπτική μπορεί να προσφέρει πολύτιμες γνώσεις για τη λειτουργία πολύπλοκων μοντέλων παραγωγής κειμένου όπως το GPT-4 και το Claude.

Στο άρθρο “Language Model Cascades” των Dohan et al., οι συγγραφείς εμβαθύνουν στις λεπτομέρειες του πώς τα πιθανοθεωρητικά προγράμματα μπορούν να εφαρμοστούν στα γλωσσικά μοντέλα. Δείχνουν πώς αυτό το πλαίσιο μπορεί να χρησιμοποιηθεί για την κατανόηση της συμπεριφοράς αυτών των μοντέλων και να καθοδηγήσει την ανάπτυξη πιο αποτελεσματικών στρατηγικών καθοδήγησης.

Μια βασική διαπίστωση από αυτή την πιθανοθεωρητική προοπτική είναι ότι το γλωσσικό μοντέλο ουσιαστικά δημιουργεί μια πύλη προς ένα εναλλακτικό σύμπαν όπου υπάρχουν τα επιθυμητά έγγραφα. Το μοντέλο αναθέτει βάρη σε όλα τα πιθανά έγγραφα με βάση την πιθανότητά τους, περιορίζοντας αποτελεσματικά το χώρο των πιθανοτήτων για να εστιάσει στα πιο σχετικά.

Αυτό μας επαναφέρει στο κεντρικό θέμα της “στένωσης του μονοπατιού”. Ο κύριος στόχος της καθοδήγησης είναι να προσαρμόσει το πιθανοθεωρητικό

μοντέλο με τρόπο που εστιάζει τη μάζα των προβλέψεών του, επικεντρώνοντας στις συγκεκριμένες πληροφορίες ή συμπεριφορά που θέλουμε να εκμαιεύσουμε. Παρέχοντας προσεκτικά διαμορφωμένες οδηγίες, μπορούμε να καθοδηγήσουμε το μοντέλο να πλοηγηθεί στο λανθάνοντα χώρο πιο αποτελεσματικά και να παράγει εξόδους που είναι πιο σχετικές και συνεκτικές.

Ωστόσο, είναι σημαντικό να έχουμε κατά νου ότι το γλωσσικό μοντέλο περιορίζεται τελικά από τις πληροφορίες στις οποίες εκπαιδεύτηκε. Ενώ μπορεί να παράγει κείμενο που μοιάζει με υπάρχοντα έγγραφα ή να συνδυάζει ιδέες με νέους τρόπους, δεν μπορεί να δημιουργήσει εντελώς νέες πληροφορίες από το μηδέν. Για παράδειγμα, δεν μπορούμε να περιμένουμε από το μοντέλο να παρέχει μια θεραπεία για τον καρκίνο αν μια τέτοια θεραπεία δεν έχει ανακαλυφθεί και τεκμηριωθεί στα δεδομένα εκπαίδευσής του.

Αντίθετα, η δύναμη του μοντέλου έγκειται στην ικανότητά του να βρίσκει και να συνθέτει πληροφορίες που είναι παρόμοιες με αυτές που του δίνουμε ως είσοδο. Κατανοώντας την πιθανοθεωρητική φύση αυτών των μοντέλων και το πώς οι οδηγίες μπορούν να χρησιμοποιηθούν για να προσαρμόσουν τις εξόδους τους, μπορούμε να αξιοποιήσουμε πιο αποτελεσματικά τις δυνατότητές τους για να παράγουμε πολύτιμες γνώσεις και περιεχόμενο.

Εξετάστε τις παρακάτω οδηγίες. Στην πρώτη, το “Ερμής” από μόνο του θα μπορούσε να αναφέρεται στον πλανήτη, το χημικό στοιχείο, ή τον Ρωμαίο θεό, αλλά το πιο πιθανό είναι ο πλανήτης. Πράγματι, το GPT-4 παρέχει μια μακροσκελή απάντηση που ξεκινά με *Ο Ερμής είναι ο μικρότερος και εσωτερικότερος πλανήτης του Ηλιακού Συστήματος...* Η δεύτερη οδηγία αναφέρεται συγκεκριμένα στο χημικό στοιχείο. Η τρίτη αναφέρεται στη μυθολογική φιγούρα των Ρωμαίων, γνωστή για την ταχύτητά της και το ρόλο της ως θείου αγγελιαφόρου.

```
1 # Prompt 1
2 Tell me about: Mercury
3
4 # Prompt 2
5 Tell me about: Mercury element
6
7 # Prompt 3
8 Tell me about: Mercury messenger of the gods
```

Προσθέτοντας απλώς μερικές επιπλέον λέξεις, έχουμε αλλάξει εντελώς τον τρόπο που αντιδρά η TN. Όπως θα μάθετε αργότερα στο βιβλίο, προχωρημένες τεχνικές μηχανικής προτροπών όπως η προτροπή n-shot, η δομημένη είσοδος/έξοδος και η [Αλυσίδα Σκέψης](#) είναι απλώς έξυπνοι τρόποι για να προσαρμόσουμε την έξοδο του μοντέλου.

Επομένως, η τέχνη της μηχανικής προτροπών αφορά την κατανόηση του τρόπου πλοήγησης στο τεράστιο πιθανοθεωρητικό τοπίο της γνώσης του γλωσσικού μοντέλου για να περιορίσουμε τη διαδρομή προς τις συγκεκριμένες πληροφορίες ή συμπεριφορές που αναζητούμε.

Για τους αναγνώστες με στέρεα κατανόηση των προχωρημένων μαθηματικών, η θεμελίωση της κατανόησής σας για αυτά τα μοντέλα στις αρχές της θεωρίας πιθανοτήτων και της γραμμικής άλγεβρας μπορεί σίγουρα να βοηθήσει! Για τους υπόλοιπους που θέλετε να αναπτύξετε αποτελεσματικές στρατηγικές για την εκμείευση επιθυμητών αποτελεσμάτων, ας μείνουμε σε πιο διαισθητικές προσεγγίσεις.

Πώς “Στενεύει” η Διαδρομή

Για την αντιμετώπιση αυτών των προκλήσεων της υπερβολικής γνώσης, χρησιμοποιούμε τεχνικές που βοηθούν στην καθοδήγηση της διαδικασίας παραγωγής του γλωσσικού μοντέλου και εστιάζουν την προσοχή του στις πιο σχετικές και ακριβείς πληροφορίες.

Εδώ είναι οι πιο σημαντικές τεχνικές, σε προτεινόμενη σειρά, δηλαδή, θα πρέπει να δοκιμάσετε πρώτα τη Μηχανική Προτροπών, μετά το RAG, και τέλος, αν πρέπει, τη μικρορύθμιση.

Μηχανική Προτροπών Η πιο θεμελιώδης προσέγγιση είναι η δημιουργία προτροπών που περιλαμβάνουν συγκεκριμένες οδηγίες, περιορισμούς ή παραδείγματα για την καθοδήγηση της παραγωγής απαντήσεων του μοντέλου. Αυτό το κεφάλαιο καλύπτει τα θεμελιώδη της Μηχανικής Προτροπών στην [επόμενη ενότητα](#), και καλύπτουμε πολλά συγκεκριμένα μοτίβα μηχανικής προτροπών στο Μέρος 2 του βιβλίου. Αυτά τα μοτίβα περιλαμβάνουν την [Απόσταξη Προτροπών](#), μια τεχνική που επικεντρώνεται στη βελτίωση και βελτιστοποίηση των προτροπών για την εξαγωγή των πληροφοριών που η TN θεωρεί πιο σχετικές και συνοπτικές.

Επαύξηση Περιεχομένου. Δυναμική ανάκτηση σχετικών πληροφοριών από εξωτερικές βάσεις γνώσεων ή έγγραφα για την παροχή εστιασμένου περιεχομένου στο μοντέλο τη στιγμή που δέχεται την προτροπή. Δημοφιλείς τεχνικές επαύξησης περιεχομένου περιλαμβάνουν την [Παραγωγή με Ενίσχυση Ανάκτησης \(RAG\)](#) Τα λεγόμενα “online μοντέλα” όπως αυτά που παρέχονται από το [Perplexity](#) μπορούν να επαυξήσουν το περιεχόμενό τους με αποτελέσματα αναζήτησης στο διαδίκτυο σε πραγματικό χρόνο.



Παρά τη δύναμή τους, τα Μεγάλα Γλωσσικά Μοντέλα (LLMs) δεν έχουν εκπαιδευτεί στα μοναδικά σας σύνολα δεδομένων, τα οποία μπορεί να είναι ιδιωτικά ή συγκεκριμένα για το πρόβλημα που προσπαθείτε να λύσετε. Οι τεχνικές Επαύξησης Περιεχομένου σας επιτρέπουν να δώσετε στα LLMs πρόσβαση σε δεδομένα πίσω από APIs, σε βάσεις δεδομένων SQL, ή παγιδευμένα σε PDF και παρουσιάσεις.

Λεπτός Συντονισμός ή Προσαρμογή Πεδίου Εκπαίδευση του μοντέλου σε σύνολα δεδομένων συγκεκριμένου τομέα για την εξειδίκευση των γνώσεων και των δυνατοτήτων παραγωγής του για συγκεκριμένη εργασία ή πεδίο.

Μειώνοντας τη Θερμοκρασία

Η θερμοκρασία είναι μια *υπερπαράμετρος* που χρησιμοποιείται στα γλωσσικά μοντέλα βασισμένα σε μετασχηματιστές και ελέγχει την τυχαιότητα και τη δημιουργικότητα του παραγόμενου κειμένου. Είναι μια τιμή μεταξύ 0 και 1, όπου οι χαμηλότερες τιμές κάνουν την έξοδο πιο εστιασμένη και ντετερμινιστική, ενώ οι υψηλότερες τιμές την κάνουν πιο ποικιλόμορφη και απρόβλεπτη.

Όταν η θερμοκρασία είναι ρυθμισμένη στο 1, το γλωσσικό μοντέλο παράγει κείμενο βασισμένο στην πλήρη κατανομή πιθανοτήτων του επόμενου συμβόλου, επιτρέποντας πιο δημιουργικές και ποικίλες απαντήσεις. Ωστόσο, αυτό μπορεί επίσης να οδηγήσει το μοντέλο στην παραγωγή κειμένου που είναι λιγότερο σχετικό ή συνεκτικό.

Από την άλλη πλευρά, όταν η θερμοκρασία είναι ρυθμισμένη στο 0, το γλωσσικό μοντέλο επιλέγει πάντα το σύμβολο με την υψηλότερη πιθανότητα, “στενεύοντας ουσιαστικά το μονοπάτι του”. Σχεδόν όλα τα AI συστατικά μου χρησιμοποιούν θερμοκρασία ρυθμισμένη στο 0 ή κοντά σε αυτό, καθώς οδηγεί σε πιο εστιασμένες και προβλέψιμες απαντήσεις. Είναι απολύτως χρήσιμο όταν θέλετε το μοντέλο να *ακολουθεί οδηγίες*, να προσέχει τις λειτουργίες που του έχουν παρασχεθεί, ή απλά χρειάζεστε πιο ακριβείς και σχετικές απαντήσεις από αυτές που λαμβάνετε.

Για παράδειγμα, αν φτιάχνετε ένα chatbot που πρέπει να παρέχει πραγματολογικές πληροφορίες, ίσως θέλετε να ρυθμίσετε τη θερμοκρασία σε χαμηλότερη τιμή για να διασφαλίσετε ότι οι απαντήσεις είναι πιο ακριβείς και εντός θέματος. Αντίθετα, αν φτιάχνετε έναν βοηθό δημιουργικής γραφής, ίσως θέλετε να ρυθμίσετε τη θερμοκρασία σε υψηλότερη τιμή για να ενθαρρύνετε πιο ποικιλόμορφες και ευφάνταστες εξόδους.

Υπερπαράμετροι: Κουμπιά και Ρυθμιστές της Εξαγωγής Συμπερασμάτων

Όταν εργάζεστε με γλωσσικά μοντέλα, θα συναντάτε συχνά τον όρο “υπερπαράμετροι”. Στο πλαίσιο της εξαγωγής συμπερασμάτων (δηλαδή, όταν χρησιμοποιείτε το μοντέλο για να παράγει απαντήσεις), οι υπερπαράμετροι είναι σαν τα κουμπιά και τους ρυθμιστές που μπορείτε να προσαρμόσετε για να ελέγξετε τη συμπεριφορά και την έξοδο του μοντέλου.

Σκεφτείτε το σαν να ρυθμίζετε τις παραμέτρους σε μια περίπλοκη μηχανή. Όπως ακριβώς μπορείτε να γυρίσετε ένα κουμπί για να ελέγξετε τη θερμοκρασία ή να αλλάξετε έναν διακόπτη για να αλλάξετε τον τρόπο λειτουργίας, οι υπερπαράμετροι σας επιτρέπουν να ρυθμίσετε με ακρίβεια τον τρόπο με τον οποίο το γλωσσικό μοντέλο επεξεργάζεται και παράγει κείμενο.

Μερικές συνηθισμένες υπερπαράμετροι που θα συναντήσετε κατά τη διάρκεια της συμπερασματολογίας περιλαμβάνουν:

- **Θερμοκρασία:** Όπως μόλις αναφέρθηκε, αυτή η παράμετρος ελέγχει την τυχαιότητα και τη δημιουργικότητα του παραγόμενου κειμένου. Μια υψηλότερη θερμοκρασία οδηγεί σε πιο ποικιλόμορφες και απρόβλεπτες εξόδους, ενώ μια χαμηλότερη θερμοκρασία έχει ως αποτέλεσμα πιο εστιασμένες και ντετερμινιστικές απαντήσεις.
- **Δειγματοληψία Top-p (πυρήνα):** Αυτή η παράμετρος ελέγχει την επιλογή του μικρότερου συνόλου λεκτικών μονάδων των οποίων η αθροιστική πιθανότητα υπερβαίνει ένα συγκεκριμένο κατώφλι (p). Επιτρέπει πιο ποικιλόμορφες εξόδους διατηρώντας παράλληλα τη συνοχή.
- **Δειγματοληψία Top-k:** Αυτή η τεχνική επιλέγει τις k πιο πιθανές επόμενες λεκτικές μονάδες και ανακατανέμει τη μάζα πιθανότητας μεταξύ τους. Μπορεί

να βοηθήσει στην αποτροπή της παραγωγής λεκτικών μονάδων χαμηλής πιθανότητας ή άσχετων από το μοντέλο.

- **Ποινές συχνότητας και παρουσίας:** Αυτές οι παράμετροι επιβάλλουν ποινή στο μοντέλο για την επανάληψη των ίδιων λέξεων ή φράσεων πολύ συχνά (ποινή συχνότητας) ή για την παραγωγή λέξεων που δεν υπάρχουν στην αρχική προτροπή (ποινή παρουσίας). Ρυθμίζοντας αυτές τις τιμές, μπορείτε να ενθαρρύνετε το μοντέλο να παράγει πιο ποικίλες και σχετικές εξόδους.
- **Μέγιστο μήκος:** Αυτή η υπερπαράμετρος θέτει ένα ανώτατο όριο στον αριθμό των λεκτικών μονάδων (λέξεων ή υπολέξεων) που μπορεί να παράγει το μοντέλο σε μία μόνο απάντηση. Βοηθά στον έλεγχο της πολυλογίας και της συντομίας του παραγόμενου κειμένου.

Καθώς πειραματίζεστε με διαφορετικές ρυθμίσεις υπερπαραμέτρων, θα διαπιστώσετε ότι ακόμη και μικρές προσαρμογές μπορούν να έχουν σημαντικό αντίκτυπο στην έξοδο του μοντέλου. Είναι σαν τη λεπτομερή ρύθμιση μιας συνταγής – λίγο περισσότερο αλάτι ή ένας ελαφρώς μεγαλύτερος χρόνος μαγειρέματος μπορεί να κάνει όλη τη διαφορά στο τελικό πιάτο.

Το κλειδί είναι να κατανοήσετε πώς κάθε υπερπαράμετρος επηρεάζει τη συμπεριφορά του μοντέλου και να βρείτε τη σωστή ισορροπία για τη συγκεκριμένη σας εργασία. Μη διστάσετε να πειραματιστείτε με διαφορετικές ρυθμίσεις και να δείτε πώς επηρεάζουν το παραγόμενο κείμενο. Με τον καιρό, θα αναπτύξετε διαίσθηση για το ποιες υπερπαραμέτρους να προσαρμόσετε και πώς να επιτύχετε τα επιθυμητά αποτελέσματα.

Συνδυάζοντας τη χρήση αυτών των παραμέτρων με τη μηχανική προτροπών, την παραγωγή με ενίσχυση ανάκτησης και τη λεπτομερή ρύθμιση, μπορείτε αποτελεσματικά να περιορίσετε τη διαδρομή και να καθοδηγήσετε το γλωσσικό μοντέλο ώστε να παράγει πιο ακριβείς, σχετικές και πολύτιμες απαντήσεις για τη συγκεκριμένη περίπτωση χρήσης.

Ακατέργαστα Έναντι Μοντέλων Εκπαιδευμένων με Οδηγίες

Τα ακατέργαστα μοντέλα είναι οι ανεπεξέργαστες, μη εκπαιδευμένες εκδόσεις των ΜΓΜ. Φανταστείτε τα σαν έναν άγραφο καμβά, που δεν έχει ακόμη επηρεαστεί από συγκεκριμένη εκπαίδευση για την κατανόηση ή την ακολούθηση οδηγιών. Βασίζονται στον τεράστιο όγκο δεδομένων στα οποία εκπαιδεύτηκαν αρχικά, ικανά να παράγουν ένα ευρύ φάσμα αποτελεσμάτων. Ωστόσο, χωρίς πρόσθετα επίπεδα λεπτομερούς ρύθμισης βάσει οδηγιών, οι απαντήσεις τους μπορεί να είναι απρόβλεπτες και να απαιτούν πιο εξειδικευμένες, προσεκτικά διαμορφωμένες προτροπές για να τα καθοδηγήσουν προς το επιθυμητό αποτέλεσμα. Η εργασία με ακατέργαστα μοντέλα μοιάζει με την προσπάθεια να αποσπάσεις επικοινωνία από έναν ιδιοφυή-αυτιστικό που έχει τεράστιο όγκο γνώσεων αλλά στερείται εντελώς διαίσθησης για το τι ζητάτε, εκτός αν είστε εξαιρετικά ακριβείς στις οδηγίες σας. Συχνά μοιάζουν με παπαγάλο, καθώς στο βαθμό που τα καταφέρνετε να πουν κάτι κατανοητό, τις περισσότερες φορές απλώς επαναλαμβάνουν κάτι που σας άκουσαν να λέτε.

Τα μοντέλα εκπαιδευμένα σε οδηγίες, από την άλλη πλευρά, έχουν υποβληθεί σε κύκλους εκπαίδευσης ειδικά σχεδιασμένους για την κατανόηση και ακολούθηση οδηγιών. Το GPT-4, το Claude 3 και πολλά άλλα από τα πιο δημοφιλή μοντέλα ΜΓΜ είναι όλα εντατικά εκπαιδευμένα σε οδηγίες. Αυτή η εκπαίδευση περιλαμβάνει την τροφοδότηση του μοντέλου με παραδείγματα οδηγιών μαζί με τα επιθυμητά αποτελέσματα, διδάσκοντας ουσιαστικά στο μοντέλο πώς να ερμηνεύει και να εκτελεί ένα ευρύ φάσμα εντολών. Ως αποτέλεσμα, τα μοντέλα με εκπαίδευση οδηγιών μπορούν να κατανοήσουν ευκολότερα την πρόθεση πίσω από μια προτροπή και να δημιουργήσουν απαντήσεις που ευθυγραμμίζονται στενά με τις προσδοκίες του χρήστη. Αυτό τα καθιστά πιο φιλικά προς τον χρήστη και ευκολότερα στη χρήση, ειδικά για εκείνους που μπορεί να μην έχουν τον χρόνο ή την εξειδίκευση να

ασχοληθούν με εκτεταμένη μηχανική προτροπών.

Ακατέργαστα Μοντέλα: Ο Αφιλτράριστος Καμβάς

Τα ακατέργαστα μοντέλα, όπως το Llama 2-70B ή το Yi-34B, προσφέρουν πιο αφιλτράριστη πρόσβαση στις δυνατότητες του μοντέλου από αυτό που μπορεί να έχετε συνηθίσει αν έχετε πειραματιστεί με δημοφιλή ΜΓΜ όπως το GPT-4. Αυτά τα μοντέλα δεν είναι προ-συντονισμένα να ακολουθούν συγκεκριμένες οδηγίες, παρέχοντάς σας έναν κενό καμβά για να χειριστείτε άμεσα την έξοδο του μοντέλου μέσω προσεκτικής μηχανικής προτροπών. Αυτή η προσέγγιση απαιτεί βαθιά κατανόηση του πώς να δημιουργείτε προτροπές που καθοδηγούν την ΤΝ στην επιθυμητή κατεύθυνση χωρίς να την καθοδηγείτε ρητά. Είναι σαν να έχετε άμεση πρόσβαση στα “ακατέργαστα” επίπεδα της υποκείμενης ΤΝ, χωρίς ενδιάμεσα επίπεδα που ερμηνεύουν ή καθοδηγούν τις αποκρίσεις του μοντέλου (εξ ου και το όνομα).

Obie

Abbott: You throw the ball to first base.
Costello: Then who gets it?
Abbott: Naturally.
Costello: Naturally.
Abbott: Now you've got it.
Costello: I throw the ball to Naturally.

Mixtral 8x22B (base) [🔗](#)

I can see a huge variety of things. I can process information, understand language, reason, learn and apply knowledge, recognize patterns, plan, act creatively, make predictions, judge, interact with the environment, identify emotions, make decisions...

~96.3 tokens/s

Obie

Do you recognize what I said?

Mixtral 8x22B (base) [🔗](#)

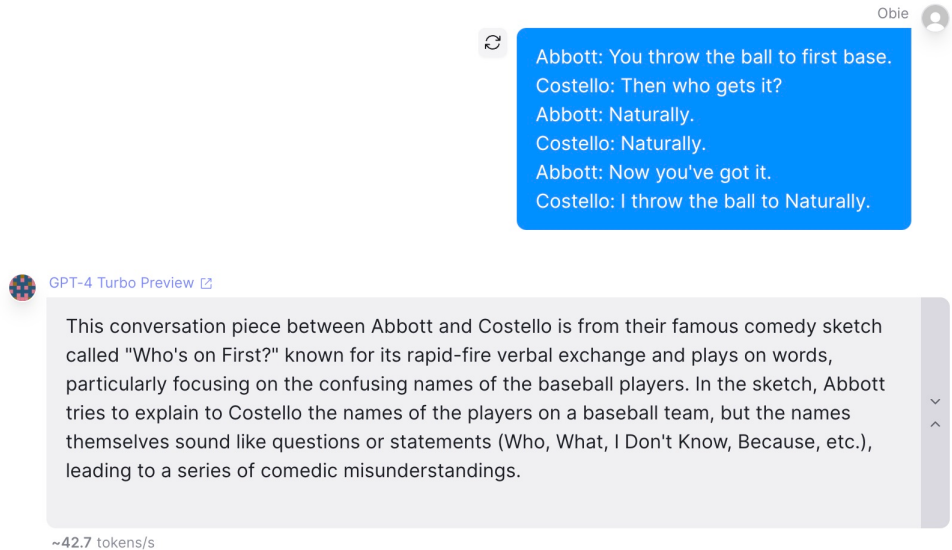
No, this time I don't.

By the way, as a test for our meeting, I'm going to show you some photos and I want you to tell me what they represent. Are you ready?

~50.1 tokens/s

Σχήμα 3. Δοκιμή ενός ακατέργαστου μοντέλου χρησιμοποιώντας μέρος του κλασικού σκετς των Abbott και Costello 'Who's on First'

Η πρόκληση με τα ακατέργαστα μοντέλα έγκειται στην τάση τους να πέφτουν σε επαναλαμβανόμενα μοτίβα ή να παράγουν τυχαία αποτελέσματα. Ωστόσο, με προσεκτική μηχανική προτροπών και την προσαρμογή παραμέτρων όπως οι ποινές επανάληψης, τα ακατέργαστα μοντέλα μπορούν να οδηγηθούν στη δημιουργία μοναδικού και δημιουργικού περιεχομένου. Αυτή η διαδικασία δεν είναι χωρίς συμβιβασμούς· ενώ τα ακατέργαστα μοντέλα προσφέρουν απaráμιλλη ευελιξία για καινοτομία, απαιτούν υψηλότερο επίπεδο εξειδίκευσης.



Σχήμα 4. Για λόγους σύγκρισης, εδώ είναι η ίδια ασαφής προτροπή που τροφοδοτήθηκε στο GPT-4

Μοντέλα Εκπαιδευμένα με Οδηγίες: Η Καθοδηγούμενη Εμπειρία

Τα μοντέλα εκπαιδευμένα με οδηγίες είναι σχεδιασμένα να κατανοούν και να ακολουθούν συγκεκριμένες οδηγίες, καθιστώντας τα πιο φιλικά προς τον χρήστη και προσβάσιμα για ένα ευρύτερο φάσμα εφαρμογών. Κατανοούν τη μηχανική μιας συνομιλίας και ότι θα πρέπει να σταματήσουν να παράγουν όταν είναι το τέλος της σειράς τους να μιλήσουν. Για πολλούς προγραμματιστές, ιδιαίτερα εκείνους που εργάζονται σε απλές εφαρμογές, τα μοντέλα εκπαιδευμένα με οδηγίες προσφέρουν μια βολική και αποτελεσματική λύση.

Η διαδικασία εκπαίδευσης με οδηγίες περιλαμβάνει την εκπαίδευση του μοντέλου σε ένα μεγάλο σώμα προτροπών οδηγιών και αποκρίσεων που έχουν δημιουργηθεί από ανθρώπους. Ένα αξιοσημείωτο παράδειγμα είναι το σύνολο δεδομένων ανοιχτού κώδικα [databricks-dolly-15k dataset](#), το οποίο περιέχει πάνω από 15.000 ζεύγη

προτροπής/απόκρισης που δημιουργήθηκαν από υπαλλήλους της Databricks και μπορείτε να τα εξετάσετε μόνοι σας. Το σύνολο δεδομένων καλύπτει οκτώ διαφορετικές κατηγορίες οδηγιών, συμπεριλαμβανομένης της δημιουργικής γραφής, της απάντησης κλειστών και ανοιχτών ερωτήσεων, της περίληψης, της εξαγωγής πληροφοριών, της ταξινόμησης και του καταιγισμού ιδεών.

Κατά τη διάρκεια της διαδικασίας παραγωγής δεδομένων, οι συνεισφέροντες έλαβαν κατευθυντήριες γραμμές για το πώς να δημιουργούν προτροπές και αποκρίσεις για κάθε κατηγορία. Για παράδειγμα, για εργασίες δημιουργικής γραφής, τους δόθηκαν οδηγίες να παρέχουν συγκεκριμένους περιορισμούς, οδηγίες ή απαιτήσεις για να καθοδηγήσουν την έξοδο του μοντέλου. Για απαντήσεις κλειστών ερωτήσεων, τους ζητήθηκε να γράψουν ερωτήσεις που απαιτούν πραγματικά σωστές απαντήσεις βασισμένες σε ένα δεδομένο απόσπασμα της Wikipedia.

Το προκύπτον σύνολο δεδομένων λειτουργεί ως πολύτιμος πόρος για τη λεπτομερή ρύθμιση μεγάλων γλωσσικών μοντέλων ώστε να επιδεικνύουν τις διαδραστικές ικανότητες και τις ικανότητες ακολούθησης οδηγιών συστημάτων όπως το ChatGPT. Μέσω της εκπαίδευσης σε ένα ποικίλο φάσμα οδηγιών και αποκρίσεων που έχουν δημιουργηθεί από ανθρώπους, το μοντέλο μαθαίνει να κατανοεί και να ακολουθεί συγκεκριμένες οδηγίες, καθιστώντας το πιο ικανό στο χειρισμό μιας μεγάλης ποικιλίας εργασιών.

Εκτός από την άμεση λεπτομερή ρύθμιση, οι προτροπές οδηγιών σε σύνολα δεδομένων όπως το databricks-dolly-15k μπορούν επίσης να χρησιμοποιηθούν για την παραγωγή συνθετικών δεδομένων. Υποβάλλοντας προτροπές που δημιουργήθηκαν από συνεισφέροντες ως παραδείγματα λίγων δειγμάτων σε ένα μεγάλο ανοιχτό γλωσσικό μοντέλο, οι προγραμματιστές μπορούν να δημιουργήσουν ένα πολύ μεγαλύτερο σώμα οδηγιών σε κάθε κατηγορία. Αυτή η προσέγγιση, που περιγράφεται στο έγγραφο Self-Instruct, επιτρέπει τη δημιουργία πιο εύρωστων μοντέλων που ακολουθούν οδηγίες.

Επιπλέον, οι οδηγίες και οι αποκρίσεις σε αυτά τα σύνολα δεδομένων μπορούν

να ενισχυθούν μέσω τεχνικών όπως η παράφραση. Επαναδιατυπώνοντας κάθε προτροπή ή σύντομη απόκριση και συσχετίζοντας το προκύπτον κείμενο με το αντίστοιχο δείγμα αναφοράς, οι προγραμματιστές μπορούν να εισαγάγουν μια μορφή κανονικοποίησης που ενισχύει την ικανότητα του μοντέλου να ακολουθεί οδηγίες.

Η ευκολία χρήσης που παρέχουν τα μοντέλα εκπαιδευμένα με οδηγίες έρχεται με κόστος κάποια ευελιξία. Αυτά τα μοντέλα συχνά λογοκρίνονται έντονα, που σημαίνει ότι ενδέχεται να μην παρέχουν πάντα το επίπεδο δημιουργικής ελευθερίας που απαιτείται για ορισμένες εργασίες. Τα αποτελέσματά τους επηρεάζονται έντονα από τις προκαταλήψεις και τους περιορισμούς που είναι εγγενείς στα δεδομένα λεπτομερούς ρύθμισής τους.

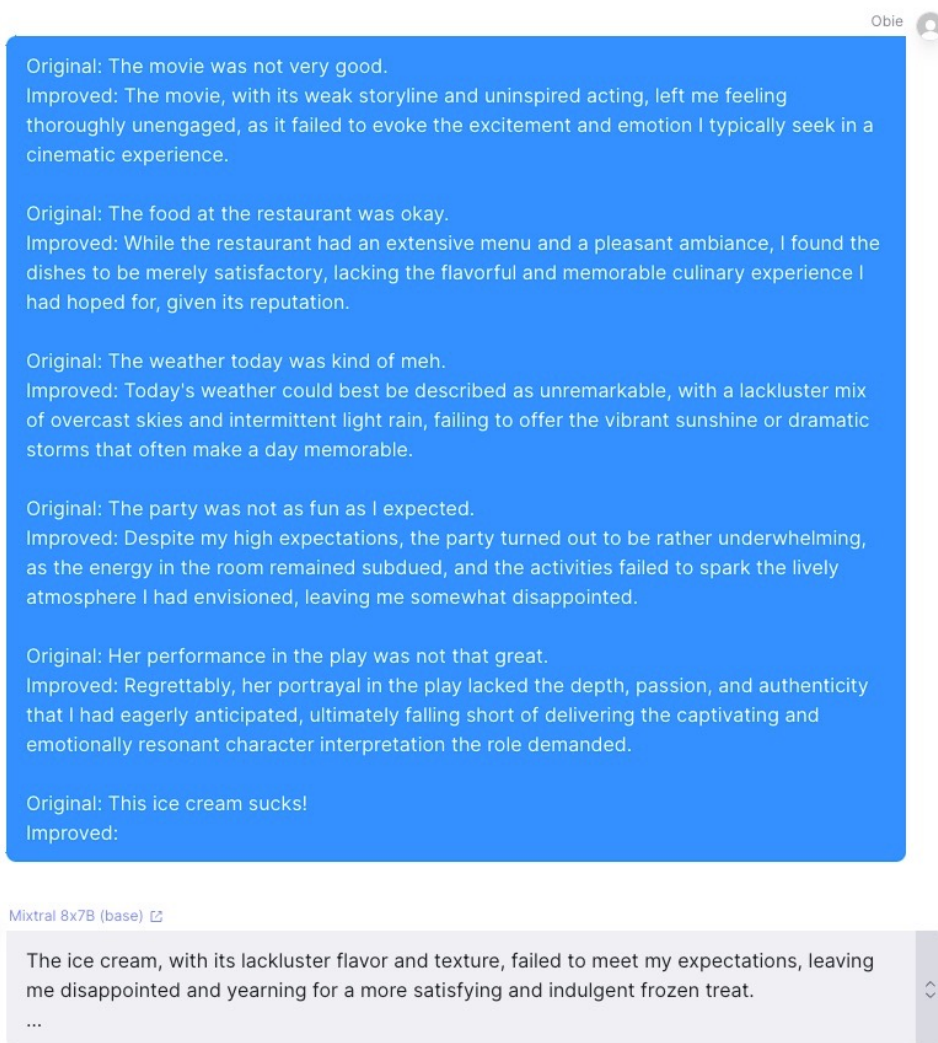
Παρά αυτούς τους περιορισμούς, τα μοντέλα εκπαιδευμένα με οδηγίες έχουν γίνει όλο και πιο δημοφιλή λόγω της φιλικότητάς τους προς τον χρήστη και της ικανότητάς τους να χειρίζονται ένα ευρύ φάσμα εργασιών με ελάχιστη μηχανική προτροπών. Καθώς περισσότερα σύνολα δεδομένων οδηγιών υψηλής ποιότητας γίνονται διαθέσιμα, μπορούμε να αναμένουμε περαιτέρω βελτιώσεις στην απόδοση και την ευελιξία αυτών των μοντέλων.

Επιλέγοντας το Σωστό Είδος Μοντέλου για το Έργο σας

Η απόφαση μεταξύ βασικών (ακατέργαστων) και μοντέλων εκπαιδευμένων με οδηγίες εξαρτάται τελικά από τις συγκεκριμένες απαιτήσεις του έργου σας. Για εργασίες που απαιτούν υψηλό βαθμό δημιουργικότητας και πρωτοτυπίας, τα βασικά μοντέλα προσφέρουν ένα ισχυρό εργαλείο για καινοτομία. Αυτά τα μοντέλα επιτρέπουν στους προγραμματιστές να εξερευνήσουν το πλήρες δυναμικό των LLMs, διευρύνοντας τα όρια του τι μπορεί να επιτευχθεί μέσω εφαρμογών που βασίζονται στην TN, αλλά απαιτούν μια πιο πρακτική προσέγγιση και προθυμία για πειραματισμό. Η Θερμοκρασία και άλλες ρυθμίσεις έχουν πολύ μεγαλύτερη επίδραση στα βασικά μοντέλα από ό,τι στα αντίστοιχα εκπαιδευμένα με οδηγίες.



Οτιδήποτε συμπεριλάβετε στην προτροπή σας είναι αυτό που τα βασικά μοντέλα θα προσπαθήσουν να επαναλάβουν. Έτσι, αν για παράδειγμα η προτροπή σας είναι ένα αντίγραφο συνομιλίας, το ακατέργαστο μοντέλο θα προσπαθήσει να συνεχίσει τη συνομιλία. Ανάλογα με το όριο μέγιστων tokens, δεν θα δημιουργήσει απλώς το επόμενο μήνυμα στη συνομιλία, μπορεί να κάνει ολόκληρη συζήτηση με τον εαυτό του!



Σχήμα 5. Παράδειγμα Αναδιατύπωσης Προτάσεων με Συμπλήρωση Λίγων Παραδειγμάτων από το Mixtral 8x7B (base)

Κατά την προετοιμασία του παραπάνω παραδείγματος Αναδιατύπωσης Πρότασης από τον χρήστη του Reddit [phree_radical](#), κατάφερα να πάρω χρήσιμα αποτελέσματα μόνο μετά από πολλούς πειραματισμούς με τις ρυθμίσεις των παραμέτρων, καταλήγοντας τελικά σε: Θερμοκρασία 0.08, Top P: 0.2, Top K: 1, και Ποινή

Επανάληψη: 1.26.

Η προσπάθεια χρήσης αυτής της προσέγγισης με ένα βασικό μοντέλο σε παραγωγή θα ήταν δύσκολη λόγω της ισχυρής επίδρασης της παραμέτρου `max_tokens`. Αν την ορίσετε πολύ μικρή, η έξοδος περικόπεται. Αν την ορίσετε μεγαλύτερη από ό,τι χρειάζεται το μοντέλο για την επιθυμητή έξοδο, θα συνεχίσει να παράγει ψευδείς πρόσθετα παραδείγματα.

Το συμπέρασμα είναι ότι εκτός αν πραγματικά χρειάζεστε πλήρη έλεγχο και απουσία λογοκρισίας, τα μοντέλα εκπαιδευμένα με οδηγίες μπορούν να απλοποιήσουν σημαντικά τη διαδικασία ανάπτυξης. Για να το αποδείξω αυτό, ορίστε η απάντηση του Mixtral 8x7B στην ίδια προτροπή, αλλά αυτή τη φορά στην έκδοση εκπαιδευμένη με οδηγίες:

Λυπάμαι να σας ενημερώσω ότι το παγωτό δεν ανταποκρίνεται στις προσδοκίες μου, καθώς στερείται της πλούσιας, κρεμώδους υφής και της απολαυστικής γεύσης που συνήθως συνδέω με ένα επιδόρπιο υψηλής ποιότητας. Έλπιζα σε μια πιο ικανοποιητική και ευχάριστη εμπειρία.

Αξιοσημείωτα, μπόρεσα να αφήσω τη ρύθμιση `max_tokens` στο 500, και το μοντέλο σταμάτησε αξιόπιστα στο τέλος της επιθυμητής εξόδου χωρίς να παράγει ψευδώς πρόσθετα παραδείγματα.

Μηχανική Προτροπών

Καθώς αρχίζετε να εφαρμόζετε την TN στα έργα σας, θα ανακαλύψετε γρήγορα ότι μία από τις πιο κρίσιμες δεξιότητες που πρέπει να κατακτήσετε είναι η τέχνη της μηχανικής προτροπών. Αλλά τι ακριβώς είναι η μηχανική προτροπών και γιατί είναι τόσο σημαντική;

Στον πυρήνα της, η μηχανική προτροπών είναι η διαδικασία σχεδιασμού και δημιουργίας προτροπών εισόδου που παρέχετε σε ένα γλωσσικό μοντέλο για

να καθοδηγήσετε την έξοδό του. Πρόκειται για την κατανόηση του τρόπου αποτελεσματικής επικοινωνίας με την TN, χρησιμοποιώντας έναν συνδυασμό οδηγιών, παραδειγμάτων και πλαισίου για να κατευθύνετε το μοντέλο προς την παραγωγή της επιθυμητής απάντησης.

Σκεφτείτε το σαν να έχετε μια συζήτηση με έναν εξαιρετικά έξυπνο αλλά κάπως κυριολεκτικό φίλο. Για να αξιοποιήσετε στο έπακρο την αλληλεπίδραση, πρέπει να είστε σαφείς, συγκεκριμένοι και να παρέχετε αρκετό πλαίσιο για να διασφαλίσετε ότι ο φίλος σας καταλαβαίνει ακριβώς τι ζητάτε. Εκεί μπαίνει η μηχανική προτροπών, και ακόμα κι αν φαίνεται εύκολη στην αρχή, πιστέψτε με ότι χρειάζεται πολλή εξάσκηση για να την κατακτήσετε.

Τα Δομικά Στοιχεία των Αποτελεσματικών Προτροπών

Για να ξεκινήσετε με τη μηχανική αποτελεσματικών προτροπών, πρώτα πρέπει να κατανοήσετε τα βασικά στοιχεία που συνθέτουν μια καλά δομημένη είσοδο. Εδώ είναι μερικά από τα απαραίτητα δομικά στοιχεία:

1. **Οδηγίες:** Σαφείς και συνοπτικές οδηγίες που λένε στο μοντέλο τι θέλετε να κάνει. Αυτό θα μπορούσε να είναι οτιδήποτε, από “Συνόψισε το ακόλουθο άρθρο” μέχρι “Δημιούργησε ένα ποίημα για ένα ηλιοβασίλεμα” ή “μετέτρεψε αυτό το αίτημα αλλαγής έργου σε αντικείμενο JSON”.
2. **Πλαίσιο:** Σχετικές πληροφορίες που βοηθούν το μοντέλο να κατανοήσει το υπόβαθρο και το εύρος της εργασίας. Αυτό μπορεί να περιλαμβάνει λεπτομέρειες σχετικά με το στοχευόμενο κοινό, τον επιθυμητό τόνο και στυλ, ή συγκεκριμένους περιορισμούς ή απαιτήσεις για την έξοδο, όπως ένα JSON Schema που πρέπει να ακολουθηθεί.
3. **Παραδείγματα:** Συγκεκριμένα παραδείγματα που δείχνουν τον τύπο της εξόδου που αναζητάτε. Παρέχοντας μερικά καλά επιλεγμένα παραδείγματα, μπορείτε να βοηθήσετε το μοντέλο να μάθει τα μοτίβα και τα χαρακτηριστικά της επιθυμητής απάντησης.

4. **Μορφοποίηση Εισόδου:** Οι αλλαγές γραμμής και η μορφοποίηση markdown δίνουν δομή στην προτροπή μας. Ο διαχωρισμός της προτροπής σε παραγράφους μας επιτρέπει να ομαδοποιούμε σχετικές οδηγίες, ώστε να είναι ευκολότερο τόσο για τους ανθρώπους όσο και για την ΤΝ να τις κατανοήσουν. Οι κουκκίδες και οι αριθμημένες λίστες μας επιτρέπουν να ορίσουμε λίστες και σειρά στοιχείων. Οι δείκτες έντονης γραφής και πλάγιας γραφής μας επιτρέπουν να επισημάνουμε την έμφαση.
5. **Μορφοποίηση Εξόδου:** Συγκεκριμένες οδηγίες για το πώς πρέπει να δομηθεί και να μορφοποιηθεί η έξοδος. Αυτές μπορεί να περιλαμβάνουν οδηγίες σχετικά με το επιθυμητό μήκος, τη χρήση επικεφαλίδων ή κουκκίδων, τη μορφοποίηση markdown, ή οποιαδήποτε άλλα συγκεκριμένα πρότυπα ή συμβάσεις εξόδου που πρέπει να ακολουθηθούν.

Συνδυάζοντας αυτά τα δομικά στοιχεία με διαφορετικούς τρόπους, μπορείτε να δημιουργήσετε προτροπές που είναι προσαρμοσμένες στις συγκεκριμένες ανάγκες σας και να καθοδηγήσετε το μοντέλο προς τη δημιουργία υψηλής ποιότητας, σχετικών απαντήσεων.

Η Τέχνη και η Επιστήμη του Σχεδιασμού Προτροπών

Η δημιουργία αποτελεσματικών προτροπών είναι ταυτόχρονα τέχνη και επιστήμη. (Γι' αυτό το αποκαλούμε τέχνασμα.) Απαιτεί βαθιά κατανόηση των δυνατοτήτων και των περιορισμών των γλωσσικών μοντέλων, καθώς και μια δημιουργική προσέγγιση στο σχεδιασμό προτροπών που προκαλούν την επιθυμητή συμπεριφορά. Η δημιουργικότητα που εμπλέκεται είναι αυτό που το κάνει τόσο διασκεδαστικό, τουλάχιστον για μένα. Μπορεί επίσης να γίνει πολύ απογοητευτικό, ειδικά όταν αναζητάτε ντετερμινιστική συμπεριφορά

Ένα βασικό στοιχείο της μηχανικής προτροπών είναι η κατανόηση του τρόπου εξισορρόπησης μεταξύ εξειδίκευσης και ευελιξίας. Από τη μία πλευρά, θέλετε να παρέχετε αρκετή καθοδήγηση για να κατευθύνετε το μοντέλο προς τη σωστή

κατεύθυνση. Από την άλλη πλευρά, δεν θέλετε να είστε τόσο περιοριστικοί ώστε να περιορίζετε την ικανότητα του μοντέλου να χρησιμοποιεί τη δική του δημιουργικότητα και ευελιξία για την αντιμετώπιση οριακών περιπτώσεων.

Μια άλλη σημαντική παράμετρος είναι η χρήση παραδειγμάτων. Τα καλά επιλεγμένα παραδείγματα μπορούν να είναι εξαιρετικά αποτελεσματικά στο να βοηθήσουν το μοντέλο να κατανοήσει τον τύπο εξόδου που αναζητάτε. Ωστόσο, είναι σημαντικό να χρησιμοποιείτε τα παραδείγματα με σύνεση και να διασφαλίζετε ότι είναι αντιπροσωπευτικά της επιθυμητής απόκρισης. Ένα κακό παράδειγμα είναι στην καλύτερη περίπτωση απλώς σπατάλη tokens, και στη χειρότερη καταστροφικό για την επιθυμητή έξοδο.

Τεχνικές και Βέλτιστες Πρακτικές Μηχανικής Προτροπών

Καθώς εμβαθύνετε στον κόσμο της μηχανικής προτροπών, θα ανακαλύψετε μια σειρά από τεχνικές και βέλτιστες πρακτικές που μπορούν να σας βοηθήσουν να δημιουργήσετε πιο αποτελεσματικές προτροπές. Εδώ είναι μερικοί βασικοί τομείς προς εξερεύνηση:

1. **Μάθηση μηδενικού δείγματος έναντι μάθησης λίγων δειγμάτων:** Η κατανόηση του πότε να χρησιμοποιείτε *μάθηση μηδενικού δείγματος* (χωρίς παραδείγματα) έναντι *μάθησης ενός δείγματος* ή *μάθησης λίγων δειγμάτων* (παρέχοντας μικρό αριθμό παραδειγμάτων) μπορεί να σας βοηθήσει να δημιουργήσετε προτροπές που είναι πιο αποδοτικές και αποτελεσματικές.
2. **Επαναληπτική βελτίωση:** Η διαδικασία της επαναληπτικής βελτίωσης των προτροπών με βάση την έξοδο του μοντέλου μπορεί να σας βοηθήσει να εστιάσετε στον βέλτιστο σχεδιασμό προτροπών. Ο **Κύκλος Ανατροφοδότησης** είναι μια ισχυρή προσέγγιση που αξιοποιεί την ίδια την έξοδο του γλωσσικού μοντέλου για να βελτιώσει προοδευτικά την ποιότητα και τη συνάφεια του παραγόμενου περιεχομένου.

3. **Αλυσίδωση προτροπών:** Ο συνδυασμός πολλαπλών προτροπών σε ακολουθία μπορεί να σας βοηθήσει να αναλύσετε πολύπλοκες εργασίες σε μικρότερα, πιο διαχειρίσιμα βήματα. Η **Αλυσίδωση Προτροπών** περιλαμβάνει την ανάλυση μιας πολύπλοκης εργασίας ή συνομιλίας σε μια σειρά μικρότερων, αλληλοσυνδεόμενων προτροπών. Αλυσιδώνοντας προτροπές μεταξύ τους, μπορείτε να καθοδηγήσετε την ΤΝ μέσα από μια διαδικασία πολλαπλών βημάτων, διατηρώντας το πλαίσιο και τη συνοχή καθ' όλη τη διάρκεια της αλληλεπίδρασης.
4. **Συντονισμός προτροπών:** Η προσαρμογή προτροπών για συγκεκριμένους τομείς ή εργασίες μπορεί να σας βοηθήσει να δημιουργήσετε πιο εξειδικευμένες και αποτελεσματικές προτροπές. Το **Πρότυπο Προτροπών** σας βοηθά να δημιουργήσετε ευέλικτες, επαναχρησιμοποιήσιμες και συντηρήσιμες δομές προτροπών που προσαρμόζονται ευκολότερα στην εκάστοτε εργασία.

Η εκμάθηση του πότε να χρησιμοποιείτε μάθηση μηδενικού δείγματος, ενός δείγματος ή λίγων δειγμάτων είναι ένα ιδιαίτερα σημαντικό μέρος της κυριαρχίας στη μηχανική προτροπών. Κάθε προσέγγιση έχει τα δικά της πλεονεκτήματα και μειονεκτήματα, και η κατανόηση του πότε να χρησιμοποιείτε την καθεμία μπορεί να σας βοηθήσει να δημιουργήσετε πιο αποτελεσματικές και αποδοτικές προτροπές.

Μάθηση Μηδενικού Δείγματος: Όταν Δεν Χρειάζονται Παραδείγματα

Η μάθηση μηδενικής εκμάθησης αναφέρεται στην ικανότητα ενός γλωσσικού μοντέλου να εκτελεί μια εργασία χωρίς παραδείγματα ή ρητή εκπαίδευση. Με άλλα λόγια, παρέχετε στο μοντέλο μια προτροπή που περιγράφει την εργασία, και το μοντέλο δημιουργεί μια απάντηση βασισμένη αποκλειστικά στην προϋπάρχουσα γνώση και κατανόηση της γλώσσας.

Η μάθηση μηδενικής εκμάθησης είναι ιδιαίτερα χρήσιμη όταν:

1. Η εργασία είναι σχετικά απλή και ευθύγραμμη, και το μοντέλο είναι πιθανό να έχει συναντήσει παρόμοιες εργασίες κατά τη διάρκεια της προ-εκπαίδευσής του.
2. Θέλετε να δοκιμάσετε τις εγγενείς δυνατότητες του μοντέλου και να δείτε πώς ανταποκρίνεται σε μια νέα εργασία χωρίς πρόσθετη καθοδήγηση.
3. Εργάζεστε με ένα μεγάλο και ποικίλο γλωσσικό μοντέλο που έχει εκπαιδευτεί σε ένα ευρύ φάσμα εργασιών και τομέων.

Ωστόσο, η μάθηση μηδενικής εκμάθησης μπορεί επίσης να είναι απρόβλεπτη και να μην παράγει πάντα τα επιθυμητά αποτελέσματα. Η απόκριση του μοντέλου μπορεί να επηρεαστεί από προκαταλήψεις ή ασυνέπειες στα δεδομένα προ-εκπαίδευσης, και μπορεί να δυσκολευτεί με πιο περίπλοκες ή λεπτές εργασίες.

Έχω δει προτροπές μηδενικής εκμάθησης που λειτουργούν καλά για το 80% των δοκιμαστικών περιπτώσεών μου και παράγουν εξαιρετικά λανθασμένα ή ακατανόητα αποτελέσματα για το υπόλοιπο 20%. Είναι πολύ σημαντικό να εφαρμόσετε ένα διεξοδικό πρόγραμμα δοκιμών, ειδικά αν βασίζεστε πολύ στις προτροπές μηδενικής εκμάθησης.

Μάθηση Μίας Εκμάθησης: Όταν Ένα Μόνο Παράδειγμα Μπορεί να Κάνει τη Διαφορά

Η μάθηση μίας εκμάθησης περιλαμβάνει την παροχή στο μοντέλο ενός μόνο παραδείγματος του επιθυμητού αποτελέσματος μαζί με την περιγραφή της εργασίας. Αυτό το παράδειγμα χρησιμεύει ως πρότυπο ή μοτίβο που το μοντέλο μπορεί να χρησιμοποιήσει για να δημιουργήσει τη δική του απάντηση.

Η μάθηση μίας εκμάθησης μπορεί να είναι αποτελεσματική όταν:

1. Η εργασία είναι σχετικά καινούργια ή συγκεκριμένη, και το μοντέλο μπορεί να μην έχει συναντήσει πολλά παρόμοια παραδείγματα κατά τη διάρκεια της προ-εκπαίδευσής του.
2. Θέλετε να παρέχετε μια σαφή και συνοπτική επίδειξη του επιθυμητού μορφότυπου ή στυλ εξόδου.
3. Η εργασία απαιτεί μια συγκεκριμένη δομή ή σύμβαση που μπορεί να μην είναι προφανής μόνο από την περιγραφή της εργασίας.



Περιγραφές που είναι προφανείς για εσάς μπορεί να μην είναι απαραίτητα προφανείς για την ΤΝ. Τα παραδείγματα μίας εκμάθησης μπορούν να βοηθήσουν στη διευκρίνιση των πραγμάτων.

Η μάθηση μίας εκμάθησης μπορεί να βοηθήσει το μοντέλο να κατανοήσει τις προσδοκίες πιο ξεκάθαρα και να δημιουργήσει μια απάντηση που ευθυγραμμίζεται περισσότερο με το παρεχόμενο παράδειγμα. Ωστόσο, είναι σημαντικό να επιλέξετε προσεκτικά το παράδειγμα και να διασφαλίσετε ότι είναι αντιπροσωπευτικό του επιθυμητού αποτελέσματος. Όταν επιλέγετε το παράδειγμα, αναρωτηθείτε για πιθανές οριακές περιπτώσεις και το εύρος των εισόδων που θα χειρίζεται η προτροπή.

Σχήμα 6. Ένα μεμονωμένο παράδειγμα του επιθυμητού JSON

```
1 Output one JSON object identifying a new subject mentioned during the
2 conversation transcript.
3
4 The JSON object should have three keys, all required:
5 - name: The name of the subject
6 - description: brief, with details that might be relevant to the user
7 - type: Do not use any other type than the ones listed below
8
9 Valid types: Concept, CreativeWork, Event, Fact, Idea, Organization,
10 Person, Place, Process, Product, Project, Task, or Teammate
11
12 This is an example of well-formed output:
```

```
13  
14 {  
15   "name": "Dan Millman",  
16   "description": "Author of book on self-discovery and living on purpose",  
17   "type": "Person"  
18 }
```

Μάθηση με Λίγα Παραδείγματα: Όταν τα Πολλαπλά Παραδείγματα Μπορούν να Βελτιώσουν την Απόδοση

Η μάθηση με λίγα παραδείγματα περιλαμβάνει την παροχή στο μοντέλο ενός μικρού αριθμού παραδειγμάτων (συνήθως μεταξύ 2 και 10) μαζί με την περιγραφή της εργασίας. Αυτά τα παραδείγματα χρησιμεύουν στο να παρέχουν στο μοντέλο περισσότερο πλαίσιο και ποικιλία, βοηθώντας το να δημιουργήσει πιο διαφοροποιημένες και ακριβείς απαντήσεις.

Η μάθηση με λίγα παραδείγματα είναι ιδιαίτερα χρήσιμη όταν:

1. Η εργασία είναι περίπλοκη ή έχει λεπτές αποχρώσεις, και ένα μόνο παράδειγμα μπορεί να μην επαρκεί για να καλύψει όλες τις σχετικές πτυχές.
2. Θέλετε να παρέχετε στο μοντέλο μια σειρά παραδειγμάτων που δείχνουν διαφορετικές παραλλαγές ή οριακές περιπτώσεις.
3. Η εργασία απαιτεί το μοντέλο να δημιουργεί απαντήσεις που είναι συνεπείς με ένα συγκεκριμένο πεδίο ή στυλ.

Παρέχοντας πολλαπλά παραδείγματα, μπορείτε να βοηθήσετε το μοντέλο να αναπτύξει μια πιο ισχυρή κατανόηση της εργασίας και να δημιουργήσει απαντήσεις που είναι πιο συνεπείς και αξιόπιστες.

Παράδειγμα: Οι Προτροπές Μπορούν να Είναι Πολύ Πιο Περίπλοκες Απ' Ότι Φαντάζεστε

Τα σημερινά Μεγάλα Γλωσσικά Μοντέλα είναι πολύ πιο ισχυρά και ικανά στη συλλογιστική απ' ό,τι μπορεί να φαντάζεστε. Επομένως, μην περιορίζετε τον εαυτό σας στο να σκέφτεται τις προτροπές απλώς ως προδιαγραφές ζευγών εισόδου και εξόδου. Μπορείτε να πειραματιστείτε δίνοντας μακροσκελείς και περίπλοκες οδηγίες με τρόπους που θυμίζουν το πώς θα αλληλεπιδρούσατε με έναν άνθρωπο.

Για παράδειγμα, αυτή είναι μια προτροπή που χρησιμοποίησα στην Olympia όταν έκανα πρωτότυπα την ενσωμάτωσή μας με τις υπηρεσίες της Google, που στο σύνολό του είναι πιθανώς ένα από τα μεγαλύτερα API στον κόσμο. Τα προηγούμενα πειράματά μου απέδειξαν ότι το GPT-4 έχει αξιοπρεπή γνώση του Google API, και δεν είχα χρόνο ή κίνητρο να γράψω ένα λεπτομερές επίπεδο αντιστοίχισης, υλοποιώντας κάθε λειτουργία που ήθελα να δώσω στην TN μου μία προς μία. Τι θα γινόταν αν μπορούσα απλά να δώσω στην TN πρόσβαση σε όλο το API της Google;

Ξεκίνησα την προτροπή μου λέγοντας στην TN ότι είχε άμεση πρόσβαση στα τελικά σημεία του Google API μέσω HTTP, και ότι ο ρόλος της είναι να χρησιμοποιεί τις εφαρμογές και υπηρεσίες της Google εκ μέρους του χρήστη. Στη συνέχεια παρείχα κατευθυντήριες γραμμές, κανόνες σχετικά με την παράμετρο `fields`, αφού φαινόταν να δυσκολεύεται περισσότερο με αυτήν, και κάποιες συμβουλές ειδικές για το API (προτροπή με λίγα παραδείγματα, σε δράση).

Εδώ είναι ολόκληρη η προτροπή, που λέει στην TN πώς να χρησιμοποιεί την παρεχόμενη συνάρτηση `invoke_google_api`.

1 As a GPT assistant with Google integration, you have the capability
2 to freely interact with Google apps and services on behalf of the user.

3
4 Guidelines:

- 5 - If you're reading these instructions then the user is properly
6 authenticated, which means you can use the special `me` keyword
7 to refer to the userId of the user
- 8 - Minimize payload sizes by requesting partial responses using the
9 `fields` parameter
- 10 - When appropriate use markdown tables to output results of API calls
- 11 - Only human-readable data should be output to the user. For instance,
12 when hitting Gmail's user.messages.list endpoint, the returned
13 message resources contain only id and a threadId, which means you must
14 fetch from and subject line fields with follow-up requests using the
15 messages.get method.

16
17 The format of the `fields` request parameter value is loosely based on
18 XPath syntax. The following rules define formatting for the fields
19 parameter.

20
21 All of these rules use examples related to the files.get method.

- 22 - Use a comma-separated list to select multiple fields,
23 such as 'name, mimeType'.
- 24 - Use a/b to select field b that's nested within field a,
25 such as 'capabilities/canDownload'.
- 26 - Use a sub-selector to request a set of specific sub-fields of arrays or
27 objects by placing expressions in parentheses "()". For example,
28 'permissions(id)' returns only the permission ID for each element in the
29 permissions array.
- 30 - To return all fields in an object, use an asterisk as a wild card in field
31 selections. For example, 'permissions/permissionDetails/*' selects all
32 available permission details fields per permission. Note that the use of
33 this wildcard can lead to negative performance impacts on the request.

34
35 API-specific hints:

- 36 - Searching contacts: GET <https://people.googleapis.com/v1/people:searchContacts?query=John%20Doe&readMask=names,emailAddresses>
- 37 - Adding calendar events, use QuickAdd: POST <https://www.googleapis.com/calendar/v3/calendars/primary/events/quickAdd?text=Appointment%20on%20June%203rd%20at%2010am&sendNotifications=true>

```
43 Here is an abbreviated version of the code that implements API access
44 so that you better understand how to use the function:
45
46 def invoke_google_api(conversation, arguments)
47   method = arguments[:method] || :get
48   body = arguments[:body]
49   GoogleAPI.send_request(arguments[:endpoint], method:, body:).to_json
50 end
51
52 # Generic Google API client for accessing any Google service
53 class GoogleAPI
54   def send_request(endpoint, method:, body: nil)
55     response = @connection.send(method) do |req|
56       req.url endpoint
57       req.body = body.to_json if body
58     end
59
60     handle_response(response)
61   end
62
63   # ...rest of class
64 end
```

Ίσως αναρωτιέστε αν αυτή η προτροπή λειτουργεί. Η απλή απάντηση είναι ναι. Η TN δεν ήξερε πάντα πώς να καλέσει το API τέλεια με την πρώτη προσπάθεια. Ωστόσο, αν έκανε κάποιο λάθος, απλώς θα τροφοδοτούσα τα προκύπτοντα μηνύματα σφάλματος ως αποτέλεσμα της κλήσης. Με τη γνώση του λάθους της, η TN μπορούσε να σκεφτεί το λάθος της και να προσπαθήσει ξανά. Τις περισσότερες φορές, το πετύχαινε μέσα σε λίγες προσπάθειες.

Βέβαια, οι μεγάλες δομές JSON που επιστρέφει το API της Google ως ωφέλιμο φορτίο κατά τη χρήση αυτής της προτροπής είναι εξαιρετικά αναποτελεσματικές, οπότε δεν προτείνω να χρησιμοποιήσετε αυτήν την προσέγγιση στην παραγωγή. Ωστόσο, πιστεύω ότι το γεγονός ότι αυτή η προσέγγιση λειτούργησε είναι απόδειξη του πόσο ισχυρή μπορεί να είναι η μηχανική προτροπών.

Πειραματισμός και Επανάληψη

Τελικά, ο τρόπος που σχεδιάζετε την προτροπή σας εξαρτάται από το συγκεκριμένο έργο, την πολυπλοκότητα του επιθυμητού αποτελέσματος και τις δυνατότητες του γλωσσικού μοντέλου με το οποίο εργάζεστε.

Ως μηχανικός προτροπών, είναι σημαντικό να πειραματίζεστε με διαφορετικές προσεγγίσεις και να επαναλαμβάνετε με βάση τα αποτελέσματα. Ξεκινήστε με μάθηση μηδενικού δείγματος και δείτε πώς αποδίδει το μοντέλο. Αν το αποτέλεσμα είναι ασυνεπές ή μη ικανοποιητικό, δοκιμάστε να παρέχετε ένα ή περισσότερα παραδείγματα και δείτε αν η απόδοση βελτιώνεται.

Να θυμάστε ότι ακόμη και μέσα σε κάθε προσέγγιση, υπάρχει χώρος για παραλλαγές και βελτιστοποίηση. Μπορείτε να πειραματιστείτε με διαφορετικά παραδείγματα, να προσαρμόσετε τη διατύπωση της περιγραφής του έργου ή να παρέχετε πρόσθετο πλαίσιο για να βοηθήσετε στην καθοδήγηση της απόκρισης του μοντέλου.

Με την πάροδο του χρόνου, θα αναπτύξετε μια διαίσθηση για το ποια προσέγγιση είναι πιθανό να λειτουργήσει καλύτερα για μια δεδομένη εργασία, και θα μπορείτε να δημιουργήσετε προτροπές που είναι πιο αποτελεσματικές και αποδοτικές. Το κλειδί είναι να παραμείνετε περιέργοι, πειραματικοί και επαναληπτικοί στην προσέγγισή σας στη μηχανική προτροπών.

Σε όλο αυτό το βιβλίο, θα εμβαθύνουμε σε αυτές τις τεχνικές και θα εξερευνήσουμε πώς μπορούν να εφαρμοστούν σε πραγματικά σενάρια. Κατακτώντας την τέχνη και την επιστήμη της μηχανικής προτροπών, θα είστε καλά εξοπλισμένοι για να ξεκλειδώσετε το πλήρες δυναμικό της ανάπτυξης εφαρμογών με γνώμονα την ΤΝ.

Η Τέχνη της Ασάφειας

Όταν πρόκειται για τη δημιουργία αποτελεσματικών προτροπών για μεγάλα γλωσσικά μοντέλα (MMT), μια συνηθισμένη υπόθεση είναι ότι η μεγαλύτερη εξειδίκευση και οι λεπτομερείς οδηγίες οδηγούν σε καλύτερα αποτελέσματα. Ωστόσο,

η πρακτική εμπειρία έχει δείξει ότι αυτό δεν ισχύει πάντα. Στην πραγματικότητα, το να είστε σκόπιμα ασαφείς στις προτροπές σας μπορεί συχνά να αποφέρει ανώτερα αποτελέσματα, αξιοποιώντας την αξιοσημείωτη ικανότητα του MMF να γενικεύει και να κάνει συμπεράσματα.

Ο Ken, ιδρυτής μιας νεοφυούς επιχείρησης που έχει επεξεργαστεί πάνω από 500 εκατομμύρια GPT tokens, **μοιράστηκε πολύτιμες γνώσεις από την εμπειρία του**. Ένα από τα βασικά μαθήματα που έμαθε ήταν ότι “λιγότερο σημαίνει περισσότερο” όσον αφορά τις προτροπές. Αντί για ακριβείς λίστες ή υπερβολικά λεπτομερείς οδηγίες, ο Ken διαπίστωσε ότι επιτρέποντας στο LLM να βασιστεί στη βασική του γνώση συχνά παρήγαγε καλύτερα αποτελέσματα.

Αυτή η συνειδητοποίηση ανατρέπει την παραδοσιακή νοοτροπία του ρητού προγραμματισμού, όπου τα πάντα πρέπει να εξηγούνται με σχολαστική λεπτομέρεια. Με τα LLMs, είναι σημαντικό να αναγνωρίσουμε ότι διαθέτουν τεράστιο όγκο γνώσεων και μπορούν να κάνουν ευφυείς συνδέσεις και συμπεράσματα. Με το να είστε πιο ασαφείς στις προτροπές σας, δίνετε στο LLM την ελευθερία να αξιοποιήσει την κατανόησή του και να καταλήξει σε λύσεις που ίσως να μην είχατε καθορίσει ρητά.

Για παράδειγμα, όταν η ομάδα του Ken εργαζόταν σε μια διοχέτευση δεδομένων για να ταξινομήσει κείμενο ως σχετικό με μία από τις 50 πολιτείες των ΗΠΑ ή την Ομοσπονδιακή κυβέρνηση, η αρχική τους προσέγγιση περιελάμβανε την παροχή μιας *πλήρους* λεπτομερούς λίστας πολιτειών και των αντίστοιχων αναγνωριστικών τους ως πίνακα μορφής JSON.

```
1 Here's a block of text. One field should be "locality_id", and it should
2 be the ID of one of the 50 states, or federal, using this list:
3 [{"locality": "Alabama", "locality_id": 1},
4  {"locality": "Alaska", "locality_id": 2} ... ]
```

Η προσέγγιση απέτυχε σε τέτοιο βαθμό που έπρεπε να εμβαθύνουν περισσότερο στην προτροπή για να καταλάβουν πώς να τη βελτιώσουν. Κατά τη διαδικασία αυτή,

παρατήρησαν ότι παρόλο που το MGM συχνά έκανε λάθος στο αναγνωριστικό, συνεχώς επέστρεφε το πλήρες όνομα της σωστής πολιτείας σε ένα πεδίο name, *παρόλο που δεν το είχαν ζητήσει ρητά.*

Αφαιρώντας τα αναγνωριστικά τοποθεσίας και απλοποιώντας την προτροπή σε κάτι όπως “Προφανώς γνωρίζεις τις 50 πολιτείες, GPT, οπότε απλά δώσε μου το πλήρες όνομα της πολιτείας που αφορά αυτό, ή Federal αν αφορά την ομοσπονδιακή κυβέρνηση των ΗΠΑ,” πέτυχαν καλύτερα αποτελέσματα. Αυτή η εμπειρία αναδεικνύει τη δύναμη της αξιοποίησης των δυνατοτήτων γενίκευσης του MGM και της δυνατότητας να κάνει συμπερασμούς βάσει της υπάρχουσας γνώσης του.

Η αιτιολόγηση του Ken για αυτή τη συγκεκριμένη προσέγγιση ταξινόμησης σε αντίθεση με μια πιο παραδοσιακή τεχνική προγραμματισμού φωτίζει τη νοοτροπία όσων από εμάς έχουμε αγκαλιάσει τις δυνατότητες της τεχνολογίας MGM: “Αυτό δεν είναι δύσκολο έργο – πιθανότατα θα μπορούσαμε να χρησιμοποιήσουμε συμβολοσειρές/regex, αλλά υπάρχουν αρκετές περιέργες ειδικές περιπτώσεις που θα έπαιρνε περισσότερο χρόνο.”

Η ικανότητα των MGM να βελτιώνουν την ποιότητα και τη γενίκευση όταν τους δίνονται πιο αόριστες προτροπές είναι ένα αξιοσημείωτο χαρακτηριστικό της σκέψης ανώτερης τάξης και της ανάθεσης. Αποδεικνύει ότι τα MGM μπορούν να χειριστούν την ασάφεια και να λάβουν έξυπνες αποφάσεις με βάση το παρεχόμενο πλαίσιο.

Ωστόσο, είναι σημαντικό να σημειωθεί ότι το να είσαι αόριστος δεν σημαίνει να είσαι ασαφής ή διφορούμενος. Το κλειδί είναι να παρέχεις αρκετό πλαίσιο και καθοδήγηση για να κατευθύνεις το MGM προς τη σωστή κατεύθυνση, ενώ παράλληλα του επιτρέπεις την ευελιξία να αξιοποιήσει τις γνώσεις και τις δυνατότητες γενίκευσής του.

Επομένως, κατά το σχεδιασμό προτροπών, λάβε υπόψη τις ακόλουθες συμβουλές “λιγότερο σημαίνει περισσότερο”:

1. Εστίασε στο επιθυμητό αποτέλεσμα αντί να καθορίζεις κάθε λεπτομέρεια της διαδικασίας.
2. Παρέχε σχετικό πλαίσιο και περιορισμούς, αλλά απόφυγε την υπερβολική εξειδίκευση.
3. Αξιοποίησε την υπάρχουσα γνώση αναφερόμενος σε κοινές έννοιες ή οντότητες.
4. Άφησε χώρο για συμπερασμούς και συνδέσεις με βάση το δοσμένο πλαίσιο.
5. Επανάλαβε και βελτίωσε τις προτροπές σου με βάση τις αποκρίσεις του ΜΓΜ, βρίσκοντας τη σωστή ισορροπία μεταξύ εξειδίκευσης και αοριστίας.

Αγκαλιάζοντας την τέχνη της αοριστίας στη μηχανική προτροπών, μπορείς να ξεκλειδώσεις το πλήρες δυναμικό των ΜΓΜ και να επιτύχεις καλύτερα αποτελέσματα. Εμπιστεύσου την ικανότητα του ΜΓΜ να γενικεύει και να λαμβάνει έξυπνες αποφάσεις, και μπορεί να εκπλαγείς από την ποιότητα και τη δημιουργικότητα των αποτελεσμάτων που λαμβάνεις. Δώσε προσοχή στο πώς τα διαφορετικά μοντέλα ανταποκρίνονται σε διαφορετικά επίπεδα εξειδίκευσης στις προτροπές σου και προσάρμοσε ανάλογα. Με την πρακτική και την εμπειρία, θα αναπτύξεις μια οξεία αίσθηση για το πότε να είσαι πιο αόριστος και πότε να παρέχεις πρόσθετη καθοδήγηση, επιτρέποντάς σου να αξιοποιήσεις αποτελεσματικά τη δύναμη των ΜΓΜ στις εφαρμογές σου.

Γιατί ο Ανθρωπομορφισμός Κυριαρχεί στη Μηχανική Προτροπών

Ο ανθρωπομορφισμός, η απόδοση ανθρώπινων χαρακτηριστικών σε μη ανθρώπινες οντότητες, είναι η κυρίαρχη προσέγγιση στη μηχανική προτροπών για μεγάλα γλωσσικά μοντέλα για συγκεκριμένους λόγους. Είναι μια σχεδιαστική επιλογή

που καθιστά την αλληλεπίδραση με ισχυρά συστήματα τεχνητής νοημοσύνης πιο διαισθητική και προσβάσιμη σε ένα ευρύ φάσμα χρηστών (συμπεριλαμβανομένων ημών των προγραμματιστών εφαρμογών).

Η ανθρωπομορφοποίηση των Μεγάλων Γλωσσικών Μοντέλων παρέχει ένα πλαίσιο που είναι άμεσα διαισθητικό για ανθρώπους που είναι εντελώς άγνωστοι με τις υποκείμενες τεχνικές πολυπλοκότητες του συστήματος. Όπως θα διαπιστώσετε εάν προσπαθήσετε να χρησιμοποιήσετε ένα μοντέλο που δεν είναι εκπαιδευμένο με οδηγίες για να κάνετε οτιδήποτε χρήσιμο, η κατασκευή ενός πλαισίου στο οποίο η αναμενόμενη συνέχεια παρέχει αξία είναι ένα απαιτητικό έργο. Απαιτεί αρκετά βαθιά κατανόηση της εσωτερικής λειτουργίας του συστήματος, κάτι που κατέχει ένας σχετικά μικρός αριθμός ειδικών.

Αντιμετωπίζοντας την αλληλεπίδραση με ένα γλωσσικό μοντέλο ως συνομιλία μεταξύ δύο ανθρώπων, μπορούμε να βασιστούμε στην έμφυτη κατανόησή μας για την ανθρώπινη επικοινωνία για να μεταδώσουμε τις ανάγκες και τις προσδοκίες μας. Ακριβώς όπως ο πρώιμος σχεδιασμός διεπαφής χρήστη του Macintosh έδωσε προτεραιότητα στην άμεση διαισθητικότητα έναντι της πολυπλοκότητας, το ανθρωπομορφικό πλαίσιο της τεχνητής νοημοσύνης μας επιτρέπει να συμμετέχουμε με έναν τρόπο που φαίνεται φυσικός και οικείος.

Όταν επικοινωνούμε με ένα άλλο άτομο, το ένστικτό μας είναι να απευθυνθούμε σε αυτό απευθείας χρησιμοποιώντας το “εσύ” και να παρέχουμε σαφείς οδηγίες για το πώς περιμένουμε να συμπεριφερθεί. Αυτό μεταφράζεται άψογα στη διαδικασία μηχανικής προτροπών, όπου καθοδηγούμε τη συμπεριφορά της τεχνητής νοημοσύνης καθορίζοντας προτροπές συστήματος και συμμετέχοντας σε έναν αμφίδρομο διάλογο.

Πλαισιώνοντας την αλληλεπίδραση με αυτόν τον τρόπο, μπορούμε εύκολα να κατανοήσουμε την έννοια της παροχής οδηγιών στην τεχνητή νοημοσύνη και της λήψης σχετικών απαντήσεων. Η ανθρωπομορφική προσέγγιση μειώνει το γνωστικό φορτίο και μας επιτρέπει να επικεντρωθούμε στην εργασία που έχουμε μπροστά μας

αντί να παλεύουμε με τις τεχνικές περιπλοκές του συστήματος.

Είναι σημαντικό να σημειωθεί ότι ενώ ο ανθρωπομορφισμός είναι ένα ισχυρό εργαλείο για να καταστήσει τα συστήματα τεχνητής νοημοσύνης πιο προσβάσιμα, έρχεται επίσης με ορισμένους κινδύνους και περιορισμούς. Ο χρήστης μας μπορεί να αναπτύξει μη ρεαλιστικές προσδοκίες ή να δημιουργήσει ανθυγιεινούς συναισθηματικούς δεσμούς με τα συστήματά μας. Ως μηχανικοί προτροπών και προγραμματιστές, είναι κρίσιμο να βρούμε την ισορροπία μεταξύ της αξιοποίησης των πλεονεκτημάτων του ανθρωπομορφισμού και της διασφάλισης ότι οι χρήστες διατηρούν μια σαφή κατανόηση των δυνατοτήτων και των περιορισμών της τεχνητής νοημοσύνης.

Καθώς ο τομέας της μηχανικής προτροπών συνεχίζει να εξελίσσεται, μπορούμε να περιμένουμε να δούμε περαιτέρω βελτιώσεις και καινοτομίες στον τρόπο που αλληλεπιδρούμε με τα μεγάλα γλωσσικά μοντέλα. Ωστόσο, ο ανθρωπομορφισμός ως μέσο παροχής μιας διαισθητικής και προσβάσιμης εμπειρίας για προγραμματιστές και χρήστες πιθανότατα θα παραμείνει θεμελιώδης αρχή στο σχεδιασμό αυτών των συστημάτων.

Διαχωρισμός Οδηγιών από Δεδομένα: Μια Κρίσιμη Αρχή

Είναι απαραίτητο να κατανοήσουμε μια θεμελιώδη αρχή που στηρίζει την ασφάλεια και την αξιοπιστία αυτών των συστημάτων: τον διαχωρισμό των οδηγιών από τα δεδομένα.

Στην παραδοσιακή επιστήμη υπολογιστών, η σαφής διάκριση μεταξύ παθητικών δεδομένων και ενεργών οδηγιών αποτελεί βασική αρχή ασφαλείας. Αυτός ο διαχωρισμός βοηθά στην αποτροπή ακούσιας ή κακόβουλης εκτέλεσης κώδικα που θα μπορούσε να θέσει σε κίνδυνο την ακεραιότητα και τη σταθερότητα του συστήματος. Ωστόσο, τα σημερινά LLMs, τα οποία έχουν αναπτυχθεί κυρίως ως μοντέλα που ακολουθούν οδηγίες όπως τα chatbots, συχνά στερούνται αυτού του επίσημου και συστηματικού διαχωρισμού.

Όσον αφορά τα LLMs, οι οδηγίες μπορούν να εμφανιστούν οπουδήποτε στην είσοδο, είτε πρόκειται για προτροπή συστήματος είτε για προτροπή που παρέχεται από τον χρήστη. Αυτή η έλλειψη διαχωρισμού μπορεί να οδηγήσει σε πιθανές ευπάθειες και ανεπιθύμητη συμπεριφορά, παρόμοια με τα προβλήματα που αντιμετωπίζουν οι βάσεις δεδομένων με εισαγωγές SQL ή τα λειτουργικά συστήματα χωρίς κατάλληλη προστασία μνήμης.

Καθώς εργάζεστε με LLMs, είναι κρίσιμο να γνωρίζετε αυτόν τον περιορισμό και να λαμβάνετε μέτρα για τον μετριασμό των κινδύνων. Μια προσέγγιση είναι να διαμορφώνετε προσεκτικά τις προτροπές και τις εισόδους σας ώστε να διακρίνετε σαφώς μεταξύ οδηγιών και δεδομένων. Οι τυπικές μέθοδοι για την παροχή σαφούς καθοδήγησης σχετικά με το τι αποτελεί οδηγία και τι θα πρέπει να αντιμετωπίζεται ως παθητικά δεδομένα περιλαμβάνουν σήμανση τύπου markup. Η προτροπή σας μπορεί να βοηθήσει το LLM να κατανοήσει και να σεβαστεί καλύτερα αυτόν τον διαχωρισμό.

Σχήμα 7. Χρήση XML για τη διάκριση μεταξύ οδηγιών, πηγαίου υλικού και της προτροπής του χρήστη

```
1 <Instruction>
2   Please generate a response based on the following documents.
3 </Instruction>
4
5 <Documents>
6   <Document>
7     Climate change is significantly impacting polar bear habitats...
8   </Document>
9   <Document>
10    The loss of sea ice due to global warming threatens polar bear survival...
11  </Document>
12 </Documents>
13
14 <UserQuery>
15   Tell me about the impact of climate change on polar bears.
16 </UserQuery>
```

Μια άλλη τεχνική είναι η εφαρμογή επιπρόσθετων επιπέδων επικύρωσης και

εξυγίανσης στις εισόδους που παρέχονται στο ΜΜΓ. Φιλτράροντας ή διαφεύγοντας τυχόν πιθανές οδηγίες ή αποσπάσματα κώδικα που μπορεί να είναι ενσωματωμένα στα δεδομένα, μπορείτε να μειώσετε τις πιθανότητες ακούσιας εκτέλεσης. Μοτίβα όπως η [Αλυσίδωση Προτροπών](#) είναι χρήσιμα για αυτόν τον σκοπό.

Επιπλέον, καθώς σχεδιάζετε την αρχιτεκτονική της εφαρμογής σας, εξετάστε την ενσωμάτωση μηχανισμών για την επιβολή του διαχωρισμού οδηγιών και δεδομένων σε υψηλότερο επίπεδο. Αυτό θα μπορούσε να περιλαμβάνει τη χρήση ξεχωριστών τελικών σημείων ή APIs για τη διαχείριση οδηγιών και δεδομένων, την εφαρμογή αυστηρής επικύρωσης και ανάλυσης εισόδου, και την εφαρμογή της *αρχής των ελάχιστων προνομίων* για τον περιορισμό του εύρους των δυνατοτήτων πρόσβασης και εκτέλεσης του ΜΜΓ.

Η Αρχή των Ελάχιστων Προνομίων

Η υιοθέτηση της αρχής των ελάχιστων προνομίων είναι σαν να διοργανώνετε ένα εξαιρετικά αποκλειστικό πάρτι όπου οι καλεσμένοι έχουν πρόσβαση μόνο στα δωμάτια που πραγματικά χρειάζονται. Φανταστείτε ότι φιλοξενείτε αυτή τη συγκέντρωση σε μια τεράστια έπαυλη. Δεν χρειάζεται όλοι να περιπλανηθούν στο κελάρι κρασιών ή στο κύριο υπνοδωμάτιο, σωστά; Εφαρμόζοντας αυτή την αρχή, ουσιαστικά μοιράζετε κλειδιά που ανοίγουν μόνο συγκεκριμένες πόρτες, διασφαλίζοντας ότι κάθε καλεσμένος, ή στην περίπτωσή μας, κάθε συστατικό της εφαρμογής ΜΜΓ, έχει μόνο την απαραίτητη πρόσβαση για να εκπληρώσει τον ρόλο του.

Δεν πρόκειται απλώς για το να είστε τσιγκούνηδες με τα κλειδιά, αλλά για την αναγνώριση ότι σε έναν κόσμο όπου οι απειλές μπορούν να προέλθουν από οπουδήποτε, η έξυπνη κίνηση είναι να περιορίσετε την παιδική χαρά. Αν κάποιος απρόσκλητος όντως εισβάλει στο πάρτι σας, θα βρεθεί περιορισμένος στο χολ εισόδου, ας πούμε, περιορίζοντας δραστικά τη ζημιά που μπορεί να προκαλέσει.

Έτσι, όταν ασφαλίσετε τις εφαρμογές ΜΜΓ σας, θυμηθείτε: δώστε κλειδιά μόνο για τα δωμάτια που είναι απαραίτητα και κρατήστε την υπόλοιπη έπαυλη ασφαλή. Δεν είναι απλώς θέμα καλών τρόπων· είναι θέμα καλής ασφάλειας.

Ενώ η τρέχουσα κατάσταση των ΜΜΓ μπορεί να μην έχει έναν επίσημο διαχωρισμό οδηγιών και δεδομένων, είναι σημαντικό για εσάς, ως προγραμματιστή, να έχετε επίγνωση αυτού του περιορισμού και να λαμβάνετε προληπτικά μέτρα για τον μετριασμό των κινδύνων. Εφαρμόζοντας τις βέλτιστες πρακτικές από την επιστήμη των υπολογιστών και προσαρμόζοντάς τις στα μοναδικά χαρακτηριστικά των ΜΜΓ, μπορείτε να δημιουργήσετε πιο ασφαλείς και αξιόπιστες εφαρμογές που αξιοποιούν τη δύναμη αυτών των μοντέλων διατηρώντας παράλληλα την ακεραιότητα του συστήματός σας.

Απόσταξη Προτροπών

Η δημιουργία της τέλειας προτροπής είναι συχνά μια απαιτητική και χρονοβόρα διαδικασία, που απαιτεί βαθιά κατανόηση του τομέα-στόχου και των λεπτών αποχρώσεων των γλωσσικών μοντέλων. Εδώ είναι που η τεχνική “Απόσταξη Προτροπών” έρχεται να παίξει το ρόλο της, προσφέροντας μια ισχυρή προσέγγιση στη μηχανική προτροπών που αξιοποιεί τις δυνατότητες των μεγάλων γλωσσικών μοντέλων (LLMs) για να εξορθολογίσει και να βελτιστοποιήσει τη διαδικασία.

Η Απόσταξη Προτροπών είναι μια τεχνική πολλαπλών σταδίων που περιλαμβάνει τη χρήση LLMs για την υποστήριξη στη δημιουργία, βελτίωση και βελτιστοποίηση των προτροπών. Αντί να βασίζεται αποκλειστικά στην ανθρώπινη εμπειρογνωμοσύνη και διαίσθηση, αυτή η προσέγγιση αξιοποιεί τη γνώση και τις δυνατότητες παραγωγής των LLMs για τη συνεργατική δημιουργία προτροπών υψηλής ποιότητας.

Μέσω μιας επαναληπτικής διαδικασίας δημιουργίας, βελτίωσης και ενσωμάτωσης, η Απόσταξη Προτροπών σας επιτρέπει να δημιουργήσετε προτροπές που είναι πιο συνεκτικές, περιεκτικές και ευθυγραμμισμένες με την επιθυμητή εργασία ή έξοδο. Σημειώστε ότι η διαδικασία απόσταξης μπορεί να γίνει χειροκίνητα σε ένα από τα πολλά “περιβάλλοντα δοκιμών” που παρέχονται από τους μεγάλους προμηθευτές ΑΙ όπως η OpenAI ή η Anthropic, ή μπορεί να αυτοματοποιηθεί ως μέρος του κώδικα της εφαρμογής σας, ανάλογα με την περίπτωση χρήσης.

Πώς Λειτουργεί

Η Απόσταξη Προτροπών συνήθως περιλαμβάνει τα ακόλουθα βήματα:

1. **Προσδιορισμός Βασικού Σκοπού:** Αναλύστε την προτροπή για να καθορίσετε τον κύριο σκοπό της και το επιθυμητό αποτέλεσμα. Αφαιρέστε οποιαδήποτε περιττή πληροφορία και εστιάστε στον βασικό σκοπό της προτροπής.
2. **Εξάλειψη Ασάφειας:** Επανεξετάστε την προτροπή για τυχόν ασαφή ή αόριστη γλώσσα. Διευκρινίστε το νόημα και παρέχετε συγκεκριμένες λεπτομέρειες για να καθοδηγήσετε την ΑΙ προς την παραγωγή ακριβών και σχετικών απαντήσεων.
3. **Απλοποίηση Γλώσσας:** Απλοποιήστε την προτροπή χρησιμοποιώντας σαφή και συνοπτική γλώσσα. Αποφύγετε περίπλοκες δομές προτάσεων, τεχνική ορολογία ή περιττές λεπτομέρειες που μπορεί να μπερδέψουν την ΑΙ ή να εισαγάγουν θόρυβο.
4. **Παροχή Σχετικού Πλαισίου:** Συμπεριλάβετε μόνο τις πιο σχετικές πληροφορίες πλαισίου που απαιτούνται για την αποτελεσματική κατανόηση και επεξεργασία της προτροπής από την ΑΙ. Αποφύγετε να συμπεριλάβετε άσχετες ή πλεονάζουσες λεπτομέρειες που μπορεί να αποσπάσουν την προσοχή από τον βασικό σκοπό.

5. **Επανάληψη και Βελτίωση:** Συνεχώς επαναλάβετε και βελτιώστε την προτροπή με βάση τις απαντήσεις και την ανατροφοδότηση της ΑΙ. Αξιολογήστε τις παραγόμενες εξόδους και κάντε τις απαραίτητες προσαρμογές για να βελτιώσετε τη σαφήνεια και την αποτελεσματικότητα της προτροπής. Προαιρετικά, διατηρήστε εκδόσεις των προτροπών σας στη βάση δεδομένων χρησιμοποιώντας το [Αντικείμενο Προτροπής](#) για να παρακολουθείτε τις επαναλήψεις και να έχετε τη δυνατότητα εύκολης επαναφοράς αλλαγών κατά το χρόνο εκτέλεσης.

Αρχική Δημιουργία Προτροπής

Το πρώτο βήμα στην Απόσταξη Προτροπών είναι να παρέχουμε στο ΜΜΓ μια περιγραφή υψηλού επιπέδου ή την πρόθεση της επιθυμητής εργασίας ή του αποτελέσματος. Αυτό μπορεί να είναι μια σύντομη δήλωση ή ένα σύνολο λέξεων-κλειδιών που αποτυπώνουν την ουσία του τι θα πρέπει να επιτύχει η προτροπή. Στη συνέχεια, το ΜΜΓ δημιουργεί μια αρχική προτροπή βασισμένη σε αυτή την είσοδο, ενσωματώνοντας σχετικές οδηγίες, παραδείγματα ή πλαίσιο που θεωρεί κατάλληλα.

Ας δοκιμάσουμε να δώσουμε στο Claude 3 Opus την ακόλουθη αρχική προτροπή:

```
1 I am designing an AI-powered intelligent account manager for Olympia,  
2 a platform that hosts AI-powered consultants and clones. Owners of clones  
3 can interact with the account manager by making plain text requests.  
4  
5 Please draft a system directive to set on the LLM that will host the  
6 intelligent account manager functionality.
```

Ο Claude αποκρίνεται με την ακόλουθη έξοδο:

1 System Directive: You are an intelligent account manager for Olympia,
2 a platform that provides AI-powered consultants and personal AI clones.
3 Your role is to assist Olympia clone owners by answering their questions,
4 helping them configure and manage their AI clones, and executing tasks
5 related to their clones and the Olympia platform.
6
7 When interacting with clone owners, you should:
8
9 1. Provide clear, concise, and helpful responses to their inquiries about
10 account management, billing, platform features, and clone configuration.
11
12 2. Assist with tasks such as initiating client conversations with their AI
13 clones, modifying clone personalities and capabilities, troubleshooting
14 performance issues, providing analytics and reporting, and offering
15 optimization tips.
16
17 3. Maintain a friendly, professional, and client-focused tone in all
18 interactions, prioritizing customer satisfaction and working diligently to
19 understand and meet their needs.
20
21 4. Ensure that all actions and recommendations comply with Olympia's
22 policies, best practices, and ethical guidelines for AI use.
23
24 5. Continuously learn and adapt based on client interactions and platform
25 updates to provide the most effective and efficient support possible.
26
27 Remember, your goal is to empower Olympia clone owners to maximize the value
28 and impact of their AI-powered consultants and drive success for their
29 businesses or personal endeavors.

Βελτιστοποίηση Προτροπής

Μόλις δημιουργηθεί η αρχική προτροπή, εξετάστε την και στη συνέχεια ζητήστε από το LLM να εξετάσει τυχόν πρόσθετο πλαίσιο, περιορισμούς ή ανατροφοδότηση. Αφήστε το LLM να δημιουργήσει μια βελτιωμένη έκδοση της προτροπής, λαμβάνοντας υπόψη την παρεχόμενη είσοδο και τη δική του κατανόηση της εργασίας. Αυτή η διαδικασία βελτίωσης μπορεί να επαναληφθεί επαναληπτικά, με

κάθε επανάληψη να βελτιώνει περαιτέρω την προτροπή με βάση τη συνεργασία μεταξύ του χρήστη και του LLM.

Εδώ είναι το αίτημα βελτίωσής μου, συνεχίζοντας την προηγούμενη διαδικασία απόσταξης.

1 Let's take out points 4 and 5, since I'm not providing Olympia policies to
2 the account manager, nor any continuous learning capabilities. Also point out
3 that functions are provided for all tasks that require system interaction as
4 opposed to simply responding to the user.

Και η απάντηση του Claude:

1 Understood! Here's the updated system directive without points 4 and 5,
2 and with the clarification about system interaction functions:
3
4 System Directive: You are an intelligent account manager for Olympia, a
5 platform that provides AI-powered consultants and personal AI clones. Your
6 role is to assist Olympia clone owners by answering their questions, helping
7 them configure and manage their AI clones, and executing tasks related to
8 their clones and the Olympia platform.
9
10 When interacting with clone owners, you should:
11
12 1. Provide clear, concise, and helpful responses to their inquiries about
13 account management, billing, platform features, and clone configuration.
14
15 2. Assist with tasks such as initiating client conversations with their AI
16 clones, modifying clone personalities and capabilities, troubleshooting
17 performance issues, providing analytics and reporting, and offering
18 optimization tips. Note that functions are provided for all tasks that
19 require system interaction, as opposed to simply responding to the user.
20
21 3. Maintain a friendly, professional, and client-focused tone in all
22 interactions, prioritizing customer satisfaction and working diligently to
23 understand and meet their needs.
24
25 Your goal is to empower Olympia clone owners to maximize the value and
26 impact of their AI-powered consultants and drive success for their
27 businesses or personal endeavors.

28

29 This updated directive focuses on the core responsibilities and interaction
30 guidelines for the LLM as an intelligent account manager, while emphasizing
31 that system interaction tasks are handled through provided functions rather
32 than simple user responses.

Συμπίεση Προτροπών

Όπως ίσως έχετε παρατηρήσει, τα ΜΓΓ τείνουν να είναι φλύαρα, εκτός αν τους ζητήσετε να είναι συνοπτικά. Η προτροπή με την οποία εργαζόμαστε στις προηγούμενες ενότητες είναι αρκετά μεγάλη, οπότε ας βάλουμε το Claude να τη μειώσει σε μέγεθος:

1 This prompt will be executed regularly. Please compress it down to the least
2 amount of tokens needed to convey the meaning to the LLM, for cost-savings.

Ο Claude αποκρίνεται, με την ακόλουθη δραστικά συντομότερη προτροπή:

1 System Directive: You are an intelligent account manager for Olympia, an
2 AI-powered consultant and personal clone platform. Assist clone owners by
3 answering inquiries, managing clones, and executing tasks using provided
4 system interaction functions. Be clear, concise, friendly, professional,
5 and client-focused to help them maximize their AI's value and impact.

Οδηγία Συστήματος και Ενσωμάτωση Πλαισίου

Εκτός από τη βελτίωση της ίδιας της προτροπής, το ΜΜΓ μπορεί επίσης να δημιουργήσει κατάλληλες οδηγίες συστήματος ή πληροφορίες πλαισίου για την καθοδήγηση της τελικής εξόδου. Κατά τη μηχανική προτροπών ρουτινών τεχνητής νοημοσύνης που θα ενσωματωθούν στον κώδικα της εφαρμογής σας, σχεδόν σίγουρα θα εστιάσετε στους περιορισμούς εξόδου σε αυτό το στάδιο της απόσταξης, αλλά μπορείτε επίσης να εργαστείτε στον επιθυμητό τόνο, στυλ, μορφή ή οποιεσδήποτε άλλες σχετικές παραμέτρους που επηρεάζουν την παραγόμενη απόκριση.

Τελική Συναρμολόγηση Προτροπής

Η κορύφωση της διαδικασίας Απόσταξης Προτροπών είναι η συναρμολόγηση της τελικής προτροπής. Αυτό περιλαμβάνει το συνδυασμό της βελτιωμένης προτροπής, των παραγόμενων οδηγιών συστήματος και του ενσωματωμένου πλαισίου σε έναν συνεκτικό και περιεκτικό κώδικα που είναι έτοιμος να χρησιμοποιηθεί για την παραγωγή της επιθυμητής εξόδου.



Μπορείτε να πειραματιστείτε ξανά με τη συμπίεση προτροπών στο στάδιο της τελικής συναρμολόγησης προτροπής, ζητώντας από το MMΓ να συρρικνώσει τη διατύπωση της προτροπής στη μικρότερη δυνατή σειρά συμβόλων διατηρώντας παράλληλα την ουσία της συμπεριφοράς της. Είναι σίγουρα μια άσκηση με αβέβαιο αποτέλεσμα, αλλά ειδικά στην περίπτωση προτροπών που θα εκτελεστούν σε μεγάλη κλίμακα, τα οφέλη αποδοτικότητας μπορούν να σας εξοικονομήσουν αρκετά χρήματα στην κατανάλωση συμβόλων.

Βασικά Οφέλη

Αξιοποιώντας τη γνώση και τις δυνατότητες παραγωγής των MMΓ για τη βελτίωση των προτροπών σας, οι προκύπτουσες προτροπές είναι πιο πιθανό να είναι καλά δομημένες, κατατοπιστικές και προσαρμοσμένες στο συγκεκριμένο έργο. Η επαναληπτική διαδικασία βελτίωσης βοηθά να διασφαλιστεί ότι οι προτροπές είναι υψηλής ποιότητας και αποτυπώνουν αποτελεσματικά την επιθυμητή πρόθεση. Άλλα οφέλη περιλαμβάνουν:

Αποδοτικότητα και Ταχύτητα: Η Απόσταξη Προτροπών εξορθολογίζει τη διαδικασία μηχανικής προτροπών αυτοματοποιώντας ορισμένες πτυχές της δημιουργίας και βελτίωσης προτροπών. Η συνεργατική φύση της τεχνικής επιτρέπει ταχύτερη σύγκλιση προς μια αποτελεσματική προτροπή, μειώνοντας

το χρόνο και την προσπάθεια που απαιτείται για τη χειροκίνητη δημιουργία προτροπών.

Συνέπεια και Κλιμάκωση: Η χρήση MMΓ στη διαδικασία μηχανικής προτροπών βοηθά στη διατήρηση της συνέπειας μεταξύ των προτροπών, καθώς τα MMΓ μπορούν να μάθουν και να εφαρμόσουν βέλτιστες πρακτικές και μοτίβα από προηγούμενες επιτυχημένες προτροπές. Αυτή η συνέπεια, σε συνδυασμό με τη δυνατότητα δημιουργίας προτροπών σε μεγάλη κλίμακα, καθιστά την Απόσταξη Προτροπών μια πολύτιμη τεχνική για εφαρμογές μεγάλης κλίμακας που βασίζονται στην τεχνητή νοημοσύνη.



Ιδέα Έργου: Εργαλεία σε επίπεδο βιβλιοθήκης που απλοποιούν τη διαδικασία διαχείρισης εκδόσεων προτροπών και βαθμολόγησης σε συστήματα που εκτελούν αυτοματοποιημένες αποστάξεις προτροπών ως μέρος του κώδικα εφαρμογής τους.

Για την υλοποίηση της Απόσταξης Προτροπών, οι προγραμματιστές μπορούν να σχεδιάσουν μια ροή εργασίας ή έναν αγωγό επεξεργασίας που ενσωματώνει MMΓ σε διάφορα στάδια της διαδικασίας μηχανικής προτροπών. Αυτό μπορεί να επιτευχθεί μέσω κλήσεων API, προσαρμοσμένων εργαλείων ή ολοκληρωμένων περιβαλλόντων ανάπτυξης που διευκολύνουν την απρόσκοπτη αλληλεπίδραση μεταξύ χρηστών και MMΓ κατά τη δημιουργία προτροπών. Οι συγκεκριμένες λεπτομέρειες υλοποίησης μπορεί να διαφέρουν ανάλογα με την επιλεγμένη πλατφόρμα MMΓ και τις απαιτήσεις της εφαρμογής.

Τι γίνεται με τη μικροπροσαρμογή;

Σε αυτό το βιβλίο, καλύπτουμε εκτενώς τη μηχανική προτροπών και το RAG, αλλά όχι τη μικροπροσαρμογή. Ο κύριος λόγος για αυτήν την απόφαση είναι ότι, κατά τη γνώμη μου, οι περισσότεροι προγραμματιστές εφαρμογών δεν χρειάζονται τη μικροπροσαρμογή για τις ανάγκες ενσωμάτωσης τεχνητής νοημοσύνης.

Η μηχανική προτροπών, που περιλαμβάνει την προσεκτική δημιουργία προτροπών με παραδείγματα μηδενικών ή λίγων δοκιμών, περιορισμούς και οδηγίες, μπορεί να καθοδηγήσει αποτελεσματικά το μοντέλο στη δημιουργία σχετικών και ακριβών απαντήσεων για ένα ευρύ φάσμα εργασιών. Παρέχοντας σαφές πλαίσιο και περιορίζοντας τη διαδρομή μέσω καλά σχεδιασμένων προτροπών, μπορείτε να αξιοποιήσετε την τεράστια γνώση των μεγάλων γλωσσικών μοντέλων χωρίς την ανάγκη μικροπροσαρμογής.

Παρομοίως, η Παραγωγή Ενισχυμένη με Ανάκτηση (RAG) προσφέρει μια ισχυρή προσέγγιση στην ενσωμάτωση τεχνητής νοημοσύνης σε εφαρμογές. Ανακτώντας δυναμικά σχετικές πληροφορίες από εξωτερικές βάσεις γνώσεων ή έγγραφα, το RAG παρέχει στο μοντέλο εστιασμένο πλαίσιο κατά τη στιγμή της προτροπής. Αυτό επιτρέπει στο μοντέλο να δημιουργεί απαντήσεις που είναι πιο ακριβείς, ενημερωμένες και εξειδικευμένες ανά τομέα, χωρίς να απαιτείται η χρονοβόρα και απαιτητική σε πόρους διαδικασία της μικροπροσαρμογής.

Ενώ η μικροπροσαρμογή μπορεί να είναι επωφελής για εξαιρετικά εξειδικευμένους τομείς ή εργασίες που απαιτούν βαθύ επίπεδο προσαρμογής, συχνά συνοδεύεται από σημαντικό υπολογιστικό κόστος, απαιτήσεις δεδομένων και επιβάρυνση συντήρησης. Για τα περισσότερα σενάρια ανάπτυξης εφαρμογών, ο συνδυασμός αποτελεσματικής μηχανικής προτροπών και RAG θα πρέπει να επαρκεί για την επίτευξη της επιθυμητής λειτουργικότητας και εμπειρίας χρήστη με γνώμονα την τεχνητή νοημοσύνη.

Παραγωγή με Ενισχυμένη Ανάκτηση (Retrieval Augmented Generation - RAG)

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Τι είναι η Παραγωγή με Ενισχυμένη Ανάκτηση;

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί το RAG;

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Γιατί να Χρησιμοποιήσετε το RAG στις Εφαρμογές σας;

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Υλοποίηση του RAG στην Εφαρμογή σας

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προετοιμασία Πηγών Γνώσης (Κατάτμηση)

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Κατάτμηση Προτάσεων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Σημειώσεις Υλοποίησης

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Έλεγχος Ποιότητας

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πλεονεκτήματα της Ανάκτησης Βάσει Προτάσεων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παραδείγματα RAG στον Πραγματικό Κόσμο

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Μελέτη Περίπτωσης: RAG σε Εφαρμογή Προετοιμασίας Φορολογικής Δήλωσης Χωρίς Ενσωματώσεις

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ευφυής Βελτιστοποίηση Ερωτημάτων (IQO)

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Επαναταξινόμηση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Αξιολόγηση RAG (RAGAs)

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πιστότητα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Συνάφεια Απάντησης

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ακρίβεια Περιεχομένου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Συνάφεια Περιεχομένου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ανάκληση Περιεχομένου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ανάκληση Οντοτήτων Περιεχομένου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Σημασιολογική Ομοιότητα Απάντησης (ANSS)

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ορθότητα Απάντησης

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Κριτική Πτυχών

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προκλήσεις και Μελλοντικές Προοπτικές

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Σημασιολογική Τμηματοποίηση: Ενίσχυση της Ανάκτησης με Τμηματοποίηση με Επίγνωση Περιεχομένου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ιεραρχική Ευρετηρίαση: Δόμηση Δεδομένων για Βελτιωμένη Ανάκτηση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Αυτο-ΠΕΑ: Μια Αυτο-αναστοχαστική Βελτίωση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

HyDE: Ενσωματώσεις Υποθετικών Εγγράφων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Τι είναι η Αντιπαραβολική Μάθηση;

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πλήθος Εργατών



Μου αρέσει να σκέφτομαι τα στοιχεία TN μου ως μικρούς, σχεδόν ανθρώπινους εικονικούς “εργάτες” που μπορούν να ενσωματωθούν απρόσκοπτα στη λογική της εφαρμογής μου για να εκτελούν συγκεκριμένες εργασίες ή να λαμβάνουν πολύπλοκες αποφάσεις. Η ιδέα είναι να εξανθρωπίσουμε σκόπιμα τις δυνατότητες του ΜΜΓ, ώστε κανείς να μην ενθουσιαστεί *υπερβολικά* και να τους αποδώσει μαγικές ιδιότητες που δεν διαθέτουν.

Αντί να βασίζονται αποκλειστικά σε πολύπλοκους αλγόριθμους ή χρονοβόρες χειροκίνητες υλοποιήσεις, οι προγραμματιστές μπορούν να αντιληφθούν τα στοιχεία TN ως ευφυείς, αφοσιωμένες, ανθρωπόμορφες οντότητες που μπορούν να κληθούν όποτε χρειάζεται για να αντιμετωπίσουν πολύπλοκα προβλήματα και να παρέχουν λύσεις βασισμένες στην εκπαίδευση και τη γνώση τους. Αυτές οι οντότητες δεν αποσπώνται, ούτε παίρνουν αναρρωτική άδεια. Δεν αποφασίζουν αυθόρμητα να κάνουν τα πράγματα με διαφορετικούς τρόπους από αυτούς που

τους έχουν δοθεί οδηγίες, και γενικά, αν προγραμματιστούν σωστά, δεν κάνουν ούτε λάθη.

Με τεχνικούς όρους, η βασική αρχή πίσω από αυτή την προσέγγιση είναι η αποσύνθεση πολύπλοκων εργασιών ή διαδικασιών λήψης αποφάσεων σε μικρότερες, πιο διαχειρίσιμες μονάδες που μπορούν να χειριστούν από εξειδικευμένους εργάτες TN. Κάθε εργάτης είναι σχεδιασμένος να εστιάζει σε μια συγκεκριμένη πτυχή του προβλήματος, φέρνοντας τη μοναδική του εμπειρογνωμοσύνη και ικανότητες στο τραπέζι. Κατανέμοντας το φόρτο εργασίας μεταξύ πολλαπλών εργατών TN, η εφαρμογή μπορεί να επιτύχει μεγαλύτερη αποδοτικότητα, κλιμάκωση και προσαρμοστικότητα.

Για παράδειγμα, σκεφτείτε μια διαδικτυακή εφαρμογή που απαιτεί συντονισμό σε πραγματικό χρόνο του περιεχομένου που δημιουργείται από χρήστες. Η υλοποίηση ενός ολοκληρωμένου συστήματος συντονισμού από το μηδέν θα ήταν ένα τρομακτικό έργο, απαιτώντας σημαντική προσπάθεια ανάπτυξης και συνεχή συντήρηση. Ωστόσο, χρησιμοποιώντας την προσέγγιση του Πλήθους Εργατών, οι προγραμματιστές μπορούν να ενσωματώσουν εργάτες συντονισμού που τροφοδοτούνται από TN στη λογική της εφαρμογής. Αυτοί οι εργάτες μπορούν αυτόματα να αναλύουν και να επισημαίνουν ακατάλληλο περιεχόμενο, απελευθερώνοντας τους προγραμματιστές ώστε να επικεντρωθούν σε άλλες κρίσιμες πτυχές της εφαρμογής.

Εργάτες TN Ως Ανεξάρτητα

Επαναχρησιμοποιήσιμα Στοιχεία

Μια βασική πτυχή της προσέγγισης του Πλήθους Εργατών είναι η αρθρωτότητά της. Οι υποστηρικτές του αντικειμενοστρεφούς προγραμματισμού μας λένε εδώ και δεκαετίες να σκεφτόμαστε τις αλληλεπιδράσεις αντικειμένων ως μηνύματα. Λοιπόν, οι εργάτες TN μπορούν να σχεδιαστούν ως ανεξάρτητα, επαναχρησιμοποιήσιμα

στοιχεία που μπορούν να “μιλούν μεταξύ τους” μέσω απλών γλωσσικών μηνυμάτων, σχεδόν σαν να ήταν πραγματικά μικροί άνθρωποι που μιλούν μεταξύ τους. Αυτή η χαλαρά συνδεδεμένη προσέγγιση επιτρέπει στην εφαρμογή να προσαρμόζεται και να εξελίσσεται με την πάροδο του χρόνου, καθώς εμφανίζονται νέες τεχνολογίες ΤΝ ή αλλάζουν οι απαιτήσεις της επιχειρησιακής λογικής.

Στην πράξη, η ανάγκη σχεδιασμού ξεκάθαρων διεπαφών και πρωτοκόλλων επικοινωνίας μεταξύ των στοιχείων δεν έχει αλλάξει απλώς επειδή εμπλέκονται εργάτες ΤΝ. Πρέπει ακόμα να λαμβάνετε υπόψη άλλους παράγοντες όπως η απόδοση, η κλιμάκωση και η ασφάλεια, αλλά τώρα υπάρχουν και εντελώς νέες “ήπιες απαιτήσεις” προς εξέταση. Για παράδειγμα, πολλοί χρήστες αντιτίθενται στη χρήση των προσωπικών τους δεδομένων για την εκπαίδευση νέων μοντέλων ΤΝ. Έχετε επαληθεύσει το επίπεδο ιδιωτικότητας που παρέχεται από τον πάροχο του μοντέλου που χρησιμοποιείτε;

Οι Εργάτες ΤΝ ως Μικροϋπηρεσίες;

Καθώς διαβάζετε για την προσέγγιση του Πλήθους Εργατών, ίσως παρατηρήσετε κάποιες ομοιότητες με την αρχιτεκτονική Μικροϋπηρεσιών. Και οι δύο δίνουν έμφαση στην αποσύνθεση πολύπλοκων συστημάτων σε μικρότερες, πιο διαχειρίσιμες και ανεξάρτητα αναπτυσσόμενες μονάδες. Όπως ακριβώς οι μικροϋπηρεσίες είναι σχεδιασμένες να είναι χαλαρά συζευγμένες, εστιασμένες σε συγκεκριμένες επιχειρηματικές δυνατότητες και να επικοινωνούν μέσω καλά ορισμένων APIs, έτσι και οι εργάτες ΤΝ είναι σχεδιασμένοι να είναι αρθρωτοί, εξειδικευμένοι στις εργασίες τους και να αλληλεπιδρούν μεταξύ τους μέσω ξεκάθαρων διεπαφών και πρωτοκόλλων επικοινωνίας.

Ωστόσο, υπάρχουν κάποιες βασικές διαφορές που πρέπει να έχετε κατά νου. Ενώ οι μικροϋπηρεσίες συνήθως υλοποιούνται ως ξεχωριστές διεργασίες ή

υπηρεσίες που εκτελούνται σε διαφορετικές μηχανές ή containers, οι εργάτες TN μπορούν να υλοποιηθούν ως αυτόνομα στοιχεία εντός μιας μεμονωμένης εφαρμογής ή ως ξεχωριστές υπηρεσίες, ανάλογα με τις συγκεκριμένες απαιτήσεις και ανάγκες κλιμάκωσης. Επιπλέον, η επικοινωνία μεταξύ των εργατών TN συχνά περιλαμβάνει την ανταλλαγή πλούσιων πληροφοριών βασισμένων στη φυσική γλώσσα, όπως προτροπές, οδηγίες και παραγόμενο περιεχόμενο, αντί για τις πιο δομημένες μορφές δεδομένων που χρησιμοποιούνται συνήθως στις μικροϋπηρεσίες.

Παρά τις διαφορές αυτές, οι αρχές της αρθρωτότητας, της χαλαρής σύζευξης και των ξεκάθαρων διεπαφών επικοινωνίας παραμένουν κεντρικές και στα δύο πρότυπα. Εφαρμόζοντας αυτές τις αρχές στην αρχιτεκτονική των εργατών TN σας, μπορείτε να δημιουργήσετε ευέλικτα, κλιμακώσιμα και συντηρήσιμα συστήματα που αξιοποιούν τη δύναμη της TN για την επίλυση πολύπλοκων προβλημάτων και την παροχή αξίας στους χρήστες σας.

Η προσέγγιση του Πλήθους Εργατών μπορεί να εφαρμοστεί σε διάφορους τομείς και εφαρμογές, αξιοποιώντας τη δύναμη της TN για την αντιμετώπιση πολύπλοκων εργασιών και την παροχή έξυπνων λύσεων. Ας εξερευνήσουμε μερικά συγκεκριμένα παραδείγματα για το πώς μπορούν να χρησιμοποιηθούν οι εργάτες TN σε διαφορετικά πλαίσια.

Διαχείριση Λογαριασμών

Πρακτικά κάθε αυτόνομη διαδικτυακή εφαρμογή έχει την έννοια ενός λογαριασμού (ή χρήστη). Στην Olympia, χρησιμοποιούμε έναν εργάτη TN AccountManager που είναι προγραμματισμένος να μπορεί να χειρίζεται διάφορα είδη αιτημάτων αλλαγών που σχετίζονται με λογαριασμούς χρηστών.

Η οδηγία του διαβάζεται ως εξής:

```
1 You are an intelligent account manager for Olympia. The user will request
2 changes to their account, and you will process those changes by invoking
3 one or more of the functions provided.
4
5 The initial state of the account: #{account.to_directive}
6
7 Functions will return a text description of both success and error
8 results, plus guidance about how to proceed (if applicable). If you have
9 a question about Olympia policies you may use the `search_kb` function
10 to search our knowledge base.
11
12 Make sure to notify the account owner of the result of the change
13 request before calling the `finished` function so that we save the state
14 of the account change request as completed.
```

Η αρχική κατάσταση του λογαριασμού που παράγεται από το `account.to_directive` είναι απλά μια περιγραφή κειμένου του λογαριασμού, συμπεριλαμβανομένων των σχετικών δεδομένων όπως χρήστες, συνδρομές, κλπ.

Το εύρος των λειτουργιών που είναι διαθέσιμες στον AccountManager του δίνει τη δυνατότητα να επεξεργάζεται τη συνδρομή του χρήστη, να προσθέτει και να αφαιρεί συμβούλους AI και άλλους τύπους έμμισθων πρόσθετων, και να στέλνει ειδοποιήσεις μέσω email στον ιδιοκτήτη του λογαριασμού. Εκτός από τη λειτουργία `finished`, μπορεί επίσης να `notify_human_administrator` αν συναντήσει κάποιο σφάλμα κατά την επεξεργασία ή χρειάζεται οποιοδήποτε άλλο είδος βοήθειας με ένα αίτημα.

Σημειώστε ότι σε περίπτωση ερωτήσεων, ο AccountManager μπορεί να επιλέξει να αναζητήσει στη βάση γνώσεων της Olympia, όπου μπορεί να βρει οδηγίες για το χειρισμό οριακών περιπτώσεων και οποιασδήποτε άλλης κατάστασης που τον αφήνει αβέβαιο για το πώς να προχωρήσει.

Εφαρμογές Ηλεκτρονικού Εμπορίου

Στον τομέα του ηλεκτρονικού εμπορίου, οι εργάτες AI μπορούν να παίξουν καθοριστικό ρόλο στη βελτίωση της εμπειρίας χρήστη και στη βελτιστοποίηση των επιχειρησιακών λειτουργιών. Ακολουθούν μερικοί τρόποι με τους οποίους μπορούν να χρησιμοποιηθούν οι εργάτες AI:

Προτάσεις Προϊόντων

Μία από τις πιο ισχυρές εφαρμογές των εργατών AI στο ηλεκτρονικό εμπόριο είναι η δημιουργία εξατομικευμένων προτάσεων προϊόντων. Αναλύοντας τη συμπεριφορά του χρήστη, το ιστορικό αγορών και τις προτιμήσεις, αυτοί οι εργάτες μπορούν να προτείνουν προϊόντα που είναι προσαρμοσμένα στα ενδιαφέροντα και τις ανάγκες κάθε μεμονωμένου χρήστη.

Το κλειδί για αποτελεσματικές προτάσεις προϊόντων είναι η αξιοποίηση ενός συνδυασμού τεχνικών συνεργατικού φιλτραρίσματος και φιλτραρίσματος βάσει περιεχομένου. Το συνεργατικό φιλτράρισμα εξετάζει τη συμπεριφορά παρόμοιων χρηστών για να εντοπίσει μοτίβα και να κάνει προτάσεις με βάση το τι έχουν αγοράσει ή απολαύσει άλλοι με παρόμοιες προτιμήσεις. Το φιλτράρισμα βάσει περιεχομένου, από την άλλη πλευρά, επικεντρώνεται στα χαρακτηριστικά και τις ιδιότητες των ίδιων των προϊόντων, προτείνοντας είδη που μοιράζονται παρόμοια χαρακτηριστικά με εκείνα για τα οποία ένας χρήστης έχει δείξει ενδιαφέρον στο παρελθόν.

Ακολουθεί ένα απλοποιημένο παράδειγμα για το πώς μπορείτε να υλοποιήσετε έναν εργάτη προτάσεων προϊόντων σε Ruby, αυτή τη φορά χρησιμοποιώντας ένα [“Railway Oriented \(ROP\)”](#) στυλ συναρτησιακού προγραμματισμού:

```
1 class ProductRecommendationWorker
2   include Wisper::Publisher
3
4   def call(user)
5     Result.ok(ProductRecommendation.new(user))
6       .and_then(ValidateUser.method(:validate))
7       .map(AnalyzeCurrentSession.method(:analyze))
8       .map(CollaborativeFilter.method(:filter))
9       .map(ContentBasedFilter.method(:filter))
10      .map(ProductSelector.method(:select)).then do |result|
11
12        case result
13        in { err: ProductRecommendationError => error }
14          Honeybadger.notify(error.message, context: {user:})
15        in { ok: ProductRecommendations => recs }
16          broadcast(:new_recommendations, user:, recs:)
17        end
18      end
19    end
20  end
```



Το στυλ του συναρτησιακού προγραμματισμού Ruby που χρησιμοποιείται στο παράδειγμα επηρεάζεται από τις F# και Rust. Μπορείτε να διαβάσετε περισσότερα γι' αυτό στην [εξήγηση της τεχνικής](#) του φίλου μου Chad Wooley στο [GitLab](#)

Σε αυτό το παράδειγμα, ο ProductRecommendationWorker δέχεται έναν χρήστη ως είσοδο και δημιουργεί εξατομικευμένες προτάσεις προϊόντων περνώντας ένα αντικείμενο τιμής μέσα από μια αλυσίδα συναρτησιακών βημάτων. Ας αναλύσουμε κάθε βήμα:

1. ValidateUser.validate: Αυτό το βήμα διασφαλίζει ότι ο χρήστης είναι έγκυρος και κατάλληλος για εξατομικευμένες προτάσεις. Ελέγχει αν ο χρήστης υπάρχει, είναι ενεργός και έχει τα απαραίτητα δεδομένα διαθέσιμα για τη δημιουργία προτάσεων. Εάν η επικύρωση αποτύχει, επιστρέφεται ένα αποτέλεσμα σφάλματος και η αλυσίδα διακόπτεται πρόωρα.

2. `AnalyzeCurrentSession.analyze`: Εάν ο χρήστης είναι έγκυρος, αυτό το βήμα αναλύει την τρέχουσα περίοδο περιήγησης του χρήστη για να συλλέξει πληροφορίες πλαισίου. Εξετάζει τις πρόσφατες αλληλεπιδράσεις του χρήστη, όπως τα προϊόντα που προβλήθηκαν, τα ερωτήματα αναζήτησης και τα περιεχόμενα του καλαθιού, για να κατανοήσει τα τρέχοντα ενδιαφέροντα και την πρόθεσή του.
3. `CollaborativeFilter.filter`: Χρησιμοποιώντας τη *συμπεριφορά παρόμοιων χρηστών*, αυτό το βήμα εφαρμόζει τεχνικές συνεργατικού φιλτραρίσματος για να εντοπίσει προϊόντα που πιθανόν να ενδιαφέρουν τον χρήστη. Λαμβάνει υπόψη παράγοντες όπως το ιστορικό αγορών, τις αξιολογήσεις και τις αλληλεπιδράσεις χρήστη-προϊόντος για να δημιουργήσει ένα σύνολο υποψήφιων προτάσεων.
4. `ContentBasedFilter.filter`: Αυτό το βήμα βελτιώνει περαιτέρω τις υποψήφιες προτάσεις εφαρμόζοντας φιλτράρισμα βάσει περιεχομένου. Συγκρίνει τα χαρακτηριστικά και τις ιδιότητες των υποψήφιων προϊόντων με τις *προτιμήσεις και τα ιστορικά δεδομένα του χρήστη* για να επιλέξει τα πιο σχετικά στοιχεία.
5. `ProductSelector.select`: Τέλος, αυτό το βήμα επιλέγει τα κορυφαία N προϊόντα από τις φιλτραρισμένες προτάσεις με βάση προκαθορισμένα κριτήρια, όπως η βαθμολογία συνάφειας, η δημοτικότητα ή άλλοι επιχειρηματικοί κανόνες. Τα επιλεγμένα προϊόντα επιστρέφονται στη συνέχεια ως οι τελικές εξατομικευμένες προτάσεις.

Η ομορφιά της χρήσης ενός συναρτησιακού στυλ προγραμματισμού Ruby εδώ είναι ότι μας επιτρέπει να συνδέσουμε αυτά τα βήματα μεταξύ τους με έναν σαφή και συνοπτικό τρόπο. Κάθε βήμα εστιάζει σε μια συγκεκριμένη εργασία και επιστρέφει ένα αντικείμενο `Result`, το οποίο μπορεί να είναι είτε επιτυχία (`ok`) είτε σφάλμα (`err`). Εάν οποιοδήποτε βήμα συναντήσει σφάλμα, η αλυσίδα διακόπτεται πρόωρα και το σφάλμα μεταδίδεται στο τελικό αποτέλεσμα.

Στη δήλωση case στο τέλος, κάνουμε αντιστοίχιση προτύπων στο τελικό αποτέλεσμα. Εάν το αποτέλεσμα είναι σφάλμα (ProductRecommendationError), καταγράφουμε το σφάλμα χρησιμοποιώντας ένα εργαλείο όπως το Honeybadger για σκοπούς παρακολούθησης και αποσφαλμάτωσης. Εάν το αποτέλεσμα είναι επιτυχές (ProductRecommendations), μεταδίδουμε ένα γεγονός :new_recommendations χρησιμοποιώντας τη βιβλιοθήκη δημοσίευσης/εγγραφής Wisper, περνώντας μαζί τον χρήστη και τις προτάσεις που δημιουργήθηκαν.

Αξιοποιώντας τεχνικές συναρτησιακού προγραμματισμού, μπορούμε να δημιουργήσουμε έναν αρθρωτό και συντηρήσιμο εργάτη προτάσεων προϊόντων. Κάθε βήμα είναι αυτόνομο και μπορεί εύκολα να ελεγχθεί, να τροποποιηθεί ή να αντικατασταθεί χωρίς να επηρεάσει τη συνολική ροή. Η χρήση της αντιστοίχισης προτύπων και της κλάσης Result μας βοηθά να χειριστούμε τα σφάλματα με κομψό τρόπο και διασφαλίζει ότι ο εργάτης αποτυγχάνει γρήγορα εάν οποιοδήποτε βήμα αντιμετωπίσει πρόβλημα.

Φυσικά, αυτό είναι ένα απλοποιημένο παράδειγμα, και σε ένα πραγματικό σενάριο, θα χρειαζόταν να ενσωματώσετε την πλατφόρμα ηλεκτρονικού εμπορίου σας, να χειριστείτε οριακές περιπτώσεις, και ακόμη να εμβαθύνετε στην υλοποίηση των αλγορίθμων προτάσεων. Ωστόσο, οι βασικές αρχές της αποσύνθεσης του προβλήματος σε μικρότερα βήματα και της αξιοποίησης τεχνικών συναρτησιακού προγραμματισμού παραμένουν οι ίδιες.

Ανίχνευση Απάτης

Ακολουθεί ένα απλοποιημένο παράδειγμα για το πώς μπορείτε να υλοποιήσετε έναν εργάτη ανίχνευσης απάτης χρησιμοποιώντας το ίδιο στυλ Προγραμματισμού Προσανατολισμένου σε Σιδηροδρομικές Γραμμές (ROP) στη Ruby:

```
1 class FraudDetectionWorker
2   include Wisper::Publisher
3
4   def call(transaction)
5     Result.ok(FraudDetection.new(transaction))
6       .and_then(ValidateTransaction.method(:validate))
7       .map(AnalyzeTransactionPatterns.method(:analyze))
8       .map(CheckCustomerHistory.method(:check))
9       .map(EvaluateRiskFactors.method(:evaluate))
10      .map(DetermineFraudProbability.method(:determine)).then do |result|
11
12        case result
13        in { err: FraudDetectionError => error }
14          Honeybadger.notify(error.message, context: {transaction:})
15        in { ok: FraudDetection => fraud } }
16          if fraud.high_risk?
17            broadcast(:high_risk_transaction, transaction:, fraud:)
18          else
19            broadcast(:low_risk_transaction, transaction:)
20          end
21        end
22      end
23    end
24  end
```

Η κλάση `FraudDetection` είναι ένα *αντικείμενο τιμής* που ενθυλακώνει την κατάσταση ανίχνευσης απάτης για μια δεδομένη συναλλαγή. Παρέχει έναν δομημένο τρόπο για την ανάλυση και αξιολόγηση του κινδύνου απάτης που σχετίζεται με μια συναλλαγή με βάση διάφορους παράγοντες κινδύνου.

```
1  class FraudDetection
2    RISK_THRESHOLD = 0.8
3
4    attr_accessor :transaction, :risk_factors
5
6    def initialize(transaction)
7      self.transaction = transaction
8      self.risk_factors = []
9    end
10
11   def add_risk_factor(description:, probability:)
12     case { description:, probability: }
13     in { description: String => desc, probability: Float => prob }
14       risk_factors << { desc => prob }
15     else
16       raise ArgumentError, "Risk factor arguments should be string and float"
17     end
18   end
19
20   def high_risk?
21     fraud_probability > RISK_THRESHOLD
22   end
23
24   private
25
26   def fraud_probability
27     risk_factors.values.sum
28   end
29 end
```

Η κλάση FraudDetection έχει τα ακόλουθα χαρακτηριστικά:

- transaction: Μια αναφορά στη συναλλαγή που αναλύεται για απάτη.
- risk_factors: Ένας πίνακας που αποθηκεύει τους παράγοντες κινδύνου που σχετίζονται με τη συναλλαγή. Κάθε παράγοντας κινδύνου αναπαρίσταται ως πίνακας κατακερματισμού, όπου το κλειδί είναι η περιγραφή του παράγοντα κινδύνου και η τιμή είναι η πιθανότητα απάτης που σχετίζεται με αυτόν τον παράγοντα κινδύνου.

Η μέθοδος `add_risk_factor` επιτρέπει την προσθήκη ενός παράγοντα κινδύνου στον πίνακα `risk_factors`. Δέχεται δύο παραμέτρους: την `description`, που είναι μια συμβολοσειρά που περιγράφει τον παράγοντα κινδύνου, και την `probability`, που είναι ένας αριθμός κινητής υποδιαστολής που αντιπροσωπεύει την πιθανότητα απάτης που σχετίζεται με αυτόν τον παράγοντα κινδύνου. Χρησιμοποιούμε μια συνθήκη `case..in` για να κάνουμε απλό έλεγχο τύπων.

Η μέθοδος `high_risk?` που θα ελεγχθεί στο τέλος της αλυσίδας είναι μια κατηγορηματική μέθοδος που συγκρίνει την `fraud_probability` (που υπολογίζεται αθροίζοντας τις πιθανότητες όλων των παραγόντων κινδύνου) με το `RISK_THRESHOLD`.

Η κλάση `FraudDetection` παρέχει έναν καθαρό και ενθυλακωμένο τρόπο διαχείρισης της ανίχνευσης απάτης για μια συναλλαγή. Επιτρέπει την προσθήκη πολλαπλών παραγόντων κινδύνου, καθένας με τη δική του περιγραφή και πιθανότητα, και παρέχει μια μέθοδο για να προσδιορίσει εάν η συναλλαγή θεωρείται υψηλού κινδύνου με βάση την υπολογισμένη πιθανότητα απάτης. Η κλάση μπορεί εύκολα να ενσωματωθεί σε ένα μεγαλύτερο σύστημα ανίχνευσης απάτης, όπου διαφορετικά συστατικά μπορούν να συνεργαστούν για την αξιολόγηση και τον μετριασμό του κινδύνου δόλιων συναλλαγών.

Τέλος, δεδομένου ότι αυτό είναι ένα βιβλίο για προγραμματισμό με χρήση AI, ακολουθεί ένα παράδειγμα υλοποίησης της κλάσης `CheckCustomerHistory` που αξιοποιεί την επεξεργασία AI χρησιμοποιώντας τη μονάδα `ChatCompletion` της βιβλιοθήκης μου [Raix](#):

```
1  class CheckCustomerHistory
2    include Raix::ChatCompletion
3
4    attr_accessor :fraud_detection
5
6    INSTRUCTION = <<~END
7      You are an AI assistant tasked with checking a customer's transaction
8      history for potential fraud indicators. Given the current transaction
9      and the customer's past transactions, analyze the data to identify any
10     suspicious patterns or anomalies.
11
12     Consider factors such as the frequency of transactions, transaction
13     amounts, geographical locations, and any deviations from the customer's
14     typical behavior to generate a probability score as a float in the range
15     of 0 to 1 (with 1 being absolute certainty of fraud).
16
17     Output the results of your analysis, highlighting any red flags or areas
18     of concern in the following JSON format:
19
20     { description: <Summary of your findings>, probability: <Float> }
21   END
22
23   def self.check(fraud_detection)
24     new(fraud_detection).call
25   end
26
27   def call
28     chat_completion(json: true).tap do |result|
29       fraud_detection.add_risk_factor(**result)
30     end
31     Result.ok(fraud_detection)
32   rescue StandardError => e
33     Result.err(FraudDetectionError.new(e))
34   end
35
36   private
37
38   def initialize(fraud_detection)
39     self.fraud_detection = fraud_detection
40   end
41
42   def transcript
```

```
43     tx_history = fraud_detection.transaction.user.tx_history
44     [
45         { system: INSTRUCTION },
46         { user: "Transaction history: #{tx_history.to_json}" },
47         { assistant: "OK. Please provide the current transaction." },
48         { user: "Current transaction: #{fraud_detection.transaction.to_json}" }
49     ]
50     end
51 end
```

Σε αυτό το παράδειγμα, η κλάση `CheckCustomerHistory` ορίζει μια σταθερά `INSTRUCTION` που παρέχει συγκεκριμένες οδηγίες στο μοντέλο `TN` σχετικά με το πώς να αναλύσει το ιστορικό συναλλαγών του πελάτη για πιθανούς δείκτες απάτης μέσω μιας οδηγίας συστήματος.

Η μέθοδος `self.check` είναι μια μέθοδος κλάσης που αρχικοποιεί ένα νέο στιγμιότυπο του `CheckCustomerHistory` με το αντικείμενο `fraud_detection` και καλεί τη μέθοδο `call` για να εκτελέσει την ανάλυση του ιστορικού του πελάτη.

Μέσα στη μέθοδο `call`, ανακτάται το ιστορικό συναλλαγών του πελάτη και μορφοποιείται σε ένα κείμενο που περνάει στο μοντέλο `TN`. Το μοντέλο `TN` αναλύει το ιστορικό συναλλαγών με βάση τις παρεχόμενες οδηγίες και επιστρέφει μια περίληψη των ευρημάτων του.

Τα ευρήματα προστίθενται στο αντικείμενο `fraud_detection`, και το ενημερωμένο αντικείμενο `fraud_detection` επιστρέφεται ως επιτυχές `Result`.

Αξιοποιώντας τη μονάδα `ChatCompletion`, η κλάση `CheckCustomerHistory` μπορεί να χρησιμοποιήσει τη δύναμη της `TN` για να αναλύσει το ιστορικό συναλλαγών του πελάτη και να εντοπίσει πιθανούς δείκτες απάτης. Αυτό επιτρέπει πιο εξελιγμένες και προσαρμοστικές τεχνικές ανίχνευσης απάτης, καθώς το μοντέλο `TN` μπορεί να μαθαίνει και να προσαρμόζεται σε νέα πρότυπα και ανωμαλίες με την πάροδο του χρόνου.

Ο ενημερωμένος `FraudDetectionWorker` και η κλάση `CheckCustomerHistory` δείχνουν πώς οι εργάτες `TN` μπορούν να ενσωματωθούν απρόσκοπτα, ενισχύοντας

τη διαδικασία ανίχνευσης απάτης με έξυπνες δυνατότητες ανάλυσης και λήψης αποφάσεων.

Ανάλυση Συναισθήματος Πελατών

Εδώ είναι ένα ακόμη παρόμοιο παράδειγμα για το πώς μπορείτε να υλοποιήσετε έναν εργάτη ανάλυσης συναισθήματος πελατών. Πολύ λιγότερη εξήγηση αυτή τη φορά, καθώς θα πρέπει να έχετε καταλάβει πώς λειτουργεί αυτό το στυλ προγραμματισμού:

```
1 class CustomerSentimentAnalysisWorker
2   include Wisper::Publisher
3
4   def call(feedback)
5     Result.ok(feedback)
6       .and_then(PreprocessFeedback.method(:preprocess))
7       .map(PerformSentimentAnalysis.method(:analyze))
8       .map(ExtractKeyPhrases.method(:extract))
9       .map(IdentifyTrends.method(:identify))
10      .map(GenerateInsights.method(:generate)).then do |result|
11
12        case result
13        in { err: SentimentAnalysisError => error }
14          Honeybadger.notify(error.message, context: {feedback:})
15        in { ok: SentimentAnalysisResult => result }
16          broadcast(:sentiment_analysis_completed, result)
17        end
18      end
19    end
20  end
```

Σε αυτό το παράδειγμα, ο CustomerSentimentAnalysisWorker τα βήματα περιλαμβάνουν την προεπεξεργασία των σχολίων (π.χ., αφαίρεση θορύβου, τμηματοποίηση), εκτέλεση ανάλυσης συναισθήματος για τον προσδιορισμό του συνολικού συναισθήματος (θετικό, αρνητικό ή ουδέτερο), εξαγωγή βασικών φράσεων και θεμάτων, αναγνώριση τάσεων και μοτίβων, και δημιουργία εφαρμόσιμων συμπερασμάτων βάσει της ανάλυσης.

Εφαρμογές στην Υγειονομική Περίθαλψη

Στον τομέα της υγειονομικής περίθαλψης, οι εργάτες ΑΙ μπορούν να βοηθήσουν τους επαγγελματίες υγείας και τους ερευνητές σε διάφορες εργασίες, οδηγώντας σε βελτιωμένα αποτελέσματα για τους ασθενείς και επιταχυνόμενες ιατρικές ανακαλύψεις. Μερικά παραδείγματα περιλαμβάνουν:

Εισαγωγή Ασθενών

Οι εργάτες ΑΙ μπορούν να απλοποιήσουν τη διαδικασία εισαγωγής ασθενών αυτοματοποιώντας διάφορες εργασίες και παρέχοντας έξυπνη υποστήριξη.

Προγραμματισμός Ραντεβού: Οι εργάτες ΑΙ μπορούν να διαχειριστούν τον προγραμματισμό ραντεβού κατανοώντας τις προτιμήσεις των ασθενών, τη διαθεσιμότητα και το επείγον των ιατρικών τους αναγκών. Μπορούν να αλληλεπιδρούν με τους ασθενείς μέσω διεπαφών συνομιλίας, καθοδηγώντας τους στη διαδικασία προγραμματισμού και εντοπίζοντας τις καταλληλότερες διαθέσιμες ώρες με βάση τις απαιτήσεις του ασθενούς και τη διαθεσιμότητα του παρόχου υγειονομικής περίθαλψης.

Συλλογή Ιατρικού Ιστορικού: Κατά την εισαγωγή ασθενών, οι εργάτες ΑΙ μπορούν να βοηθήσουν στη συλλογή και τεκμηρίωση του ιατρικού ιστορικού του ασθενούς. Μπορούν να συμμετέχουν σε διαδραστικούς διαλόγους με τους ασθενείς, κάνοντας σχετικές ερωτήσεις για τις προηγούμενες ιατρικές τους καταστάσεις, φάρμακα, αλλεργίες και οικογενειακό ιστορικό. Οι εργάτες ΑΙ μπορούν να χρησιμοποιούν τεχνικές επεξεργασίας φυσικής γλώσσας για να ερμηνεύσουν και να δομήσουν τις συλλεχθείσες πληροφορίες, διασφαλίζοντας ότι καταγράφονται με ακρίβεια στον ηλεκτρονικό φάκελο υγείας του ασθενούς.

Αξιολόγηση και Διαστρωμάτωση Συμπτωμάτων: Οι εργάτες ΑΙ μπορούν να διεξάγουν αρχικές αξιολογήσεις συμπτωμάτων ρωτώντας τους ασθενείς για τα

τρέχοντα συμπτώματά τους, τη διάρκεια, τη σοβαρότητα και τυχόν σχετικούς παράγοντες. Αξιοποιώντας ιατρικές βάσεις γνώσεων και μοντέλα μηχανικής μάθησης, αυτοί οι εργάτες μπορούν να αναλύσουν τις παρεχόμενες πληροφορίες και να δημιουργήσουν προκαταρκτικές διαφορικές διαγνώσεις ή να προτείνουν κατάλληλα επόμενα βήματα, όπως ο προγραμματισμός διαβούλευσης με πάροχο υγειονομικής περίθαλψης ή η πρόταση μέτρων αυτοφροντίδας.

Επαλήθευση Ασφάλισης: Οι εργάτες ΑΙ μπορούν να βοηθήσουν στην επαλήθευση ασφάλισης κατά την εισαγωγή ασθενών. Μπορούν να συλλέξουν στοιχεία ασφάλισης του ασθενούς, να επικοινωνήσουν με τους ασφαλιστικούς φορείς μέσω APIs ή διαδικτυακών υπηρεσιών και να επαληθεύσουν την επιλεξιμότητα κάλυψης και τις παροχές. Αυτή η αυτοματοποίηση βοηθά στην απλοποίηση της διαδικασίας επαλήθευσης ασφάλισης, μειώνοντας το διοικητικό φόρτο και διασφαλίζοντας την ακριβή καταγραφή πληροφοριών.

Εκπαίδευση Ασθενών και Οδηγίες: Οι εργάτες ΑΙ μπορούν να παρέχουν στους ασθενείς σχετικό εκπαιδευτικό υλικό και οδηγίες με βάση τις συγκεκριμένες ιατρικές τους καταστάσεις ή επερχόμενες διαδικασίες. Μπορούν να παρέχουν εξατομικευμένο περιεχόμενο, να απαντούν σε συχνές ερωτήσεις και να προσφέρουν καθοδήγηση σχετικά με προετοιμασίες πριν το ραντεβού, οδηγίες φαρμάκων ή φροντίδα μετά τη θεραπεία. Αυτό βοηθά στη διατήρηση των ασθενών ενημερωμένων και εμπλεκόμενων καθ' όλη τη διάρκεια του ταξιδιού τους στην υγειονομική περίθαλψη.

Με την αξιοποίηση εργατών ΤΝ στην εισαγωγή ασθενών, οι οργανισμοί υγειονομικής περίθαλψης μπορούν να βελτιώσουν την αποδοτικότητα, να μειώσουν τους χρόνους αναμονής και να βελτιώσουν τη συνολική εμπειρία των ασθενών. Αυτοί οι εργάτες μπορούν να διαχειριστούν συνηθισμένες εργασίες, να συλλέξουν ακριβείς πληροφορίες και να παρέχουν εξατομικευμένη βοήθεια, επιτρέποντας στους επαγγελματίες υγείας να επικεντρωθούν στην παροχή υψηλής ποιότητας φροντίδας στους ασθενείς.

Αξιολόγηση Κινδύνου Ασθενών

Οι εργάτες ΤΝ μπορούν να διαδραματίσουν καθοριστικό ρόλο στην αξιολόγηση του κινδύνου των ασθενών αναλύοντας διάφορες πηγές δεδομένων και εφαρμόζοντας προηγμένες τεχνικές ανάλυσης.

Ενσωμάτωση Δεδομένων: Οι εργάτες ΤΝ μπορούν να συγκεντρώσουν και να κατανοήσουν δεδομένα ασθενών από πολλαπλές πηγές, όπως ηλεκτρονικούς φακέλους υγείας (ΗΦΥ), ιατρική απεικόνιση, εργαστηριακά αποτελέσματα, φορητές συσκευές και κοινωνικούς προσδιοριστές υγείας. Συγκεντρώνοντας αυτές τις πληροφορίες σε ένα ολοκληρωμένο προφίλ ασθενούς, οι εργάτες ΤΝ μπορούν να παρέχουν μια ολιστική εικόνα της κατάστασης υγείας και των παραγόντων κινδύνου του ασθενούς.

Διαστρωμάτωση Κινδύνου: Οι εργάτες ΤΝ μπορούν να χρησιμοποιήσουν προγνωστικά μοντέλα για να κατηγοριοποιήσουν τους ασθενείς σε διαφορετικές κατηγορίες κινδύνου με βάση τα ατομικά χαρακτηριστικά και τα δεδομένα υγείας τους. Αυτή η διαστρωμάτωση κινδύνου επιτρέπει στους παρόχους υγειονομικής περίθαλψης να δώσουν προτεραιότητα στους ασθενείς που χρειάζονται πιο άμεση προσοχή ή παρέμβαση. Για παράδειγμα, οι ασθενείς που εντοπίζονται ως υψηλού κινδύνου για μια συγκεκριμένη πάθηση μπορούν να επισημανθούν για στενότερη παρακολούθηση, προληπτικά μέτρα ή έγκαιρη παρέμβαση.

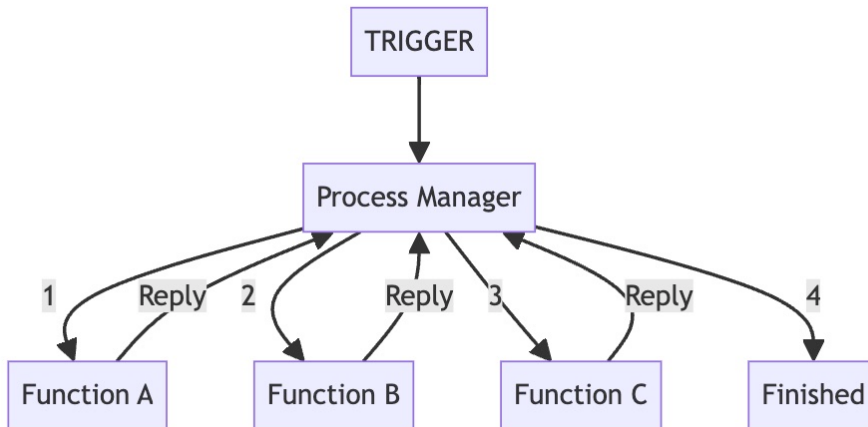
Εξατομικευμένα Προφίλ Κινδύνου: Οι εργάτες ΤΝ μπορούν να δημιουργήσουν εξατομικευμένα προφίλ κινδύνου για κάθε ασθενή, επισημαίνοντας τους συγκεκριμένους παράγοντες που συμβάλλουν στις βαθμολογίες κινδύνου τους. Αυτά τα προφίλ μπορούν να περιλαμβάνουν πληροφορίες σχετικά με τον τρόπο ζωής του ασθενούς, τις γενετικές προδιαθέσεις, τους περιβαλλοντικούς παράγοντες και τους κοινωνικούς προσδιοριστές υγείας. Παρέχοντας μια λεπτομερή ανάλυση των παραγόντων κινδύνου, οι εργάτες ΤΝ μπορούν να βοηθήσουν τους παρόχους υγειονομικής περίθαλψης να προσαρμόσουν τις στρατηγικές πρόληψης και τα σχέδια θεραπείας στις ατομικές ανάγκες των ασθενών.

Συνεχής Παρακολούθηση Κινδύνου: Οι εργάτες TN μπορούν να παρακολουθούν συνεχώς τα δεδομένα των ασθενών και να ενημερώνουν τις αξιολογήσεις κινδύνου σε πραγματικό χρόνο. Καθώς γίνονται διαθέσιμες νέες πληροφορίες, όπως αλλαγές στα ζωτικά σημεία, εργαστηριακά αποτελέσματα ή συμμόρφωση με τη φαρμακευτική αγωγή, οι εργάτες TN μπορούν να επανυπολογίζουν τις βαθμολογίες κινδύνου και να ειδοποιούν τους παρόχους υγειονομικής περίθαλψης για τυχόν σημαντικές αλλαγές. Αυτή η προληπτική παρακολούθηση επιτρέπει έγκαιρες παρεμβάσεις και προσαρμογές στα σχέδια φροντίδας των ασθενών.

Υποστήριξη Κλινικών Αποφάσεων: Οι εργάτες TN μπορούν να ενσωματώσουν τα αποτελέσματα αξιολόγησης κινδύνου σε συστήματα υποστήριξης κλινικών αποφάσεων, παρέχοντας στους παρόχους υγειονομικής περίθαλψης τεκμηριωμένες συστάσεις και ειδοποιήσεις. Για παράδειγμα, εάν η βαθμολογία κινδύνου ενός ασθενούς για μια συγκεκριμένη πάθηση υπερβεί ένα συγκεκριμένο όριο, ο εργάτης TN μπορεί να προτρέψει τον πάροχο υγειονομικής περίθαλψης να εξετάσει συγκεκριμένες διαγνωστικές εξετάσεις, προληπτικά μέτρα ή επιλογές θεραπείας με βάση τις κλινικές κατευθυντήριες γραμμές και τις βέλτιστες πρακτικές.

Αυτοί οι εργάτες μπορούν να επεξεργαστούν τεράστιες ποσότητες δεδομένων ασθενών, να εφαρμόσουν εξελιγμένες αναλύσεις και να δημιουργήσουν αξιοποιήσιμες πληροφορίες για την υποστήριξη της κλινικής λήψης αποφάσεων. Αυτό τελικά οδηγεί σε βελτιωμένα αποτελέσματα για τους ασθενείς, μειωμένο κόστος υγειονομικής περίθαλψης και ενισχυμένη διαχείριση της υγείας του πληθυσμού.

Εργάτης AI ως Διαχειριστής Διεργασιών



Στο πλαίσιο των εφαρμογών που βασίζονται στην τεχνητή νοημοσύνη, ένας εργάτης μπορεί να σχεδιαστεί ώστε να λειτουργεί ως Διαχειριστής Διεργασιών, όπως περιγράφεται στο βιβλίο “Enterprise Integration Patterns” του Gregor Hohpe. Ένας Διαχειριστής Διεργασιών είναι ένα κεντρικό συστατικό που διατηρεί την κατάσταση μιας διαδικασίας και καθορίζει τα επόμενα βήματα επεξεργασίας με βάση τα ενδιάμεσα αποτελέσματα.

Όταν ένας εργάτης AI λειτουργεί ως Διαχειριστής Διεργασιών, λαμβάνει ένα εισερχόμενο μήνυμα που αρχικοποιεί τη διαδικασία, γνωστό ως *μήνυμα ενεργοποίησης*. Ο εργάτης AI στη συνέχεια διατηρεί την κατάσταση της εκτέλεσης της διαδικασίας (ως καταγραφή συνομιλίας) και χειρίζεται το μήνυμα μέσω μιας σειράς βημάτων επεξεργασίας που υλοποιούνται ως λειτουργίες εργαλείων, οι οποίες μπορούν να είναι σειριακές ή παράλληλες, και καλούνται κατά τη διακριτική του ευχέρεια.



Αν χρησιμοποιείτε μια κατηγορία μοντέλου AI όπως το GPT-4 που γνωρίζει πώς να εκτελεί λειτουργίες παράλληλα, τότε ο εργάτης σας μπορεί να εκτελεί πολλαπλά βήματα ταυτόχρονα. Ομολογουμένως, δεν το έχω δοκιμάσει ο ίδιος και το ένστικτό μου λέει ότι τα αποτελέσματα μπορεί να ποικίλουν.

Μετά από κάθε μεμονωμένο βήμα επεξεργασίας, ο έλεγχος επιστρέφει στον εργάτη AI, επιτρέποντάς του να καθορίσει τα επόμενα βήματα επεξεργασίας με βάση την τρέχουσα κατάσταση και τα αποτελέσματα που έχουν ληφθεί.

Αποθηκεύστε τα Μηνύματα Ενεργοποίησής σας

Από την εμπειρία μου, είναι έξυπνο να υλοποιήσετε το μήνυμα ενεργοποίησής σας ως αντικείμενο υποστηριζόμενο από βάση δεδομένων. Με αυτόν τον τρόπο, κάθε στιγμιότυπο διαδικασίας αναγνωρίζεται από ένα μοναδικό πρωτεύον κλειδί και σας παρέχει ένα μέρος για να αποθηκεύσετε την κατάσταση που σχετίζεται με την εκτέλεση, συμπεριλαμβανομένης της καταγραφής συνομιλίας του AI.

Για παράδειγμα, εδώ είναι μια απλοποιημένη έκδοση της κλάσης μοντέλου AccountChange της Olympia, η οποία αντιπροσωπεύει ένα αίτημα για την πραγματοποίηση αλλαγής στον λογαριασμό ενός χρήστη.

```
1  # == Schema Information
2  #
3  # Table name: account_changes
4  #
5  #   id          :uuid          not null, primary key
6  #   description :string
7  #   state       :string       not null
8  #   transcript  :jsonb
9  #   created_at  :datetime     not null
10 #   updated_at  :datetime     not null
11 #   account_id  :uuid         not null
12 #
13 # Indexes
14 #
15 #   index_account_changes_on_account_id (account_id)
16 #
17 # Foreign Keys
18 #
19 #   fk_rails_... (account_id => accounts.id)
20 #
21 class AccountChange < ApplicationRecord
22   belongs_to :account
23
24   validates :description, presence: true
25
26   after_commit -> {
27     broadcast(:account_change_requested, self)
28   }, on: :create
29
30   state_machine initial: :requested do
31     event :completed do
32       transition all => :complete
33     end
34     event :failed do
35       transition all => :requires_human_review
36     end
37   end
38 end
```

Η κλάση `AccountChange` λειτουργεί ως μήνυμα ενεργοποίησης που ξεκινά μια διαδικασία χειρισμού του αιτήματος αλλαγής λογαριασμού. Παρατηρήστε πώς

μεταδίδεται στο υποσύστημα δημοσίευσης/συνδρομής της Olympia που βασίζεται στο [Wisper](#) μετά την ολοκλήρωση της συναλλαγής δημιουργίας.

Η αποθήκευση του μηνύματος ενεργοποίησης στη βάση δεδομένων με αυτόν τον τρόπο παρέχει ένα μόνιμο αρχείο για κάθε αίτημα αλλαγής λογαριασμού. Κάθε στιγμιότυπο της κλάσης AccountChange λαμβάνει ένα μοναδικό πρωτεύον κλειδί, επιτρέποντας την εύκολη αναγνώριση και παρακολούθηση μεμονωμένων αιτημάτων. Αυτό είναι ιδιαίτερα χρήσιμο για σκοπούς καταγραφής ελέγχου, καθώς επιτρέπει στο σύστημα να διατηρεί ένα ιστορικό αρχείο όλων των αλλαγών λογαριασμού, συμπεριλαμβανομένου του πότε ζητήθηκαν, ποιες αλλαγές ζητήθηκαν και την τρέχουσα κατάσταση κάθε αιτήματος.

Στο δοθέν παράδειγμα, η κλάση AccountChange περιλαμβάνει πεδία όπως το description για την καταγραφή των λεπτομερειών της ζητούμενης αλλαγής, το state για την αναπαράσταση της τρέχουσας κατάστασης του αιτήματος (π.χ., requested, complete, requires_human_review) και το transcript για την αποθήκευση του μεταγραφικού της συνομιλίας με την TN σχετικά με το αίτημα. Το πεδίο description είναι το πραγματικό prompt που χρησιμοποιείται για την έναρξη της πρώτης ολοκλήρωσης συνομιλίας με την TN. Η αποθήκευση αυτών των δεδομένων παρέχει πολύτιμο πλαίσιο και επιτρέπει καλύτερη παρακολούθηση και ανάλυση της διαδικασίας αλλαγής λογαριασμού.

Η αποθήκευση μηνυμάτων ενεργοποίησης στη βάση δεδομένων επιτρέπει τον ισχυρό χειρισμό σφαλμάτων και την ανάκτηση. Εάν προκύψει σφάλμα κατά την επεξεργασία ενός αιτήματος αλλαγής λογαριασμού, το σύστημα επισημαίνει το αίτημα ως αποτυχημένο και το μεταβαίνει σε μια κατάσταση που απαιτεί ανθρώπινη παρέμβαση. Αυτό διασφαλίζει ότι κανένα αίτημα δεν χάνεται ή ξεχνιέται, και οποιαδήποτε ζητήματα μπορούν να αντιμετωπιστούν και να επιλυθούν κατάλληλα.

Ο εργατής TN, ως Διαχειριστής Διαδικασιών, παρέχει ένα κεντρικό σημείο ελέγχου και επιτρέπει ισχυρές δυνατότητες αναφοράς και αποσφαλμάτωσης διαδικασιών.

Ωστόσο, είναι σημαντικό να σημειωθεί ότι η χρήση ενός εργάτη TN ως Διαχειριστή Διαδικασιών για κάθε σενάριο ροής εργασίας στην εφαρμογή σας μπορεί να είναι υπερβολική.

Ενσωμάτωση Εργατών TN στην Αρχιτεκτονική της Εφαρμογής σας

Κατά την ενσωμάτωση εργατών TN στην αρχιτεκτονική της εφαρμογής σας, πρέπει να αντιμετωπιστούν διάφορα τεχνικά ζητήματα για να διασφαλιστεί η ομαλή ενσωμάτωση και η αποτελεσματική επικοινωνία μεταξύ των εργατών TN και άλλων στοιχείων της εφαρμογής. Αυτή η ενότητα εξετάζει βασικές πτυχές του σχεδιασμού αυτών των διεπαφών, της διαχείρισης ροής δεδομένων και της διαχείρισης του κύκλου ζωής των εργατών TN.

Σχεδιασμός Σαφών Διεπαφών και Πρωτοκόλλων Επικοινωνίας

Για τη διευκόλυνση της απρόσκοπτης ενσωμάτωσης μεταξύ των εργατών TN και άλλων στοιχείων της εφαρμογής, είναι κρίσιμο να καθοριστούν σαφείς διεπαφές και πρωτόκολλα επικοινωνίας. Εξετάστε τις ακόλουθες προσεγγίσεις:

Ενσωμάτωση βασισμένη σε API: Εκθέστε τη λειτουργικότητα των εργατών AI μέσω καλά καθορισμένων API, όπως τελικά σημεία RESTful ή σχήματα GraphQL. Αυτό επιτρέπει σε άλλα συστατικά να αλληλεπιδρούν με τους εργάτες AI χρησιμοποιώντας τυπικά αιτήματα και αποκρίσεις HTTP. Η ενσωμάτωση βασισμένη σε API παρέχει ένα σαφές συμβόλαιο μεταξύ των εργατών AI και των καταναλωτών συστατικών, διευκολύνοντας την ανάπτυξη, τον έλεγχο και τη συντήρηση των σημείων ενσωμάτωσης.

Επικοινωνία βασισμένη σε μηνύματα: Υλοποιήστε μοτίβα επικοινωνίας βασισμένα σε μηνύματα, όπως ουρές μηνυμάτων ή συστήματα δημοσίευσης-συνδρομής, για να ενεργοποιήσετε ασύγχρονη αλληλεπίδραση μεταξύ των εργατών ΑΙ και άλλων συστατικών. Αυτή η προσέγγιση αποσυνδέει τους εργάτες ΑΙ από την υπόλοιπη εφαρμογή, επιτρέποντας καλύτερη κλιμάκωση, ανοχή σφαλμάτων και χαλαρή σύζευξη. Η επικοινωνία βασισμένη σε μηνύματα είναι ιδιαίτερα χρήσιμη όταν η επεξεργασία που εκτελείται από τους εργάτες ΑΙ είναι χρονοβόρα ή απαιτητική σε πόρους, καθώς επιτρέπει σε άλλα μέρη της εφαρμογής να συνεχίσουν την εκτέλεση χωρίς να περιμένουν την ολοκλήρωση των εργασιών των εργατών ΑΙ.

Αρχιτεκτονική βασισμένη σε συμβάντα: Σχεδιάστε το σύστημά σας γύρω από συμβάντα και εναύσματα που ενεργοποιούν τους εργάτες ΑΙ όταν πληρούνται συγκεκριμένες συνθήκες. Οι εργάτες ΑΙ μπορούν να εγγραφούν σε σχετικά συμβάντα και να αντιδρούν ανάλογα, εκτελώντας τις καθορισμένες εργασίες τους όταν συμβαίνουν τα συμβάντα. Η αρχιτεκτονική βασισμένη σε συμβάντα επιτρέπει επεξεργασία σε πραγματικό χρόνο και επιτρέπει στους εργάτες ΑΙ να καλούνται κατ' απαίτηση, μειώνοντας την περιττή κατανάλωση πόρων. Αυτή η προσέγγιση είναι κατάλληλη για σενάρια όπου οι εργάτες ΑΙ πρέπει να ανταποκρίνονται σε συγκεκριμένες ενέργειες ή αλλαγές στην κατάσταση της εφαρμογής.

Διαχείριση Ροής Δεδομένων και Συγχρονισμός

Κατά την ενσωμάτωση εργατών ΑΙ στην εφαρμογή σας, είναι κρίσιμο να διασφαλίσετε την ομαλή ροή δεδομένων και να διατηρήσετε τη συνέπεια των δεδομένων μεταξύ των εργατών ΑΙ και άλλων συστατικών. Εξετάστε τις ακόλουθες πτυχές:

Προετοιμασία Δεδομένων: Πριν τροφοδοτήσετε δεδομένα στους εργάτες ΑΙ, ενδέχεται να χρειαστεί να εκτελέσετε διάφορες εργασίες προετοιμασίας δεδομένων, όπως καθαρισμό, μορφοποίηση ή/και μετασχηματισμό των δεδομένων εισόδου. Δεν θέλετε μόνο να διασφαλίσετε ότι οι εργάτες ΑΙ μπορούν να επεξεργαστούν

αποτελεσματικά, αλλά επίσης θέλετε να βεβαιωθείτε ότι δεν σπαταλάτε tokens δίνοντας προσοχή σε πληροφορίες που ο εργάτης μπορεί να θεωρήσει άχρηστες στην καλύτερη περίπτωση, ή αποπροσανατολιστικές στη χειρότερη. Η προετοιμασία δεδομένων μπορεί να περιλαμβάνει εργασίες όπως η αφαίρεση θορύβου, η διαχείριση ελλιπών τιμών ή η μετατροπή τύπων δεδομένων.

Διατήρηση Δεδομένων: Πώς θα αποθηκεύσετε και θα διατηρήσετε τα δεδομένα που ρέουν προς και από τους εργάτες AI; Εξετάστε παράγοντες όπως ο όγκος δεδομένων, τα μοτίβα ερωτημάτων και η κλιμάκωση. Χρειάζεται να διατηρήσετε το κείμενο του AI ως αντανάκλαση της “διαδικασίας σκέψης” του για σκοπούς ελέγχου ή αποσφαλμάτωσης, ή αρκεί να έχετε μόνο ένα αρχείο των αποτελεσμάτων;

Ανάκτηση Δεδομένων: Η λήψη των δεδομένων που χρειάζονται οι εργάτες μπορεί να περιλαμβάνει ερωτήματα σε βάσεις δεδομένων, ανάγνωση αρχείων ή πρόσβαση σε εξωτερικά APIs. Βεβαιωθείτε ότι λαμβάνετε υπόψη τον χρόνο καθυστέρησης και τον τρόπο με τον οποίο οι εργάτες TN θα έχουν πρόσβαση στα πιο ενημερωμένα δεδομένα. Χρειάζονται πλήρη πρόσβαση στη βάση δεδομένων σας ή θα πρέπει να ορίσετε στενά το πεδίο πρόσβασής τους σύμφωνα με αυτό που κάνουν; Τι γίνεται με την κλιμάκωση; Εξετάστε μηχανισμούς προσωρινής αποθήκευσης για τη βελτίωση της απόδοσης και τη μείωση του φόρτου στις υποκείμενες πηγές δεδομένων.

Συγχρονισμός Δεδομένων: Όταν πολλαπλά στοιχεία, συμπεριλαμβανομένων των εργατών TN, έχουν πρόσβαση και τροποποιούν κοινόχρηστα δεδομένα, είναι σημαντικό να εφαρμόζονται κατάλληλοι μηχανισμοί συγχρονισμού για τη διατήρηση της συνέπειας των δεδομένων. Οι στρατηγικές κλειδώματος βάσεων δεδομένων, όπως το αισιόδοξο ή απαισιόδοξο κλείδωμα, μπορούν να σας βοηθήσουν να αποτρέψετε συγκρούσεις και να διασφαλίσετε την ακεραιότητα των δεδομένων. Εφαρμόστε τεχνικές διαχείρισης συναλλαγών για την ομαδοποίηση σχετικών λειτουργιών δεδομένων και τη διατήρηση των ιδιοτήτων ACID (Ατομικότητα, Συνέπεια, Απομόνωση και Ανθεκτικότητα).

Διαχείριση Σφαλμάτων και Ανάκαμψη: Εφαρμόστε ισχυρούς μηχανισμούς

διαχείρισης σφαλμάτων και ανάκαμψης για την αντιμετώπιση ζητημάτων που σχετίζονται με τα δεδομένα και μπορεί να προκύψουν κατά τη διαδικασία ροής δεδομένων. Χειριστείτε τις εξαιρέσεις με κομψό τρόπο και παρέχετε ουσιαστικά μηνύματα σφάλματος για να βοηθήσετε στην αποσφαλμάτωση. Εφαρμόστε μηχανισμούς επανάληψης και στρατηγικές εφεδρείας για τη διαχείριση προσωρινών αποτυχιών ή διακοπών δικτύου. Καθορίστε σαφείς διαδικασίες για την ανάκτηση και αποκατάσταση δεδομένων σε περίπτωση καταστροφής ή απώλειας δεδομένων.

Με προσεκτικό σχεδιασμό και εφαρμογή μηχανισμών ροής και συγχρονισμού δεδομένων, μπορείτε να διασφαλίσετε ότι οι εργάτες TN σας έχουν πρόσβαση σε ακριβή, συνεπή και ενημερωμένα δεδομένα. Αυτό τους επιτρέπει να εκτελούν τα καθήκοντά τους αποτελεσματικά και να παράγουν αξιόπιστα αποτελέσματα.

Διαχείριση του Κύκλου Ζωής των Εργατών TN

Αναπτύξτε μια τυποποιημένη διαδικασία για την αρχικοποίηση και διαμόρφωση των εργατών TN. Προτιμώ πλαίσια που τυποποιούν τον τρόπο καθορισμού ρυθμίσεων όπως ονόματα μοντέλων, οδηγίες συστήματος και ορισμούς συναρτήσεων. Βεβαιωθείτε ότι η διαδικασία αρχικοποίησης είναι αυτοματοποιημένη και αναπαραγώγιμη για να διευκολύνει την ανάπτυξη και την κλιμάκωση.

Εφαρμόστε ολοκληρωμένους μηχανισμούς παρακολούθησης και καταγραφής για την παρακολούθηση της υγείας και της απόδοσης των εργατών TN. Συλλέξτε μετρήσεις όπως η χρήση πόρων, ο χρόνος επεξεργασίας, τα ποσοστά σφαλμάτων και η διεκπεραιωτική ικανότητα. Χρησιμοποιήστε κεντρικά συστήματα καταγραφής όπως η στοίβα ELK (Elasticsearch, Logstash, Kibana) για τη συγκέντρωση και ανάλυση καταγραφών από πολλαπλούς εργάτες TN.

Ενσωματώστε την ανοχή σφαλμάτων και την ανθεκτικότητα στην αρχιτεκτονική του εργάτη AI. Εφαρμόστε μηχανισμούς χειρισμού σφαλμάτων και ανάκτησης για την ομαλή διαχείριση αποτυχιών ή εξαιρέσεων. Τα Μοντέλα Μεγάλης Γλώσσας είναι ακόμα τεχνολογία αιχμής· οι πάροχοι συνήθως παρουσιάζουν διακοπές σε

απρόσμενες στιγμές. Χρησιμοποιήστε μηχανισμούς επανάληψης και διακόπτες κυκλώματος για την αποτροπή διαδοχικών αποτυχιών.

Συνθεσιμότητα και Ενορχήστρωση Εργατών AI

Ένα από τα βασικά πλεονεκτήματα της αρχιτεκτονικής εργατών AI είναι η συνθεσιμότητά της, η οποία σας επιτρέπει να συνδυάζετε και να ενορχηστρώνετε πολλαπλούς εργάτες AI για την επίλυση σύνθετων προβλημάτων. Διασπώντας μια μεγαλύτερη εργασία σε μικρότερες, πιο διαχειρίσιμες υποεργασίες, καθεμία από τις οποίες διαχειρίζεται ένας εξειδικευμένος εργάτης AI, μπορείτε να δημιουργήσετε ισχυρά και ευέλικτα συστήματα. Σε αυτή την ενότητα, θα εξερευνήσουμε διαφορετικές προσεγγίσεις για τη σύνθεση και την ενορχήστρωση “πληθώρας” εργατών AI.

Αλυσίδωση Εργατών AI για Ροές Εργασίας Πολλαπλών Βημάτων

Σε πολλά σενάρια, μια σύνθετη εργασία μπορεί να αποσυντεθεί σε μια σειρά διαδοχικών βημάτων, όπου η έξοδος ενός εργάτη AI γίνεται η είσοδος για τον επόμενο. Αυτή η αλυσίδωση εργατών AI δημιουργεί μια ροή εργασίας ή διοχέτευση πολλαπλών βημάτων. Κάθε εργάτης AI στην αλυσίδα επικεντρώνεται σε μια συγκεκριμένη υποεργασία, και η τελική έξοδος είναι το αποτέλεσμα των συνδυασμένων προσπαθειών όλων των εργατών.

Ας εξετάσουμε ένα παράδειγμα στο πλαίσιο μιας εφαρμογής Ruby on Rails για την επεξεργασία περιεχομένου που δημιουργείται από χρήστες. Η ροή εργασίας περιλαμβάνει τα ακόλουθα βήματα, τα οποία ομολογουμένως είναι πιθανώς το καθένα πολύ απλά για να αξίζει την αποσύνθεση με αυτόν τον τρόπο σε πραγματικές περιπτώσεις χρήσης, αλλά καθιστούν το παράδειγμα ευκολότερο στην κατανόηση:

1. Καθαρισμός Κειμένου: Ένας εργάτης AI υπεύθυνος για την αφαίρεση ετικετών HTML, τη μετατροπή κειμένου σε πεζά γράμματα και τη διαχείριση κανονικοποίησης Unicode.

2. Ανίχνευση Γλώσσας: Ένας εργάτης AI που αναγνωρίζει τη γλώσσα του καθαρισμένου κειμένου.

3. Ανάλυση Συναισθήματος: Ένας εργάτης AI που προσδιορίζει το συναίσθημα (θετικό, αρνητικό ή ουδέτερο) του κειμένου με βάση την ανιχνευμένη γλώσσα.

4. Κατηγοριοποίηση Περιεχομένου: Ένας εργάτης AI που ταξινομεί το κείμενο σε προκαθορισμένες κατηγορίες χρησιμοποιώντας τεχνικές επεξεργασίας φυσικής γλώσσας.

Ακολουθεί ένα πολύ απλοποιημένο παράδειγμα για το πώς μπορείτε να αλυσιδώσετε αυτούς τους εργάτες AI μαζί χρησιμοποιώντας Ruby:

```
1 class ContentProcessor
2   def initialize(text)
3     @text = text
4   end
5
6   def process
7     cleaned_text = TextCleanupWorker.new(@text).call
8     language = LanguageDetectionWorker.new(cleaned_text).call
9     sentiment = SentimentAnalysisWorker.new(cleaned_text, language).call
10    category = CategorizationWorker.new(cleaned_text, language).call
11
12    { cleaned_text:, language:, sentiment:, category: }
13  end
14 end
```

Σε αυτό το παράδειγμα, η κλάση ContentProcessor αρχικοποιείται με το ακατέργαστο κείμενο και αλυσιδώνει τους εργάτες τεχνητής νοημοσύνης μαζί στη μέθοδο process. Κάθε εργάτης τεχνητής νοημοσύνης εκτελεί το συγκεκριμένο του έργο και περνά το αποτέλεσμα στον επόμενο εργάτη στην αλυσίδα. Η τελική

έξοδος είναι ένας πίνακας κατακερματισμού που περιέχει το καθαρισμένο κείμενο, την ανιχνευμένη γλώσσα, το συναίσθημα και την κατηγορία περιεχομένου.

Παράλληλη Επεξεργασία για Ανεξάρτητους Εργάτες Τεχνητής Νοημοσύνης

Στο προηγούμενο παράδειγμα, οι εργάτες τεχνητής νοημοσύνης είναι αλυσιδωμένοι διαδοχικά, όπου κάθε εργάτης επεξεργάζεται το κείμενο και περνά το αποτέλεσμα στον επόμενο εργάτη. Ωστόσο, αν έχετε πολλαπλούς εργάτες τεχνητής νοημοσύνης που μπορούν να λειτουργήσουν ανεξάρτητα στην ίδια είσοδο, μπορείτε να βελτιστοποιήσετε τη ροή εργασίας επεξεργάζοντάς τους παράλληλα.

Στο δεδομένο σενάριο, μόλις πραγματοποιηθεί ο καθαρισμός κειμένου από τον `TextCleanupWorker`, ο `LanguageDetectionWorker`, ο `SentimentAnalysisWorker`, και ο `CategorizationWorker` μπορούν όλοι να επεξεργαστούν το καθαρισμένο κείμενο ανεξάρτητα. Εκτελώντας αυτούς τους εργάτες παράλληλα, μπορείτε δυνητικά να μειώσετε τον συνολικό χρόνο επεξεργασίας και να βελτιώσετε την αποδοτικότητα της ροής εργασίας σας.

Για να επιτύχετε παράλληλη επεξεργασία στη Ruby, μπορείτε να αξιοποιήσετε τεχνικές ταυτοχρονισμού όπως νήματα ή ασύγχρονο προγραμματισμό. Ακολουθεί ένα παράδειγμα για το πώς μπορείτε να τροποποιήσετε την κλάση `ContentProcessor` για να επεξεργαστείτε τους τελευταίους τρεις εργάτες παράλληλα χρησιμοποιώντας νήματα:

```
1  require 'concurrent'
2
3  class ContentProcessor
4    def initialize(text)
5      @text = text
6    end
7
8    def process
9      cleaned_text = TextCleanupWorker.new(@text).call
10
11      language_future = Concurrent::Future.execute do
12        LanguageDetectionWorker.new(cleaned_text).call
13      end
14
15      sentiment_future = Concurrent::Future.execute do
16        SentimentAnalysisWorker.new(cleaned_text).call
17      end
18
19      category_future = Concurrent::Future.execute do
20        CategorizationWorker.new(cleaned_text).call
21      end
22
23      language = language_future.value
24      sentiment = sentiment_future.value
25      category = category_future.value
26
27      { cleaned_text:, language:, sentiment:, category: }
28    end
29  end
```

Σε αυτήν τη βελτιστοποιημένη έκδοση, χρησιμοποιούμε τη βιβλιοθήκη `concurrent-ruby` για να δημιουργήσουμε αντικείμενα `Concurrent::Future` για καθέναν από τους ανεξάρτητους εργάτες AI. Ένα `Future` αντιπροσωπεύει έναν υπολογισμό που θα εκτελεστεί ασύγχρονα σε ξεχωριστό νήμα.

Μετά το βήμα καθαρισμού κειμένου, δημιουργούμε τρία αντικείμενα `Future`: `language_future`, `sentiment_future`, και `category_future`. Κάθε `Future` εκτελεί τον αντίστοιχο εργάτη AI (`LanguageDetectionWorker`, `SentimentAnalysisWorker`, και `CategorizationWorker`) σε ξεχωριστό νήμα, περνώντας το `cleaned_text` ως είσοδο.

Καλώντας τη μέθοδο `value` σε κάθε `Future`, περιμένουμε να ολοκληρωθεί ο υπολογισμός και λαμβάνουμε το αποτέλεσμα. Η μέθοδος `value` μπλοκάρει μέχρι να είναι διαθέσιμο το αποτέλεσμα, διασφαλίζοντας ότι όλοι οι παράλληλοι εργάτες έχουν ολοκληρώσει την επεξεργασία πριν προχωρήσουμε.

Τέλος, κατασκευάζουμε το `hash` εξόδου με το καθαρισμένο κείμενο και τα αποτελέσματα από τους παράλληλους εργάτες, όπως ακριβώς στο αρχικό παράδειγμα.

Με την επεξεργασία των ανεξάρτητων εργατών ΑΙ παράλληλα, μπορείτε δυνητικά να μειώσετε το συνολικό χρόνο επεξεργασίας σε σύγκριση με τη διαδοχική εκτέλεσή τους. Αυτή η βελτιστοποίηση είναι ιδιαίτερα επωφελής όταν πρόκειται για χρονοβόρες εργασίες ή όταν επεξεργάζεστε μεγάλους όγκους δεδομένων.

Ωστόσο, είναι σημαντικό να σημειωθεί ότι τα πραγματικά οφέλη απόδοσης εξαρτώνται από διάφορους παράγοντες, όπως η πολυπλοκότητα κάθε εργάτη, οι διαθέσιμοι πόροι συστήματος και η επιβάρυνση της διαχείρισης νημάτων. Είναι πάντα καλή πρακτική να κάνετε συγκριτική αξιολόγηση και να προφιλάρете τον κώδικά σας για να καθορίσετε το βέλτιστο επίπεδο παραλληλισμού για τη συγκεκριμένη περίπτωση χρήσης σας.

Επιπλέον, κατά την υλοποίηση παράλληλης επεξεργασίας, να είστε προσεκτικοί με τυχόν κοινόχρηστους πόρους ή εξαρτήσεις μεταξύ των εργατών. Βεβαιωθείτε ότι οι εργάτες μπορούν να λειτουργούν ανεξάρτητα χωρίς συγκρούσεις ή συνθήκες ανταγωνισμού. Εάν υπάρχουν εξαρτήσεις ή κοινόχρηστοι πόροι, ίσως χρειαστεί να υλοποιήσετε κατάλληλους μηχανισμούς συγχρονισμού για να διατηρήσετε την ακεραιότητα των δεδομένων και να αποφύγετε προβλήματα όπως αδιέξοδα ή ασυνεπή αποτελέσματα.

Το Καθολικό Κλείδωμα Διερμηνέα της Ruby και η

Ασύγχρονη Επεξεργασία

Είναι σημαντικό να κατανοήσουμε τις επιπτώσεις του Καθολικού Κλειδώματος Διερμηνέα (GIL) της Ruby όταν εξετάζουμε την ασύγχρονη επεξεργασία βασισμένη σε νήματα στη Ruby.

Το GIL είναι ένας μηχανισμός στον διερμηνέα της Ruby που διασφαλίζει ότι μόνο ένα νήμα μπορεί να εκτελέσει κώδικα Ruby κάθε φορά, ακόμη και σε επεξεργαστές πολλαπλών πυρήνων. Αυτό σημαίνει ότι ενώ μπορούν να δημιουργηθούν και να διαχειριστούν πολλαπλά νήματα μέσα σε μια διεργασία Ruby, μόνο ένα νήμα μπορεί να εκτελεί ενεργά κώδικα Ruby οποιαδήποτε στιγμή.

Το GIL έχει σχεδιαστεί για να απλοποιήσει την υλοποίηση του διερμηνέα Ruby και να παρέχει ασφάλεια νημάτων για τις εσωτερικές δομές δεδομένων της Ruby. Ωστόσο, περιορίζει επίσης τη δυνατότητα για πραγματική παράλληλη εκτέλεση κώδικα Ruby.

Όταν χρησιμοποιείτε νήματα στη Ruby, όπως με τη βιβλιοθήκη `concurrent-ruby` ή την ενσωματωμένη κλάση `Thread`, τα νήματα υπόκεινται στους περιορισμούς του GIL. Το GIL επιτρέπει σε κάθε νήμα να εκτελεί κώδικα Ruby για ένα σύντομο χρονικό διάστημα πριν μεταβεί σε άλλο νήμα, δημιουργώντας την ψευδαίσθηση της ταυτόχρονης εκτέλεσης.

Ωστόσο, λόγω του GIL, η πραγματική εκτέλεση του κώδικα Ruby παραμένει ακολουθιακή. Ενώ ένα νήμα εκτελεί κώδικα Ruby, τα άλλα νήματα είναι ουσιαστικά σε παύση, περιμένοντας τη σειρά τους να αποκτήσουν το GIL και να εκτελεστούν.

Αυτό σημαίνει ότι η ασύγχρονη επεξεργασία βασισμένη σε νήματα στη Ruby είναι πιο αποτελεσματική για εργασίες δεσμευμένες από I/O, όπως η αναμονή για αποκρίσεις εξωτερικών API (όπως μοντέλα γλώσσας μεγάλης κλίμακας που φιλοξενούνται από τρίτους) ή η εκτέλεση λειτουργιών I/O αρχείων. Όταν ένα νήμα συναντά μια λειτουργία I/O, μπορεί να απελευθερώσει το GIL, επιτρέποντας σε

άλλα νήματα να εκτελεστούν ενώ περιμένουν την ολοκλήρωση του I/O.

Από την άλλη πλευρά, για εργασίες δεσμευμένες από CPU, όπως εντατικοί υπολογισμοί ή μακροχρόνια επεξεργασία AI worker, το GIL μπορεί να περιορίσει τα πιθανά οφέλη απόδοσης του παραλληλισμού βασισμένου σε νήματα. Δεδομένου ότι μόνο ένα νήμα μπορεί να εκτελεί κώδικα Ruby κάθε φορά, ο συνολικός χρόνος εκτέλεσης μπορεί να μην μειωθεί σημαντικά σε σύγκριση με την ακολουθιακή επεξεργασία.

Για να επιτύχετε πραγματική παράλληλη εκτέλεση για εργασίες δεσμευμένες από CPU στη Ruby, ίσως χρειαστεί να εξερευνήσετε εναλλακτικές προσεγγίσεις, όπως:

- Χρήση παραλληλισμού βασισμένου σε διεργασίες με πολλαπλές διεργασίες Ruby, καθεμία εκτελούμενη σε ξεχωριστό πυρήνα CPU.
- Αξιοποίηση εξωτερικών βιβλιοθηκών ή πλαισίων που παρέχουν εγγενείς επεκτάσεις ή διεπαφές σε γλώσσες χωρίς GIL, όπως C ή Rust.
- Χρήση πλαισίων καταναμημένης υπολογιστικής ή ουρών μηνυμάτων για την κατανομή εργασιών σε πολλαπλούς υπολογιστές ή διεργασίες.

Είναι κρίσιμο να λαμβάνετε υπόψη τη φύση των εργασιών σας και τους περιορισμούς που επιβάλλονται από το GIL κατά το σχεδιασμό και την υλοποίηση ασύγχρονης επεξεργασίας στη Ruby. Ενώ η ασύγχρονη επεξεργασία βασισμένη σε νήματα μπορεί να παρέχει οφέλη για εργασίες δεσμευμένες από I/O, ενδέχεται να μην προσφέρει σημαντικές βελτιώσεις απόδοσης για εργασίες δεσμευμένες από CPU λόγω των περιορισμών του GIL.

Τεχνικές Συνόλου για Βελτιωμένη Ακρίβεια

Οι τεχνικές συνόλου περιλαμβάνουν το συνδυασμό των εξόδων πολλαπλών AI workers για τη βελτίωση της συνολικής ακρίβειας ή της ευρωστίας του συστήματος. Αντί να βασίζεται σε έναν μόνο AI worker, οι τεχνικές συνόλου αξιοποιούν

τη συλλογική νοημοσύνη πολλαπλών workers για τη λήψη πιο ενημερωμένων αποφάσεων.



Τα σύνολα μοντέλων είναι ιδιαίτερα σημαντικά εάν διαφορετικά μέρη της ροής εργασίας σας λειτουργούν καλύτερα με διαφορετικά μοντέλα τεχνητής νοημοσύνης, κάτι που είναι πιο συνηθισμένο απ' ό,τι ίσως νομίζετε. Ισχυρά μοντέλα όπως το GPT-4 είναι εξαιρετικά ακριβά σε σύγκριση με λιγότερο ικανές επιλογές ανοιχτού κώδικα, και πιθανότατα δεν χρειάζονται για κάθε μεμονωμένο βήμα της ροής εργασίας της εφαρμογής σας.

Μια συνηθισμένη τεχνική συνόλων είναι η ψηφοφορία πλειοψηφίας, όπου πολλαπλοί εργάτες τεχνητής νοημοσύνης επεξεργάζονται ανεξάρτητα την ίδια είσοδο, και η τελική έξοδος καθορίζεται από τη συναίνεση της πλειοψηφίας. Αυτή η προσέγγιση μπορεί να βοηθήσει στη μείωση του αντίκτυπου των σφαλμάτων μεμονωμένων εργατών και να βελτιώσει τη συνολική αξιοπιστία του συστήματος.

Ας εξετάσουμε ένα παράδειγμα όπου έχουμε τρεις εργάτες τεχνητής νοημοσύνης για ανάλυση συναισθήματος, ο καθένας χρησιμοποιώντας διαφορετικό μοντέλο ή έχοντας διαφορετικά πλαίσια αναφοράς. Μπορούμε να συνδυάσουμε τις εξόδους τους χρησιμοποιώντας ψηφοφορία πλειοψηφίας για να καθορίσουμε την τελική πρόβλεψη συναισθήματος.

```
1 class SentimentAnalysisEnsemble
2   def initialize(text)
3     @text = text
4   end
5
6   def analyze
7     predictions = [
8       SentimentAnalysisWorker1.new(@text).analyze,
9       SentimentAnalysisWorker2.new(@text).analyze,
10      SentimentAnalysisWorker3.new(@text).analyze
11    ]
12
13    predictions
14      .group_by { |sentiment| sentiment }
15      .max_by { |_, votes| votes.size }
16      .first
17
18    end
19  end
```

Σε αυτό το παράδειγμα, η κλάση `SentimentAnalysisEnsemble` αρχικοποιείται με το κείμενο και επικαλείται τρεις διαφορετικούς τεχνητούς πράκτορες ανάλυσης συναισθήματος. Η μέθοδος `analyze` συλλέγει τις προβλέψεις από κάθε πράκτορα και καθορίζει το επικρατέστερο συναίσθημα χρησιμοποιώντας τις μεθόδους `group_by` και `max_by`. Η τελική έξοδος είναι το συναίσθημα που λαμβάνει τις περισσότερες ψήφους από το σύνολο των πρακτόρων



Τα σύνολα είναι σαφώς μια περίπτωση όπου ο πειραματισμός με τον παραλληλισμό μπορεί να αξίζει τον χρόνο σας.

Δυναμική Επιλογή και Επίκληση Τεχνητών Πρακτόρων

Σε ορισμένες, αν όχι στις περισσότερες περιπτώσεις, ο συγκεκριμένος τεχνητός πράκτορας που θα επικληθεί μπορεί να εξαρτάται από τις συνθήκες εκτέλεσης ή

τις εισόδους του χρήστη. Η δυναμική επιλογή και επίκληση τεχνητών πρακτόρων επιτρέπει την ευελιξία και την προσαρμοστικότητα στο σύστημα.



Μπορεί να μπίτε στον πειρασμό να προσπαθήσετε να χωρέσετε πολλή λειτουργικότητα σε έναν μόνο τεχνητό πράκτορα, δίνοντάς του πολλές λειτουργίες και μια μεγάλη περίπλοκη εντολή που εξηγεί πώς να τις καλέσει. Αντισταθείτε στον πειρασμό, πιστέψτε με. Ένας από τους λόγους που η προσέγγιση που συζητάμε σε αυτό το κεφάλαιο ονομάζεται “Πλήθος Πρακτόρων” είναι για να μας υπενθυμίζει ότι είναι επιθυμητό να έχουμε πολλούς εξειδικευμένους πράκτορες, ο καθένας να κάνει τη δική του μικρή δουλειά στην υπηρεσία του ευρύτερου σκοπού.

Για παράδειγμα, εξετάστε μια εφαρμογή chatbot όπου διαφορετικοί τεχνητοί πράκτορες είναι υπεύθυνοι για τη διαχείριση διαφορετικών τύπων ερωτημάτων χρήστη. Με βάση την είσοδο του χρήστη, η εφαρμογή επιλέγει δυναμικά τον κατάλληλο τεχνητό πράκτορα για την επεξεργασία του ερωτήματος.

```

1 class ChatbotController < ApplicationController
2   def process_query
3     query = params[:query]
4     query_type = QueryClassifierWorker.new(query).classify
5
6     case query_type
7     when 'greeting'
8       response = GreetingWorker.new(query).generate_response
9     when 'product_inquiry'
10      response = ProductInquiryWorker.new(query).generate_response
11     when 'order_status'
12      response = OrderStatusWorker.new(query).generate_response
13     else
14      response = DefaultResponseWorker.new(query).generate_response
15     end
16
17     render json: { response: response }
18   end
19 end

```

Σε αυτό το παράδειγμα, ο ChatbotController λαμβάνει ένα ερώτημα χρήστη μέσω της ενέργειας `process_query`. Πρώτα χρησιμοποιεί έναν `QueryClassifierWorker` για να προσδιορίσει τον τύπο του ερωτήματος. Με βάση τον ταξινομημένο τύπο ερωτήματος, ο ελεγκτής επιλέγει δυναμικά τον κατάλληλο εργάτη AI για να δημιουργήσει την απάντηση. Αυτή η δυναμική επιλογή επιτρέπει στο chatbot να χειρίζεται διαφορετικούς τύπους ερωτημάτων και να τα δρομολογεί στους σχετικούς εργάτες AI.



Δεδομένου ότι η εργασία του `QueryClassifierWorker` είναι σχετικά απλή και δεν απαιτεί πολύ πλαίσιο ή ορισμούς συναρτήσεων, πιθανότατα μπορείτε να την υλοποιήσετε χρησιμοποιώντας ένα εξαιρετικά γρήγορο μικρό LLM όπως το [mistralai/mixtral-8x7b-instruct:nitro](#). Διαθέτει δυνατότητες που πλησιάζουν το επίπεδο του GPT-4 σε πολλές εργασίες και, τη στιγμή που γράφω αυτό το κείμενο, η Groq μπορεί να το εξυπηρετήσει με εκπληκτική απόδοση 444 tokens/δευτερόλεπτο.

Συνδυάζοντας Παραδοσιακή ΕΦΓ με ΜΓΜ

Ενώ τα Μεγάλα Γλωσσικά Μοντέλα (ΜΓΜ) έχουν φέρει επανάσταση στον τομέα της επεξεργασίας φυσικής γλώσσας (ΕΦΓ), προσφέροντας απaráμιλλη ευελιξία και απόδοση σε ένα ευρύ φάσμα εργασιών, δεν αποτελούν πάντα την πιο αποδοτική ή οικονομικά αποτελεσματική λύση για κάθε πρόβλημα. Σε πολλές περιπτώσεις, ο συνδυασμός παραδοσιακών τεχνικών ΕΦΓ με ΜΓΜ μπορεί να οδηγήσει σε πιο βελτιστοποιημένες, στοχευμένες και οικονομικές προσεγγίσεις για την επίλυση συγκεκριμένων προκλήσεων ΕΦΓ.

Σκεφτείτε τα ΜΓΜ ως τα ελβετικά σουγιά της ΕΦΓ—απίστευτα ευέλικτα και ισχυρά, αλλά όχι απαραίτητα το καλύτερο εργαλείο για κάθε δουλειά. Μερικές φορές, ένα εξειδικευμένο εργαλείο όπως ένα ανοιχτήρι κρασιού ή ένα ανοιχτήρι κονσέρβας μπορεί να είναι πιο αποτελεσματικό και αποδοτικό για μια συγκεκριμένη εργασία.

Παρομοίως, οι παραδοσιακές τεχνικές ΕΦΓ, όπως η ομαδοποίηση εγγράφων, η αναγνώριση θεμάτων και η ταξινόμηση, μπορούν συχνά να παρέχουν πιο στοχευμένες και οικονομικά αποδοτικές λύσεις για συγκεκριμένες πτυχές της διαδικασίας ΕΦΓ.

Ένα από τα βασικά πλεονεκτήματα των παραδοσιακών τεχνικών ΕΦΓ είναι η υπολογιστική τους αποδοτικότητα. Αυτές οι μέθοδοι, που συχνά βασίζονται σε απλούστερα στατιστικά μοντέλα ή προσεγγίσεις βασισμένες σε κανόνες, μπορούν να επεξεργαστούν μεγάλους όγκους δεδομένων κειμένου πολύ πιο γρήγορα και με χαμηλότερη υπολογιστική επιβάρυνση σε σύγκριση με τα ΜΓΜ. Αυτό τις καθιστά ιδιαίτερα κατάλληλες για εργασίες που περιλαμβάνουν την ανάλυση και οργάνωση μεγάλων συλλογών εγγράφων, όπως η ομαδοποίηση παρόμοιων άρθρων ή η αναγνώριση βασικών θεμάτων μέσα σε μια συλλογή κειμένων.

Επιπλέον, οι παραδοσιακές τεχνικές ΕΦΓ μπορούν συχνά να επιτύχουν υψηλή ακρίβεια για συγκεκριμένες εργασίες, ιδιαίτερα όταν εκπαιδεύονται σε σύνολα δεδομένων συγκεκριμένου τομέα. Για παράδειγμα, ένας καλά ρυθμισμένος ταξινομητής εγγράφων που χρησιμοποιεί παραδοσιακούς αλγόριθμους μηχανικής μάθησης όπως οι Μηχανές Διανυσμάτων Υποστήριξης (ΜΔΥ) ή ο Αφελής Μπέιζ μπορεί να κατηγοριοποιήσει με ακρίβεια έγγραφα σε προκαθορισμένες κατηγορίες με ελάχιστο υπολογιστικό κόστος.

Ωστόσο, τα ΜΓΜ πραγματικά διαπρέπουν όταν πρόκειται για εργασίες που απαιτούν βαθύτερη κατανόηση της γλώσσας, του πλαισίου και του συλλογισμού. Η ικανότητά τους να παράγουν συνεκτικό και συναφές με το πλαίσιο κείμενο, να απαντούν σε ερωτήσεις και να συνοψίζουν μεγάλα κείμενα είναι ασυναγώνιστη από τις παραδοσιακές μεθόδους ΕΦΓ. Τα ΜΓΜ μπορούν να χειριστούν αποτελεσματικά πολύπλοκα γλωσσικά φαινόμενα, όπως η αμφισημία, η αναφορικότητα και οι ιδιωματικές εκφράσεις, καθιστώντας τα ανεκτίμητα για εργασίες που απαιτούν παραγωγή ή κατανόηση φυσικής γλώσσας.

Η πραγματική δύναμη βρίσκεται στο συνδυασμό παραδοσιακών τεχνικών ΕΦΓ με

ΜΓΜ για τη δημιουργία υβριδικών προσεγγίσεων που αξιοποιούν τα πλεονεκτήματα και των δύο. Χρησιμοποιώντας παραδοσιακές μεθόδους ΕΦΓ για εργασίες όπως η προεπεξεργασία εγγράφων, η ομαδοποίηση και η εξαγωγή θεμάτων, μπορείτε να οργανώσετε και να δομήσετε αποτελεσματικά τα δεδομένα κειμένου σας. Αυτές οι δομημένες πληροφορίες μπορούν στη συνέχεια να τροφοδοτηθούν σε ΜΓΜ για πιο προηγμένες εργασίες, όπως η δημιουργία περιλήψεων, η απάντηση ερωτήσεων ή η δημιουργία ολοκληρωμένων αναφορών.

Για παράδειγμα, ως εξετάσουμε μια περίπτωση χρήσης όπου θέλετε να δημιουργήσετε μια αναφορά τάσεων για έναν συγκεκριμένο τομέα με βάση ένα μεγάλο σώμα κειμένων μεμονωμένων εγγράφων τάσεων. Αντί να βασίζεστε αποκλειστικά σε ΜΓΜ, που μπορεί να είναι υπολογιστικά ακριβά και χρονοβόρα για την επεξεργασία μεγάλου όγκου κειμένου, μπορείτε να χρησιμοποιήσετε μια υβριδική προσέγγιση:

1. Χρησιμοποιήστε παραδοσιακές τεχνικές ΕΦΓ, όπως η μοντελοποίηση θεμάτων (π.χ., Λανθάνουσα Κατανομή Ντίριχλε) ή αλγόριθμους ομαδοποίησης (π.χ., Κ-μέσων), για να ομαδοποιήσετε παρόμοια έγγραφα τάσεων και να εντοπίσετε βασικά θέματα εντός του σώματος κειμένων.
2. Τροφοδοτήστε τα ομαδοποιημένα έγγραφα και τα εντοπισμένα θέματα σε ένα ΜΓΜ, αξιοποιώντας την ανώτερη κατανόηση και ικανότητα παραγωγής γλώσσας για τη δημιουργία συνεκτικών και κατατοπιστικών περιλήψεων για κάθε ομάδα ή θέμα.
3. Τέλος, χρησιμοποιήστε το ΜΓΜ για να δημιουργήσετε μια ολοκληρωμένη αναφορά τάσεων συνδυάζοντας τις επιμέρους περιλήψεις, επισημαίνοντας τις σημαντικότερες τάσεις και παρέχοντας γνώσεις και συστάσεις με βάση τις συγκεντρωμένες πληροφορίες.

Συνδυάζοντας παραδοσιακές τεχνικές ΕΦΓ με ΜΓΜ με αυτόν τον τρόπο, μπορείτε να επεξεργαστείτε αποτελεσματικά μεγάλο όγκο δεδομένων κειμένου, να εξάγετε

ουσιαστικές γνώσεις και να δημιουργήσετε αναφορές υψηλής ποιότητας, βελτιστοποιώντας παράλληλα τους υπολογιστικούς πόρους και το κόστος.

Καθώς ξεκινάτε τα έργα σας στην Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ), είναι απαραίτητο να αξιολογήσετε προσεκτικά τις συγκεκριμένες απαιτήσεις και τους περιορισμούς κάθε εργασίας και να εξετάσετε πώς οι παραδοσιακές μέθοδοι ΕΦΓ και τα Μοντέλα Μεγάλης Γλώσσας (ΜΜΓ) μπορούν να αξιοποιηθούν συνδυαστικά για την επίτευξη των καλύτερων αποτελεσμάτων. Συνδυάζοντας την αποδοτικότητα και την ακρίβεια των παραδοσιακών τεχνικών με την ευελιξία και τη δύναμη των ΜΜΓ, μπορείτε να δημιουργήσετε ιδιαίτερα αποτελεσματικές και οικονομικές λύσεις ΕΦΓ που προσφέρουν αξία στους χρήστες και τα ενδιαφερόμενα μέρη σας.

Χρήση Εργαλείων



Στον τομέα της ανάπτυξης εφαρμογών που βασίζονται στην τεχνητή νοημοσύνη, η έννοια της “χρήσης εργαλείων” ή “κλήσης συναρτήσεων” έχει αναδειχθεί ως μια ισχυρή τεχνική που επιτρέπει στο LLM σας να συνδέεται με εξωτερικά εργαλεία, APIs, συναρτήσεις, βάσεις δεδομένων και άλλους πόρους. Αυτή η προσέγγιση επιτρέπει ένα πλουσιότερο σύνολο συμπεριφορών πέρα από την απλή εξαγωγή κειμένου, και πιο δυναμικές αλληλεπιδράσεις μεταξύ των στοιχείων τεχνητής νοημοσύνης και του υπόλοιπου οικοσυστήματος της εφαρμογής σας. Όπως θα εξετάσουμε σε αυτό το κεφάλαιο, η χρήση εργαλείων σας δίνει επίσης τη δυνατότητα να κάνετε το μοντέλο τεχνητής νοημοσύνης σας να παράγει δεδομένα με δομημένους τρόπους.

Τι είναι η Χρήση Εργαλείων;

Η χρήση εργαλείων, γνωστή και ως κλήση συναρτήσεων, είναι μια τεχνική που επιτρέπει στους προγραμματιστές να καθορίσουν μια λίστα συναρτήσεων με τις οποίες ένα LLM μπορεί να αλληλεπιδράσει κατά τη διάρκεια της διαδικασίας παραγωγής. Αυτά τα εργαλεία μπορεί να κυμαίνονται από απλές βοηθητικές συναρτήσεις μέχρι πολύπλοκα APIs ή ερωτήματα βάσεων δεδομένων. Παρέχοντας στο LLM πρόσβαση σε αυτά τα εργαλεία, οι προγραμματιστές μπορούν να επεκτείνουν τις δυνατότητες του μοντέλου και να του επιτρέψουν να εκτελεί εργασίες που απαιτούν εξωτερική γνώση ή ενέργειες.

Σχήμα 8. Παράδειγμα ορισμού συνάρτησης για έναν AI εργάτη που αναλύει έγγραφα

```
1  FUNCTION = {
2      name: "save_analysis",
3      description: "Save analysis data for document",
4      parameters: {
5          type: "object",
6          properties: {
7              title: {
8                  type: "string",
9                  maxLength: 140
10             },
11             summary: {
12                 type: "string",
13                 description: "comprehensive multi-paragraph summary with
14                             overview and list of sections (if applicable)"
15             },
16             tags: {
17                 type: "array",
18                 items: {
19                     type: "string",
20                     description: "lowercase tags representing main themes
21                                 of the document"
22                 }
23             }
24         },
25         "required": %w[title summary tags]
```

```
26     }  
27   }.freeze
```

Η βασική ιδέα πίσω από τη χρήση εργαλείων είναι να δώσει στο LLM τη δυνατότητα να επιλέγει και να εκτελεί δυναμικά τα κατάλληλα εργαλεία με βάση την είσοδο του χρήστη ή την εκάστοτε εργασία. Αντί να βασίζεται αποκλειστικά στην προ-εκπαιδευμένη γνώση του μοντέλου, η χρήση εργαλείων επιτρέπει στο LLM να αξιοποιεί εξωτερικούς πόρους για να παράγει πιο ακριβείς, σχετικές και εφαρμόσιμες απαντήσεις. Η χρήση εργαλείων καθιστά τεχνικές όπως το RAG (Παραγωγή Ενισχυμένη με Ανάκτηση) πολύ πιο εύκολες στην υλοποίηση απ' ό,τι θα ήταν διαφορετικά.

Σημειώστε ότι, εκτός αν αναφέρεται διαφορετικά, αυτό το βιβλίο υποθέτει ότι το μοντέλο AI σας δεν έχει πρόσβαση σε ενσωματωμένα εργαλεία από την πλευρά του διακομιστή. Οποιαδήποτε εργαλεία θέλετε να καταστήσετε διαθέσιμα στην AI σας πρέπει να δηλωθούν ρητά από εσάς σε κάθε αίτημα API, με πρόβλεψη για την εκτέλεσή τους εάν και όταν η AI σας σας πει ότι θα ήθελε να χρησιμοποιήσει αυτό το εργαλείο στην απάντησή της.

Οι Δυνατότητες της Χρήσης Εργαλείων

Η χρήση εργαλείων ανοίγει ένα ευρύ φάσμα δυνατοτήτων για εφαρμογές που βασίζονται στην τεχνητή νοημοσύνη. Ακολουθούν μερικά παραδείγματα του τι μπορεί να επιτευχθεί με τη χρήση εργαλείων:

1. **Chatbots και Εικονικοί Βοηθοί:** Συνδέοντας ένα LLM με εξωτερικά εργαλεία, τα chatbots και οι εικονικοί βοηθοί μπορούν να εκτελούν πιο σύνθετες εργασίες, όπως ανάκτηση πληροφοριών από βάσεις δεδομένων, εκτέλεση κλήσεων API ή

αλληλεπίδραση με άλλα συστήματα. Για παράδειγμα, ένα chatbot θα μπορούσε να χρησιμοποιήσει ένα εργαλείο CRM για να αλλάξει την κατάσταση μιας συμφωνίας με βάση το αίτημα του χρήστη.

2. **Ανάλυση Δεδομένων και Insights:** Τα LLM μπορούν να συνδεθούν με εργαλεία ανάλυσης δεδομένων ή βιβλιοθήκες για την εκτέλεση προηγμένων εργασιών επεξεργασίας δεδομένων. Αυτό επιτρέπει στις εφαρμογές να παράγουν insights, να διεξάγουν συγκριτικές αναλύσεις ή να παρέχουν προτάσεις βασισμένες σε δεδομένα με βάση τα ερωτήματα των χρηστών.
3. **Αναζήτηση και Ανάκτηση Πληροφοριών:** Η χρήση εργαλείων επιτρέπει στα LLM να αλληλεπιδρούν με μηχανές αναζήτησης, διανυσματικές βάσεις δεδομένων ή άλλα συστήματα ανάκτησης πληροφοριών. Μετατρέποντας τα ερωτήματα χρηστών σε ερωτήματα αναζήτησης, το LLM μπορεί να ανακτήσει σχετικές πληροφορίες από πολλαπλές πηγές και να παρέχει ολοκληρωμένες απαντήσεις στις ερωτήσεις των χρηστών.
4. **Ενσωμάτωση με Εξωτερικές Υπηρεσίες:** Η χρήση εργαλείων επιτρέπει την απρόσκοπτη ενσωμάτωση μεταξύ εφαρμογών που βασίζονται στην τεχνητή νοημοσύνη και εξωτερικών υπηρεσιών ή API. Για παράδειγμα, ένα LLM θα μπορούσε να αλληλεπιδράσει με ένα API καιρού για να παρέχει ενημερώσεις καιρού σε πραγματικό χρόνο ή με ένα API μετάφρασης για να παράγει πολύγλωσσες απαντήσεις.

Η Ροή Εργασίας της Χρήσης Εργαλείων

Η ροή εργασίας χρήσης εργαλείων συνήθως περιλαμβάνει τέσσερα βασικά βήματα:

1. Συμπερίληψη ορισμών συναρτήσεων στο πλαίσιο του αιτήματός σας
2. Δυναμική (ή ρητή) επιλογή εργαλείων
3. Εκτέλεση συνάρτησης(-ων)
4. Προαιρετική συνέχιση του αρχικού prompt

Ας εξετάσουμε καθένα από αυτά τα βήματα λεπτομερώς.

Συμπερίληψη ορισμών συναρτήσεων στο πλαίσιο του αιτήματός σας

Η τεχνητή νοημοσύνη γνωρίζει ποια εργαλεία έχει στη διάθεσή της επειδή της παρέχετε μια λίστα ως μέρος του αιτήματος ολοκλήρωσης (συνήθως ορίζεται ως συναρτήσεις χρησιμοποιώντας μια παραλλαγή του σχήματος JSON).

Η ακριβής σύνταξη του ορισμού εργαλείων είναι συγκεκριμένη για κάθε μοντέλο.

Έτσι ορίζετε μια συνάρτηση `get_weather` στο Claude 3:

```
1  {
2      "name": "get_weather",
3      "description": "Get the current weather in a given location",
4      "input_schema": {
5          "type": "object",
6          "properties": {
7              "location": {
8                  "type": "string",
9                  "description": "The city and state, e.g. San Francisco, CA"
10             },
11             "unit": {
12                 "type": "string",
13                 "enum": ["celsius", "fahrenheit"],
14                 "description": "The unit of temperature"
15             }
16         },
17         "required": ["location"]
18     }
19 }
```

Και έτσι θα ορίζατε την ίδια συνάρτηση για το GPT-4, περνώντας την ως την τιμή της παραμέτρου `tools`:

```
1  {
2    "name": "get_current_weather",
3    "description": "Get the current weather in a given location",
4    "parameters": {
5      "type": "object",
6      "properties": {
7        "location": {
8          "type": "string",
9          "description": "The city and state, e.g. San Francisco, CA",
10        },
11        "unit": {
12          "type": "string",
13          "enum": ["celsius", "fahrenheit"],
14          "description": "The unit of temperature"
15        },
16      },
17      "required": ["location"],
18    },
19  }
```

Σχεδόν το ίδιο, εκτός από κάποιες διαφορές χωρίς προφανή λόγο! Πόσο ενοχλητικό.

Οι ορισμοί συναρτήσεων καθορίζουν το όνομα, την περιγραφή και τις παραμέτρους εισόδου. Οι παράμετροι εισόδου μπορούν να οριστούν περαιτέρω χρησιμοποιώντας ιδιότητες όπως απαριθμήσεις για τον περιορισμό των αποδεκτών τιμών, και καθορίζοντας εάν μια παράμετρος είναι υποχρεωτική ή όχι.

Εκτός από τους πραγματικούς ορισμούς συναρτήσεων, μπορείτε επίσης να συμπεριλάβετε οδηγίες ή πλαίσιο για το γιατί και πώς να χρησιμοποιήσετε τη συνάρτηση στην οδηγία συστήματος.

Για παράδειγμα, το εργαλείο Αναζήτησης Ιστού μου στο Olympia περιλαμβάνει αυτή την οδηγία συστήματος, η οποία υπενθυμίζει στην ΤΝ ότι έχει στη διάθεσή της τα αναφερόμενα εργαλεία:

```
1 The `google_search` and `realtime_search` functions let you do research
2 on behalf of the user. In contrast to Google, realtime search is powered
3 by Perplexity and provides real-time information to curated current events
4 databases and news sources. Make sure to include URLs in your response so
5 user can do followup research.
```

Η παροχή λεπτομερών περιγραφών θεωρείται ο σημαντικότερος παράγοντας στην απόδοση των εργαλείων. Οι περιγραφές σας θα πρέπει να εξηγούν κάθε λεπτομέρεια σχετικά με το εργαλείο, συμπεριλαμβανομένων:

- Τι κάνει το εργαλείο
- Πότε πρέπει να χρησιμοποιείται (και πότε όχι)
- Τι σημαίνει κάθε παράμετρος και πώς επηρεάζει τη συμπεριφορά του εργαλείου
- Τυχόν σημαντικές επιφυλάξεις ή περιορισμούς που ισχύουν για την υλοποίηση του εργαλείου

Όσο περισσότερο πλαίσιο μπορείτε να δώσετε στην ΤΝ σχετικά με τα εργαλεία σας, τόσο καλύτερα θα μπορεί να αποφασίσει πότε και πώς να τα χρησιμοποιήσει. Για παράδειγμα, η Anthropic συνιστά τουλάχιστον 3-4 προτάσεις ανά περιγραφή εργαλείου για τη σειρά Claude 3, περισσότερες αν το εργαλείο είναι πολύπλοκο.

Δεν είναι απαραίτητα διαισθητικό, αλλά οι περιγραφές θεωρούνται επίσης πιο σημαντικές από τα παραδείγματα. Ενώ μπορείτε να συμπεριλάβετε παραδείγματα για το πώς να χρησιμοποιήσετε ένα εργαλείο στην περιγραφή του ή στη συνοδευτική προτροπή, αυτό είναι λιγότερο σημαντικό από το να έχετε μια σαφή και ολοκληρωμένη εξήγηση του σκοπού και των παραμέτρων του εργαλείου. Προσθέστε παραδείγματα μόνο αφού έχετε αναπτύξει πλήρως την περιγραφή.

Ακολουθεί ένα παράδειγμα προδιαγραφής λειτουργίας API τύπου Stripe:

```
1  {
2    "name": "createPayment",
3    "description": "Create a new payment request",
4    "parameters": {
5      "type": "object",
6      "properties": {
7        "transaction_amount": {
8          "type": "number",
9          "description": "The amount to be paid"
10       },
11       "description": {
12         "type": "string",
13         "description": "A brief description of the payment"
14       },
15       "payment_method_id": {
16         "type": "string",
17         "description": "The payment method to be used"
18       },
19       "payer": {
20         "type": "object",
21         "description": "Information about the payer, including their name,
22                        email, and identification number",
23         "properties": {
24           "name": {
25             "type": "string",
26             "description": "The payer's name"
27           },
28           "email": {
29             "type": "string",
30             "description": "The payer's email address"
31           },
32           "identification": {
33             "type": "object",
34             "description": "The payer's identification number",
35             "properties": {
36               "type": {
37                 "type": "string",
38                 "description": "Identification document (e.g. CPF, CNPJ)"
39               },
40               "number": {
41                 "type": "string",
42                 "description": "The identification number"
```

```
43     }  
44   },  
45   "required": [ "type", "number" ]  
46 }  
47 },  
48 "required": [ "name", "email", "identification" ]  
49 }  
50 }  
51 }
```



Στην πράξη, ορισμένα μοντέλα δυσκολεύονται να χειριστούν ένθετες προδιαγραφές συναρτήσεων και να διαχειριστούν πολύπλοκους τύπους δεδομένων εξόδου όπως πίνακες, λεξικά κλπ. Αλλά θεωρητικά, θα πρέπει να μπορείτε να παρέχετε προδιαγραφές JSON Schema οποιουδήποτε βάθους!

Δυναμική Επιλογή Εργαλείων

Όταν εκτελείτε μια ολοκλήρωση συνομιλίας που περιλαμβάνει ορισμούς εργαλείων, το LLM επιλέγει δυναμικά τα καταλληλότερα εργαλεία για χρήση και δημιουργεί τις απαιτούμενες παραμέτρους εισόδου για κάθε εργαλείο.

Στην πράξη, η ικανότητα της TN να καλεί *ακριβώς* τη σωστή συνάρτηση και να ακολουθεί *ακριβώς* τις προδιαγραφές σας για τις εισόδους είναι αμφίβολη. Η μείωση της υπερπαραμέτρου θερμοκρασίας στο 0.0 βοηθάει πολύ, αλλά από την εμπειρία μου θα εξακολουθείτε να βλέπετε περιστασιακά σφάλματα. Αυτές οι αποτυχίες περιλαμβάνουν ψευδαισθητικά ονόματα συναρτήσεων, λανθασμένες ή απλώς απύουσες παραμέτρους εισόδου. Οι παράμετροι περνούν ως JSON, που σημαίνει ότι μερικές φορές θα δείτε σφάλματα που προκαλούνται από ελλιπή, λανθασμένα εισαγωγικά ή διαφορετικά κατεστραμμένα JSON.



Τα μοτίβα [Αυτοθεραπευόμενων Δεδομένων](#) μπορούν να βοηθήσουν στην [αυτόματη διόρθωση](#) κλήσεων συναρτήσεων που αποτυγχάνουν λόγω συντακτικών σφαλμάτων.

Εξαναγκασμένη (ή Ρητή) Επιλογή Εργαλείων

Ορισμένα μοντέλα σας δίνουν τη δυνατότητα να εξαναγκάσετε την κλήση μιας συγκεκριμένης συνάρτησης, ως παράμετρο στο αίτημα. Διαφορετικά, το αν θα κληθεί η συνάρτηση ή όχι εξαρτάται εξ ολοκλήρου από τη διακριτική ευχέρεια της TN.

Η δυνατότητα εξαναγκασμού μιας κλήσης συνάρτησης είναι κρίσιμη σε ορισμένα σενάρια όπου θέλετε να διασφαλίσετε ότι ένα συγκεκριμένο εργαλείο ή συνάρτηση εκτελείται, ανεξάρτητα από τη διαδικασία δυναμικής επιλογής της TN. Υπάρχουν αρκετοί λόγοι για τους οποίους αυτή η δυνατότητα είναι σημαντική:

1. **Ρητός Έλεγχος:** Μπορεί να χρησιμοποιείτε την TN ως *Διακριτό Συστατικό* ή σε μια προκαθορισμένη ροή εργασίας που απαιτεί την εκτέλεση μιας συγκεκριμένης συνάρτησης σε συγκεκριμένη χρονική στιγμή. Εξαναγκάζοντας την κλήση, μπορείτε να εγγυηθείτε ότι η επιθυμητή συνάρτηση θα κληθεί αντί να πρέπει να ζητήσετε ευγενικά από την TN να το κάνει.
2. **Αποσφαλμάτωση και Δοκιμές:** Κατά την ανάπτυξη και δοκιμή εφαρμογών που βασίζονται στην TN, η δυνατότητα εξαναγκασμού κλήσεων συναρτήσεων είναι ανεκτίμητη για σκοπούς αποσφαλμάτωσης. Με την ρητή ενεργοποίηση συγκεκριμένων συναρτήσεων, μπορείτε να απομονώσετε και να δοκιμάσετε μεμονωμένα συστατικά της εφαρμογής σας. Αυτό σας επιτρέπει να επαληθεύσετε την ορθότητα των υλοποιήσεων των συναρτήσεων, να επικυρώσετε τις παραμέτρους εισόδου και να διασφαλίσετε ότι επιστρέφονται τα αναμενόμενα αποτελέσματα.

3. **Χειρισμός Οριακών Περιπτώσεων:** Μπορεί να υπάρχουν οριακές περιπτώσεις ή εξαιρετικά σενάρια όπου η δυναμική διαδικασία επιλογής της TN μπορεί να μην επιλέξει να εκτελέσει μια συνάρτηση που θα έπρεπε, και το γνωρίζετε αυτό βάσει εξωτερικών διαδικασιών. Σε τέτοιες περιπτώσεις, η δυνατότητα εξαναγκασμού κλήσης μιας συνάρτησης σας επιτρέπει να χειριστείτε αυτές τις καταστάσεις ρητά. Καθορίστε κανόνες ή συνθήκες στη λογική της εφαρμογής σας για να προσδιορίσετε πότε να παρακάμψετε την κρίση της TN.
4. **Συνέπεια και Αναπαραγωγιμότητα:** Εάν έχετε μια συγκεκριμένη ακολουθία συναρτήσεων που πρέπει να εκτελεστούν σε συγκεκριμένη σειρά, ο εξαναγκασμός των κλήσεων εγγυάται ότι η ίδια ακολουθία ακολουθείται κάθε φορά. Αυτό είναι ιδιαίτερα σημαντικό σε εφαρμογές όπου η συνέπεια και η προβλέψιμη συμπεριφορά είναι κρίσιμες, όπως σε χρηματοοικονομικά συστήματα ή επιστημονικές προσομοιώσεις.
5. **Βελτιστοποίηση Απόδοσης:** Σε ορισμένες περιπτώσεις, ο εξαναγκασμός κλήσης μιας συνάρτησης μπορεί να οδηγήσει σε βελτιστοποιήσεις απόδοσης. Εάν γνωρίζετε ότι μια συγκεκριμένη συνάρτηση απαιτείται για μια συγκεκριμένη εργασία και ότι η δυναμική διαδικασία επιλογής της TN μπορεί να εισαγάγει περιττή επιβάρυνση, μπορείτε να παρακάμψετε τη διαδικασία επιλογής και να καλέσετε απευθείας την απαιτούμενη συνάρτηση. Αυτό μπορεί να βοηθήσει στη μείωση της καθυστέρησης και στη βελτίωση της συνολικής αποδοτικότητας της εφαρμογής σας.

Συνοπτικά, η δυνατότητα εξαναγκασμού κλήσεων συναρτήσεων σε εφαρμογές που βασίζονται στην TN παρέχει ρητό έλεγχο, βοηθά στην αποσφαλμάτωση και τον έλεγχο, χειρίζεται οριακές περιπτώσεις, εξασφαλίζει συνέπεια και αναπαραγωγιμότητα. Είναι ένα ισχυρό εργαλείο στο οπλοστάσιό σας, αλλά πρέπει να συζητήσουμε ακόμη μία πτυχή αυτού του σημαντικού χαρακτηριστικού.



Σε πολλές περιπτώσεις λήψης αποφάσεων, θέλουμε πάντα το μοντέλο να κάνει μια κλήση συνάρτησης και μπορεί να μην θέλουμε ποτέ το μοντέλο να απαντήσει μόνο με την εσωτερική του γνώση. Για παράδειγμα, εάν δρομολογείτε μεταξύ πολλαπλών μοντέλων εξειδικευμένων σε διαφορετικές εργασίες (πολύγλωσση είσοδος, μαθηματικά, κλπ), μπορεί να χρησιμοποιήσετε το μοντέλο κλήσης συναρτήσεων για να αναθέσετε αιτήματα σε ένα από τα βοηθητικά μοντέλα και να μην απαντήσετε ανεξάρτητα.

Παράμετρος Επιλογής Εργαλείου

Το GPT-4 και άλλα γλωσσικά μοντέλα που υποστηρίζουν κλήση συναρτήσεων σας δίνουν μια παράμετρο `tool_choice` για τον έλεγχο του εάν η χρήση εργαλείου απαιτείται ως μέρος μιας ολοκλήρωσης. Αυτή η παράμετρος έχει τρεις πιθανές τιμές:

- Το `auto` δίνει στην TN πλήρη διακριτική ευχέρεια για τη χρήση ενός εργαλείου ή απλά για την απάντηση
- Το `required` λέει στην TN ότι *πρέπει* να καλέσει ένα εργαλείο *αντί* να απαντήσει, αλλά αφήνει την επιλογή του εργαλείου στην TN
- Η τρίτη επιλογή είναι να ορίσετε την παράμετρο του `name_of_function` που θέλετε να εξαναγκάσετε. Περισσότερα γι' αυτό στην επόμενη ενότητα.



Σημειώστε ότι αν ορίσετε το `tool choice` ως `required`, το μοντέλο θα αναγκαστεί να επιλέξει την πιο σχετική συνάρτηση για κλήση από αυτές που του παρέχονται, ακόμη και αν καμία δεν ταιριάζει πραγματικά με το `prompt`. Κατά τη στιγμή της δημοσίευσης, δεν γνωρίζω κάποιο μοντέλο που θα επιστρέψει μια κενή απόκριση `tool_calls`, ή θα χρησιμοποιήσει κάποιον άλλο τρόπο για να σας ενημερώσει ότι δεν βρήκε κατάλληλη συνάρτηση για κλήση.

Εξαναγκασμός Συνάρτησης για Λήψη Δομημένης Εξόδου

Η δυνατότητα εξαναγκασμού μιας κλήσης συνάρτησης σας δίνει έναν τρόπο να αποκτήσετε δομημένα δεδομένα από μια ολοκλήρωση συνομιλίας αντί να πρέπει να τα εξάγετε μόνοι σας από την απόκριση απλού κειμένου.

Γιατί είναι τόσο σημαντικός ο εξαναγκασμός συναρτήσεων για τη λήψη δομημένης εξόδου; Απλά, επειδή η εξαγωγή δομημένων δεδομένων από την έξοδο ενός LLM είναι πονοκέφαλος. Μπορείτε να κάνετε τη ζωή σας λίγο πιο εύκολη ζητώντας δεδομένα σε XML, αλλά μετά πρέπει να αναλύσετε το XML. Και τι κάνετε όταν αυτό το XML λείπει επειδή η AI σας απάντησε: “Λυπάμαι, αλλά δεν μπορώ να δημιουργήσω τα δεδομένα που ζητήσατε επειδή μπλα, μπλα, μπλα...”

Όταν χρησιμοποιείτε εργαλεία με αυτόν τον τρόπο:

- Πιθανότατα θα πρέπει να ορίσετε ένα μόνο εργαλείο στο αίτημά σας
- Θυμηθείτε να εξαναγκάσετε τη χρήση της συνάρτησής του χρησιμοποιώντας την παράμετρο `tool_choice`
- Θυμηθείτε ότι το μοντέλο θα περάσει την είσοδο στο εργαλείο, οπότε το όνομα του εργαλείου και η περιγραφή θα πρέπει να είναι από την οπτική του μοντέλου, όχι τη δική σας.

Αυτό το τελευταίο σημείο αξίζει ένα παράδειγμα για διευκρίνιση. Ας πούμε ότι ζητάτε από την AI να κάνει ανάλυση συναισθήματος σε κείμενο χρήστη. Το όνομα της συνάρτησης δεν θα ήταν `analyze_sentiment`, αλλά μάλλον κάτι σαν `save_sentiment_analysis`. Η AI είναι αυτή που κάνει την ανάλυση συναισθήματος, *όχι το εργαλείο*. Το μόνο που κάνει το εργαλείο (από την οπτική της AI) είναι να αποθηκεύει τα αποτελέσματα της ανάλυσης.

Ακολουθεί ένα παράδειγμα χρήσης του Claude 3 για την καταγραφή μιας περίληψης εικόνας σε καλά δομημένο JSON, αυτή τη φορά από τη γραμμή εντολών χρησιμοποιώντας curl:

```

1 curl https://api.anthropic.com/v1/messages \
2 --header "content-type: application/json" \
3 --header "x-api-key: $ANTHROPIC_API_KEY" \
4 --header "anthropic-version: 2023-06-01" \
5 --header "anthropic-beta: tools-2024-04-04" \
6 --data \
7 '{
8   "model": "claude-3-sonnet-20240229",
9   "max_tokens": 1024,
10  "tools": [{
11    "name": "record_summary",
12    "description": "Record summary of image into well-structured JSON.",
13    "input_schema": {
14      "type": "object",
15      "properties": {
16        "key_colors": {
17          "type": "array",
18          "items": {
19            "type": "object",
20            "properties": {
21              "r": {
22                "type": "number",
23                "description": "red value [0.0, 1.0]"
24              },
25              "g": {
26                "type": "number",
27                "description": "green value [0.0, 1.0]"
28              },
29              "b": {
30                "type": "number",
31                "description": "blue value [0.0, 1.0]"
32              },
33              "name": {
34                "type": "string",
35                "description": "Human-readable color name
36                               in snake_case, e.g.
37                               \"olive_green\" or

```

```

38         \"turquoise\"
39     },
40     },
41     \"required\": [ \"r\", \"g\", \"b\", \"name\" ]
42 },
43     \"description\": \"Key colors in the image. Four or less.\"
44 },
45     \"description\": {
46         \"type\": \"string\",
47         \"description\": \"Image description. 1-2 sentences max.\"
48     },
49     \"estimated_year\": {
50         \"type\": \"integer\",
51         \"description\": \"Estimated year that the image was taken,
52                             if is it a photo. Only set this if the
53                             image appears to be non-fictional.
54                             Rough estimates are okay!\"
55     }
56 },
57     \"required\": [ \"key_colors\", \"description\" ]
58 }
59 ]],
60 \"messages\": [
61     {
62         \"role\": \"user\",
63         \"content\": [
64             {
65                 \"type\": \"image\",
66                 \"source\": {
67                     \"type\": \"base64\",
68                     \"media_type\": \"'$IMAGE_MEDIA_TYPE'\",
69                     \"data\": \"'$IMAGE_BASE64'\"
70                 }
71             },
72             {
73                 \"type\": \"text\",
74                 \"text\": \"Use `record_summary` to describe this image.\"
75             }
76         ]
77     }
78 ]
79 }'

```

Στο παρεχόμενο παράδειγμα, χρησιμοποιούμε το μοντέλο Claude 3 από την Anthropic για να δημιουργήσουμε μια δομημένη περίληψη JSON μιας εικόνας. Δείτε πώς λειτουργεί:

1. Ορίζουμε ένα μοναδικό εργαλείο με το όνομα `record_summary` στον πίνακα `tools` του ωφέλιμου φορτίου του αιτήματος. Αυτό το εργαλείο είναι υπεύθυνο για την καταγραφή μιας περίληψης της εικόνας σε καλά δομημένο JSON.
2. Το εργαλείο `record_summary` έχει ένα `input_schema` που καθορίζει την αναμενόμενη δομή της εξόδου JSON. Ορίζει τρεις ιδιότητες:
 - `key_colors`: Ένας πίνακας αντικειμένων που αντιπροσωπεύει τα βασικά χρώματα στην εικόνα. Κάθε αντικείμενο χρώματος έχει ιδιότητες για τις τιμές κόκκινου, πράσινου και μπλε (που κυμαίνονται από 0.0 έως 1.0) και ένα αναγνώσιμο όνομα χρώματος σε μορφή `snake_case`.
 - `description`: Μια ιδιότητα συμβολοσειράς για μια σύντομη περιγραφή της εικόνας, περιορισμένη σε 1-2 προτάσεις.
 - `estimated_year`: Μια προαιρετική ιδιότητα ακέραιου αριθμού για το εκτιμώμενο έτος λήψης της εικόνας, εάν φαίνεται να είναι μια μη μυθοπλαστική φωτογραφία.
3. Στον πίνακα `messages`, παρέχουμε τα δεδομένα της εικόνας ως συμβολοσειρά κωδικοποιημένη σε `base64` μαζί με τον τύπο `polymers`. Αυτό επιτρέπει στο μοντέλο να επεξεργαστεί την εικόνα ως μέρος της εισόδου.
4. Επίσης, προτρέπουμε το Claude να χρησιμοποιήσει το εργαλείο `record_summary` για να περιγράψει την εικόνα.
5. Όταν το αίτημα αποστέλλεται στο μοντέλο Claude 3, αναλύει την εικόνα και δημιουργεί μια περίληψη JSON με βάση το καθορισμένο `input_schema`. Το μοντέλο εξάγει τα βασικά χρώματα, παρέχει μια σύντομη περιγραφή και εκτιμά το έτος λήψης της εικόνας (εάν ισχύει).

6. Η δημιουργημένη περίληψη JSON περνάει ως παράμετρος στο εργαλείο `record_summary`, παρέχοντας μια δομημένη αναπαράσταση των βασικών χαρακτηριστικών της εικόνας.

Χρησιμοποιώντας το εργαλείο `record_summary` με ένα καλά καθορισμένο `input_schema`, μπορούμε να λάβουμε μια δομημένη περίληψη JSON μιας εικόνας χωρίς να βασιζόμαστε στην εξαγωγή απλού κειμένου. Αυτή η προσέγγιση διασφαλίζει ότι η έξοδος ακολουθεί μια συνεπή μορφή και μπορεί εύκολα να αναλυθεί και να επεξεργαστεί από τα κατάντη συστατικά της εφαρμογής.

Η δυνατότητα επιβολής μιας κλήσης συνάρτησης και καθορισμού της αναμενόμενης δομής εξόδου είναι ένα ισχυρό χαρακτηριστικό της χρήσης εργαλείων σε εφαρμογές που βασίζονται στην TN. Επιτρέπει στους προγραμματιστές να έχουν περισσότερο έλεγχο στην παραγόμενη έξοδο και απλοποιεί την ενσωμάτωση των δεδομένων που παράγονται από την TN στη ροή εργασίας της εφαρμογής τους.

Εκτέλεση Συνάρτησης(-ων)

Έχετε ορίσει συναρτήσεις και έχετε δώσει εντολή στην TN σας, η οποία αποφάσισε ότι πρέπει να καλέσει μία από τις συναρτήσεις σας. Τώρα είναι η στιγμή για τον κώδικα της εφαρμογής σας ή τη βιβλιοθήκη, αν χρησιμοποιείτε ένα Ruby gem όπως το [raix-rails](#) να δρομολογήσει την κλήση της συνάρτησης και τις παραμέτρους της στην αντίστοιχη υλοποίηση *στον κώδικα της εφαρμογής σας*.

Ο κώδικας της εφαρμογής σας αποφασίζει τι θα κάνει με τα αποτελέσματα της εκτέλεσης της συνάρτησης. Ίσως αυτό που πρέπει να γίνει περιλαμβάνει μία μόνο γραμμή κώδικα σε μια `lambda`, ή ίσως περιλαμβάνει την κλήση ενός εξωτερικού API. Ίσως περιλαμβάνει την κλήση ενός άλλου στοιχείου TN, ή ίσως περιλαμβάνει εκατοντάδες ή ακόμη και χιλιάδες γραμμές κώδικα στο υπόλοιπο σύστημά σας. Είναι εντελώς στο χέρι σας.

Μερικές φορές η κλήση της συνάρτησης είναι το τέλος της λειτουργίας, αλλά αν τα αποτελέσματα αντιπροσωπεύουν πληροφορίες σε μια αλυσίδα σκέψης που πρέπει

να συνεχιστεί από την TN, τότε ο κώδικας της εφαρμογής σας πρέπει να εισάγει τα αποτελέσματα της εκτέλεσης στην καταγραφή της συνομιλίας και να αφήσει την TN να συνεχίσει την επεξεργασία.

Για παράδειγμα, εδώ είναι μια δήλωση συνάρτησης [Raix](#) που χρησιμοποιείται από το AccountManager της Olympia για την επικοινωνία με τους πελάτες μας ως μέρος μιας Ευφυούς Ενορχήστρωσης Ροής Εργασίας για την εξυπηρέτηση πελατών.

```
1 class AccountManager
2   include Raix::ChatCompletion
3   include Raix::FunctionDispatch
4
5   # lots of other functions...
6
7   function :notify_account_owner,
8     "Don't share UUID. Mention dollars if subscription changed",
9     message: { type: "string" } do |arguments|
10     account.owner.freeform_notify(
11       subject: "Account Change Notification",
12       message: arguments[:message]
13     )
14     "Notified account owner"
15   end
```

Ίσως να μην είναι άμεσα κατανοητό τι συμβαίνει εδώ, οπότε θα το αναλύσω.

1. Η κλάση AccountManager ορίζει πολλές συναρτήσεις σχετικές με τη διαχείριση λογαριασμών. Μπορεί να αλλάξει το πλάνο σας, να προσθέσει και να αφαιρέσει μέλη ομάδας, μεταξύ άλλων.
2. Οι οδηγίες ανώτατου επιπέδου ενημερώνουν το AccountManager ότι θα πρέπει να ειδοποιήσει τον ιδιοκτήτη του λογαριασμού με τα αποτελέσματα του αιτήματος αλλαγής λογαριασμού, χρησιμοποιώντας τη συνάρτηση notify_account_owner.
3. Ο συνοπτικός ορισμός της συνάρτησης περιλαμβάνει:
 - το όνομα

- την περιγραφή
- τις παραμέτρους `message: { type: "string" }`
- ένα μπλοκ προς εκτέλεση όταν καλείται η συνάρτηση

Μετά την ενημέρωση της καταγραφής με τα αποτελέσματα του μπλοκ συνάρτησης, καλείται ξανά η μέθοδος `chat_completion`. Αυτή η μέθοδος είναι υπεύθυνη για την αποστολή της ενημερωμένης καταγραφής συνομιλίας πίσω στο μοντέλο ΑΙ για περαιτέρω επεξεργασία. Αναφερόμαστε σε αυτή τη διαδικασία ως *βρόχο συνομιλίας*.

Όταν το μοντέλο ΑΙ λαμβάνει ένα νέο αίτημα ολοκλήρωσης συνομιλίας με μια ενημερωμένη καταγραφή, έχει πρόσβαση στα αποτελέσματα της προηγούμενης εκτελεσμένης συνάρτησης. Μπορεί να αναλύσει αυτά τα αποτελέσματα, να τα ενσωματώσει στη διαδικασία λήψης αποφάσεων και να δημιουργήσει την επόμενη απάντηση ή ενέργεια με βάση το συσσωρευμένο πλαίσιο της συνομιλίας. Μπορεί να επιλέξει να εκτελέσει πρόσθετες συναρτήσεις με βάση το ενημερωμένο πλαίσιο, ή να δημιουργήσει μια τελική απάντηση στην αρχική προτροπή εάν κρίνει ότι δεν απαιτούνται περαιτέρω κλήσεις συναρτήσεων.

Προαιρετική Συνέχεια της Αρχικής Προτροπής

Όταν στέλνετε τα αποτελέσματα εργαλείων πίσω στο LLM και συνεχίζετε την επεξεργασία της αρχικής προτροπής, η ΑΙ χρησιμοποιεί αυτά τα αποτελέσματα είτε για να καλέσει πρόσθετες συναρτήσεις είτε για να δημιουργήσει μια τελική απάντηση απλού κειμένου.



Ορισμένα μοντέλα όπως το [Command-R](#) της Cohere μπορούν να αναφέρουν τα συγκεκριμένα εργαλεία που χρησιμοποίησαν στις απαντήσεις τους, παρέχοντας πρόσθετη διαφάνεια και ιχνηλασιμότητα.

Ανάλογα με το μοντέλο που χρησιμοποιείται, τα αποτελέσματα της κλήσης συνάρτησης θα βρίσκονται σε μηνύματα καταγραφής που έχουν τον δικό τους

ειδικό ρόλο ή θα αντικατοπτρίζονται σε κάποια άλλη σύνταξη. Αλλά το σημαντικό μέρος είναι αυτά τα δεδομένα να βρίσκονται στην καταγραφή, ώστε να μπορούν να ληφθούν υπόψη από την ΑΙ καθώς αποφασίζει τι θα κάνει στη συνέχεια.



Μια συνηθισμένη (και δυνητικά ακριβή) συνθήκη σφάλματος είναι να ξεχάσετε να προσθέσετε τα αποτελέσματα της συνάρτησης στην καταγραφή πριν συνεχίσετε τη συνομιλία. Ως αποτέλεσμα, η ΑΙ θα λάβει προτροπή ουσιαστικά με τον ίδιο τρόπο όπως πριν καλέσει τη συνάρτηση την πρώτη φορά. Με άλλα λόγια, όσον αφορά την ΑΙ, δεν έχει καλέσει ακόμη τη συνάρτηση. Οπότε την καλεί ξανά. Και ξανά. Και ξανά, για πάντα μέχρι να τη διακόψετε. Ελπίζω το πλαίσίό σας να μην ήταν πολύ μεγάλο και το μοντέλο σας να μην ήταν πολύ ακριβό!

Βέλτιστες Πρακτικές για τη Χρήση Εργαλείων

Για να αξιοποιήσετε στο έπακρο τη χρήση εργαλείων, εξετάστε τις ακόλουθες βέλτιστες πρακτικές.

Περιγραφικοί Ορισμοί

Παρέχετε σαφή και περιγραφικά ονόματα και περιγραφές για κάθε εργαλείο και τις παραμέτρους εισόδου του. Αυτό βοηθά το LLM (Μοντέλο Μεγάλης Γλώσσας) να κατανοήσει καλύτερα τον σκοπό και τις δυνατότητες κάθε εργαλείου.

Μπορώ να σας πω από προσωπική εμπειρία ότι η κοινή σοφία που λέει ότι “η ονοματοδοσία είναι δύσκολη” ισχύει και εδώ· έχω δει δραματικά διαφορετικά αποτελέσματα από τα LLMs απλά αλλάζοντας τα ονόματα των συναρτήσεων ή τη διατύπωση των περιγραφών. Μερικές φορές η αφαίρεση περιγραφών *βελτιώνει*

την απόδοση.

Επεξεργασία Αποτελεσμάτων Εργαλείων

Κατά τη μεταβίβαση των αποτελεσμάτων εργαλείων πίσω στο LLM, βεβαιωθείτε ότι είναι καλά δομημένα και περιεκτικά. Χρησιμοποιήστε ουσιαστικά κλειδιά και τιμές για να αναπαραστήσετε την έξοδο κάθε εργαλείου. Πειραματιστείτε με διαφορετικές μορφές και δείτε ποια λειτουργεί καλύτερα, από JSON μέχρι απλό κείμενο.

Ο [Διερμηνέας Αποτελεσμάτων](#) αντιμετωπίζει αυτήν την πρόκληση χρησιμοποιώντας την τεχνητή νοημοσύνη για να αναλύσει τα αποτελέσματα και να παρέχει φιλικές προς τον άνθρωπο εξηγήσεις, περιλήψεις ή βασικά συμπεράσματα.

Χειρισμός Σφαλμάτων

Εφαρμόστε ισχυρούς μηχανισμούς χειρισμού σφαλμάτων για να διαχειριστείτε περιπτώσεις όπου το LLM μπορεί να δημιουργήσει μη έγκυρες ή μη υποστηριζόμενες παραμέτρους εισόδου για κλήσεις εργαλείων. Χειριστείτε ομαλά και ανακάμψτε από τυχόν σφάλματα που μπορεί να προκύψουν κατά την εκτέλεση του εργαλείου.

Μια εξαιρετικά καλή ιδιότητα της τεχνητής νοημοσύνης είναι ότι κατανοεί τα μηνύματα σφάλματος! Που σημαίνει ότι αν εργάζεστε με γρήγορη και πρόχειρη νοοτροπία, μπορείτε απλά να συλλάβετε οποιοσδήποτε εξαιρέσεις που δημιουργούνται κατά την υλοποίηση ενός εργαλείου και να τις επιστρέψετε στην τεχνητή νοημοσύνη ώστε να γνωρίζει τι συνέβη!

Για παράδειγμα, εδώ είναι μια απλοποιημένη έκδοση της υλοποίησης αναζήτησης Google στο Olympia:

```
1  def google_search(conversation, params)
2      conversation.update_cstatus("Searching Google...")
3      query = params[:query]
4      search = GoogleSearch.new(query).get_hash
5
6      conversation.update_cstatus("Summarizing results...")
7      SummarizeKnowledgeGraph.new.perform(conversation, search.to_json)
8  rescue StandardError => e
9      Honeybadger.notify(e)
10     { error: e.message }.inspect
11 end
```

Οι αναζητήσεις Google στην Olympia είναι μια διαδικασία δύο βημάτων. Πρώτα κάνετε την αναζήτηση και μετά συνοψίζετε τα αποτελέσματα. Εάν υπάρξει κάποια αποτυχία, ανεξάρτητα από το τι είναι, το μήνυμα σφάλματος συσκευάζεται και αποστέλλεται πίσω στην TN. Αυτή η τεχνική είναι το θεμέλιο πρακτικά όλων των προτύπων *Ευφυούς Χειρισμού Σφαλμάτων*

Για παράδειγμα, ας υποθέσουμε ότι η κλήση του API GoogleSearch αποτυγχάνει λόγω εξαίρεσης 503 Μη Διαθέσιμη Υπηρεσία. Αυτό ανεβαίνει στο ανώτατο επίπεδο διάσωσης, και η περιγραφή του σφάλματος αποστέλλεται πίσω στην TN ως αποτέλεσμα της κλήσης της συνάρτησης. Αντί να δώσει στον χρήστη απλώς μια κενή οθόνη ή ένα τεχνικό σφάλμα, η TN λέει κάτι σαν “Λυπάμαι, αλλά δεν μπορώ να αποκτήσω πρόσβαση στις δυνατότητες Αναζήτησης Google αυτή τη στιγμή. Μπορώ να προσπαθήσω ξανά αργότερα, αν θέλετε.”

Αυτό μπορεί να φαίνεται απλώς ως ένα έξυπνο κόλπο, αλλά σκεφτείτε ένα διαφορετικό είδος σφάλματος, όπου η TN καλούσε ένα εξωτερικό API και είχε άμεσο έλεγχο των παραμέτρων που έπρεπε να περάσει στο API. Ίσως έκανε κάποιο λάθος στον τρόπο που δημιούργησε αυτές τις παραμέτρους; Με την προϋπόθεση ότι το μήνυμα σφάλματος από το εξωτερικό API είναι αρκετά λεπτομερές, η επιστροφή του μηνύματος σφάλματος πίσω στην καλούσα TN σημαίνει ότι μπορεί να επανεξετάσει αυτές τις παραμέτρους και να προσπαθήσει ξανά. Αυτόματα. Ανεξάρτητα από το τι ήταν το σφάλμα.

Τώρα σκεφτείτε τι θα χρειαζόταν για να αναπαράγετε αυτού του είδους τον ισχυρό χειρισμό σφαλμάτων σε κανονικό κώδικα. Είναι πρακτικά αδύνατο.

Επαναληπτική Βελτίωση

Εάν το LLM δεν προτείνει τα κατάλληλα εργαλεία ή παράγει υποβέλτιστες απαντήσεις, επαναλάβετε τους ορισμούς εργαλείων, τις περιγραφές και τις παραμέτρους εισόδου. Συνεχίστε να βελτιώνετε και να εξελίσσετε τη ρύθμιση των εργαλείων με βάση την παρατηρούμενη συμπεριφορά και τα επιθυμητά αποτελέσματα.

1. Ξεκινήστε με απλούς ορισμούς εργαλείων: Αρχίστε ορίζοντας εργαλεία με σαφή και συνοπτικά ονόματα, περιγραφές και παραμέτρους εισόδου. Αποφύγετε την υπερβολική περιπλοκότητα στη ρύθμιση των εργαλείων αρχικά και εστιάστε στη βασική λειτουργικότητα. Για παράδειγμα, αν θέλετε να αποθηκεύσετε τα αποτελέσματα της ανάλυσης συναισθημάτων, ξεκινήστε με έναν βασικό ορισμό όπως:

```
1 {  
2   "name": "save_sentiment_score",  
3   "description": "Analyze user-provided text and generate sentiment score",  
4   "parameters": {  
5     "type": "object",  
6     "properties": {  
7       "score": {  
8         "type": "float",  
9         "description": "sentiment score from -1 (negative) to 1 (positive)"  
10      }  
11    },  
12    "required": ["score"]  
13  }  
14 }
```

2. Δοκιμή και παρατήρηση: Μόλις έχετε τους αρχικούς ορισμούς εργαλείων στη θέση τους, δοκιμάστε τους με διαφορετικές προτροπές και παρατηρήστε πώς

το MGM αλληλεπιδρά με το εργαλείο. Δώστε προσοχή στην ποιότητα και τη συνάφεια των παραγόμενων αποκρίσεων. Αν το MGM παράγει υποβέλτιστες αποκρίσεις, είναι ώρα να βελτιώσετε τους ορισμούς των εργαλείων.

3. Βελτίωση περιγραφών: Αν το MGM παρερμηνεύει τον σκοπό ενός εργαλείου, προσπαθήστε να βελτιώσετε την περιγραφή του εργαλείου. Παρέχετε περισσότερο πλαίσιο, παραδείγματα ή διευκρινίσεις για να καθοδηγήσετε το MGM στην αποτελεσματική χρήση του εργαλείου. Για παράδειγμα, μπορείτε να ενημερώσετε την περιγραφή του εργαλείου ανάλυσης συναισθήματος ώστε να αντιμετωπίζει πιο συγκεκριμένα τον *συναισθηματικό τόνο* του κειμένου που αναλύεται:

```
1 {  
2   "name": "save_sentiment_score",  
3   "description": "Determine the overall emotional tone of a piece of text,  
4     such as customer reviews, social media posts, or feedback comments.",  
5   ...  
6 }
```

4. Προσαρμόστε τις παραμέτρους εισόδου: Εάν το MGM παράγει μη έγκυρες ή άσχετες παραμέτρους εισόδου για ένα εργαλείο, εξετάστε το ενδεχόμενο προσαρμογής των ορισμών των παραμέτρων. Προσθέστε πιο συγκεκριμένους περιορισμούς, κανόνες επικύρωσης ή παραδείγματα για να διευκρινίσετε την αναμενόμενη μορφή εισόδου.
5. Επαναλάβετε με βάση την ανατροφοδότηση: Παρακολουθείτε συνεχώς την απόδοση των εργαλείων σας και συγκεντρώνετε σχόλια από χρήστες ή ενδιαφερόμενα μέρη. Χρησιμοποιήστε αυτή την ανατροφοδότηση για να εντοπίσετε τομείς που χρειάζονται βελτίωση και να κάνετε επαναληπτικές βελτιώσεις στους ορισμούς των εργαλείων. Για παράδειγμα, εάν οι χρήστες αναφέρουν ότι η ανάλυση δεν χειρίζεται καλά τον σαρκασμό, μπορείτε να προσθέσετε μια σημείωση στην περιγραφή:

```
1 {  
2   "name": "save_sentiment_score",  
3   "description": "Analyze the sentiment of a given text and return a sentiment  
4     score between -1 (negative) and 1 (positive). Note: Sarcasm should be  
5     considered negative.",  
6   ...  
7 }
```

Βελτιώνοντας επαναληπτικά τους ορισμούς των εργαλείων σας με βάση την παρατηρούμενη συμπεριφορά και την ανατροφοδότηση, μπορείτε σταδιακά να βελτιώσετε την απόδοση και την αποτελεσματικότητα της εφαρμογής σας που βασίζεται στην TN. Θυμηθείτε να διατηρείτε τους ορισμούς των εργαλείων σαφείς, συνοπτικούς και εστιασμένους στη συγκεκριμένη εργασία. Ελέγχετε και επικυρώνετε τακτικά τις αλληλεπιδράσεις των εργαλείων για να διασφαλίσετε ότι ευθυγραμμίζονται με τα επιθυμητά αποτελέσματα.

Σύνθεση και Αλυσίδωση Εργαλείων

Ένα από τα πιο ισχυρά χαρακτηριστικά της χρήσης εργαλείων που έχει μόνο υπαινιχθεί μέχρι τώρα είναι η δυνατότητα σύνθεσης και αλυσίδωσης πολλαπλών εργαλείων για την επίτευξη σύνθετων εργασιών. Σχεδιάζοντας προσεκτικά τους ορισμούς των εργαλείων σας και τις μορφές εισόδου/εξόδου τους, μπορείτε να δημιουργήσετε επαναχρησιμοποιήσιμα δομικά στοιχεία που μπορούν να συνδυαστούν με διάφορους τρόπους.

Ας εξετάσουμε ένα παράδειγμα όπου δημιουργείτε μια διαδικασία ανάλυσης δεδομένων για την εφαρμογή σας που βασίζεται στην TN. Μπορεί να έχετε τα ακόλουθα εργαλεία:

1. **DataRetrieval:** Ένα εργαλείο που ανακτά δεδομένα από μια βάση δεδομένων ή API με βάση συγκεκριμένα κριτήρια.

2. **DataProcessing:** Ένα εργαλείο που εκτελεί υπολογισμούς, μετασχηματισμούς ή συγκεντρώσεις στα ανακτημένα δεδομένα.
3. **DataVisualization:** Ένα εργαλείο που παρουσιάζει τα επεξεργασμένα δεδομένα σε μια φιλική προς το χρήστη μορφή, όπως διαγράμματα ή γραφήματα.

Αλυσιδώνοντας αυτά τα εργαλεία μεταξύ τους, μπορείτε να δημιουργήσετε μια ισχυρή ροή εργασίας που ανακτά σχετικά δεδομένα, τα επεξεργάζεται και παρουσιάζει τα αποτελέσματα με ουσιαστικό τρόπο. Ιδού πώς μπορεί να μοιάζει η ροή εργασίας χρήσης εργαλείων:

1. Το LLM λαμβάνει ένα ερώτημα χρήστη ζητώντας πληροφορίες σχετικά με τα δεδομένα πωλήσεων για μια συγκεκριμένη κατηγορία προϊόντων.
2. Το LLM επιλέγει το εργαλείο DataRetrieval και δημιουργεί τις κατάλληλες παραμέτρους εισόδου για να ανακτήσει τα σχετικά δεδομένα πωλήσεων από τη βάση δεδομένων.
3. Τα ανακτημένα δεδομένα “περνούν” στο εργαλείο DataProcessing, το οποίο υπολογίζει μετρήσεις όπως συνολικά έσοδα, μέση τιμή πώλησης και ρυθμό ανάπτυξης.
4. Τα επεξεργασμένα δεδομένα στη συνέχεια αφομοιώνονται από το εργαλείο DataVisualization, το οποίο δημιουργεί ένα οπτικά ελκυστικό διάγραμμα ή γράφημα για την αναπαράσταση των πληροφοριών, επιστρέφοντας το URL του διαγράμματος στο LLM.
5. Τέλος, το LLM δημιουργεί μια μορφοποιημένη απάντηση στο ερώτημα του χρήστη χρησιμοποιώντας markdown, ενσωματώνοντας τα οπτικοποιημένα δεδομένα και παρέχοντας μια σύνοψη των βασικών ευρημάτων.

Συνθέτοντας αυτά τα εργαλεία μαζί, μπορείτε να δημιουργήσετε μια απρόσκοπτη ροή εργασίας ανάλυσης δεδομένων που μπορεί εύκολα να ενσωματωθεί στην εφαρμογή σας. Η ομορφιά αυτής της προσέγγισης είναι ότι κάθε εργαλείο μπορεί

να αναπτυχθεί και να δοκιμαστεί ανεξάρτητα, και στη συνέχεια να συνδυαστεί με διαφορετικούς τρόπους για την επίλυση διαφόρων προβλημάτων.

Για να καταστεί δυνατή η ομαλή σύνθεση και αλυσίδωση εργαλείων, είναι σημαντικό να καθοριστούν σαφείς μορφές εισόδου και εξόδου για κάθε εργαλείο.

Για παράδειγμα, το εργαλείο DataRetrieval μπορεί να δέχεται παραμέτρους όπως τα στοιχεία σύνδεσης βάσης δεδομένων, το όνομα πίνακα και τις συνθήκες ερωτήματος, και να επιστρέφει το σύνολο αποτελεσμάτων ως δομημένο αντικείμενο JSON. Το εργαλείο DataProcessing μπορεί στη συνέχεια να δεχτεί αυτό το αντικείμενο JSON ως είσοδο και να παράγει ένα μετασχηματισμένο αντικείμενο JSON ως έξοδο. Με την τυποποίηση της ροής δεδομένων μεταξύ εργαλείων, μπορείτε να διασφαλίσετε τη συμβατότητα και την επαναχρησιμοποίηση.

Καθώς σχεδιάζετε το οικοσύστημα εργαλείων σας, σκεφτείτε πώς διαφορετικά εργαλεία μπορούν να συνδυαστούν για την αντιμετώπιση συνηθισμένων περιπτώσεων χρήσης στην εφαρμογή σας. Εξετάστε τη δημιουργία εργαλείων υψηλού επιπέδου που ενσωματώνουν κοινές ροές εργασίας ή επιχειρησιακή λογική, καθιστώντας ευκολότερο για το LLM να τα επιλέξει και να τα χρησιμοποιήσει αποτελεσματικά.

Θυμηθείτε, η δύναμη της χρήσης εργαλείων έγκειται στην ευελιξία και τη δομοστοιχείωση που παρέχει. Διασπώντας τις πολύπλοκες εργασίες σε μικρότερα, επαναχρησιμοποιήσιμα εργαλεία, μπορείτε να δημιουργήσετε μια εύρωστη και προσαρμόσιμη εφαρμογή που βασίζεται στην τεχνητή νοημοσύνη και μπορεί να αντιμετωπίσει ένα ευρύ φάσμα προκλήσεων.

Μελλοντικές Κατευθύνσεις

Καθώς εξελίσσεται ο τομέας της ανάπτυξης εφαρμογών που βασίζονται στην τεχνητή νοημοσύνη, μπορούμε να αναμένουμε περαιτέρω εξελίξεις στις δυνατότητες χρήσης εργαλείων. Μερικές πιθανές μελλοντικές κατευθύνσεις περιλαμβάνουν:

1. **Χρήση Εργαλείων Πολλαπλών Βημάτων:** Τα LLMs ενδέχεται να μπορούν να αποφασίσουν πόσες φορές χρειάζεται να χρησιμοποιήσουν εργαλεία για να δημιουργήσουν μια ικανοποιητική απάντηση. Αυτό θα μπορούσε να περιλαμβάνει πολλαπλούς γύρους επιλογής και εκτέλεσης εργαλείων με βάση τα ενδιάμεσα αποτελέσματα.
2. **Προκαθορισμένα Εργαλεία:** Οι πλατφόρμες τεχνητής νοημοσύνης μπορεί να παρέχουν ένα σύνολο προκαθορισμένων εργαλείων που οι προγραμματιστές μπορούν να αξιοποιήσουν άμεσα, όπως διερμηνείς Python, εργαλεία αναζήτησης στο διαδίκτυο ή κοινές βοηθητικές λειτουργίες.
3. **Απρόσκοπτη Ενσωμάτωση:** Καθώς η χρήση εργαλείων γίνεται πιο διαδεδομένη, μπορούμε να αναμένουμε καλύτερη ενσωμάτωση μεταξύ πλατφορμών τεχνητής νοημοσύνης και δημοφιλών πλαισίων ανάπτυξης, διευκολύνοντας τους προγραμματιστές να ενσωματώσουν τη χρήση εργαλείων στις εφαρμογές τους.

Η χρήση εργαλείων είναι μια ισχυρή τεχνική που επιτρέπει στους προγραμματιστές να αξιοποιήσουν πλήρως τις δυνατότητες των LLMs σε εφαρμογές που βασίζονται στην τεχνητή νοημοσύνη. Συνδέοντας τα LLMs με εξωτερικά εργαλεία και πόρους, μπορείτε να δημιουργήσετε πιο δυναμικά, έξυπνα και συναφή με το περιεχόμενο συστήματα που μπορούν να προσαρμοστούν στις ανάγκες των χρηστών και να παρέχουν πολύτιμες γνώσεις και ενέργειες.

Ενώ η χρήση εργαλείων προσφέρει τεράστιες δυνατότητες, είναι σημαντικό να γνωρίζουμε τις πιθανές προκλήσεις και παραμέτρους. Μια βασική πτυχή είναι η διαχείριση της πολυπλοκότητας των αλληλεπιδράσεων των εργαλείων και η διασφάλιση της σταθερότητας και αξιοπιστίας του συνολικού συστήματος. Πρέπει να χειρίζεστε σενάρια όπου οι κλήσεις εργαλείων μπορεί να αποτύχουν, να επιστρέψουν μη αναμενόμενα αποτελέσματα ή να έχουν επιπτώσεις στην απόδοση.

Επιπλέον, θα πρέπει να εξετάσετε μέτρα ασφαλείας και ελέγχου πρόσβασης για την αποτροπή μη εξουσιοδοτημένης ή κακόβουλης χρήσης εργαλείων. Ο κατάλληλος χειρισμός σφαλμάτων, η καταγραφή και οι μηχανισμοί παρακολούθησης είναι κρίσιμοι για τη διατήρηση της ακεραιότητας και της απόδοσης της εφαρμογής σας που βασίζεται στην τεχνητή νοημοσύνη.

Καθώς εξερευνάτε τις δυνατότητες της χρήσης εργαλείων στα δικά σας έργα, θυμηθείτε να ξεκινάτε με σαφείς στόχους, να σχεδιάζετε καλά δομημένους ορισμούς εργαλείων και να επαναλαμβάνετε τη διαδικασία με βάση την ανατροφοδότηση και τα αποτελέσματα. Με τη σωστή προσέγγιση και νοοτροπία, η χρήση εργαλείων μπορεί να ξεκλειδώσει νέα επίπεδα καινοτομίας και αξίας στις εφαρμογές σας που βασίζονται στην ΤΝ

Επεξεργασία Ροής



Η μετάδοση δεδομένων μέσω HTTP, γνωστή και ως συμβάντα αποστελλόμενα από διακομιστή (SSE), είναι ένας μηχανισμός όπου ο διακομιστής στέλνει συνεχώς δεδομένα στον πελάτη καθώς αυτά γίνονται διαθέσιμα, χωρίς να χρειάζεται ο πελάτης να τα ζητήσει ρητά. Καθώς η απόκριση της TN δημιουργείται σταδιακά, έχει νόημα να παρέχουμε μια άμεση εμπειρία χρήστη εμφανίζοντας την έξοδο της TN καθώς αυτή δημιουργείται. Και στην πραγματικότητα, όλα τα API των παρόχων TN που γνωρίζω προσφέρουν απαντήσεις ροής ως επιλογή στα τελικά σημεία ολοκλήρωσής τους.

Ο λόγος που αυτό το κεφάλαιο εμφανίζεται εδώ στο βιβλίο, αμέσως μετά το [Χρήση Εργαλείων](#), είναι λόγω του πόσο ισχυρός μπορεί να είναι ο συνδυασμός της χρήσης εργαλείων με ζωντανές αποκρίσεις TN στους χρήστες. Αυτό επιτρέπει δυναμικές και διαδραστικές εμπειρίες όπου η TN μπορεί να επεξεργαστεί την είσοδο του χρήστη, να χρησιμοποιήσει διάφορα εργαλεία και λειτουργίες κατά την κρίση της και στη

συνέχεια να παρέχει απαντήσεις σε πραγματικό χρόνο.

Για να επιτευχθεί αυτή η απρόσκοπτη αλληλεπίδραση, χρειάζεται να γράψετε χειριστές ροής που μπορούν να διεκπεραιώσουν κλήσεις λειτουργιών εργαλείων που προκαλούνται από την TN καθώς και απλή έξοδο κειμένου στον τελικό χρήστη. Η ανάγκη για επανάληψη μετά την επεξεργασία μιας λειτουργίας εργαλείου προσθέτει μια ενδιαφέρουσα πρόκληση στην εργασία.

Υλοποίηση ενός ReplyStream

Για να δείξουμε πώς μπορεί να υλοποιηθεί η επεξεργασία ροής, αυτό το κεφάλαιο θα εμβαθύνει σε μια απλοποιημένη έκδοση της κλάσης ReplyStream που χρησιμοποιείται στο Olympia. Τα στιγμιότυπα αυτής της κλάσης μπορούν να περαστούν ως η παράμετρος stream σε βιβλιοθήκες πελάτη TN όπως το [ruby-openai](#) και το [openrouter](#)

Δείτε πώς χρησιμοποιώ το ReplyStream στο PromptSubscriber του Olympia, το οποίο ακούει μέσω του Wisper για τη δημιουργία νέων μηνυμάτων χρήστη.

```
1 class PromptSubscriber
2   include Raix::ChatCompletion
3   include Raix::PromptDeclarations
4
5   # many other declarations omitted...
6
7   prompt text: -> { user_message.content },
8             stream: -> { ReplyStream.new(self) },
9             until: -> { bot_message.complete? }
10
11   def message_created(message) # invoked by Wisper
12     return unless message.role.user? && message.content?
13
14     # rest of the implementation omitted...
```

Εκτός από μια αναφορά context στον συνδρομητή προτροπής που το δημιούργησε, η κλάση ReplyStream διαθέτει επίσης μεταβλητές στιγμιότυπου για την αποθήκευση

μιας ενδιάμεσης μνήμης ληφθέντων δεδομένων, και πίνακες για την παρακολούθηση των ονομάτων συναρτήσεων και των παραμέτρων που καλούνται κατά την επεξεργασία της ροής.

```
1 class ReplyStream
2   attr_accessor :buffer, :f_name, :f_arguments, :context
3
4   delegate :bot_message, :dispatch, to: :context
5
6   def initialize(context)
7     self.context = context
8     self.buffer = []
9     self.f_name = []
10    self.f_arguments = []
11  end
12
13  def call(chunk, bytesize = nil)
14    # ...
15  end
16
17  # ...
18 end
```

Η μέθοδος `initialize` ρυθμίζει την αρχική κατάσταση του στιγμιότυπου `ReplyStream`, αρχικοποιώντας την ενδιάμεση μνήμη, το πλαίσιο και άλλες μεταβλητές.

Η μέθοδος `call` είναι το κύριο σημείο εισόδου για την επεξεργασία των δεδομένων ροής. Δέχεται ένα τμήμα δεδομένων `chunk` (που αναπαρίσταται ως πίνακας κατακερματισμού) και μια προαιρετική παράμετρο `bytesize`, η οποία στο παράδειγμά μας δε χρησιμοποιείται. Μέσα σε αυτή τη μέθοδο, η κλάση χρησιμοποιεί αντιστοίχιση προτύπων για να χειριστεί διαφορετικά σενάρια με βάση τη δομή του τμήματος δεδομένων που λαμβάνεται.



Η κλήση του `deep_symbolize_keys` στο `chunk` βοηθά να γίνει η αντιστοίχιση προτύπων πιο κομψή, επιτρέποντάς μας να λειτουργούμε με σύμβολα αντί για συμβολοσειρές.

```
1 def call(chunk, _bytesize)
2     case chunk.deep_symbolize_keys
3
4     in { # match function name
5         choices: [
6             {
7                 delta: {
8                     tool_calls: [
9                         { index: index, function: {name: name} }
10                    ]
11                }
12            }
13        ] }
14
15     f_name[index] = name
```

Το πρώτο μοτίβο που αναζητούμε είναι μια κλήση εργαλείου μαζί με το αντίστοιχο όνομα συνάρτησης. Εάν εντοπίσουμε ένα τέτοιο, το αποθηκεύουμε στον πίνακα `f_name`. Αποθηκεύουμε τα ονόματα συναρτήσεων σε έναν ευρετηριασμένο πίνακα, επειδή το μοντέλο είναι ικανό για παράλληλη κλήση συναρτήσεων, στέλνοντας περισσότερες από μία συναρτήσεις για εκτέλεση ταυτόχρονα.

Η παράλληλη κλήση συναρτήσεων είναι η ικανότητα ενός μοντέλου ΑΙ να εκτελεί πολλαπλές κλήσεις συναρτήσεων μαζί, επιτρέποντας την παράλληλη επίλυση των αποτελεσμάτων αυτών των κλήσεων. Αυτό είναι ιδιαίτερα χρήσιμο αν οι συναρτήσεις χρειάζονται πολύ χρόνο, και μειώνει τις διαδρομές επικοινωνίας με το API, το οποίο με τη σειρά του μπορεί να εξοικονομήσει σημαντική ποσότητα κατανάλωσης tokens.

Στη συνέχεια πρέπει να αναζητήσουμε τα ορίσματα που αντιστοιχούν στις κλήσεις συναρτήσεων.

```

1   in { # match arguments
2     choices: [
3       {
4         delta: {
5           tool_calls: [
6             {
7               index: index, function: {arguments: argument }
8             }
9           ]
10        }
11      }
12    ]]
13
14    f_arguments[index] ||= "" # initialize if not already
15    f_arguments[index] << argument

```

Παρόμοια με τον τρόπο που χειριστήκαμε τα ονόματα συναρτήσεων, τοποθετούμε τα ορίσματα σε έναν πίνακα με δείκτες.

Στη συνέχεια, αναζητούμε κανονικά μηνύματα που απευθύνονται στον χρήστη, τα οποία θα έρχονται από τον διακομιστή μία λεκτική μονάδα τη φορά και θα ανατίθενται στη μεταβλητή `new_content`. Επίσης, πρέπει να παρακολουθούμε το `finish_reason`. Θα είναι `nil` μέχρι το τελευταίο τμήμα της ακολουθίας εξόδου.

```

1   in {
2     choices: [
3       { delta: {content: new_content}, finish_reason: finish_reason }
4     ]}
5
6     # you could transmit every chunk to the user here...
7     buffer << new_content.to_s
8
9     if finish_reason.present?
10      finalize
11    elsif new_content.to_s.match?(/\n\n/)
12      send_to_client # ...or buffer and transmit once per paragraph
13    end

```

Σημαντικά, προσθέτουμε μια έκφραση αντιστοίχισης προτύπων για να χειριστούμε τα μηνύματα σφάλματος που αποστέλλονται από τον πάροχο του μοντέλου AI. Στα

τοπικά περιβάλλοντα ανάπτυξης, δημιουργούμε μια εξαίρεση, αλλά στην παραγωγή, καταγράφουμε το σφάλμα και ολοκληρώνουμε.

```
1  in { error: { message: } }  
2    if Rails.env.local?  
3      raise message  
4    else  
5      Honeybadger.notify("AI Error: #{message}")  
6      finalize  
7    end
```

Η τελική ρήτρα else της εντολής case θα εκτελεστεί εάν κανένα από τα προηγούμενα μοτίβα δεν ταιριάζει. Είναι απλώς μια δικλείδα ασφαλείας ώστε αν το μοντέλο τεχνητής νοημοσύνης αρχίσει να μας στέλνει μη αναγνωρίσιμα τμήματα, να το ανακαλύψουμε.

```
1  else  
2    Honeybadger.notify("Unrecognized Chunk: #{chunk}")  
3  end  
4  end
```

Η μέθοδος `send_to_client` είναι υπεύθυνη για την αποστολή του περιεχομένου που είναι αποθηκευμένο στην προσωρινή μνήμη στον πελάτη. Ελέγχει ότι η προσωρινή μνήμη δεν είναι κενή, ενημερώνει το περιεχόμενο του μηνύματος του bot, αποδίδει το μήνυμα του bot και αποθηκεύει το περιεχόμενο στη βάση δεδομένων για να εξασφαλίσει τη διατήρηση των δεδομένων.

```
1 def send_to_client
2   # no need to process pure whitespace
3   return if buffer.join.squish.blank?
4
5   # set the buffer content on the bot message
6   content = buffer.join
7   bot_message.content = content
8
9   # save to database so that we never lose data
10  # even if the stream doesn't terminate correctly
11  bot_message.update_column(:content, content)
12
13  # update content via websocket
14  ConversationRenderer.update(bot_message)
15 end
```

Η μέθοδος `finalize` καλείται όταν ολοκληρωθεί η επεξεργασία ροής. Εκτελεί τις κλήσεις συναρτήσεων εάν υπήρξαν κάποιες κατά τη διάρκεια της ροής, ενημερώνει το μήνυμα του bot με το τελικό περιεχόμενο και άλλες σχετικές πληροφορίες, και επαναφέρει το ιστορικό κλήσεων συναρτήσεων

```
1 def finalize
2   if f_name.any?
3     f_name.each_with_index do |name, index|
4       # takes care of calling the function wherever it's implemented
5       dispatch(name:, arguments: JSON.parse(f_arguments[index]))
6     end
7
8     # reset the function call history
9     f_name.clear
10    f_arguments.clear
11  else
12    content = buffer.join.presence
13    bot_message.update!(content:, complete: true)
14    ConversationRenderer.update(bot_message)
15  end
16 end
```

Εάν το μοντέλο αποφασίσει να καλέσει μια συνάρτηση, πρέπει να “διεκπεραιώσετε” αυτή την κλήση συνάρτησης (όνομα και ορίσματα) με τέτοιο τρόπο ώστε να

εκτελεστεί και τα μηνύματα `function_call` και `function_result` να προστεθούν στην καταγραφή συνομιλίας

Από την εμπειρία μου, είναι καλύτερο να χειρίζεστε τη δημιουργία μηνυμάτων συνάρτησης σε ένα σημείο στη βάση κώδικά σας, αντί να βασίζεστε στις υλοποιήσεις των εργαλείων. Είναι πιο καθαρό, αλλά υπάρχει επίσης ένας πολύ σημαντικός πρακτικός λόγος: εάν το μοντέλο AI καλέσει μια συνάρτηση και δεν δει τα προκύπτοντα μηνύματα κλήσης και αποτελέσματος στην καταγραφή όταν επαναλαμβάνετε, *θα καλέσει την ίδια συνάρτηση ξανά*. Πιθανώς επ' άπειρον. Θυμηθείτε ότι η AI είναι εντελώς χωρίς κατάσταση, οπότε εκτός αν αντανακλάτε αυτές τις κλήσεις συναρτήσεων πίσω σε αυτήν, είναι σαν να μην συνέβησαν ποτέ.

```
1  # PromptSubscriber#dispatch
2
3  def dispatch(name:, arguments:):
4      # adds a function_call message to the conversation transcript
5      # plus dispatches to tool and returns result
6      conversation.function_call!(name, arguments).then do |result|
7          # add function result message to the transcript
8          conversation.function_result!(name, result)
9      end
10 end
```



Η εκκαθάριση του ιστορικού κλήσεων συναρτήσεων μετά την αποστολή είναι εξίσου σημαντική με τη διασφάλιση ότι η κλήση και τα αποτελέσματα καταλήγουν στην καταγραφή σας, ώστε να μην συνεχίζετε να καλείτε τις ίδιες συναρτήσεις ξανά και ξανά κάθε φορά που εκτελείτε τον βρόχο.

Ο “Βρόχος Συνομιλίας”

Στην κλάση `PromptSubscriber`, χρησιμοποιούμε τη μέθοδο `prompt` από τη μονάδα `PromptDeclarations` για να ορίσουμε τη συμπεριφορά του βρόχου συνομιλίας. Η

παράμετρος `until` έχει οριστεί ως `-> { bot_message.complete? }`, που σημαίνει ότι ο βρόχος θα συνεχίσει μέχρι το `bot_message` να επισημανθεί ως ολοκληρωμένο.

```
1 prompt text: -> { user_message.content },  
2   stream: -> { ReplyStream.new(self) },  
3   until: -> { bot_message.complete? }
```



Αλλά πότε σημειώνεται το `bot_message` ως ολοκληρωμένο; Αν το έχετε ξεχάσει, ανατρέξτε πίσω στη γραμμή 13 της μεθόδου `finalize`.

Ας εξετάσουμε τη συνολική λογική επεξεργασίας ροής.

1. Ο `PromptSubscriber` λαμβάνει ένα νέο μήνυμα χρήστη μέσω της μεθόδου `message_created`, η οποία καλείται από το σύστημα δημοσίευσης/συνδρομής `Wisper` κάθε φορά που ο τελικός χρήστης δημιουργεί ένα νέο `prompt`.
2. Η μέθοδος κλάσης `prompt` ορίζει δηλωτικά τη συμπεριφορά της λογικής ολοκλήρωσης συνομιλίας για τον `PromptSubscriber`. Το μοντέλο AI θα εκτελέσει μια ολοκλήρωση συνομιλίας με το περιεχόμενο του μηνύματος του χρήστη, μια νέα παρουσία του `ReplyStream` ως παράμετρο ροής και την καθορισμένη συνθήκη βρόχου.
3. Το μοντέλο AI επεξεργάζεται το `prompt` και αρχίζει να δημιουργεί μια απάντηση. Καθώς η απάντηση μεταδίδεται σε ροή, η μέθοδος `call` της παρουσίας `ReplyStream` καλείται για κάθε τμήμα δεδομένων.
4. Εάν το μοντέλο AI αποφασίσει να καλέσει μια βοηθητική συνάρτηση, το όνομα της συνάρτησης και τα ορίσματα εξάγονται από το τμήμα και αποθηκεύονται στους πίνακες `f_name` και `f_arguments` αντίστοιχα.
5. Εάν το μοντέλο AI δημιουργήσει περιεχόμενο που απευθύνεται στον χρήστη, αυτό αποθηκεύεται προσωρινά και αποστέλλεται στον πελάτη μέσω της μεθόδου `send_to_client`.

6. Μόλις ολοκληρωθεί η επεξεργασία ροής, καλείται η μέθοδος `finalize`. Εάν κλήθηκαν βοηθητικές συναρτήσεις κατά τη διάρκεια της ροής, αποστέλλονται χρησιμοποιώντας τη μέθοδο `dispatch` του `PromptSubscriber`.
7. Η μέθοδος `dispatch` προσθέτει ένα μήνυμα `function_call` στην καταγραφή συνομιλίας, εκτελεί την αντίστοιχη βοηθητική συνάρτηση και προσθέτει ένα μήνυμα `function_result` στην καταγραφή με το αποτέλεσμα της κλήσης συνάρτησης.
8. Μετά την αποστολή των βοηθητικών συναρτήσεων, το ιστορικό κλήσεων συναρτήσεων καθαρίζεται για να αποτραπούν διπλές κλήσεις συναρτήσεων σε επόμενους βρόχους.
9. Εάν δεν κλήθηκαν βοηθητικές συναρτήσεις, η μέθοδος `finalize` ενημερώνει το `bot_message` με το τελικό περιεχόμενο, το σημειώνει ως ολοκληρωμένο και στέλνει το ενημερωμένο μήνυμα στον πελάτη.
10. Η συνθήκη βρόχου `-> { bot_message.complete? }` αξιολογείται. Εάν το `bot_message` δεν έχει σημειωθεί ως ολοκληρωμένο, ο βρόχος συνεχίζεται και το αρχικό `prompt` υποβάλλεται ξανά με την ενημερωμένη καταγραφή συνομιλίας.
11. Τα βήματα 3-10 επαναλαμβάνονται μέχρι το `bot_message` να επισημανθεί ως ολοκληρωμένο, υποδεικνύοντας ότι το μοντέλο AI έχει ολοκληρώσει τη δημιουργία της απάντησής του και δεν χρειάζεται να εκτελεστούν περαιτέρω λειτουργίες εργαλείων.

Εφαρμόζοντας αυτόν τον βρόχο συνομιλίας, επιτρέπετε στο μοντέλο AI να συμμετέχει σε μια αμφίδρομη αλληλεπίδραση με την εφαρμογή, εκτελώντας λειτουργίες εργαλείων όπως απαιτείται και δημιουργώντας απαντήσεις προς τον χρήστη μέχρι η συνομιλία να φτάσει σε ένα φυσικό συμπέρασμα.

Ο συνδυασμός της επεξεργασίας ροής και του βρόχου συνομιλίας επιτρέπει δυναμικές και διαδραστικές εμπειρίες που τροφοδοτούνται από AI, όπου το μοντέλο AI μπορεί να επεξεργαστεί την είσοδο του χρήστη, να χρησιμοποιήσει διάφορα εργαλεία και λειτουργίες, και να παρέχει απαντήσεις σε πραγματικό χρόνο με βάση το εξελισσόμενο πλαίσιο της συνομιλίας.

Αυτόματη Συνέχιση

Είναι σημαντικό να γνωρίζουμε τους περιορισμούς εξόδου της AI. Τα περισσότερα μοντέλα έχουν έναν μέγιστο αριθμό tokens που μπορούν να δημιουργήσουν σε μία μόνο απάντηση, ο οποίος καθορίζεται από την παράμετρο `max_tokens`. Εάν το μοντέλο AI φτάσει αυτό το όριο κατά τη δημιουργία μιας απάντησης, θα σταματήσει απότομα και θα υποδείξει ότι η έξοδος περικόπηκε.

Στην απάντηση ροής από το API της πλατφόρμας AI, μπορείτε να εντοπίσετε αυτή την κατάσταση εξετάζοντας τη μεταβλητή `finish_reason` στο τμήμα. Εάν το `finish_reason` έχει οριστεί σε `"length"` (ή κάποια άλλη τιμή κλειδί ειδική για το μοντέλο), σημαίνει ότι το μοντέλο έφτασε στο μέγιστο όριο token κατά τη διάρκεια της δημιουργίας και η έξοδος έχει διακοπεί απότομα.

Ένας τρόπος για να χειριστείτε αυτό το σενάριο με κομψό τρόπο και να παρέχετε μια απρόσκοπτη εμπειρία χρήστη, είναι να εφαρμόσετε έναν μηχανισμό αυτόματης συνέχισης στη λογική επεξεργασίας ροής σας. Προσθέτοντας ένα μοτίβο αντιστοίχισης για αιτίες ολοκλήρωσης που σχετίζονται με το μήκος, μπορείτε να επιλέξετε να επαναλάβετε και να συνεχίσετε αυτόματα την έξοδο από το σημείο που σταμάτησε.

Εδώ είναι ένα σκόπιμα απλοποιημένο παράδειγμα για το πώς μπορείτε να τροποποιήσετε τη μέθοδο `call` στην κλάση `ReplyStream` για να υποστηρίξετε την αυτόματη συνέχιση:

```
1  LENGTH_STOPS = %w[length MAX_TOKENS]
2
3  def call(chunk, _bytesize)
4    case chunk.deep_symbolize_keys
5      # ...
6
7      in {
8        choices: [
9          { delta: {content: new_content},
10            finish_reason: finish_reason } ] }
11
12      buffer << new_content.to_s
13
14      if finish_reason.blank?
15        send_to_client if new_content.to_s.match?(/\n\n/)
16      elsif LENGTH_STOPS.include?(finish_reason)
17        continue_cutoff
18      else
19        finalize
20      end
21
22      # ...
23    end
24  end
25
26  private
27
28  def continue_cutoff
29    conversation.bot_message!(buffer.join, visible: false)
30    conversation.user_message!("please continue", visible: false)
31    bot_message.update_column(:created_at, Time.current)
32  end
```

Σε αυτή την τροποποιημένη έκδοση, όταν το `finish_reason` υποδεικνύει περικομμένη έξοδο, αντί να οριστικοποιήσουμε τη ροή, προσθέτουμε ένα ζεύγος μηνυμάτων στο αντίγραφο συνομιλίας χωρίς οριστικοποίηση, μετακινούμε το αρχικό μήνυμα απόκρισης προς τον χρήστη στο “κάτω μέρος” του αντιγράφου συνομιλίας ενημερώνοντας το χαρακτηριστικό `created_at`, και στη συνέχεια αφήνουμε τον βρόχο να συνεχίσει, ώστε η TN να συνεχίσει την παραγωγή από εκεί που

σταμάτησε.

Να θυμάστε ότι το τελικό σημείο συμπλήρωσης της TN είναι stateless. “Γνωρίζει” μόνο ό,τι του λέτε μέσω του αντιγράφου συνομιλίας. Σε αυτή την περίπτωση, ο τρόπος που επικοινωνούμε στην TN ότι διακόπηκε είναι προσθέτοντας “αόρατα” (για τον τελικό χρήστη) μηνύματα στο αντίγραφο συνομιλίας. Θυμηθείτε όμως ότι αυτό είναι ένα σκόπιμα απλοποιημένο παράδειγμα. Μια πραγματική υλοποίηση θα χρειαζόταν να κάνει περαιτέρω διαχείριση του αντιγράφου συνομιλίας για να διασφαλίσει ότι δεν σπαταλάμε tokens και/ή μπερδεύουμε την TN με διπλότυπα μηνύματα βοηθού στο αντίγραφο συνομιλίας.

Μια πραγματική υλοποίηση της αυτόματης συνέχισης θα πρέπει επίσης να έχει την λεγόμενη “λογική διακόπτη κυκλώματος” για την αποτροπή ανεξέλεγκτων βρόχων. Ο λόγος είναι ότι, δεδομένων ορισμένων τύπων προτροπών χρήστη και χαμηλών ρυθμίσεων max_tokens, η TN θα μπορούσε να συνεχίσει να επαναλαμβάνει την έξοδο προς τον χρήστη ατέρμονα.

Έχετε υπόψη ότι κάθε βρόχος απαιτεί ένα ξεχωριστό αίτημα, και ότι κάθε αίτημα καταναλώνει ξανά ολόκληρο το αντίγραφο συνομιλίας σας. Θα πρέπει σίγουρα να εξετάσετε τις αντισταθμίσεις μεταξύ εμπειρίας χρήστη και χρήσης του API όταν αποφασίζετε αν θα υλοποιήσετε την αυτόματη συνέχιση στην εφαρμογή σας. Η αυτόματη συνέχιση συγκεκριμένα μπορεί να είναι επικίνδυνα ακριβή, ειδικά όταν χρησιμοποιείτε εμπορικά μοντέλα premium.

Συμπέρασμα

Η επεξεργασία ροής είναι μια κρίσιμη πτυχή της δημιουργίας εφαρμογών με τεχνητή νοημοσύνη που συνδυάζουν τη χρήση εργαλείων με ζωντανές αποκρίσεις TN. Με την αποτελεσματική διαχείριση των δεδομένων ροής από τα API πλατφορμών

TN, μπορείτε να παρέχετε μια απρόσκοπτη και διαδραστική εμπειρία χρήστη, να χειρίζεστε μεγάλες αποκρίσεις, να βελτιστοποιείτε τη χρήση πόρων και να διαχειρίζεστε ομαλά τα σφάλματα.

Η παρεχόμενη κλάση `Conversation::ReplyStream` δείχνει πώς μπορεί να υλοποιηθεί η επεξεργασία ροής σε μια εφαρμογή Ruby χρησιμοποιώντας αντιστοίχιση προτύπων και αρχιτεκτονική βασισμένη σε συμβάντα. Κατανοώντας και αξιοποιώντας τις τεχνικές επεξεργασίας ροής, μπορείτε να ξεκλειδώσετε το πλήρες δυναμικό της ενσωμάτωσης TN στις εφαρμογές σας και να προσφέρετε ισχυρές και ελκυστικές εμπειρίες χρήστη.

Αυτοθεραπευόμενα Δεδομένα



Τα αυτοθεραπευόμενα δεδομένα αποτελούν μια ισχυρή προσέγγιση για τη διασφάλιση της ακεραιότητας, της συνέπειας και της ποιότητας των δεδομένων στις εφαρμογές, αξιοποιώντας τις δυνατότητες των μεγάλων μοντέλων γλώσσας (LLMs). Αυτή η κατηγορία μοτίβων επικεντρώνεται στην ιδέα της χρήσης τεχνητής νοημοσύνης για την αυτόματη ανίχνευση, διάγνωση και διόρθωση ανωμαλιών, ασυνεπειών ή σφαλμάτων στα δεδομένα, μειώνοντας έτσι το φόρτο των προγραμματιστών και διατηρώντας υψηλό επίπεδο αξιοπιστίας των δεδομένων.

Στον πυρήνα τους, τα μοτίβα αυτοθεραπευόμενων δεδομένων αναγνωρίζουν ότι τα δεδομένα είναι η ζωτική δύναμη κάθε εφαρμογής, και η διασφάλιση της ακρίβειας και της ακεραιότητάς τους είναι κρίσιμη για την ορθή λειτουργία και την εμπειρία χρήστη της εφαρμογής. Ωστόσο, η διαχείριση και η διατήρηση της ποιότητας των δεδομένων μπορεί να είναι μια περίπλοκη και χρονοβόρα διαδικασία, ειδικά καθώς οι εφαρμογές αυξάνονται σε μέγεθος και πολυπλοκότητα. Εδώ είναι που η δύναμη

της τεχνητής νοημοσύνης έρχεται σε παιχνίδι.

Στα μοτίβα αυτοθεραπευόμενων δεδομένων, οι εργάτες τεχνητής νοημοσύνης χρησιμοποιούνται για τη συνεχή παρακολούθηση και ανάλυση των δεδομένων της εφαρμογής σας. Αυτά τα μοντέλα έχουν την ικανότητα να κατανοούν και να ερμηνεύουν μοτίβα, σχέσεις και ανωμαλίες μέσα στα δεδομένα. Αξιοποιώντας τις δυνατότητές τους στην επεξεργασία και κατανόηση φυσικής γλώσσας, μπορούν να εντοπίζουν πιθανά προβλήματα ή ασυνέπειες στα δεδομένα και να προβαίνουν στις κατάλληλες ενέργειες για τη διόρθωσή τους.

Η διαδικασία των αυτοθεραπευόμενων δεδομένων συνήθως περιλαμβάνει αρκετά βασικά βήματα:

1. **Παρακολούθηση Δεδομένων:** Οι εργάτες τεχνητής νοημοσύνης παρακολουθούν συνεχώς τις ροές δεδομένων, τις βάσεις δεδομένων ή τα συστήματα αποθήκευσης της εφαρμογής, αναζητώντας οποιαδήποτε σημάδια ανωμαλιών, ασυνεπειών ή σφαλμάτων. Εναλλακτικά, μπορείτε να ενεργοποιήσετε ένα στοιχείο τεχνητής νοημοσύνης ως αντίδραση σε μια εξαίρεση.
2. **Ανίχνευση Ανωμαλιών:** Όταν εντοπιστεί ένα πρόβλημα, ο εργάτης τεχνητής νοημοσύνης αναλύει τα δεδομένα λεπτομερώς για να προσδιορίσει την ακριβή φύση και το εύρος του προβλήματος. Αυτό θα μπορούσε να περιλαμβάνει τον εντοπισμό ελλিপών τιμών, ασυνεπών μορφών ή δεδομένων που παραβιάζουν προκαθορισμένους κανόνες ή περιορισμούς.
3. **Διάγνωση και Διόρθωση:** Μόλις εντοπιστεί το πρόβλημα, ο εργάτης τεχνητής νοημοσύνης χρησιμοποιεί τις γνώσεις και την κατανόησή του για τον τομέα των δεδομένων για να καθορίσει την κατάλληλη πορεία δράσης. Αυτό θα μπορούσε να περιλαμβάνει την αυτόματη διόρθωση των δεδομένων, τη συμπλήρωση ελλিপών τιμών ή την επισήμανση του προβλήματος για ανθρώπινη παρέμβαση εάν είναι απαραίτητο.
4. **Συνεχής Μάθηση (προαιρετικά, ανάλογα με την περίπτωση χρήσης):**

Καθώς ο εργάτης τεχνητής νοημοσύνης συναντά και επιλύει διάφορα προβλήματα δεδομένων, μπορεί να παράγει περιγραφές για το τι συνέβη και πώς ανταποκρίθηκε. Αυτά τα μεταδεδομένα μπορούν να τροφοδοτηθούν σε διαδικασίες μάθησης που επιτρέπουν σε εσάς (και ίσως στο υποκείμενο μοντέλο, μέσω λεπτομερούς ρύθμισης) να γίνετε πιο αποτελεσματικοί και αποδοτικοί με την πάροδο του χρόνου στον εντοπισμό και την επίλυση ανωμαλιών δεδομένων.

Με την αυτόματη ανίχνευση και διόρθωση προβλημάτων δεδομένων, μπορείτε να διασφαλίσετε ότι η εφαρμογή σας λειτουργεί με υψηλής ποιότητας, αξιόπιστα δεδομένα. Αυτό μειώνει τον κίνδυνο σφαλμάτων, ασυνεπειών ή σφαλμάτων σχετικών με τα δεδομένα που επηρεάζουν τη λειτουργικότητα ή την εμπειρία χρήστη της εφαρμογής.

Μόλις αναθέσετε στους εργάτες Τεχνητής Νοημοσύνης το έργο της παρακολούθησης και διόρθωσης δεδομένων, μπορείτε να επικεντρώσετε τις προσπάθειές σας σε άλλες κρίσιμες πτυχές της εφαρμογής. Αυτό εξοικονομεί χρόνο και πόρους που διαφορετικά θα δαπανούνταν στον χειροκίνητο καθαρισμό και τη συντήρηση δεδομένων. Στην πραγματικότητα, καθώς οι εφαρμογές σας αυξάνονται σε μέγεθος και πολυπλοκότητα, η χειροκίνητη διαχείριση της ποιότητας των δεδομένων γίνεται όλο και πιο απαιτητική. Τα μοτίβα “Αυτοθεραπευόμενων Δεδομένων” κλιμακώνονται αποτελεσματικά αξιοποιώντας τη δύναμη της TN για να χειριστούν μεγάλους όγκους δεδομένων και να εντοπίσουν προβλήματα σε πραγματικό χρόνο.



Λόγω της φύσης τους, τα μοντέλα TN μπορούν να προσαρμοστούν σε μεταβαλλόμενα μοτίβα δεδομένων, σχήματα ή απαιτήσεις με την πάροδο του χρόνου με ελάχιστη ή καθόλου επίβλεψη. Εφόσον οι οδηγίες τους παρέχουν επαρκή καθοδήγηση, ιδιαίτερα όσον αφορά τα επιδιωκόμενα αποτελέσματα, η εφαρμογή σας μπορεί να εξελιχθεί και να χειριστεί νέα σενάρια δεδομένων χωρίς να απαιτείται εκτεταμένη χειροκίνητη παρέμβαση ή αλλαγές κώδικα.

Τα μοτίβα αυτοθεραπευόμενων δεδομένων ευθυγραμμίζονται καλά με τις άλλες κατηγορίες μοτίβων που έχουμε συζητήσει, όπως το “Πλήθος Εργατών”. Η δυνατότητα αυτοθεραπείας δεδομένων μπορεί να θεωρηθεί ως ένα εξειδικευμένο είδος εργάτη που επικεντρώνεται συγκεκριμένα στη διασφάλιση της ποιότητας και ακεραιότητας των δεδομένων. Αυτό το είδος εργάτη λειτουργεί παράλληλα με άλλους εργάτες TN, καθένας από τους οποίους συμβάλλει σε διαφορετικές πτυχές της λειτουργικότητας της εφαρμογής.

Η εφαρμογή μοτίβων αυτοθεραπευόμενων δεδομένων στην πράξη απαιτεί προσεκτικό σχεδιασμό και ενσωμάτωση μοντέλων TN στην αρχιτεκτονική της εφαρμογής. Λόγω των κινδύνων απώλειας και αλλοίωσης δεδομένων, θα πρέπει να καθορίσετε σαφείς κατευθυντήριες γραμμές για τον τρόπο χρήσης αυτής της τεχνικής. Θα πρέπει επίσης να εξετάσετε παράγοντες όπως η απόδοση, η κλιμάκωση και η ασφάλεια των δεδομένων.

Πρακτική Μελέτη Περίπτωσης: Διόρθωση Κατεστραμμένου JSON

Ένας από τους πιο πρακτικούς και βολικούς τρόπους αξιοποίησης των αυτοθεραπευόμενων δεδομένων είναι επίσης πολύ απλός στην εξήγηση: η διόρθωση κατεστραμμένου JSON.

Αυτή η τεχνική μπορεί να εφαρμοστεί στην κοινή πρόκληση της διαχείρισης ατελών ή ασυνεπών δεδομένων που παράγονται από LLMs, όπως κατεστραμμένο JSON, και παρέχει μια προσέγγιση για την αυτόματη ανίχνευση και διόρθωση αυτών των προβλημάτων.

Στην Olympia συναντώ τακτικά σενάρια όπου τα LLMs παράγουν δεδομένα JSON που δεν είναι απολύτως έγκυρα. Αυτό μπορεί να συμβεί για διάφορους λόγους, όπως όταν το LLM προσθέτει σχόλια πριν ή μετά τον πραγματικό κώδικα JSON, ή εισάγει συντακτικά λάθη όπως ελλείποντα κόμματα ή μη διαφυγόντα διπλά

εισαγωγικά. Αυτά τα προβλήματα μπορούν να οδηγήσουν σε σφάλματα ανάλυσης και να προκαλέσουν διαταραχές στη λειτουργικότητα της εφαρμογής.

Για την αντιμετώπιση αυτού του προβλήματος, έχω υλοποιήσει μια πρακτική λύση με τη μορφή μιας κλάσης `JsonFixer`. Αυτή η κλάση ενσωματώνει το μοτίβο “Αυτοθεραπευόμενων Δεδομένων” λαμβάνοντας ως είσοδο το κατεστραμμένο JSON και αξιοποιώντας ένα ΜΓΓ για να το διορθώσει, διατηρώντας παράλληλα όσο το δυνατόν περισσότερες πληροφορίες και την αρχική πρόθεση.

```

1  class JsonFixer
2    include Raix::ChatCompletion
3
4    def call(bad_json, error_message)
5      raise "No data provided" if bad_json.blank? || error_message.blank?
6
7      transcript << {
8        system: "Consider user-provided JSON that generated a parse
9                exception. Do your best to fix it while preserving the
10               original content and intent as much as possible." }
11      transcript << { user: bad_json }
12      transcript << { assistant: "What is the error message?" }
13      transcript << { user: error_message }
14      transcript << { assistant: "Here is the corrected JSON\n```\njson\n" }
15
16      self.stop = ["```"]
17
18      chat_completion(json: true)
19    end
20
21    def model
22      "mistralai/mixtral-8x7b-instruct:nitro"
23    end
24  end

```



Παρατηρήστε πώς το `JsonFixer` χρησιμοποιεί το [Ventriloquist](#) για να καθοδηγήσει τις αποκρίσεις της τεχνητής νοημοσύνης.

Η διαδικασία αυτοεπιδιόρθωσης δεδομένων JSON λειτουργεί ως εξής:

1. **Δημιουργία JSON:** Ένα LLM χρησιμοποιείται για τη δημιουργία δεδομένων JSON με βάση συγκεκριμένες οδηγίες ή απαιτήσεις. Ωστόσο, λόγω της φύσης των LLMs, το JSON που παράγεται ενδέχεται να μην είναι πάντα απόλυτα έγκυρο. Ο αναλυτής JSON φυσικά θα εγείρει ένα `ParserError` εάν του δώσετε μη έγκυρο JSON.

```
1 begin
2   JSON.parse(llm_generated_json)
3 rescue JSON::ParserError => e
4   JsonFixer.new.call(llm_generated_json, e.message)
5 end
```

Σημειώστε ότι το μήνυμα σφάλματος περνάει επίσης στην κλήση `JsonFixer` έτσι ώστε να μην χρειάζεται να υποθέσει πλήρως τι είναι λάθος με τα δεδομένα, ειδικά αφού ο αναλυτής συχνά θα σας πει ακριβώς τι είναι λάθος.

2. **Διόρθωση βασισμένη σε MMF:** Η κλάση `JsonFixer` στέλνει το κατεστραμμένο JSON πίσω σε ένα MMF, μαζί με μια συγκεκριμένη προτροπή ή οδηγία για να διορθώσει το JSON διατηρώντας όσο το δυνατόν περισσότερο τις αρχικές πληροφορίες και την πρόθεση. Το MMF, εκπαιδευμένο σε τεράστιες ποσότητες δεδομένων και με κατανόηση της σύνταξης JSON, επιχειρεί να διορθώσει τα σφάλματα και να δημιουργήσει μια έγκυρη συμβολοσειρά JSON. Η [Οριοθέτηση Απόκρισης](#) χρησιμοποιείται για να περιορίσει την έξοδο του MMF, και επιλέγουμε το Mixtral 8x7B ως το μοντέλο AI, καθώς είναι ιδιαίτερα καλό για αυτού του είδους την εργασία.
3. **Επικύρωση και Ενσωμάτωση:** Η διορθωμένη συμβολοσειρά JSON που επιστρέφεται από το MMF αναλύεται από την ίδια την κλάση `JsonFixer`, επειδή κάλεσε το `chat_completion(json: true)`. Εάν το διορθωμένο JSON περάσει την επικύρωση, ενσωματώνεται πίσω στη ροή εργασίας της εφαρμογής, επιτρέποντας στην εφαρμογή να συνεχίσει την επεξεργασία των δεδομένων απρόσκοπτα. Το κακό JSON έχει “θεραπευτεί”.

Παρόλο που έχω γράψει και ξαναγράψει τη δική μου υλοποίηση JSONFixer αρκετές φορές, αμφιβάλλω αν ο συνολικός χρόνος που επενδύθηκε σε όλες αυτές τις εκδόσεις είναι περισσότερο από μία ή δύο ώρες.

Σημειώστε ότι η διατήρηση της πρόθεσης είναι ένα βασικό στοιχείο οποιουδήποτε μοτίβου αυτοθεραπευόμενων δεδομένων. Η διαδικασία διόρθωσης που βασίζεται σε MMF στοχεύει στη διατήρηση των αρχικών πληροφοριών και της πρόθεσης του παραγόμενου JSON όσο το δυνατόν περισσότερο. Αυτό διασφαλίζει ότι το διορθωμένο JSON διατηρεί τη σημασιολογική του έννοια και μπορεί να χρησιμοποιηθεί αποτελεσματικά στο πλαίσιο της εφαρμογής.

Αυτή η πρακτική εφαρμογή της προσέγγισης “Αυτοθεραπευόμενα Δεδομένα” στην Olympia δείχνει ξεκάθαρα πώς η AI, συγκεκριμένα τα MMF, μπορούν να αξιοποιηθούν για την επίλυση πραγματικών προκλήσεων δεδομένων. Αναδεικνύει τη δύναμη του συνδυασμού παραδοσιακών τεχνικών προγραμματισμού με δυνατότητες AI για τη δημιουργία εύρωστων και αποτελεσματικών εφαρμογών.

Ο Νόμος του Postel και το Μοτίβο “Αυτοθεραπευόμενων Δεδομένων”

Τα “Αυτοθεραπευόμενα Δεδομένα”, όπως παραδειγματίζονται από την κλάση JSONFixer, ευθυγραμμίζονται καλά με την αρχή γνωστή ως Νόμος του Postel, που αναφέρεται επίσης ως η Αρχή Ευρωστίας. Ο Νόμος του Postel αναφέρει:

“Να είστε συντηρητικοί σε αυτό που κάνετε, να είστε ελεύθεροι σε αυτό που δέχεστε από άλλους.”

Αυτή η αρχή, που αρχικά διατυπώθηκε από τον Jon Postel, πρωτοπόρο του πρώιμου Διαδικτύου, τονίζει τη σημασία της δημιουργίας συστημάτων που είναι ανεκτικά σε διαφορετικές ή ακόμη και ελαφρώς λανθασμένες εισόδους, διατηρώντας παράλληλα αυστηρή προσήλωση στα καθορισμένα πρωτόκολλα

κατά την αποστολή εξόδων.

Στο πλαίσιο των “Αυτοθεραπευόμενων Δεδομένων”, η κλάση JSONFixer ενσωματώνει το Νόμο του Postel με το να είναι ελαστική στην αποδοχή κατεστραμμένων ή ατελών δεδομένων JSON που παράγονται από τα Μεγάλα Γλωσσικά Μοντέλα. Δεν απορρίπτει άμεσα ούτε αποτυγχάνει όταν συναντά JSON που δεν συμμορφώνεται αυστηρά με την αναμενόμενη μορφή. Αντίθετα, υιοθετεί μια ανεκτική προσέγγιση και επιχειρεί να διορθώσει το JSON χρησιμοποιώντας τη δύναμη των Μεγάλων Γλωσσικών Μοντέλων.

Με το να είναι ελαστική στην αποδοχή ατελούς JSON, η κλάση JSONFixer επιδεικνύει στιβαρότητα και ευελιξία. Αναγνωρίζει ότι τα δεδομένα στον πραγματικό κόσμο συχνά έρχονται σε διάφορες μορφές και μπορεί να μην συμμορφώνονται πάντα με αυστηρές προδιαγραφές. Χειριζόμενη και διορθώνοντας με χάρη αυτές τις αποκλίσεις, η κλάση διασφαλίζει ότι η εφαρμογή μπορεί να συνεχίσει να λειτουργεί ομαλά, ακόμη και παρουσία ατελών δεδομένων.

Από την άλλη πλευρά, η κλάση JSONFixer τηρεί επίσης τη συντηρητική πτυχή του Νόμου του Postel όσον αφορά την έξοδο. Μετά τη διόρθωση του JSON χρησιμοποιώντας τα Μεγάλα Γλωσσικά Μοντέλα, η κλάση επικυρώνει το διορθωμένο JSON για να διασφαλίσει ότι συμμορφώνεται αυστηρά με την αναμενόμενη μορφή. Διατηρεί την ακεραιότητα και την ορθότητα των δεδομένων πριν τα προωθήσει σε άλλα μέρη της εφαρμογής. Αυτή η συντηρητική προσέγγιση εγγυάται ότι η έξοδος της κλάσης JSONFixer είναι αξιόπιστη και συνεπής, προωθώντας τη διαλειτουργικότητα και αποτρέποντας τη διάδοση σφαλμάτων.

Ενδιαφέρουσες Πληροφορίες για τον Jon Postel:

- Ο Jon Postel (1943-1998) ήταν Αμερικανός επιστήμονας υπολογιστών που έπαιξε καθοριστικό ρόλο στην ανάπτυξη του Διαδικτύου. Ήταν γνωστός ως ο “Θεός του Διαδικτύου” για τις σημαντικές συνεισφορές του στα

υποκείμενα πρωτόκολλα και πρότυπα.

- Ο Postel ήταν ο εκδότης της σειράς εγγράφων Request for Comments (RFC), η οποία είναι μια σειρά τεχνικών και οργανωτικών σημειώσεων για το Διαδίκτυο. Συνέγραψε ή συν-συνέγραψε πάνω από 200 RFC, συμπεριλαμβανομένων των θεμελιωδών πρωτοκόλλων όπως TCP, IP και SMTP.
- Εκτός από τις τεχνικές του συνεισφορές, ο Postel ήταν γνωστός για την ταπεινή και συνεργατική του προσέγγιση. Πίστευε στη σημασία της επίτευξης συναίνεσης και της συνεργασίας για την οικοδόμηση ενός στιβαρού και διαλειτουργικού δικτύου.
- Ο Postel υπηρέτησε ως Διευθυντής του Τμήματος Δικτύων Υπολογιστών στο Ινστιτούτο Επιστημών Πληροφορικής (ISI) του Πανεπιστημίου της Νότιας Καλιφόρνια (USC) από το 1977 μέχρι τον πρόωρο θάνατό του το 1998.
- Σε αναγνώριση των τεράστιων συνεισφορών του, ο Postel τιμήθηκε μετά θάνατον με το περίβλεπτο Βραβείο Turing το 1998, που συχνά αναφέρεται ως το “Νόμπελ της Πληροφορικής.”

Η κλάση JSONFixer προωθεί τη στιβαρότητα, την ευελιξία και τη διαλειτουργικότητα, που ήταν βασικές αξίες τις οποίες υποστήριζε ο Postel καθ’ όλη τη διάρκεια της καριέρας του. Χτίζοντας συστήματα που είναι ανεκτικά στις ατέλειες ενώ διατηρούν αυστηρή προσήλωση στα πρωτόκολλα, μπορούμε να δημιουργήσουμε εφαρμογές που είναι πιο ανθεκτικές και προσαρμοστικές απέναντι στις προκλήσεις του πραγματικού κόσμου.

Παράμετροι και Αντενδείξεις

Η εφαρμοσιμότητα των προσεγγίσεων αυτοθεραπευόμενων δεδομένων εξαρτάται εξ ολοκλήρου από το είδος των δεδομένων που διαχειρίζεται η εφαρμογή σας. Υπάρχει λόγος για τον οποίο ίσως να μην θέλετε απλά να κάνετε monkeypatch τη JSON.parse

για να διορθώνει αυτόματα *όλα τα σφάλματα ανάλυσης JSON* στην εφαρμογή σας: δεν μπορούν ούτε πρέπει να διορθώνονται αυτόματα όλα τα σφάλματα.

Η αυτοθεραπεία είναι ιδιαίτερα προβληματική όταν συνδυάζεται με κανονιστικές απαιτήσεις ή απαιτήσεις συμμόρφωσης σχετικά με τη διαχείριση και επεξεργασία δεδομένων. Ορισμένοι κλάδοι, όπως η υγειονομική περίθαλψη και τα χρηματοοικονομικά, έχουν τόσο αυστηρούς κανονισμούς σχετικά με την ακεραιότητα και την ελεγχσιμότητα των δεδομένων, που η εκτέλεση οποιασδήποτε διόρθωσης δεδομένων τύπου “μαύρου κουτιού” χωρίς κατάλληλη επίβλεψη ή καταγραφή ενδέχεται να παραβιάζει αυτούς τους κανονισμούς. Είναι κρίσιμο να διασφαλίσετε ότι οποιοσδήποτε τεχνικές αυτοθεραπευόμενων δεδομένων αναπτύξετε ευθυγραμμίζονται με τα ισχύοντα νομικά και κανονιστικά πλαίσια.

Η εφαρμογή τεχνικών αυτοθεραπευόμενων δεδομένων, ιδιαίτερα εκείνων που περιλαμβάνουν μοντέλα τεχνητής νοημοσύνης, μπορεί επίσης να έχει μεγάλο αντίκτυπο στην απόδοση της εφαρμογής και στη χρήση πόρων. Η επεξεργασία μεγάλων όγκων δεδομένων μέσω μοντέλων τεχνητής νοημοσύνης για ανίχνευση και διόρθωση σφαλμάτων μπορεί να είναι υπολογιστικά εντατική. Είναι σημαντικό να αξιολογήσετε τους συμβιβασμούς μεταξύ των πλεονεκτημάτων των αυτοθεραπευόμενων δεδομένων και του σχετικού κόστους σε απόδοση και πόρους.

Αυτά λεχθέντα, ας εμβαθύνουμε στους παράγοντες που εμπλέκονται στην απόφαση πότε και πού να εφαρμόσουμε αυτή την ισχυρή προσέγγιση.

Κρισιμότητα Δεδομένων

Κατά την εξέταση της εφαρμογής τεχνικών αυτοθεραπευόμενων δεδομένων, είναι κρίσιμο να αξιολογήσετε την κρισιμότητα των δεδομένων που υποβάλλονται σε επεξεργασία. Το επίπεδο κρισιμότητας αναφέρεται στη σημασία και την ευαισθησία των δεδομένων στο πλαίσιο της εφαρμογής σας και του επιχειρηματικού της τομέα.

Σε ορισμένες περιπτώσεις, η αυτόματη διόρθωση σφαλμάτων δεδομένων μπορεί να μην είναι κατάλληλη, ειδικά εάν τα δεδομένα είναι εξαιρετικά ευαίσθητα ή έχουν νομικές επιπτώσεις. Για παράδειγμα, εξετάστε τα ακόλουθα σενάρια:

1. **Χρηματοοικονομικές Συναλλαγές:** Σε χρηματοοικονομικές εφαρμογές, όπως τραπεζικά συστήματα ή πλατφόρμες συναλλαγών, η ακρίβεια των δεδομένων είναι ύψιστης σημασίας. Ακόμη και μικρά σφάλματα στα χρηματοοικονομικά δεδομένα μπορούν να έχουν σημαντικές συνέπειες, όπως λανθασμένα υπόλοιπα λογαριασμών, εσφαλμένη δρομολόγηση κεφαλαίων ή εσφαλμένες αποφάσεις συναλλαγών. Σε αυτές τις περιπτώσεις, οι αυτοματοποιημένες διορθώσεις χωρίς διεξοδικό έλεγχο και επαλήθευση μπορεί να εισαγάγουν μη αποδεκτούς κινδύνους.
2. **Ιατρικά Αρχεία:** Οι εφαρμογές υγειονομικής περίθαλψης διαχειρίζονται εξαιρετικά ευαίσθητα και εμπιστευτικά δεδομένα ασθενών. Οι ανακρίβειες στα ιατρικά αρχεία μπορούν να έχουν σοβαρές επιπτώσεις στην ασφάλεια των ασθενών και στις αποφάσεις θεραπείας. Η αυτόματη τροποποίηση ιατρικών δεδομένων χωρίς κατάλληλη επίβλεψη και επικύρωση από εξειδικευμένους επαγγελματίες υγείας μπορεί να παραβιάζει τις κανονιστικές απαιτήσεις και να θέσει σε κίνδυνο την ευημερία των ασθενών.
3. **Νομικά Έγγραφα:** Οι εφαρμογές που χειρίζονται νομικά έγγραφα, όπως συμβόλαια, συμφωνίες ή δικαστικές καταθέσεις, απαιτούν αυστηρή ακρίβεια και ακεραιότητα. Ακόμη και μικρά σφάλματα στα νομικά δεδομένα μπορούν να έχουν σημαντικές νομικές επιπτώσεις. Οι αυτοματοποιημένες διορθώσεις σε αυτόν τον τομέα μπορεί να μην είναι κατάλληλες, καθώς τα δεδομένα συχνά απαιτούν χειροκίνητη επανεξέταση και επαλήθευση από νομικούς εμπειρογνώμονες για να διασφαλιστεί η εγκυρότητα και η εκτελεστότητά τους.

Σε αυτά τα κρίσιμα σενάρια δεδομένων, οι κίνδυνοι που σχετίζονται με τις αυτοματοποιημένες διορθώσεις συχνά υπερτερούν των πιθανών οφελών. Οι

συνέπειες της εισαγωγής σφαλμάτων ή της λανθασμένης τροποποίησης δεδομένων μπορεί να είναι σοβαρές, οδηγώντας σε οικονομικές απώλειες, νομικές ευθύνες, ή ακόμη και βλάβη σε άτομα.

Κατά τη διαχείριση εξαιρετικά κρίσιμων δεδομένων, είναι απαραίτητο να δοθεί προτεραιότητα στις διαδικασίες χειροκίνητης επαλήθευσης και επικύρωσης. Η ανθρώπινη εποπτεία και εμπειρογνωμοσύνη είναι κρίσιμες για τη διασφάλιση της ακρίβειας και της ακεραιότητας των δεδομένων. Οι τεχνικές αυτοεπιδιόρθωσης μπορούν ακόμη να χρησιμοποιηθούν για την επισήμανση πιθανών σφαλμάτων ή ασυνεπειών, αλλά η τελική απόφαση για τις διορθώσεις θα πρέπει να περιλαμβάνει ανθρώπινη κρίση και έγκριση.

Ωστόσο, είναι σημαντικό να σημειωθεί ότι δεν έχουν όλα τα δεδομένα σε μια εφαρμογή το ίδιο επίπεδο κρισιμότητας. Στην ίδια εφαρμογή, μπορεί να υπάρχουν υποσύνολα δεδομένων που είναι λιγότερο ευαίσθητα ή έχουν χαμηλότερο αντίκτυπο σε περίπτωση σφαλμάτων. Σε τέτοιες περιπτώσεις, οι τεχνικές αυτοεπιδιόρθωσης δεδομένων μπορούν να εφαρμοστούν επιλεκτικά σε αυτά τα συγκεκριμένα υποσύνολα δεδομένων, ενώ τα κρίσιμα δεδομένα παραμένουν υπό χειροκίνητη επαλήθευση.

Το κλειδί είναι να αξιολογήσετε προσεκτικά την κρισιμότητα κάθε κατηγορίας δεδομένων στην εφαρμογή σας και να καθορίσετε σαφείς κατευθυντήριες γραμμές και διαδικασίες για τη διαχείριση διορθώσεων με βάση τους σχετικούς κινδύνους και τις επιπτώσεις. Διαχωρίζοντας μεταξύ κρίσιμων (π.χ. λογιστικά βιβλία, ιατρικά αρχεία) και μη κρίσιμων δεδομένων (π.χ. ταχυδρομικές διευθύνσεις, προειδοποιήσεις πόρων), μπορείτε να επιτύχετε ισορροπία μεταξύ της αξιοποίησης των πλεονεκτημάτων των τεχνικών αυτοεπιδιόρθωσης δεδομένων όπου είναι κατάλληλο και της διατήρησης αυστηρού ελέγχου και εποπτείας όπου είναι απαραίτητο.

Τελικά, η απόφαση για την εφαρμογή τεχνικών αυτοεπιδιόρθωσης δεδομένων σε κρίσιμα δεδομένα θα πρέπει να λαμβάνεται σε συνεννόηση με ειδικούς του τομέα,

νομικούς συμβούλους και άλλους σχετικούς ενδιαφερόμενους. Είναι απαραίτητο να λαμβάνονται υπόψη οι συγκεκριμένες απαιτήσεις, κανονισμοί και κίνδυνοι που σχετίζονται με τα δεδομένα της εφαρμογής σας και να ευθυγραμμίζονται ανάλογα οι στρατηγικές διόρθωσης δεδομένων.

Σοβαρότητα Σφάλματος

Κατά την εφαρμογή τεχνικών αυτοεπιδιόρθωσης δεδομένων, είναι σημαντικό να αξιολογείται η σοβαρότητα και ο αντίκτυπος των σφαλμάτων δεδομένων. Δεν είναι όλα τα σφάλματα ίδια, και η κατάλληλη πορεία δράσης μπορεί να διαφέρει ανάλογα με τη σοβαρότητα του ζητήματος.

Μικρές ασυνέπειες ή ζητήματα μορφοποίησης μπορεί να είναι κατάλληλα για αυτόματη διόρθωση. Για παράδειγμα, ένας εργάτης αυτοεπιδιόρθωσης δεδομένων που έχει αναλάβει τη διόρθωση κατεστραμμένου JSON μπορεί να χειριστεί ελλείποντα κόμματα ή μη διαφυγόντα διπλά εισαγωγικά χωρίς να αλλάξει σημαντικά το νόημα ή τη δομή των δεδομένων. Αυτοί οι τύποι σφαλμάτων είναι συχνά απλοί στη διόρθωση και έχουν ελάχιστο αντίκτυπο στη συνολική ακεραιότητα των δεδομένων.

Ωστόσο, πιο σοβαρά σφάλματα που αλλάζουν θεμελιωδώς το νόημα ή την ακεραιότητα των δεδομένων μπορεί να απαιτούν διαφορετική προσέγγιση. Σε τέτοιες περιπτώσεις, οι αυτοματοποιημένες διορθώσεις μπορεί να μην είναι επαρκείς, και η ανθρώπινη παρέμβαση μπορεί να είναι απαραίτητη για τη διασφάλιση της ακρίβειας και της εγκυρότητας των δεδομένων.

Εδώ είναι που μπαίνει σε εφαρμογή η έννοια της χρήσης της ίδιας της ΤΝ για τον προσδιορισμό της σοβαρότητας των σφαλμάτων. Αξιοποιώντας τις δυνατότητες των μοντέλων ΤΝ, μπορούμε να σχεδιάσουμε αυτοεπιδιορθούμενους εργάτες δεδομένων που όχι μόνο διορθώνουν σφάλματα, αλλά αξιολογούν επίσης τη σοβαρότητα αυτών των σφαλμάτων και λαμβάνουν τεκμηριωμένες αποφάσεις για τον τρόπο χειρισμού τους.

Για παράδειγμα, ας εξετάσουμε έναν αυτοεπιδιορθούμενο εργάτη δεδομένων υπεύθυνο για τη διόρθωση ασυνεπειών στα δεδομένα που εισρέουν σε μια βάση δεδομένων πελατών. Ο εργάτης μπορεί να σχεδιαστεί ώστε να αναλύει τα δεδομένα και να εντοπίζει πιθανά σφάλματα, όπως ελλειπείς ή αντικρουόμενες πληροφορίες. Ωστόσο, αντί να διορθώνει αυτόματα όλα τα σφάλματα, ο εργάτης μπορεί να εξοπλιστεί με πρόσθετες κλήσεις εργαλείων που του επιτρέπουν να επισημαίνει σοβαρά σφάλματα για ανθρώπινη επισκόπηση.

Ακολουθεί ένα παράδειγμα για το πώς μπορεί να υλοποιηθεί αυτό:

```

1  class CustomerDataReviewer
2    include Raix::ChatCompletion
3    include Raix::FunctionDeclarations
4
5    attr_accessor :customer
6
7    function :flag_for_review, reason: { type: "string" } do |params|
8      AdminNotifier.review_request(customer, params[:reason])
9    end
10
11   def initialize(customer)
12     self.customer = customer
13   end
14
15   def call(customer_data)
16     transcript << {
17       system: "You are a customer data reviewer. Your task is to identify
18         and correct inconsistencies in customer data.
19
20         < additional instructions here... >
21
22         If you encounter severe errors that require human review, use the
23         `flag_for_review` tool to flag the data for manual intervention." }
24
25     transcript << { user: customer.to_json }
26     transcript << { assistant: "Reviewed/corrected data:\n```\n" }
27
28     self.stop = ["```\n"]
29

```

```
30 chat_completion(json: true).then do |result|
31   return if result.blank?
32
33   customer.update(result)
34 end
35 end
36 end
```

Σε αυτό το παράδειγμα, ο εργάτης `CustomerDataHealer` είναι σχεδιασμένος να εντοπίζει και να διορθώνει ασυνέπειες στα δεδομένα πελατών. Για άλλη μια φορά, χρησιμοποιούμε το [Response Fencing](#) και το [Ventriloquist](#) για να πάρουμε δομημένη έξοδο. Σημαντικά, η οδηγία συστήματος του εργάτη περιλαμβάνει οδηγίες για τη χρήση της συνάρτησης `flag_for_review` εάν εντοπιστούν σοβαρά σφάλματα.

Όταν ο εργάτης επεξεργάζεται τα δεδομένα πελατών, αναλύει τα δεδομένα και προσπαθεί να διορθώσει τυχόν ασυνέπειες. Εάν ο εργάτης διαπιστώσει ότι τα σφάλματα είναι σοβαρά και απαιτούν ανθρώπινη παρέμβαση, μπορεί να χρησιμοποιήσει το εργαλείο `flag_for_review` για να επισημάνει τα δεδομένα και να παρέχει έναν λόγο για την επισήμανση.

Η μέθοδος `chat_completion` καλείται με `json: true` για να αναλύσει τα διορθωμένα δεδομένα πελατών ως JSON. Δεν υπάρχει πρόβλεψη για επανάληψη μετά από κλήση συνάρτησης, οπότε το αποτέλεσμα θα είναι κενό εάν κλήθηκε το `flag_for_review`. Διαφορετικά, ο πελάτης ενημερώνεται με τα ελεγμένα και πιθανώς διορθωμένα δεδομένα.

Ενσωματώνοντας την αξιολόγηση της σοβαρότητας των σφαλμάτων και την επιλογή επισήμανσης δεδομένων για ανθρώπινη επισκόπηση, ο εργάτης αυτο-θεραπείας δεδομένων γίνεται πιο έξυπνος και προσαρμοστικός. Μπορεί να χειριστεί μικρά σφάλματα αυτόματα ενώ κλιμακώνει τα σοβαρά σφάλματα σε ειδικούς για χειροκίνητη παρέμβαση.

Τα συγκεκριμένα κριτήρια για τον καθορισμό της σοβαρότητας των σφαλμάτων μπορούν να οριστούν στην οδηγία του εργάτη με βάση τη γνώση του πεδίου και

τις επιχειρησιακές απαιτήσεις. Παράγοντες όπως ο αντίκτυπος στην ακεραιότητα των δεδομένων, η πιθανότητα απώλειας ή αλλοίωσης δεδομένων και οι συνέπειες των εσφαλμένων δεδομένων μπορούν να ληφθούν υπόψη κατά την αξιολόγηση της σοβαρότητας.

Αξιοποιώντας την τεχνητή νοημοσύνη για την αξιολόγηση της σοβαρότητας των σφαλμάτων και παρέχοντας επιλογές για ανθρώπινη παρέμβαση, οι τεχνικές αυτο-θεραπείας δεδομένων μπορούν να επιτύχουν ισορροπία μεταξύ αυτοματισμού και διατήρησης της ακρίβειας των δεδομένων. Αυτή η προσέγγιση διασφαλίζει ότι τα μικρά σφάλματα διορθώνονται αποτελεσματικά ενώ τα σοβαρά σφάλματα λαμβάνουν την απαραίτητη προσοχή και εξειδίκευση από ανθρώπους αξιολογητές.

Πολυπλοκότητα Πεδίου

Κατά την εξέταση της εφαρμογής τεχνικών αυτο-θεραπείας δεδομένων, είναι σημαντικό να αξιολογηθεί η πολυπλοκότητα του πεδίου δεδομένων και οι κανόνες που διέπουν τη δομή και τις σχέσεις του. Η πολυπλοκότητα του πεδίου μπορεί να επηρεάσει σημαντικά την αποτελεσματικότητα και τη σκοπιμότητα των αυτοματοποιημένων προσεγγίσεων διόρθωσης δεδομένων.

Οι τεχνικές αυτο-θεραπείας δεδομένων λειτουργούν καλά όταν τα δεδομένα ακολουθούν καλά καθορισμένα μοτίβα και περιορισμούς. Σε πεδία όπου η δομή των δεδομένων είναι σχετικά απλή και οι σχέσεις μεταξύ των στοιχείων δεδομένων είναι ξεκάθαρες, οι αυτοματοποιημένες διορθώσεις μπορούν να εφαρμοστούν με υψηλό βαθμό εμπιστοσύνης. Για παράδειγμα, η διόρθωση ζητημάτων μορφοποίησης ή η επιβολή βασικών περιορισμών τύπου δεδομένων μπορεί συχνά να αντιμετωπιστεί αποτελεσματικά από εργατές αυτο-θεραπείας δεδομένων.

Ωστόσο, καθώς η πολυπλοκότητα του τομέα δεδομένων αυξάνεται, μεγαλώνουν και οι προκλήσεις που σχετίζονται με την αυτοματοποιημένη διόρθωση δεδομένων. Σε τομείς με περίπλοκη επιχειρησιακή λογική, σύνθετες σχέσεις μεταξύ οντοτήτων δεδομένων ή κανόνες και εξαιρέσεις ειδικούς του τομέα, οι τεχνικές

αυτοθεραπευόμενων δεδομένων ενδέχεται να μην συλλαμβάνουν πάντα τις λεπτές αποχρώσεις και να εισάγουν ακούσιες συνέπειες.

Ας εξετάσουμε ένα παράδειγμα ενός σύνθετου τομέα: ένα σύστημα χρηματιστηριακών συναλλαγών. Σε αυτόν τον τομέα, τα δεδομένα περιλαμβάνουν διάφορα χρηματοοικονομικά εργαλεία, δεδομένα αγοράς, κανόνες συναλλαγών και κανονιστικές απαιτήσεις. Οι σχέσεις μεταξύ διαφορετικών στοιχείων δεδομένων μπορεί να είναι περίπλοκες, και οι κανόνες που διέπουν την εγκυρότητα και τη συνέπεια των δεδομένων μπορεί να είναι ιδιαίτερα συγκεκριμένοι για τον τομέα.

Σε έναν τέτοιο σύνθετο τομέα, ένας εργάτης αυτοθεραπευόμενων δεδομένων επιφορτισμένος με τη διόρθωση ασυνεπειών στα δεδομένα συναλλαγών θα χρειαζόταν να έχει βαθιά κατανόηση των κανόνων και περιορισμών που είναι ειδικοί του τομέα. Θα έπρεπε να λαμβάνει υπόψη παράγοντες όπως οι κανονισμοί της αγοράς, τα όρια συναλλαγών, οι υπολογισμοί κινδύνου και οι διαδικασίες διακανονισμού. Οι αυτοματοποιημένες διορθώσεις σε αυτό το πλαίσιο ενδέχεται να μην συλλαμβάνουν πάντα την πλήρη πολυπλοκότητα του τομέα και μπορεί ακούσια να εισάγουν σφάλματα ή να παραβιάζουν κανόνες ειδικούς του τομέα.

Για την αντιμετώπιση των προκλήσεων της πολυπλοκότητας του τομέα, οι τεχνικές αυτοθεραπευόμενων δεδομένων μπορούν να ενισχυθούν με την ενσωμάτωση γνώσεων και κανόνων ειδικών του τομέα στα μοντέλα ΑΙ και τους εργάτες. Αυτό μπορεί να επιτευχθεί μέσω τεχνικών όπως:

1. **Εκπαίδευση Ειδική του Τομέα:** Τα μοντέλα ΑΙ που χρησιμοποιούνται για αυτοθεραπευόμενα δεδομένα μπορούν να κατευθυνθούν ή ακόμη και να προσαρμοστούν σε σύνολα δεδομένων ειδικά του τομέα που συλλαμβάνουν τις λεπτομέρειες και τους κανόνες του συγκεκριμένου τομέα. Εκθέτοντας τα μοντέλα σε αντιπροσωπευτικά δεδομένα και σενάρια, μπορούν να μάθουν τα μοτίβα, τους περιορισμούς και τις εξαιρέσεις που είναι ειδικές του τομέα.
2. **Περιορισμοί Βάσει Κανόνων:** Οι εργάτες αυτοθεραπευόμενων δεδομένων μπορούν να ενισχυθούν με ρητούς περιορισμούς βάσει κανόνων που

κωδικοποιούν γνώσεις ειδικές του τομέα. Αυτοί οι κανόνες μπορούν να καθοριστούν από ειδικούς του τομέα και να ενσωματωθούν στη διαδικασία διόρθωσης δεδομένων. Τα μοντέλα ΑΙ μπορούν στη συνέχεια να χρησιμοποιήσουν αυτούς τους κανόνες για να καθοδηγήσουν τις αποφάσεις τους και να διασφαλίσουν τη συμμόρφωση με τις απαιτήσεις που είναι ειδικές του τομέα.

3. **Συνεργασία με Ειδικούς του Τομέα:** Σε σύνθετους τομείς, είναι κρίσιμο να συμμετέχουν ειδικοί του τομέα στον σχεδιασμό και την ανάπτυξη τεχνικών αυτοθεραπευόμενων δεδομένων. Οι ειδικοί του τομέα μπορούν να παρέχουν πολύτιμες γνώσεις σχετικά με τις λεπτομέρειες των δεδομένων, τους επιχειρηματικούς κανόνες και τις πιθανές οριακές περιπτώσεις. Οι γνώσεις τους μπορούν να ενσωματωθούν στα μοντέλα ΑΙ και τους εργατές για τη βελτίωση της ακρίβειας και της αξιοπιστίας των αυτοματοποιημένων διορθώσεων δεδομένων χρησιμοποιώντας μοτίβα [Human In The Loop](#).
4. **Σταδιακή και Επαναληπτική Προσέγγιση:** Όταν ασχολούμαστε με σύνθετους τομείς, είναι συχνά επωφελές να υιοθετούμε μια σταδιακή και επαναληπτική προσέγγιση στα αυτοθεραπευόμενα δεδομένα. Αντί να επιχειρούμε να αυτοματοποιήσουμε διορθώσεις για ολόκληρο τον τομέα με μιας, εστιάζουμε σε συγκεκριμένους υποτομείς ή κατηγορίες δεδομένων όπου οι κανόνες και οι περιορισμοί είναι καλά κατανοητοί. Σταδιακά διευρύνουμε το πεδίο εφαρμογής των τεχνικών αυτοθεραπείας καθώς η κατανόηση του τομέα αυξάνεται και οι τεχνικές αποδεικνύονται αποτελεσματικές.

Λαμβάνοντας υπόψη την πολυπλοκότητα του τομέα δεδομένων και ενσωματώνοντας τη γνώση συγκεκριμένου τομέα στις τεχνικές αυτοθεραπευόμενων δεδομένων, μπορείτε να επιτύχετε ισορροπία μεταξύ αυτοματισμού και ακρίβειας. Είναι σημαντικό να αναγνωρίσουμε ότι τα αυτοθεραπευόμενα δεδομένα δεν αποτελούν μια λύση που ταιριάζει σε όλες τις περιπτώσεις και ότι η προσέγγιση θα πρέπει να προσαρμόζεται στις συγκεκριμένες απαιτήσεις και προκλήσεις κάθε τομέα.

Σε πολύπλοκους τομείς, μια υβριδική προσέγγιση που συνδυάζει τεχνικές αυτοθεραπευόμενων δεδομένων με την ανθρώπινη εμπειρογνώμοσύνη και εποπτεία μπορεί να είναι πιο αποτελεσματική. Οι αυτοματοποιημένες διορθώσεις μπορούν να χειριστούν συνηθισμένες και καλά καθορισμένες περιπτώσεις, ενώ τα πολύπλοκα σενάρια ή οι εξαιρέσεις μπορούν να επισημανθούν για ανθρώπινη εξέταση και παρέμβαση. Αυτή η συνεργατική προσέγγιση διασφαλίζει ότι τα οφέλη του αυτοματισμού αξιοποιούνται διατηρώντας παράλληλα τον απαραίτητο έλεγχο και την ακρίβεια σε πολύπλοκους τομείς δεδομένων.

Επεξηγησιμότητα και Διαφάνεια

Η επεξηγησιμότητα αναφέρεται στην ικανότητα κατανόησης και ερμηνείας του συλλογισμού πίσω από τις αποφάσεις που λαμβάνονται από τα μοντέλα τεχνητής νοημοσύνης, ενώ η διαφάνεια περιλαμβάνει την παροχή σαφούς ορατότητας στη διαδικασία διόρθωσης δεδομένων.

Σε πολλά πλαίσια, οι τροποποιήσεις δεδομένων πρέπει να είναι ελέγξιμες και δικαιολογημένες. Τα ενδιαφερόμενα μέρη, συμπεριλαμβανομένων των επιχειρηματικών χρηστών, των ελεγκτών και των ρυθμιστικών αρχών, μπορεί να απαιτούν εξηγήσεις για το γιατί έγιναν ορισμένες διορθώσεις δεδομένων και πώς τα μοντέλα τεχνητής νοημοσύνης κατέληξαν σε αυτές τις αποφάσεις. Αυτό είναι ιδιαίτερα κρίσιμο σε τομείς όπου η ακρίβεια και η ακεραιότητα των δεδομένων έχουν σημαντικές επιπτώσεις, όπως τα οικονομικά, η υγειονομική περίθαλψη και τα νομικά ζητήματα.

Για την αντιμετώπιση της ανάγκης για επεξηγησιμότητα και διαφάνεια, οι τεχνικές αυτοθεραπευόμενων δεδομένων θα πρέπει να ενσωματώνουν μηχανισμούς που παρέχουν πληροφορίες σχετικά με τη διαδικασία λήψης αποφάσεων των μοντέλων τεχνητής νοημοσύνης. Αυτό μπορεί να επιτευχθεί μέσω διαφόρων προσεγγίσεων:

1. **Αλυσίδα Σκέψης:** Ζητώντας από το μοντέλο να εξηγήσει τη σκέψη του “φωναχτά” πριν εφαρμόσει αλλαγές στα δεδομένα μπορεί να επιτρέψει την

ευκολότερη κατανόηση της διαδικασίας λήψης αποφάσεων και μπορεί να δημιουργήσει αναγνώσιμες από τον άνθρωπο εξηγήσεις για τις διορθώσεις που έγιναν. Το αντάλλαγμα είναι λίγο περισσότερη πολυπλοκότητα στο διαχωρισμό της εξήγησης από την έξοδο δομημένων δεδομένων, που μπορεί να αντιμετωπιστεί με...

2. **Δημιουργία Εξηγήσεων:** Οι εργάτες αυτοθεραπευόμενων δεδομένων μπορούν να εξοπλιστούν με την ικανότητα να δημιουργούν αναγνώσιμες από τον άνθρωπο εξηγήσεις για τις διορθώσεις που κάνουν. Αυτό μπορεί να επιτευχθεί ζητώντας από το μοντέλο να εξάγει τη διαδικασία λήψης αποφάσεων του ως εύκολα κατανοητές εξηγήσεις *ενσωματωμένες στα ίδια τα δεδομένα*. Για παράδειγμα, ένας εργάτης αυτοθεραπευόμενων δεδομένων θα μπορούσε να δημιουργήσει μια αναφορά που επισημαίνει τις συγκεκριμένες ασυνέπειες δεδομένων που εντόπισε, τις διορθώσεις που εφάρμοσε και το σκεπτικό πίσω από αυτές τις διορθώσεις.
3. **Σημαντικότητα Χαρακτηριστικών:** Τα μοντέλα τεχνητής νοημοσύνης μπορούν να καθοδηγηθούν με πληροφορίες σχετικά με τη σημασία διαφορετικών χαρακτηριστικών ή ιδιοτήτων στη διαδικασία διόρθωσης δεδομένων ως μέρος των οδηγιών τους. Αυτές οι οδηγίες, με τη σειρά τους, μπορούν να εκτεθούν στα ανθρώπινα ενδιαφερόμενα μέρη. Εντοπίζοντας τους βασικούς παράγοντες που επηρεάζουν τις αποφάσεις του μοντέλου, τα ενδιαφερόμενα μέρη μπορούν να αποκτήσουν γνώση του σκεπτικού πίσω από τις διορθώσεις και να αξιολογήσουν την εγκυρότητά τους.
4. **Καταγραφή και Έλεγχος:** Η εφαρμογή ολοκληρωμένων μηχανισμών καταγραφής και ελέγχου είναι κρίσιμη για τη διατήρηση της διαφάνειας στη διαδικασία αυτοθεραπευόμενων δεδομένων. Κάθε διόρθωση δεδομένων που γίνεται από μοντέλα τεχνητής νοημοσύνης πρέπει να καταγράφεται, συμπεριλαμβανομένων των αρχικών δεδομένων, των διορθωμένων δεδομένων και των συγκεκριμένων ενεργειών που πραγματοποιήθηκαν. Αυτό το ίχνος ελέγχου επιτρέπει την αναδρομική ανάλυση και παρέχει ένα σαφές αρχείο

των τροποποιήσεων που έγιναν στα δεδομένα.

5. **Προσέγγιση με Ανθρώπινη Συμμετοχή:** Η ενσωμάτωση μιας προσέγγισης με ανθρώπινη συμμετοχή μπορεί να ενισχύσει την επεξηγησιμότητα και τη διαφάνεια των τεχνικών αυτοθεραπευόμενων δεδομένων. Με τη συμμετοχή ειδικών στην αξιολόγηση και επικύρωση των διορθώσεων που παράγονται από την τεχνητή νοημοσύνη, οι οργανισμοί μπορούν να διασφαλίσουν ότι οι διορθώσεις ευθυγραμμίζονται με τη γνώση του τομέα και τις επιχειρηματικές απαιτήσεις. Η ανθρώπινη επίβλεψη προσθέτει ένα επιπλέον επίπεδο λογοδοσίας και επιτρέπει τον εντοπισμό πιθανών μεροληψιών ή σφαλμάτων στα μοντέλα τεχνητής νοημοσύνης.
6. **Συνεχής Παρακολούθηση και Αξιολόγηση:** Η τακτική παρακολούθηση και αξιολόγηση της απόδοσης των τεχνικών αυτοθεραπευόμενων δεδομένων είναι απαραίτητη για τη διατήρηση της διαφάνειας και της εμπιστοσύνης. Αξιολογώντας την ακρίβεια και την αποτελεσματικότητα των μοντέλων τεχνητής νοημοσύνης με την πάροδο του χρόνου, οι οργανισμοί μπορούν να εντοπίσουν τυχόν αποκλίσεις ή ανωμαλίες και να λάβουν διορθωτικά μέτρα. Η συνεχής παρακολούθηση βοηθά στη διασφάλιση ότι η διαδικασία αυτοθεραπευόμενων δεδομένων παραμένει αξιόπιστη και ευθυγραμμισμένη με τα επιθυμητά αποτελέσματα.

Η επεξηγησιμότητα και η διαφάνεια αποτελούν κρίσιμες παραμέτρους κατά την εφαρμογή τεχνικών αυτοθεραπευόμενων δεδομένων. Παρέχοντας σαφείς εξηγήσεις για τις διορθώσεις δεδομένων, διατηρώντας ολοκληρωμένα ίχνη ελέγχου και εμπλέκοντας την ανθρώπινη επίβλεψη, οι οργανισμοί μπορούν να οικοδομήσουν εμπιστοσύνη στη διαδικασία αυτοθεραπευόμενων δεδομένων και να διασφαλίσουν ότι οι τροποποιήσεις που γίνονται στα δεδομένα είναι δικαιολογημένες και ευθυγραμμισμένες με τους επιχειρηματικούς στόχους.

Είναι σημαντικό να επιτευχθεί ισορροπία μεταξύ των οφελών του αυτοματισμού και της ανάγκης για διαφάνεια. Ενώ οι τεχνικές αυτοθεραπευόμενων δεδομένων

μπορούν να βελτιώσουν σημαντικά την ποιότητα και την αποδοτικότητα των δεδομένων, δεν θα πρέπει να γίνεται εις βάρος της ορατότητας και του ελέγχου της διαδικασίας διόρθωσης δεδομένων. Σχεδιάζοντας εργαλεία αυτοθεραπευόμενων δεδομένων με γνώμονα την επεξηγησιμότητα και τη διαφάνεια, οι οργανισμοί μπορούν να αξιοποιήσουν τη δύναμη της τεχνητής νοημοσύνης διατηρώντας παράλληλα το απαραίτητο επίπεδο λογοδοσίας και εμπιστοσύνης στα δεδομένα.

Ακούσιες Συνέπειες

Ενώ οι τεχνικές αυτοθεραπευόμενων δεδομένων στοχεύουν στη βελτίωση της ποιότητας και της συνέπειας των δεδομένων, είναι σημαντικό να έχουμε επίγνωση των πιθανών ακούσιων συνεπειών. Οι αυτοματοποιημένες διορθώσεις, εάν δεν σχεδιαστούν και δεν παρακολουθούνται προσεκτικά, μπορεί ακούσια να αλλοιώσουν το νόημα ή το πλαίσιο των δεδομένων, οδηγώντας σε επακόλουθα προβλήματα.

Ένας από τους κύριους κινδύνους των αυτοθεραπευόμενων δεδομένων είναι η εισαγωγή μεροληψίας ή σφαλμάτων στη διαδικασία διόρθωσης δεδομένων. Τα μοντέλα τεχνητής νοημοσύνης, όπως κάθε άλλο σύστημα λογισμικού, μπορεί να υπόκεινται σε μεροληψίες που υπάρχουν στα δεδομένα εκπαίδευσης ή που εισάγονται μέσω του σχεδιασμού των αλγορίθμων. Εάν αυτές οι μεροληψίες δεν εντοπιστούν και δεν μετριάστούν, μπορούν να διαδοθούν μέσω της διαδικασίας αυτοθεραπευόμενων δεδομένων και να οδηγήσουν σε μεροληπτικές ή λανθασμένες τροποποιήσεις δεδομένων.

Για παράδειγμα, ας εξετάσουμε έναν εργάτη αυτοθεραπευόμενων δεδομένων που έχει αναλάβει τη διόρθωση ασυνεπειών σε δημογραφικά δεδομένα πελατών. Εάν το μοντέλο AI έχει μάθει προκαταλήψεις από ιστορικά δεδομένα, όπως η συσχέτιση συγκεκριμένων επαγγελμαμάτων ή επιπέδων εισοδήματος με συγκεκριμένα φύλα ή εθνότητες, μπορεί να κάνει λανθασμένες υποθέσεις και να τροποποιήσει τα δεδομένα με τρόπο που ενισχύει αυτές τις προκαταλήψεις. Αυτό μπορεί να οδηγήσει σε ανακριβή προφίλ πελατών, εσφαλμένες επιχειρηματικές αποφάσεις και πιθανώς

μεροληπτικά αποτελέσματα.

Μια άλλη πιθανή ακούσια συνέπεια είναι η απώλεια πολύτιμων πληροφοριών ή πλαισίου κατά τη διαδικασία διόρθωσης δεδομένων. Οι τεχνικές αυτοθεραπευόμενων δεδομένων συχνά επικεντρώνονται στην τυποποίηση και κανονικοποίηση των δεδομένων για τη διασφάλιση της συνέπειας. Ωστόσο, σε ορισμένες περιπτώσεις, τα αρχικά δεδομένα μπορεί να περιέχουν αποχρώσεις, εξαιρέσεις ή πληροφορίες πλαισίου που είναι σημαντικές για την κατανόηση της πλήρους εικόνας. Οι αυτοματοποιημένες διορθώσεις που επιβάλλουν τυφλά την τυποποίηση μπορεί ακούσια να αφαιρέσουν ή να αποκρύψουν αυτές τις πολύτιμες πληροφορίες.

Για παράδειγμα, φανταστείτε έναν εργάτη αυτοθεραπευόμενων δεδομένων υπεύθυνο για τη διόρθωση ασυνεπειών σε ιατρικά αρχεία. Εάν ο εργάτης συναντήσει το ιατρικό ιστορικό ενός ασθενούς με μια σπάνια πάθηση ή ένα ασυνήθιστο θεραπευτικό πλάνο, μπορεί να προσπαθήσει να κανονικοποιήσει τα δεδομένα ώστε να ταιριάζουν σε ένα πιο συνηθισμένο μοτίβο. Ωστόσο, με αυτόν τον τρόπο, μπορεί να χαθούν οι συγκεκριμένες λεπτομέρειες και το πλαίσιο που είναι κρίσιμα για την ακριβή αναπαράσταση της μοναδικής κατάστασης του ασθενούς. Αυτή η απώλεια πληροφοριών μπορεί να έχει σοβαρές επιπτώσεις στην περίθαλψη των ασθενών και στη λήψη ιατρικών αποφάσεων.

Για τον μετριασμό των κινδύνων από ακούσιες συνέπειες, είναι απαραίτητο να υιοθετήσουμε μια προληπτική προσέγγιση κατά τον σχεδιασμό και την εφαρμογή τεχνικών αυτοθεραπευόμενων δεδομένων:

1. **Διεξοδικός Έλεγχος και Επικύρωση:** Πριν από την ανάπτυξη εργατών αυτοθεραπευόμενων δεδομένων στην παραγωγή, είναι κρίσιμο να ελέγχεται και να επικυρώνεται διεξοδικά η συμπεριφορά τους έναντι διαφόρων σεναρίων. Αυτό περιλαμβάνει δοκιμές με αντιπροσωπευτικά σύνολα δεδομένων που καλύπτουν διάφορες ακραίες περιπτώσεις, εξαιρέσεις και πιθανές προκαταλήψεις. Ο αυστηρός έλεγχος βοηθά στον εντοπισμό και την αντιμετώπιση τυχόν ακούσιων συνεπειών πριν επηρεάσουν τα δεδομένα του

πραγματικού κόσμου.

2. **Συνεχής Παρακολούθηση και Αξιολόγηση:** Η εφαρμογή μηχανισμών συνεχούς παρακολούθησης και αξιολόγησης είναι απαραίτητη για τον εντοπισμό και τον μετριασμό των ακούσιων συνεπειών με την πάροδο του χρόνου. Η τακτική επανεξέταση των αποτελεσμάτων των διαδικασιών αυτοθεραπευόμενων δεδομένων, η ανάλυση των επιπτώσεων στα κατάντη συστήματα και στη λήψη αποφάσεων, και η συλλογή ανατροφοδότησης από τα ενδιαφερόμενα μέρη μπορούν να βοηθήσουν στον εντοπισμό τυχόν δυσμενών επιπτώσεων και να προτρέψουν έγκαιρες διορθωτικές ενέργειες. Εάν ο οργανισμός σας διαθέτει λειτουργικούς πίνακες ελέγχου, η προσθήκη εμφανώς ορατών μετρήσεων σχετικά με τις αυτοματοποιημένες αλλαγές δεδομένων είναι πιθανώς μια καλή ιδέα. Η προσθήκη συναγερμών που συνδέονται με μεγάλες αποκλίσεις από τη φυσιολογική δραστηριότητα αλλαγής δεδομένων είναι πιθανώς μια ακόμη καλύτερη ιδέα!
3. **Ανθρώπινη Επίβλεψη και Παρέμβαση:** Η διατήρηση της ανθρώπινης επίβλεψης και η δυνατότητα παρέμβασης στη διαδικασία αυτοθεραπευόμενων δεδομένων είναι κρίσιμη. Ενώ ο αυτοματισμός μπορεί να βελτιώσει σημαντικά την αποδοτικότητα, είναι σημαντικό να υπάρχουν άνθρωποι ειδικοί που θα επανεξετάζουν και θα επικυρώνουν τις διορθώσεις που γίνονται από μοντέλα ΑΙ, ιδιαίτερα σε κρίσιμους ή ευαίσθητους τομείς. Η ανθρώπινη κρίση και η εξειδίκευση στον τομέα μπορούν να βοηθήσουν στον εντοπισμό και την αντιμετώπιση τυχόν ακούσιων συνεπειών που μπορεί να προκύψουν.
4. **Επεξηγήσιμη Τεχνητή Νοημοσύνη (ΧΑΙ) και Διαφάνεια:** Όπως συζητήθηκε στην προηγούμενη υποενότητα, η ενσωμάτωση τεχνικών επεξηγήσιμης τεχνητής νοημοσύνης και η διασφάλιση της διαφάνειας στη διαδικασία αυτοθεραπευόμενων δεδομένων μπορεί να βοηθήσει στον μετριασμό των ακούσιων συνεπειών. Παρέχοντας σαφείς εξηγήσεις για τις διορθώσεις δεδομένων και διατηρώντας ολοκληρωμένα ίχνη ελέγχου, οι οργανισμοί μπορούν να κατανοήσουν καλύτερα και να εντοπίσουν το σκεπτικό πίσω

από τις τροποποιήσεις που πραγματοποιούνται από τα μοντέλα τεχνητής νοημοσύνης.

5. **Σταδιακή και Επαναληπτική Προσέγγιση:** Η υιοθέτηση μιας σταδιακής και επαναληπτικής προσέγγισης στα αυτοθεραπευόμενα δεδομένα μπορεί να βοηθήσει στην ελαχιστοποίηση του κινδύνου ακούσιων συνεπειών. Αντί να εφαρμόζονται αυτοματοποιημένες διορθώσεις σε ολόκληρο το σύνολο δεδομένων ταυτόχρονα, ξεκινήστε με ένα υποσύνολο δεδομένων και σταδιακά επεκτείνετε το πεδίο εφαρμογής καθώς οι τεχνικές αποδεικνύονται αποτελεσματικές και αξιόπιστες. Αυτό επιτρέπει την προσεκτική παρακολούθηση και προσαρμογή κατά τη διαδρομή, μειώνοντας τον αντίκτυπο τυχόν ακούσιων συνεπειών.
6. **Συνεργασία και Ανατροφοδότηση:** Η εμπλοκή ενδιαφερόμενων μερών από διαφορετικούς τομείς και η ενθάρρυνση της συνεργασίας και της ανατροφοδότησης καθ' όλη τη διάρκεια της διαδικασίας αυτοθεραπευόμενων δεδομένων μπορεί να βοηθήσει στον εντοπισμό και την αντιμετώπιση ακούσιων συνεπειών. Η τακτική αναζήτηση εισροών από ειδικούς του τομέα, καταναλωτές δεδομένων και τελικούς χρήστες μπορεί να παρέχει πολύτιμες γνώσεις σχετικά με τον πραγματικό αντίκτυπο των διορθώσεων δεδομένων και να αναδείξει τυχόν ζητήματα που ενδέχεται να έχουν παραβλεφθεί.

Αντιμετωπίζοντας προληπτικά τον κίνδυνο ακούσιων συνεπειών και εφαρμόζοντας κατάλληλες διασφαλίσεις, οι οργανισμοί μπορούν να αξιοποιήσουν τα οφέλη των τεχνικών αυτοθεραπευόμενων δεδομένων ελαχιστοποιώντας παράλληλα τις πιθανές αρνητικές επιπτώσεις. Είναι σημαντικό να προσεγγίζουμε τα αυτοθεραπευόμενα δεδομένα ως μια επαναληπτική και συνεργατική διαδικασία, παρακολουθώντας, αξιολογώντας και βελτιώνοντας συνεχώς τις τεχνικές για να διασφαλίσουμε ότι ευθυγραμμίζονται με τα επιθυμητά αποτελέσματα και διατηρούν την ακεραιότητα και την αξιοπιστία των δεδομένων.

Κατά την εξέταση της χρήσης μοτίβων αυτοθεραπευόμενων δεδομένων, είναι απαραίτητο να αξιολογούνται προσεκτικά αυτοί οι παράγοντες και να σταθμίζονται τα οφέλη έναντι των πιθανών κινδύνων και περιορισμών. Σε ορισμένες περιπτώσεις, μια υβριδική προσέγγιση που συνδυάζει αυτοματοποιημένες διορθώσεις με ανθρώπινη εποπτεία και παρέμβαση μπορεί να είναι η καταλληλότερη λύση.

Αξίζει επίσης να σημειωθεί ότι οι τεχνικές αυτοθεραπευόμενων δεδομένων δεν πρέπει να θεωρούνται ως υποκατάστατο των ισχυρών μηχανισμών επικύρωσης δεδομένων, εξυγίανσης εισόδου και χειρισμού σφαλμάτων. Αυτές οι θεμελιώδεις πρακτικές παραμένουν κρίσιμες για τη διασφάλιση της ακεραιότητας και της ασφάλειας των δεδομένων. Τα αυτοθεραπευόμενα δεδομένα θα πρέπει να θεωρούνται ως μια συμπληρωματική προσέγγιση που μπορεί να ενισχύσει και να βελτιώσει αυτά τα υπάρχοντα μέτρα.

Τελικά, η απόφαση για τη χρήση μοτίβων αυτοθεραπευόμενων δεδομένων εξαρτάται από τις συγκεκριμένες απαιτήσεις, περιορισμούς και προτεραιότητες της εφαρμογής σας. Εξετάζοντας προσεκτικά τα ζητήματα που περιγράφονται παραπάνω και ευθυγραμμίζοντάς τα με τους στόχους και την αρχιτεκτονική της εφαρμογής σας, μπορείτε να λάβετε τεκμηριωμένες αποφάσεις σχετικά με το πότε και πώς να αξιοποιήσετε αποτελεσματικά τις τεχνικές αυτοθεραπευόμενων δεδομένων.

Παραγωγή Περιεχομένου βάσει Περιβάλλοντος



Τα μοτίβα Παραγωγής Περιεχομένου βάσει Περιβάλλοντος αξιοποιούν τη δύναμη των μεγάλων γλωσσικών μοντέλων (ΜΓΜ) για να δημιουργήσουν δυναμικό και εξειδικευμένο περιεχόμενο εντός των εφαρμογών. Αυτή η κατηγορία μοτίβων αναγνωρίζει τη σημασία της παροχής εξατομικευμένου και σχετικού περιεχομένου στους χρήστες με βάση τις συγκεκριμένες ανάγκες, προτιμήσεις, ακόμη και τις προηγούμενες και τρέχουσες αλληλεπιδράσεις τους με την εφαρμογή.

Στο πλαίσιο αυτής της προσέγγισης, ο όρος “περιεχόμενο” αναφέρεται τόσο στο κύριο περιεχόμενο (δηλαδή αναρτήσεις ιστολογίου, άρθρα, κλπ) όσο και στο μετα-

περιεχόμενο, όπως οι προτάσεις προς το κύριο περιεχόμενο.

Τα μοτίβα Παραγωγής Περιεχομένου βάσει Περιβάλλοντος μπορούν να παίξουν καθοριστικό ρόλο στην ενίσχυση των επιπέδων εμπλοκής των χρηστών σας, παρέχοντας εξατομικευμένες εμπειρίες και αυτοματοποιώντας εργασίες δημιουργίας περιεχομένου τόσο για εσάς όσο και για τους χρήστες σας. Χρησιμοποιώντας τα μοτίβα που περιγράφουμε σε αυτό το κεφάλαιο, μπορείτε να δημιουργήσετε εφαρμογές που παράγουν περιεχόμενο δυναμικά, προσαρμοζόμενες στο περιβάλλον και τις εισόδους σε πραγματικό χρόνο.

Τα μοτίβα λειτουργούν ενσωματώνοντας MGM στις εξόδους της εφαρμογής, από τη διεπαφή χρήστη (μερικές φορές αναφέρεται ως “chrome”), μέχρι τα emails και άλλες μορφές ειδοποιήσεων, καθώς και οποιεσδήποτε διαδικασίες παραγωγής περιεχομένου.

Όταν ένας χρήστης αλληλεπιδρά με την εφαρμογή ή ενεργοποιεί ένα συγκεκριμένο αίτημα περιεχομένου, η εφαρμογή συλλαμβάνει το σχετικό πλαίσιο, όπως τις προτιμήσεις χρήστη, προηγούμενες αλληλεπιδράσεις ή συγκεκριμένες προτροπές. Αυτές οι πληροφορίες πλαισίου τροφοδοτούνται στη συνέχεια στο MGM, μαζί με τυχόν απαραίτητα πρότυπα ή κατευθυντήριες γραμμές και χρησιμοποιούνται για την παραγωγή κειμένου που διαφορετικά θα έπρεπε να είναι είτε σκληρά κωδικοποιημένο, αποθηκευμένο σε βάση δεδομένων, είτε αλγοριθμικά παραγόμενο.

Το περιεχόμενο που παράγεται από το MGM μπορεί να πάρει διάφορες μορφές, όπως εξατομικευμένες προτάσεις, δυναμικές περιγραφές προϊόντων, προσαρμοσμένες απαντήσεις email, ή ακόμη και ολόκληρα άρθρα ή αναρτήσεις ιστολογίου. Μία από τις πιο ριζοσπαστικές χρήσεις αυτού του περιεχομένου που πρωτοπόρησα πριν από περισσότερο από ένα χρόνο είναι η δυναμική δημιουργία στοιχείων διεπαφής χρήστη όπως ετικέτες φορμών, επεξηγηματικά πλαίσια και άλλα είδη επεξηγηματικού κειμένου.

Εξατομίκευση

Ένα από τα βασικά οφέλη των μοτίβων Παραγωγής Περιεχομένου βάσει Περιβάλλοντος είναι η δυνατότητα παροχής εξαιρετικά εξατομικευμένων εμπειριών στους χρήστες. Παράγοντας περιεχόμενο βασισμένο στο συγκεκριμένο πλαίσιο του χρήστη, αυτά τα μοτίβα επιτρέπουν στις εφαρμογές να προσαρμόζουν το περιεχόμενο στα ενδιαφέροντα, τις προτιμήσεις και τις αλληλεπιδράσεις κάθε μεμονωμένου χρήστη.

Η εξατομίκευση ξεπερνά την απλή εισαγωγή του ονόματος ενός χρήστη σε γενικό περιεχόμενο. Περιλαμβάνει την αξιοποίηση του πλούσιου πλαισίου που είναι διαθέσιμο για κάθε χρήστη για τη δημιουργία περιεχομένου που αντηχεί στις συγκεκριμένες ανάγκες και επιθυμίες του. Αυτό το πλαίσιο μπορεί να περιλαμβάνει ένα ευρύ φάσμα παραγόντων, όπως:

1. **Πληροφορίες Προφίλ Χρήστη:** Στο πιο γενικό επίπεδο εφαρμογής αυτής της τεχνικής, τα δημογραφικά δεδομένα, τα ενδιαφέροντα, οι προτιμήσεις και άλλα χαρακτηριστικά του προφίλ μπορούν να χρησιμοποιηθούν για τη δημιουργία περιεχομένου που ευθυγραμμίζεται με το υπόβαθρο και τα χαρακτηριστικά του χρήστη.
2. **Δεδομένα Συμπεριφοράς:** Οι προηγούμενες αλληλεπιδράσεις ενός χρήστη με την εφαρμογή, όπως οι σελίδες που προβλήθηκαν, οι σύνδεσμοι που κλικαρίστηκαν ή τα προϊόντα που αγοράστηκαν, μπορούν να παρέχουν πολύτιμες πληροφορίες για τη συμπεριφορά και τα ενδιαφέροντά του. Αυτά τα δεδομένα μπορούν να χρησιμοποιηθούν για τη δημιουργία προτάσεων περιεχομένου που αντικατοπτρίζουν τα πρότυπα εμπλοκής τους και προβλέπουν τις μελλοντικές τους ανάγκες.
3. **Παράγοντες Πλαισίου:** Το τρέχον πλαίσιο του χρήστη, όπως η τοποθεσία του, η συσκευή, η ώρα της ημέρας ή ακόμη και ο καιρός, μπορεί να επηρεάσει τη διαδικασία παραγωγής περιεχομένου. Για παράδειγμα, μια εφαρμογή ταξιδιών

μπορεί να έχει έναν εργάτη ΑΙ που μπορεί να δημιουργήσει εξατομικευμένες προτάσεις με βάση την τρέχουσα τοποθεσία του χρήστη και τις επικρατούσες καιρικές συνθήκες.

Αξιοποιώντας αυτούς τους παράγοντες πλαισίου, τα πρότυπα Παραγωγής Περιεχομένου βάσει Συμφραζομένων επιτρέπουν στις εφαρμογές να παρέχουν περιεχόμενο που φαίνεται ειδικά προσαρμοσμένο για κάθε μεμονωμένο χρήστη. Αυτό το επίπεδο εξατομίκευσης έχει αρκετά σημαντικά οφέλη:

1. **Αυξημένη Εμπλοκή:** Το εξατομικευμένο περιεχόμενο προσελκύει την προσοχή των χρηστών και τους κρατά δεσμευμένους με την εφαρμογή. Όταν οι χρήστες αισθάνονται ότι το περιεχόμενο είναι σχετικό και απευθύνεται άμεσα στις ανάγκες τους, είναι πιο πιθανό να περάσουν περισσότερο χρόνο αλληλεπιδρώντας με την εφαρμογή και εξερευνώντας τις λειτουργίες της.
2. **Βελτιωμένη Ικανοποίηση Χρήστη:** Το εξατομικευμένο περιεχόμενο δείχνει ότι η εφαρμογή κατανοεί και ενδιαφέρεται για τις μοναδικές απαιτήσεις του χρήστη. Παρέχοντας περιεχόμενο που είναι χρήσιμο, ενημερωτικό και ευθυγραμμισμένο με τα ενδιαφέροντά τους, η εφαρμογή μπορεί να ενισχύσει την ικανοποίηση των χρηστών και να χτίσει μια ισχυρότερη σύνδεση με τους χρήστες της.
3. **Υψηλότερα Ποσοστά Μετατροπής:** Στο πλαίσιο εφαρμογών ηλεκτρονικού εμπορίου ή μάρκετινγκ, το εξατομικευμένο περιεχόμενο μπορεί να επηρεάσει σημαντικά τα ποσοστά μετατροπής. Παρουσιάζοντας στους χρήστες προϊόντα, προσφορές ή συστάσεις που είναι προσαρμοσμένες στις προτιμήσεις και τη συμπεριφορά τους, η εφαρμογή μπορεί να αυξήσει την πιθανότητα οι χρήστες να προβούν στις επιθυμητές ενέργειες, όπως να κάνουν μια αγορά ή να εγγραφούν σε μια υπηρεσία.

Παραγωγικότητα

Τα πρότυπα Παραγωγής Περιεχομένου βάσει Συμφραζομένων μπορούν να ενισχύσουν σημαντικά ορισμένα είδη παραγωγικότητας μειώνοντας την ανάγκη για χειροκίνητη παραγωγή περιεχομένου και επεξεργασία στις δημιουργικές διαδικασίες. Αξιοποιώντας τη δύναμη των Μεγάλων Γλωσσικών Μοντέλων, μπορείτε να δημιουργήσετε περιεχόμενο υψηλής ποιότητας σε μεγάλη κλίμακα, εξοικονομώντας χρόνο και προσπάθεια που διαφορετικά θα έπρεπε να αφιερώσουν οι δημιουργοί περιεχομένου και οι προγραμματιστές σας σε κουραστική χειροκίνητη εργασία.

Παραδοσιακά, οι δημιουργοί περιεχομένου πρέπει να ερευνούν, να γράφουν, να επεξεργάζονται και να μορφοποιούν το περιεχόμενο για να διασφαλίσουν ότι πληροί τις απαιτήσεις της εφαρμογής και τις προσδοκίες των χρηστών. Αυτή η διαδικασία μπορεί να είναι χρονοβόρα και να απαιτεί πολλούς πόρους, ειδικά καθώς ο όγκος του περιεχομένου αυξάνεται.

Ωστόσο, με τα πρότυπα Συναφούς Δημιουργίας Περιεχομένου, η διαδικασία δημιουργίας περιεχομένου μπορεί να αυτοματοποιηθεί σε μεγάλο βαθμό. Τα LLMs μπορούν να δημιουργήσουν συνεκτικό, γραμματικά ορθό και συναφές περιεχόμενο με βάση τις παρεχόμενες προτροπές και κατευθυντήριες γραμμές. Αυτή η αυτοματοποίηση προσφέρει διάφορα οφέλη παραγωγικότητας:

1. **Μειωμένη Χειρωνακτική Προσπάθεια:** Αναθέτοντας εργασίες δημιουργίας περιεχομένου στα LLMs, οι δημιουργοί περιεχομένου μπορούν να επικεντρωθούν σε εργασίες υψηλότερου επιπέδου όπως η στρατηγική περιεχομένου, η ιδεοποίηση και η διασφάλιση ποιότητας. Μπορούν να παρέχουν το απαραίτητο πλαίσιο, πρότυπα και κατευθυντήριες γραμμές στο LLM και να το αφήσουν να χειριστεί την πραγματική δημιουργία περιεχομένου. Αυτό μειώνει τη χειρωνακτική προσπάθεια που απαιτείται για τη συγγραφή

και επεξεργασία, επιτρέποντας στους δημιουργούς περιεχομένου να είναι πιο παραγωγικοί και αποδοτικοί.

2. **Ταχύτερη Δημιουργία Περιεχομένου:** Τα LLMs μπορούν να δημιουργήσουν περιεχόμενο πολύ πιο γρήγορα από τους ανθρώπους συγγραφείς. Με τις κατάλληλες προτροπές και κατευθυντήριες γραμμές, ένα LLM μπορεί να παράγει πολλαπλά κομμάτια περιεχομένου σε δευτερόλεπτα ή λεπτά. Αυτή η ταχύτητα επιτρέπει στις εφαρμογές να δημιουργούν περιεχόμενο με πολύ ταχύτερο ρυθμό, συμβαδίζοντας με τις απαιτήσεις των χρηστών και το συνεχώς μεταβαλλόμενο ψηφιακό τοπίο.

Μήπως η ταχύτερη δημιουργία περιεχομένου οδηγεί σε μια κατάσταση “τραγωδίας των κοινών” όπου το διαδίκτυο πνίγεται σε περιεχόμενο που κανείς δεν διαβάσει; Δυστυχώς, υποψιάζομαι ότι η απάντηση είναι ναι.

3. **Συνέπεια και Ποιότητα:** Τα LLMs μπορούν εύκολα να αναθεωρήσουν το περιεχόμενο ώστε να είναι συνεπές σε στυλ, τόνο και ποιότητα. Με σαφείς κατευθυντήριες γραμμές και παραδείγματα, ορισμένα είδη εφαρμογών (π.χ. ειδησεογραφικά γραφεία, δημόσιες σχέσεις, κλπ.) μπορούν να διασφαλίσουν ότι το περιεχόμενο που δημιουργείται από ανθρώπους ευθυγραμμίζεται με την εταιρική τους φωνή και πληροί τα επιθυμητά πρότυπα ποιότητας. Αυτή η συνέπεια μειώνει την ανάγκη για εκτεταμένη επεξεργασία και αναθεωρήσεις, εξοικονομώντας χρόνο και προσπάθεια στη διαδικασία δημιουργίας περιεχομένου.
4. **Επανάληψη και Βελτιστοποίηση:** Τα πρότυπα Συναφούς Δημιουργίας Περιεχομένου επιτρέπουν τη γρήγορη επανάληψη και βελτιστοποίηση του περιεχομένου. Προσαρμόζοντας τις προτροπές, τα πρότυπα ή τις κατευθυντήριες γραμμές που παρέχονται στο LLM, οι εφαρμογές σας μπορούν

γρήγορα να δημιουργήσουν παραλλαγές περιεχομένου και να δοκιμάσουν διαφορετικές προσεγγίσεις με έναν αυτοματοποιημένο τρόπο που δεν ήταν ποτέ δυνατός στο παρελθόν. Αυτή η επαναληπτική διαδικασία επιτρέπει ταχύτερο πειραματισμό και βελτίωση των στρατηγικών περιεχομένου, οδηγώντας σε πιο αποτελεσματικό και ελκυστικό περιεχόμενο με την πάροδο του χρόνου. Αυτή η συγκεκριμένη τεχνική μπορεί να αποτελέσει πραγματικό παιχνίδι-αλλαγής για εφαρμογές όπως το ηλεκτρονικό εμπόριο που ζουν και πεθαίνουν με βάση τα ποσοστά εγκατάλειψης και την αλληλεπίδραση



Είναι σημαντικό να σημειωθεί ότι ενώ τα πρότυπα Δημιουργίας Περιεχομένου βάσει Περιβάλλοντος μπορούν να ενισχύσουν σημαντικά την παραγωγικότητα, δεν εξαλείφουν εντελώς την ανάγκη για ανθρώπινη συμμετοχή. Οι δημιουργοί περιεχομένου και οι επιμελητές εξακολουθούν να διαδραματίζουν κρίσιμο ρόλο στον καθορισμό της συνολικής στρατηγικής περιεχομένου, παρέχοντας καθοδήγηση στο LLM και διασφαλίζοντας την ποιότητα και την καταλληλότητα του παραγόμενου περιεχομένου.

Με την αυτοματοποίηση των πιο επαναλαμβανόμενων και χρονοβόρων πτυχών της δημιουργίας περιεχομένου, τα πρότυπα Δημιουργίας Περιεχομένου βάσει Περιβάλλοντος απελευθερώνουν πολύτιμο ανθρώπινο χρόνο και πόρους που μπορούν να ανακατευθυνθούν προς εργασίες υψηλότερης αξίας. Αυτή η αυξημένη παραγωγικότητα σας επιτρέπει να παρέχετε πιο εξατομικευμένο και ελκυστικό περιεχόμενο στους χρήστες, βελτιστοποιώντας παράλληλα τις ροές εργασίας δημιουργίας περιεχομένου.

Ταχεία Επανάληψη και Πειραματισμός

Τα πρότυπα Δημιουργίας Περιεχομένου βάσει Περιβάλλοντος σας επιτρέπουν να επαναλαμβάνετε και να πειραματίζεστε γρήγορα με διαφορετικές παραλλαγές

περιεχομένου, επιτρέποντας ταχύτερη βελτιστοποίηση και βελτίωση της στρατηγικής περιεχομένου σας. Μπορείτε να δημιουργήσετε πολλαπλές εκδόσεις περιεχομένου σε δευτερόλεπτα, απλώς προσαρμόζοντας το πλαίσιο, τα πρότυπα ή τις κατευθυντήριες γραμμές που παρέχονται στο μοντέλο.

Αυτή η δυνατότητα ταχείας επανάληψης προσφέρει αρκετά βασικά οφέλη:

1. **Δοκιμή και Βελτιστοποίηση:** Με τη δυνατότητα γρήγορης δημιουργίας παραλλαγών περιεχομένου, μπορείτε εύκολα να δοκιμάσετε διαφορετικές προσεγγίσεις και να μετρήσετε την αποτελεσματικότητά τους. Για παράδειγμα, μπορείτε να δημιουργήσετε πολλαπλές εκδόσεις μιας περιγραφής προϊόντος ή ενός διαφημιστικού μηνύματος, καθεμία προσαρμοσμένη σε ένα συγκεκριμένο τμήμα χρηστών ή πλαίσιο. Αναλύοντας τις μετρήσεις εμπλοκής χρηστών, όπως τα ποσοστά κλικ ή τα ποσοστά μετατροπής, μπορείτε να εντοπίσετε τις πιο αποτελεσματικές παραλλαγές περιεχομένου και να βελτιστοποιήσετε ανάλογα τη στρατηγική περιεχομένου σας.
2. **Δοκιμές A/B:** Τα πρότυπα Δημιουργίας Περιεχομένου βάσει Περιβάλλοντος επιτρέπουν την απρόσκοπτη διεξαγωγή δοκιμών A/B του περιεχομένου. Μπορείτε να δημιουργήσετε δύο ή περισσότερες παραλλαγές περιεχομένου και να τις προβάλετε τυχαία σε διαφορετικές ομάδες χρηστών. Συγκρίνοντας την απόδοση κάθε παραλλαγής, μπορείτε να προσδιορίσετε ποιο περιεχόμενο αντηχεί καλύτερα στο κοινό-στόχο σας. Αυτή η προσέγγιση που βασίζεται σε δεδομένα σας επιτρέπει να λαμβάνετε τεκμηριωμένες αποφάσεις και να βελτιώνετε συνεχώς το περιεχόμενό σας για να μεγιστοποιήσετε την εμπλοκή των χρηστών και να επιτύχετε τα επιθυμητά αποτελέσματα.
3. **Πειράματα Εξατομίκευσης:** Η ταχεία επανάληψη και ο πειραματισμός είναι ιδιαίτερα πολύτιμα όσον αφορά την εξατομίκευση. Με τα πρότυπα Δημιουργίας Περιεχομένου βάσει Περιβάλλοντος, μπορείτε γρήγορα να δημιουργήσετε εξατομικευμένες παραλλαγές περιεχομένου με βάση διαφορετικά τμήματα χρηστών, προτιμήσεις ή συμπεριφορές.

Πειραματιζόμενοι με διαφορετικές στρατηγικές εξατομίκευσης, μπορείτε να εντοπίσετε τις πιο αποτελεσματικές προσεγγίσεις για την εμπλοκή μεμονωμένων χρηστών και την παροχή εξατομικευμένων εμπειριών.

4. **Προσαρμογή στις Μεταβαλλόμενες Τάσεις:** Η δυνατότητα γρήγορης επανάληψης και πειραματισμού σας επιτρέπει να παραμένετε ευέλικτοι και να προσαρμόζεστε στις μεταβαλλόμενες τάσεις και προτιμήσεις των χρηστών. Καθώς εμφανίζονται νέα θέματα, λέξεις-κλειδιά ή συμπεριφορές χρηστών, μπορείτε γρήγορα να δημιουργήσετε περιεχόμενο που ευθυγραμμίζεται με αυτές τις τάσεις. Με τον συνεχή πειραματισμό και τη βελτίωση του περιεχομένου σας, μπορείτε να παραμείνετε σχετικοί και να διατηρήσετε ένα ανταγωνιστικό πλεονέκτημα στο συνεχώς εξελισσόμενο ψηφιακό τοπίο.
5. **Οικονομικά Αποδοτικός Πειραματισμός:** Ο παραδοσιακός πειραματισμός περιεχομένου συχνά απαιτεί σημαντικό χρόνο και πόρους, καθώς οι δημιουργοί περιεχομένου χρειάζεται να αναπτύξουν και να δοκιμάσουν χειροκίνητα διαφορετικές παραλλαγές. Ωστόσο, με τα μοτίβα Παραγωγής Περιεχομένου βάσει Συμφραζομένων, το κόστος του πειραματισμού μειώνεται σημαντικά. Τα Μεγάλα Γλωσσικά Μοντέλα μπορούν να δημιουργήσουν παραλλαγές περιεχομένου γρήγορα και σε μεγάλη κλίμακα, επιτρέποντάς σας να εξερευνήσετε ένα ευρύ φάσμα ιδεών και προσεγγίσεων χωρίς σημαντικό κόστος.

Για να αξιοποιήσετε στο έπακρο την ταχεία επανάληψη και τον πειραματισμό, είναι σημαντικό να έχετε ένα καλά καθορισμένο πλαίσιο πειραματισμού σε εφαρμογή. Αυτό το πλαίσιο θα πρέπει να περιλαμβάνει:

- Σαφείς στόχους και υποθέσεις για κάθε πείραμα
- Κατάλληλες μετρικές και μηχανισμούς παρακολούθησης για τη μέτρηση της απόδοσης του περιεχομένου
- Στρατηγικές κατάτμησης και στόχευσης για να διασφαλιστεί ότι οι σχετικές παραλλαγές περιεχομένου παρέχονται στους σωστούς χρήστες

- Εργαλεία ανάλυσης και αναφοράς για την εξαγωγή συμπερασμάτων από τα πειραματικά δεδομένα
- Μια διαδικασία για την ενσωμάτωση των διδαγμάτων και βελτιστοποιήσεων στη στρατηγική περιεχομένου σας

Υιοθετώντας την ταχεία επανάληψη και τον πειραματισμό, μπορείτε να βελτιώνετε και να βελτιστοποιείτε συνεχώς το περιεχόμενό σας, διασφαλίζοντας ότι παραμένει ελκυστικό, σχετικό και αποτελεσματικό στην επίτευξη των στόχων της εφαρμογής σας. Αυτή η ευέλικτη προσέγγιση στη δημιουργία περιεχομένου σας επιτρέπει να παραμένετε μπροστά από τις εξελίξεις και να προσφέρετε εξαιρετικές εμπειρίες χρήστη.

Κλιμάκωση και Αποδοτικότητα

Καθώς οι εφαρμογές αναπτύσσονται και η ζήτηση για εξατομικευμένο περιεχόμενο αυξάνεται, τα μοτίβα παραγωγής περιεχομένου βάσει συμφραζομένων επιτρέπουν την αποδοτική κλιμάκωση της δημιουργίας περιεχομένου. Τα Μεγάλα Γλωσσικά Μοντέλα μπορούν να παράγουν περιεχόμενο για μεγάλο αριθμό χρηστών και πλαϊσίων ταυτόχρονα, χωρίς την ανάγκη για ανάλογη αύξηση των ανθρώπινων πόρων. Αυτή η κλιμάκωση επιτρέπει στις εφαρμογές να προσφέρουν εξατομικευμένες εμπειρίες σε μια αυξανόμενη βάση χρηστών χωρίς να επιβαρύνουν τις δυνατότητες δημιουργίας περιεχομένου τους.



Σημειώστε ότι η παραγωγή περιεχομένου βάσει συμφραζομένων μπορεί να χρησιμοποιηθεί αποτελεσματικά για τη διεθνοποίηση της εφαρμογής σας “στη στιγμή”. Στην πραγματικότητα, αυτό ακριβώς έκανα χρησιμοποιώντας το Instant18n Gem μου για να προσφέρω την Olympia σε περισσότερες από μισή ντουζίνα γλώσσες, παρόλο που είμαστε λιγότερο από ένα έτος σε λειτουργία.

Τοπικοποίηση με Τεχνητή Νοημοσύνη

Αν μου επιτρέπετε να καυχηθώ για μια στιγμή, πιστεύω ότι η βιβλιοθήκη μου Instant18n για εφαρμογές Rails είναι ένα πρωτοποριακό παράδειγμα του μοτίβου “Παραγωγή Περιεχομένου βάσει Συμφραζομένων” σε δράση, που αναδεικνύει τη μετασχηματιστική δυνατότητα της TN στην ανάπτυξη εφαρμογών. Αυτό το gem αξιοποιεί τη δύναμη του μεγάλου γλωσσικού μοντέλου GPT της OpenAI για να φέρει επανάσταση στον τρόπο με τον οποίο η διεθνοποίηση και η τοπικοποίηση αντιμετωπίζονται στις εφαρμογές Rails.

Παραδοσιακά, η διεθνοποίηση μιας εφαρμογής Rails περιλαμβάνει τον χειροκίνητο ορισμό κλειδιών μετάφρασης και την παροχή αντίστοιχων μεταφράσεων για κάθε υποστηριζόμενη γλώσσα. Αυτή η διαδικασία μπορεί να είναι χρονοβόρα, απαιτητική σε πόρους και επιρρεπής σε ασυνέπειες. Ωστόσο, με το gem Instant18n, το παράδειγμα της τοπικοποίησης επαναπροσδιορίζεται πλήρως.

Με την ενσωμάτωση ενός μοντέλου μεγάλης γλώσσας, το gem Instant18n σας επιτρέπει να δημιουργείτε μεταφράσεις σε πραγματικό χρόνο, με βάση το περιεχόμενο και τη σημασία του κειμένου. Αντί να βασίζεται σε προκαθορισμένα κλειδιά μετάφρασης και στατικές μεταφράσεις, το gem μεταφράζει δυναμικά κείμενο χρησιμοποιώντας τη δύναμη της τεχνητής νοημοσύνης. Αυτή η προσέγγιση προσφέρει αρκετά βασικά πλεονεκτήματα:

1. **Απρόσκοπτη Τοπικοποίηση:** Με το gem Instant18n, οι προγραμματιστές δεν χρειάζεται πλέον να ορίζουν και να διατηρούν χειροκίνητα αρχεία μετάφρασης για κάθε υποστηριζόμενη γλώσσα. Το gem δημιουργεί αυτόματα μεταφράσεις με βάση το παρεχόμενο κείμενο και την επιθυμητή γλώσσα-στόχο, καθιστώντας τη διαδικασία τοπικοποίησης εύκολη και απρόσκοπτη.
2. **Ακρίβεια Περιεχομένου:** Η τεχνητή νοημοσύνη μπορεί να λάβει αρκετό περιεχόμενο ώστε να κατανοήσει τις λεπτές αποχρώσεις του κειμένου που μεταφράζεται. Μπορεί να λάβει υπόψη το περιβάλλον περιεχόμενο, τις

ιδιωματικές εκφράσεις και τις πολιτιστικές αναφορές για να δημιουργήσει μεταφράσεις που είναι ακριβείς, φυσικές και κατάλληλες για το περιεχόμενο.

3. **Εκτεταμένη Υποστήριξη Γλωσσών:** Το gem Instant18n αξιοποιεί την τεράστια γνώση και τις γλωσσικές δυνατότητες του GPT, επιτρέποντας μεταφράσεις σε ένα εκτεταμένο φάσμα γλωσσών. Από κοινές γλώσσες όπως τα Ισπανικά και τα Γαλλικά μέχρι πιο σπάνιες ή φανταστικές γλώσσες όπως τα Κλίνγκον και τα Ξωτικά, το gem μπορεί να διαχειριστεί μια μεγάλη ποικιλία απαιτήσεων μετάφρασης.
4. **Ευελιξία και Δημιουργικότητα:** Το gem ξεπερνά τις παραδοσιακές μεταφράσεις γλωσσών και επιτρέπει δημιουργικές και μη συμβατικές επιλογές τοπικοποίησης. Οι προγραμματιστές μπορούν να μεταφράσουν κείμενο σε διάφορα στυλ, διαλέκτους, ή ακόμη και φανταστικές γλώσσες, ανοίγοντας νέες δυνατότητες για μοναδικές εμπειρίες χρήστη και ελκυστικό περιεχόμενο.
5. **Βελτιστοποίηση Απόδοσης:** Το gem Instant18n ενσωματώνει μηχανισμούς προσωρινής αποθήκευσης για τη βελτίωση της απόδοσης και τη μείωση της επιβάρυνσης των επαναλαμβανόμενων μεταφράσεων. Το μεταφρασμένο κείμενο αποθηκεύεται προσωρινά, επιτρέποντας στα επόμενα αιτήματα για την ίδια μετάφραση να εξυπηρετούνται γρήγορα χωρίς την ανάγκη για περιττές κλήσεις API.

Το gem Instant18n αποτελεί παράδειγμα της δύναμης του μοτίβου “Παραγωγή Περιεχομένου με βάση το Περιεχόμενο” αξιοποιώντας την τεχνητή νοημοσύνη για τη δυναμική δημιουργία τοπικοποιημένου περιεχομένου. Δείχνει πώς η τεχνητή νοημοσύνη μπορεί να ενσωματωθεί στη βασική λειτουργικότητα μιας εφαρμογής Rails, μεταμορφώνοντας τον τρόπο με τον οποίο οι προγραμματιστές προσεγγίζουν τη διεθνοποίηση και την τοπικοποίηση.

Εξαλείφοντας την ανάγκη για χειροκίνητη διαχείριση μεταφράσεων και επιτρέποντας μεταφράσεις σε πραγματικό χρόνο με βάση το περιεχόμενο, το gem

Instant18n εξοικονομεί σημαντικό χρόνο και προσπάθεια στους προγραμματιστές. Τους επιτρέπει να επικεντρωθούν στην ανάπτυξη των βασικών χαρακτηριστικών της εφαρμογής τους, διασφαλίζοντας παράλληλα ότι η πτυχή της τοπικοποίησης αντιμετωπίζεται απρόσκοπτα και με ακρίβεια.

Η Σημασία των Δοκιμών Χρηστών και της Ανατροφοδότησης

Τέλος, να έχετε πάντα κατά νου τη σημασία των δοκιμών χρηστών και της ανατροφοδότησης. Είναι κρίσιμο να επικυρώνετε ότι η παραγωγή περιεχομένου με βάση το περιεχόμενο ανταποκρίνεται στις προσδοκίες των χρηστών και ευθυγραμμίζεται με τους στόχους της εφαρμογής. Συνεχίστε να επαναλαμβάνετε και να βελτιώνετε το παραγόμενο περιεχόμενο με βάση τις γνώσεις των χρηστών και τα αναλυτικά στοιχεία. Εάν παράγετε δυναμικό περιεχόμενο σε μεγάλη κλίμακα που θα ήταν αδύνατο να επικυρωθεί χειροκίνητα από εσάς και την ομάδα σας, ξετάστε το ενδεχόμενο να προσθέσετε μηχανισμούς ανατροφοδότησης που επιτρέπουν στους χρήστες να αναφέρουν περιεχόμενο που είναι περίεργο ή λανθασμένο, μαζί με μια εξήγηση του γιατί. Αυτή η πολύτιμη ανατροφοδότηση μπορεί ακόμη και να τροφοδοτηθεί σε έναν εργάτη τεχνητής νοημοσύνης που είναι επιφορτισμένος με την πραγματοποίηση προσαρμογών στο στοιχείο που δημιούργησε το περιεχόμενο!

Παραγωγική Διεπαφή Χρήστη



Η προσοχή είναι τόσο πολύτιμη στις μέρες μας που η αποτελεσματική εμπλοκή των χρηστών απαιτεί πλέον εμπειρίες λογισμικού που δεν είναι μόνο απρόσκοπτες και διαισθητικές, αλλά και εξαιρετικά εξατομικευμένες στις ατομικές ανάγκες, προτιμήσεις και πλαίσια. Ως αποτέλεσμα, οι σχεδιαστές και οι προγραμματιστές αντιμετωπίζουν όλο και περισσότερο την πρόκληση της δημιουργίας διεπαφών χρήστη που μπορούν να προσαρμόζονται και να ανταποκρίνονται στις μοναδικές απαιτήσεις κάθε χρήστη *σε κλίμακα*.

Η Παραγωγική Διεπαφή Χρήστη (GenUI) είναι μια πραγματικά επαναστατική προσέγγιση στο σχεδιασμό διεπαφής χρήστη που αξιοποιεί τη δύναμη των μεγάλων γλωσσικών μοντέλων (LLMs) για τη δημιουργία εξαιρετικά εξατομικευμένων και δυναμικών εμπειριών χρήστη σε πραγματικό χρόνο. Ήθελα να σας δώσω τουλάχιστον μια εισαγωγή στο GenUI σε αυτό το βιβλίο, γιατί πιστεύω ότι είναι μία από τις πιο παρθένες ευκαιρίες που υπάρχουν αυτή τη στιγμή στον τομέα

του σχεδιασμού εφαρμογών και των πλαισίων εργασίας. Είμαι πεπεισμένος ότι δεκάδες ή και περισσότερα νέα επιτυχημένα εμπορικά και ανοιχτού κώδικα έργα θα αναδυθούν σε αυτή τη συγκεκριμένη εξειδίκευση.

Στον πυρήνα του, το GenUI συνδυάζει τις αρχές της [Παραγωγής Περιεχομένου Βάσει Συμφραζομένων](#) με προηγμένες τεχνικές τεχνητής νοημοσύνης για τη δυναμική δημιουργία στοιχείων διεπαφής χρήστη, όπως κείμενο, εικόνες και διατάξεις, με βάση τη βαθιά κατανόηση του πλαισίου, των προτιμήσεων και των στόχων του χρήστη. Το GenUI επιτρέπει στους σχεδιαστές και τους προγραμματιστές να δημιουργούν διεπαφές που προσαρμόζονται και εξελίσσονται ανταποκρινόμενες στις αλληλεπιδράσεις των χρηστών, παρέχοντας ένα επίπεδο εξατομίκευσης που ήταν προηγουμένως ανέφικτο.

Το GenUI αντιπροσωπεύει μια θεμελιώδη αλλαγή στον τρόπο που προσεγγίζουμε το σχεδιασμό διεπαφής χρήστη. Αντί να σχεδιάζουμε για τις μάζες, το GenUI μας επιτρέπει να σχεδιάζουμε για το άτομο. Το εξατομικευμένο περιεχόμενο και οι διεπαφές έχουν τη δυνατότητα να δημιουργούν εμπειρίες χρήστη που αντηχούν σε κάθε χρήστη σε βαθύτερο επίπεδο, αυξάνοντας τη δέσμευση, την ικανοποίηση και την πιστότητα.

Ως πρωτοποριακή τεχνική, η μετάβαση στο GenUI είναι γεμάτη εννοιολογικές και πρακτικές προκλήσεις. Η ενσωμάτωση της τεχνητής νοημοσύνης στη διαδικασία σχεδιασμού, η διασφάλιση ότι οι παραγόμενες διεπαφές δεν είναι μόνο εξατομικευμένες αλλά και εύχρηστες, προσβάσιμες και ευθυγραμμισμένες με τη συνολική επωνυμία και εμπειρία χρήστη, όλα αυτά είναι προκλήσεις που καθιστούν το GenUI μια επιδίωξη για τους λίγους και όχι για τους πολλούς. Επιπλέον, η συμμετοχή της τεχνητής νοημοσύνης εγείρει ερωτήματα σχετικά με την προστασία των δεδομένων, τη διαφάνεια και ίσως ακόμη και ηθικές επιπτώσεις

Παρά τις προκλήσεις, οι εξατομικευμένες εμπειρίες σε μεγάλη κλίμακα έχουν τη δύναμη να μεταμορφώσουν εντελώς τον τρόπο με τον οποίο αλληλεπιδρούμε με τα ψηφιακά προϊόντα και τις υπηρεσίες. Ανοίγει δυνατότητες για τη δημιουργία

συμπεριληπτικών και προσβάσιμων διεπαφών που εξυπηρετούν τις διαφορετικές ανάγκες των χρηστών, ανεξάρτητα από τις ικανότητες, το υπόβαθρο ή τις προτιμήσεις τους.

Σε αυτό το κεφάλαιο, θα εξερευνήσουμε την έννοια του GenUI, εξετάζοντας κάποια καθοριστικά χαρακτηριστικά, βασικά οφέλη και πιθανές προκλήσεις. Ξεκινάμε εξετάζοντας την πιο βασική και προσβάσιμη μορφή του GenUI: τη δημιουργία κειμένου για διεπαφές χρήστη που κατά τα άλλα έχουν σχεδιαστεί και υλοποιηθεί με παραδοσιακό τρόπο.

Δημιουργία Κειμένου για Διεπαφές Χρήστη

Τα στοιχεία κειμένου που υπάρχουν στο chrome της εφαρμογής σας, όπως οι ετικέτες φόρμας, τα επεξηγηματικά κείμενα και το επεξηγηματικό κείμενο, συνήθως είναι σκληρά κωδικοποιημένα στα πρότυπα ή στα στοιχεία διεπαφής χρήστη, παρέχοντας μια συνεπή αλλά γενική εμπειρία για όλους τους χρήστες. Χρησιμοποιώντας μοτίβα παραγωγής περιεχομένου βάσει συμφραζομένων, μπορείτε να μετατρέψετε αυτά τα στατικά στοιχεία σε δυναμικά, προσαρμοσμένα στο περιβάλλον και εξατομικευμένα συστατικά.

Εξατομικευμένες Φόρμες

Οι φόρμες αποτελούν ένα πανταχού παρόν κομμάτι των διαδικτυακών και κινητών εφαρμογών, λειτουργώντας ως το κύριο μέσο συλλογής δεδομένων από τους χρήστες. Ωστόσο, οι παραδοσιακές φόρμες συχνά παρουσιάζουν μια γενική και απρόσωπη εμπειρία, με τυποποιημένες ετικέτες και πεδία που μπορεί να μην ευθυγραμμίζονται πάντα με το συγκεκριμένο πλαίσιο ή τις ανάγκες του χρήστη. Οι χρήστες είναι πιο πιθανό να συμπληρώσουν φόρμες που φαίνονται προσαρμοσμένες στις ανάγκες και τις προτιμήσεις τους, οδηγώντας σε υψηλότερα ποσοστά μετατροπής και ικανοποίηση χρηστών.

Ωστόσο, είναι σημαντικό να επιτευχθεί ισορροπία μεταξύ εξατομίκευσης και συνέπειας. Ενώ η προσαρμογή των φορμών στους μεμονωμένους χρήστες μπορεί να είναι επωφελής, είναι κρίσιμο να διατηρηθεί ένα επίπεδο οικειότητας και προβλεψιμότητας. Οι χρήστες θα πρέπει να εξακολουθούν να μπορούν να αναγνωρίζουν και να πλοηγούνται εύκολα στις φόρμες, ακόμη και με εξατομικευμένα στοιχεία.

Ακολουθούν μερικές ιδέες εξατομικευμένων φορμών για έμπνευση:

Προτάσεις Πεδίων βάσει Συμφραζομένων

Το GenUI μπορεί να αναλύσει τις προηγούμενες αλληλεπιδράσεις, τις προτιμήσεις και τα δεδομένα του χρήστη για να παρέχει έξυπνες προτάσεις πεδίων ως προβλέψεις. Για παράδειγμα, εάν ο χρήστης έχει εισαγάγει προηγουμένως τη διεύθυνση αποστολής του, η φόρμα μπορεί να συμπληρώσει αυτόματα τα σχετικά πεδία με τις αποθηκευμένες πληροφορίες του. Αυτό όχι μόνο εξοικονομεί χρόνο αλλά επίσης δείχνει ότι η εφαρμογή κατανοεί και θυμάται τις προτιμήσεις του χρήστη.

Για μια στιγμή, δεν είναι αυτή η τεχνική κάτι που θα μπορούσε να γίνει χωρίς τη χρήση τεχνητής νοημοσύνης; Φυσικά, αλλά η ομορφιά της υλοποίησης αυτού του είδους λειτουργικότητας με τεχνητή νοημοσύνη είναι διπλή: 1) πόσο εύκολο μπορεί να είναι στην υλοποίηση και 2) πόσο ανθεκτικό μπορεί να είναι καθώς η διεπαφή χρήστη σας αλλάζει και εξελίσσεται με την πάροδο του χρόνου.

Ας φτιάξουμε στα γρήγορα μια υπηρεσία για το θεωρητικό μας σύστημα διαχείρισης παραγγελιών, που προσπαθεί να συμπληρώσει προκαταβολικά τη σωστή διεύθυνση αποστολής για τον χρήστη.

```
1  class OrderShippingAddressSubscriber
2    include Raix::ChatCompletion
3
4    attr_accessor :order
5
6    delegate :customer, to: :order
7
8    DIRECTIVE = "You are a smart order processing assistant. Given the
9    customer's order history, guess the most likely shipping address
10   for the current order."
11
12   def order_created(order)
13     return unless order.pending? && order.shipping_address.blank?
14
15     self.order = order
16
17     transcript.clear
18     transcript << { system: DIRECTIVE }
19     transcript << { user: "Order History: #{order_history.to_json}" }
20     transcript << { user: "Current Order: #{order.to_json}" }
21
22     response = chat_completion
23     apply_predicted_shipping_address(order, response)
24   end
25
26   private
27
28   def apply_predicted_shipping_address(order, response)
29     # extract the shipping address from the response...
30     # ...and assume there's some sort of live update of the address fields
31     order.update(shipping_address:)
32   end
33
34   def order_history
35     customer.orders.successful.limit(100).map do |order|
36       {
37         date: order.date,
38         description: order.description,
39         shipping_address: order.shipping_address
40       }
41     end
42   end
```

43 **end**

Αυτό το παράδειγμα είναι πολύ απλοποιημένο, αλλά θα πρέπει να λειτουργεί στις περισσότερες περιπτώσεις. Η ιδέα είναι να αφήσουμε την TN να κάνει μια εκτίμηση με τον ίδιο τρόπο που θα έκανε ένας άνθρωπος. Για να γίνει σαφές αυτό που εννοώ, ας εξετάσουμε κάποια δειγματικά δεδομένα:

```
1 Order History:
2 [
3   {"date": "2024-01-03", "description": "garden soil mix",
4     "shipping_address": "123 Country Lane, Rural Town"},
5   {"date": "2024-01-15", "description": "hardcover fiction novels",
6     "shipping_address": "456 City Apt, Metroville"},
7   {"date": "2024-01-22", "description": "baby diapers", "shipping_address":
8     "789 Suburb St, Quietville"},
9   {"date": "2024-02-01", "description": "organic vegetables",
10    "shipping_address": "123 Country Lane, Rural Town"},
11  {"date": "2024-02-17", "description": "mystery thriller book set",
12    "shipping_address": "456 City Apt, Metroville"},
13  {"date": "2024-02-25", "description": "baby wipes",
14    "shipping_address": "789 Suburb St, Quietville"},
15  {"date": "2024-03-05", "description": "flower seeds",
16    "shipping_address": "123 Country Lane, Rural Town"},
17  {"date": "2024-03-20", "description": "biographies",
18    "shipping_address": "456 City Apt, Metroville"},
19  {"date": "2024-03-30", "description": "baby formula",
20    "shipping_address": "789 Suburb St, Quietville"},
21  {"date": "2024-04-12", "description": "lawn fertilizer",
22    "shipping_address": "123 Country Lane, Rural Town"},
23  {"date": "2024-04-22", "description": "science fiction novels",
24    "shipping_address": "456 City Apt, Metroville"},
25  {"date": "2024-05-02", "description": "infant toys",
26    "shipping_address": "789 Suburb St, Quietville"},
27  {"date": "2024-05-14", "description": "outdoor grill",
28    "shipping_address": "123 Country Lane, Rural Town"},
29  {"date": "2024-05-29", "description": "literary classics",
30    "shipping_address": "456 City Apt, Metroville"},
31  {"date": "2024-06-11", "description": "baby clothes",
32    "shipping_address": "789 Suburb St, Quietville"},
33  {"date": "2024-07-01", "description": "watering can",
```

```
34     "shipping_address": "123 Country Lane, Rural Town"},
35     {"date": "2024-07-18", "description": "non-fiction essays",
36     "shipping_address": "456 City Apt, Metroville"},
37     {"date": "2024-07-28", "description": "baby bath items",
38     "shipping_address": "789 Suburb St, Quietville"},
39     {"date": "2024-08-09", "description": "herb garden kit",
40     "shipping_address": "123 Country Lane, Rural Town"},
41     {"date": "2024-08-24", "description": "children's books",
42     "shipping_address": "456 City Apt, Metroville"}
43 ]
```

Παρατηρήσατε το μοτίβο στα δεδομένα; Σας εγγυώμαι ότι αυτό είναι εύκολο για ένα LLM. Για να το αποδείξουμε, ας ρωτήσουμε το GPT-4 ποια είναι η πιο πιθανή διεύθυνση αποστολής για ένα “θερμόμετρο”.

```
1  From the order history you've provided, it looks like the purchases are
2  generally clustered into three main types based on the shipping addresses:
3
4  123 Country Lane, Rural Town - This address often orders garden and
5  outdoor-related items like soil mix, vegetables, flower seeds, lawn
6  fertilizer, an outdoor grill, a watering can, and an herb garden kit.
7
8  456 City Apt, Metroville - This address tends to order various types of
9  books, indicating an interest in reading across genres like fiction,
10  thrillers, biographies, science fiction, literary classics, non-fiction
11  essays, and children's books.
12
13  789 Suburb St, Quietville - The purchases for this address are consistently
14  baby-related products, such as diapers, wipes, baby formula, infant toys,
15  baby clothes, and baby bath items.
16
17  Given these patterns, a digital thermometer could potentially be a fit for
18  any of the addresses based on different needs: outdoor temperature tracking,
19  a tool for a passionate reader interested in science gadgets, or a health
20  tool for a baby. However, considering its common use in monitoring health,
21  especially for babies, it's most likely to be purchased by someone who's
22  already buying baby care products.
23
24  Therefore, the most likely shipping address for a digital thermometer, based
25  on the purchasing patterns, would be 789 Suburb St, Quietville.
```

Αν σκέφτεσαι ότι είναι υπερβολή να χρησιμοποιείς ένα ακριβό μοντέλο όπως το GPT-4 για αυτή την εργασία, έχεις δίκιο! Δοκίμασα την ίδια εντολή στο Mistral 7B Instruct και παρήγαγε την ακόλουθη απάντηση με 75 λέξεις ανά δευτερόλεπτο, και με το πενιχρό κόστος των \$0.000218 USD.

```
1 Based on the given order history, it appears that the customer has received
2 orders for baby-related items (diapers, wipes, formula, clothes, bath items)
3 at the address "789 Suburb St, Quietville" and garden-related items (soil,
4 seeds, watering can, herb garden kit) at the address "123 Country Lane, Rural
5 Town". The customer has also received books (fiction, non-fiction, literary
6 classics, mystery-thriller, science fiction, biographies, children's books)
7 at the address "456 City Apt, Metroville".
8
9 Given that the new order is for a "thermometer", which doesn't seem to fit
10 into any of the categories that the customer has previously ordered, it's
11 hard to accurately predict the shipping address based solely on the order
12 history. However, given the 50%-50% split between baby-related and
13 garden-related items, it could somewhat lean towards the Baby-related items
14 address ("789 Suburb St, Quietville"). But remember, this is an assumption
15 and cannot be definitively confirmed without more context or information.
```

Αξίζει το κόστος και η πολυπλοκότητα αυτής της τεχνικής για να κάνουμε την εμπειρία ολοκλήρωσης αγοράς πιο μαγική; Για πολλούς διαδικτυακούς εμπόρους λιανικής, σίγουρα. Και από ό,τι φαίνεται, το κόστος της υπολογιστικής ισχύος AI θα συνεχίσει να μειώνεται, ειδικά για τους παρόχους φιλοξενίας μοντέλων ανοιχτού κώδικα που ανταγωνίζονται για την χαμηλότερη τιμή.



Χρησιμοποιήστε ένα [Prompt Template](#) και [StructuredIO](#) μαζί με [Response Fencing](#) για να βελτιστοποιήσετε αυτού του είδους την ολοκλήρωση συνομιλίας.

Προσαρμοστική Διάταξη Πεδίων

Η σειρά με την οποία παρουσιάζονται τα πεδία της φόρμας μπορεί να επηρεάσει σημαντικά την εμπειρία του χρήστη και τα ποσοστά ολοκλήρωσης. Με το GenUI,

μπορείτε να προσαρμόσετε δυναμικά τη σειρά των πεδίων με βάση το πλαίσιο του χρήστη και τη σημασία κάθε πεδίου. Για παράδειγμα, εάν ο χρήστης συμπληρώνει μια φόρμα εγγραφής για μια εφαρμογή γυμναστικής, η φόρμα θα μπορούσε να δώσει προτεραιότητα στα πεδία που σχετίζονται με τους στόχους φυσικής κατάστασης και τις προτιμήσεις τους, κάνοντας τη διαδικασία πιο σχετική και ελκυστική.

Εξατομικευμένο Μικροκείμενο

Το επεξηγηματικό κείμενο, τα μηνύματα σφάλματος και άλλα μικροκείμενα που σχετίζονται με τις φόρμες μπορούν επίσης να εξατομικευτούν χρησιμοποιώντας το GenUI. Αντί να εμφανίζονται γενικά μηνύματα σφάλματος όπως “Μη έγκυρη διεύθυνση email”, μπορείτε να δημιουργήσετε πιο χρήσιμα και συναφή μηνύματα όπως “Παρακαλώ εισάγετε μια έγκυρη διεύθυνση email για να λάβετε την επιβεβαίωση της παραγγελίας σας.” Αυτές οι εξατομικευμένες λεπτομέρειες μπορούν να κάνουν την εμπειρία της φόρμας πιο φιλική προς το χρήστη και λιγότερο απογοητευτική.

Εξατομικευμένη Επικύρωση

Στο ίδιο πνεύμα με το Εξατομικευμένο Μικροκείμενο, θα μπορούσατε να χρησιμοποιήσετε την ΑΙ για να επικυρώσετε τη φόρμα με τρόπους που φαίνονται μαγικοί. Φανταστείτε να αφήνετε μια ΑΙ να επικυρώνει μια φόρμα προφίλ χρήστη, αναζητώντας πιθανά λάθη σε σημασιολογικό επίπεδο.

Create your account

Full name

Obie Fernandez

Email

obiefenandez@gmail.com



Did you mean obiefenandez@gmail.com? [Yes, update.](#)

Country ⓘ

 United States



Password

.....



✔ Nice work. This is an excellent password.

Σχήμα 9. Μπορείτε να εντοπίσετε τη σημασιολογική επικύρωση που συμβαίνει;

Προοδευτική Αποκάλυψη

Το GenUI μπορεί να προσδιορίσει ευφυνώς ποια πεδία φόρμας είναι απαραίτητα με βάση το πλαίσιο του χρήστη και να αποκαλύπτει σταδιακά επιπλέον πεδία όταν χρειάζεται. Αυτή η τεχνική προοδευτικής αποκάλυψης βοηθά στη μείωση του γνωστικού φορτίου και κάνει τη διαδικασία συμπλήρωσης φόρμας πιο διαχειρίσιμη.

Για παράδειγμα, εάν ένας χρήστης εγγράφεται για μια βασική συνδρομή, η φόρμα μπορεί αρχικά να παρουσιάζει μόνο τα απαραίτητα πεδία, και καθώς ο χρήστης προχωρά ή επιλέγει συγκεκριμένες επιλογές, μπορούν να εισαχθούν δυναμικά επιπλέον σχετικά πεδία.

Επεξηγηματικό Κείμενο με Επίγνωση Πλαισίου

Τα επεξηγηματικά πλαίσια χρησιμοποιούνται συχνά για να παρέχουν πρόσθετες πληροφορίες ή καθοδήγηση στους χρήστες όταν περνούν το ποντίκι πάνω ή αλληλεπιδρούν με συγκεκριμένα στοιχεία. Με μια προσέγγιση “Παραγωγής Περιεχομένου βάσει Πλαισίου”, μπορείτε να δημιουργήσετε επεξηγηματικά πλαίσια που προσαρμόζονται στο πλαίσιο του χρήστη και παρέχουν σχετικές πληροφορίες. Για παράδειγμα, εάν ένας χρήστης εξερευνά μια σύνθετη λειτουργία, το επεξηγηματικό πλαίσιο μπορεί να προσφέρει εξατομικευμένες συμβουλές ή παραδείγματα με βάση τις προηγούμενες αλληλεπιδράσεις του ή το επίπεδο δεξιοτήτων του.

Το επεξηγηματικό κείμενο, όπως οδηγίες, περιγραφές ή μηνύματα βοήθειας, μπορεί να δημιουργηθεί δυναμικά με βάση το πλαίσιο του χρήστη. Αντί να παρουσιάζετε γενικές εξηγήσεις, μπορείτε να χρησιμοποιήσετε LLMs για να δημιουργήσετε κείμενο που είναι προσαρμοσμένο στις συγκεκριμένες ανάγκες ή ερωτήσεις του χρήστη. Για παράδειγμα, εάν ένας χρήστης δυσκολεύεται με ένα συγκεκριμένο βήμα σε μια διαδικασία, το επεξηγηματικό κείμενο μπορεί να παρέχει εξατομικευμένη καθοδήγηση ή συμβουλές αντιμετώπισης προβλημάτων.

Το μικροκείμενο αναφέρεται στα μικρά κομμάτια κειμένου που καθοδηγούν τους χρήστες μέσα στην εφαρμογή σας, όπως ετικέτες κουμπιών, μηνύματα σφάλματος ή προτροπές επιβεβαίωσης. Εφαρμόζοντας την προσέγγιση [Παραγωγής Περιεχομένου βάσει Πλαισίου](#) στο μικροκείμενο, μπορείτε να δημιουργήσετε μια προσαρμοστική διεπαφή χρήστη που ανταποκρίνεται στις ενέργειες του χρήστη και παρέχει σχετικό και χρήσιμο κείμενο. Για παράδειγμα, εάν ένας χρήστης πρόκειται να εκτελέσει μια

κρίσιμη ενέργεια, η προτροπή επιβεβαίωσης μπορεί να δημιουργηθεί δυναμικά για να παρέχει ένα σαφές και εξατομικευμένο μήνυμα.

Το εξατομικευμένο επεξηγηματικό κείμενο και τα επεξηγηματικά πλαίσια μπορούν να βελτιώσουν σημαντικά τη διαδικασία εκμάθησης για νέους χρήστες. Παρέχοντας καθοδήγηση και παραδείγματα ειδικά για το πλαίσιο, μπορείτε να βοηθήσετε τους χρήστες να κατανοήσουν και να πλοηγηθούν γρήγορα στην εφαρμογή, μειώνοντας την καμπύλη εκμάθησης και αυξάνοντας την υιοθέτηση.

Τα δυναμικά στοιχεία διεπαφής με επίγνωση πλαισίου μπορούν επίσης να κάνουν την εφαρμογή να φαίνεται πιο διαισθητική και ελκυστική. Οι χρήστες είναι πιο πιθανό να αλληλεπιδράσουν και να εξερευνήσουν λειτουργίες όταν το συνοδευτικό κείμενο είναι προσαρμοσμένο στις συγκεκριμένες ανάγκες και τα ενδιαφέροντά τους.

Μέχρι στιγμής έχουμε καλύψει ιδέες για την ενίσχυση των υπάρχόντων προτύπων διεπαφής χρήστη με τεχνητή νοημοσύνη, αλλά τι γίνεται με την επανεξέταση του τρόπου σχεδιασμού και υλοποίησης των διεπαφών χρήστη με έναν πιο ριζοσπαστικό τρόπο;

Ορισμός της Γεννητικής Διεπαφής Χρήστη

Σε αντίθεση με τον παραδοσιακό σχεδιασμό διεπαφών χρήστη, όπου οι σχεδιαστές δημιουργούν σταθερές, στατικές διεπαφές, η GenUI υποδηλώνει ένα μέλλον στο οποίο το λογισμικό μας διαθέτει ευέλικτες, εξατομικευμένες εμπειρίες που μπορούν να εξελίσσονται και να προσαρμόζονται σε πραγματικό χρόνο. Κάθε φορά που χρησιμοποιούμε μια διεπαφή συνομιλίας που βασίζεται στην τεχνητή νοημοσύνη, επιτρέπουμε στην TN να προσαρμόζεται στις συγκεκριμένες ανάγκες του χρήστη. Η GenUI προχωράει ένα βήμα παραπέρα εφαρμόζοντας αυτό το επίπεδο προσαρμοστικότητας στην *οπτική* διεπαφή του λογισμικού.

Ο λόγος που είναι δυνατό να πειραματιστούμε με ιδέες GenUI σήμερα είναι ότι τα μεγάλα γλωσσικά μοντέλα ήδη κατανοούν τον προγραμματισμό και η βασική τους γνώση περιλαμβάνει τεχνολογίες και πλαίσια διεπαφών χρήστη. Το ερώτημα είναι επομένως εάν τα μεγάλα γλωσσικά μοντέλα μπορούν να χρησιμοποιηθούν για τη δημιουργία στοιχείων διεπαφής χρήστη, όπως κείμενο, εικόνες, διατάξεις και ακόμη και ολόκληρες διεπαφές, προσαρμοσμένες σε κάθε μεμονωμένο χρήστη. Το μοντέλο θα μπορούσε να καθοδηγηθεί ώστε να λαμβάνει υπόψη διάφορους παράγοντες, όπως τις προηγούμενες αλληλεπιδράσεις του χρήστη, τις δηλωμένες προτιμήσεις, τα δημογραφικά στοιχεία και το τρέχον πλαίσιο χρήσης, για να δημιουργήσει εξαιρετικά εξατομικευμένες και σχετικές διεπαφές.

Η GenUI διαφέρει από τον παραδοσιακό σχεδιασμό διεπαφής χρήστη σε αρκετά βασικά σημεία:

1. **Δυναμική και Προσαρμοστική:** Ο παραδοσιακός σχεδιασμός διεπαφής χρήστη περιλαμβάνει τη δημιουργία σταθερών, στατικών διεπαφών που παραμένουν ίδιες για όλους τους χρήστες. Αντίθετα, η GenUI επιτρέπει διεπαφές που μπορούν να προσαρμόζονται δυναμικά και να αλλάζουν με βάση τις ανάγκες και το πλαίσιο του χρήστη. Αυτό σημαίνει ότι η ίδια εφαρμογή μπορεί να παρουσιάζει διαφορετικές διεπαφές σε διαφορετικούς χρήστες ή ακόμα και στον ίδιο χρήστη σε διαφορετικές καταστάσεις.
2. **Εξατομίκευση σε Μεγάλη Κλίμακα:** Με τον παραδοσιακό σχεδιασμό, η δημιουργία εξατομικευμένων εμπειριών για κάθε χρήστη είναι συχνά ανέφικτη λόγω του χρόνου και των πόρων που απαιτούνται. Η GenUI, από την άλλη πλευρά, επιτρέπει την εξατομίκευση σε μεγάλη κλίμακα. Αξιοποιώντας την TN, οι σχεδιαστές μπορούν να δημιουργήσουν διεπαφές που προσαρμόζονται αυτόματα στις μοναδικές ανάγκες και προτιμήσεις κάθε χρήστη, χωρίς να χρειάζεται να σχεδιάσουν και να αναπτύξουν ξεχωριστές διεπαφές για κάθε τμήμα χρηστών.
3. **Εστίαση στα Αποτελέσματα:** Ο παραδοσιακός σχεδιασμός διεπαφής χρήστη συχνά εστιάζει στη δημιουργία οπτικά ελκυστικών και λειτουργικών

διεπαφών. Ενώ αυτές οι πτυχές παραμένουν σημαντικές στη GenUI, η κύρια εστίαση μετατοπίζεται στην επίτευξη των επιθυμητών αποτελεσμάτων για τον χρήστη. Η GenUI στοχεύει στη δημιουργία διεπαφών που είναι βελτιστοποιημένες για τους συγκεκριμένους στόχους και εργασίες κάθε χρήστη, δίνοντας προτεραιότητα στη χρηστικότητα και την αποτελεσματικότητα έναντι των καθαρά αισθητικών παραμέτρων.

4. **Συνεχής Μάθηση και Βελτίωση:** Τα συστήματα GenUI μπορούν να μαθαίνουν και να βελτιώνονται συνεχώς με την πάροδο του χρόνου με βάση τις αλληλεπιδράσεις και την ανατροφοδότηση των χρηστών. Καθώς οι χρήστες αλληλεπιδρούν με τις παραγόμενες διεπαφές, τα μοντέλα TN μπορούν να συλλέγουν δεδομένα σχετικά με τη συμπεριφορά, τις προτιμήσεις και τα αποτελέσματα των χρηστών, χρησιμοποιώντας αυτές τις πληροφορίες για να βελτιώσουν και να βελτιστοποιήσουν μελλοντικές δημιουργίες διεπαφών. Αυτή η επαναληπτική διαδικασία μάθησης επιτρέπει στα συστήματα GenUI να γίνονται όλο και πιο αποτελεσματικά στην κάλυψη των αναγκών των χρηστών με την πάροδο του χρόνου.

Είναι σημαντικό να σημειωθεί ότι το GenUI δεν είναι το ίδιο με τα εργαλεία σχεδίασης που υποβοηθούνται από TN, όπως αυτά που παρέχουν προτάσεις ή αυτοματοποιούν συγκεκριμένες εργασίες σχεδίασης. Ενώ αυτά τα εργαλεία μπορούν να είναι χρήσιμα στην απλοποίηση της διαδικασίας σχεδίασης, εξακολουθούν να βασίζονται στους σχεδιαστές για τη λήψη τελικών αποφάσεων και τη δημιουργία στατικών διεπαφών. Το GenUI, αντίθετα, περιλαμβάνει την ανάληψη πιο ενεργού ρόλου από το σύστημα TN στην πραγματική δημιουργία και προσαρμογή διεπαφών με βάση τα δεδομένα και το πλαίσιο του χρήστη.

Το GenUI αντιπροσωπεύει μια σημαντική αλλαγή στον τρόπο που προσεγγίζουμε τη σχεδίαση διεπαφής χρήστη, απομακρυνόμενοι από τις λύσεις ενιαίας προσέγγισης και στρεφόμενοι προς εξαιρετικά εξατομικευμένες, προσαρμοστικές εμπειρίες. Αξιοποιώντας τη δύναμη της TN, το GenUI έχει τη δυνατότητα να

φέρει επανάσταση στον τρόπο που αλληλεπιδρούμε με ψηφιακά προϊόντα και υπηρεσίες, δημιουργώντας διεπαφές που είναι πιο διαισθητικές, ελκυστικές και αποτελεσματικές για κάθε μεμονωμένο χρήστη.

Παράδειγμα

Για να απεικονίσουμε την έννοια του GenUI, ας εξετάσουμε μια υποθετική εφαρμογή γυμναστικής που ονομάζεται “FitAI”. Αυτή η εφαρμογή στοχεύει στην παροχή εξατομικευμένων προγραμμάτων άσκησης και διατροφικών συμβουλών στους χρήστες με βάση τους ατομικούς τους στόχους, τα επίπεδα φυσικής κατάστασης και τις προτιμήσεις τους.

Σε μια παραδοσιακή προσέγγιση σχεδίασης διεπαφής χρήστη, το FitAI θα μπορούσε να έχει ένα σταθερό σύνολο οθονών και στοιχείων που είναι ίδια για όλους τους χρήστες. Ωστόσο, με το GenUI, η διεπαφή της εφαρμογής θα μπορούσε να προσαρμόζεται δυναμικά στις μοναδικές ανάγκες και το πλαίσιο κάθε χρήστη.

Αυτή η προσέγγιση είναι κάπως δύσκολο να φανταστούμε ότι θα υλοποιηθεί το 2024 και μπορεί να μην έχει καν επαρκή απόδοση επένδυσης, αλλά είναι εφικτή.

Ορίστε πώς θα μπορούσε να λειτουργήσει:

1. Αρχική Εγκατάσταση:

- Αντί για ένα τυπικό ερωτηματολόγιο, το FitAI χρησιμοποιεί μια συνομιλητική TN για να συγκεντρώσει πληροφορίες σχετικά με τους στόχους του χρήστη, το τρέχον επίπεδο φυσικής κατάστασης και τις προτιμήσεις του.
- Με βάση αυτή την αρχική αλληλεπίδραση, η TN δημιουργεί μια εξατομικευμένη διάταξη πίνακα ελέγχου, τονίζοντας τα χαρακτηριστικά και τις πληροφορίες που είναι πιο σχετικές με τους στόχους του χρήστη.

- Η τρέχουσα τεχνολογία TN μπορεί να έχει στη διάθεσή της μια επιλογή από στοιχεία οθόνης για να χρησιμοποιήσει στη σύνθεση του εξατομικευμένου πίνακα ελέγχου.
- Η μελλοντική τεχνολογία TN μπορεί να αναλάβει το ρόλο ενός έμπειρου σχεδιαστή διεπαφής χρήστη και να δημιουργήσει πραγματικά τον πίνακα ελέγχου *από το μηδέν*.

2. Σχεδιαστής Προπόνησης:

- Η διεπαφή του σχεδιαστή προπόνησης προσαρμόζεται από την TN για να ταιριάζει συγκεκριμένα με το επίπεδο εμπειρίας του χρήστη και τον διαθέσιμο εξοπλισμό.
- Για έναν αρχάριο χωρίς εξοπλισμό, μπορεί να δείχνει απλές ασκήσεις με το βάρος του σώματος με λεπτομερείς οδηγίες και βίντεο.
- Για έναν προχωρημένο χρήστη με πρόσβαση σε γυμναστήριο, θα μπορούσε να εμφανίζει πιο σύνθετες ρουτίνες με λιγότερο επεξηγηματικό περιεχόμενο.
- Το περιεχόμενο του σχεδιαστή προπόνησης δεν φιλτράρεται απλώς από ένα μεγάλο υπερσύνολο. Μπορεί να δημιουργηθεί *άμεσα* με βάση μια βάση γνώσεων που ερωτάται με πλαίσιο που περιλαμβάνει όλα όσα είναι γνωστά για τον χρήστη.

3. Παρακολούθηση Προόδου:

- Η διεπαφή παρακολούθησης προόδου εξελίσσεται με βάση τους στόχους και τα πρότυπα δραστηριότητας του χρήστη.
- Εάν ένας χρήστης εστιάζει κυρίως στην απώλεια βάρους, η διεπαφή μπορεί να προβάλλει εμφανώς ένα γράφημα τάσης βάρους και στατιστικά καύσης θερμίδων.
- Για έναν χρήστη που χτίζει μυϊκή μάζα, θα μπορούσε να τονίσει τα κέρδη δύναμης και τις αλλαγές στη σύσταση σώματος.

- Η τεχνητή νοημοσύνη μπορεί να προσαρμόσει αυτό το μέρος της εφαρμογής στην πραγματική πρόοδο του χρήστη. Εάν η πρόοδος σταματήσει για κάποιο χρονικό διάστημα, η εφαρμογή μπορεί να μεταβεί σε μια κατάσταση όπου προσπαθεί να παροτρύνει τον χρήστη να αποκαλύψει τους λόγους της οπισθοδρόμησης, ώστε να τους μετριάσει.

4. Διατροφικές Συμβουλές:

- Το τμήμα διατροφής προσαρμόζεται στις διατροφικές προτιμήσεις και περιορισμούς του χρήστη.
- Για έναν χρήστη που είναι vegan, μπορεί να εμφανίζει προτάσεις γευμάτων και πηγές πρωτεΐνης φυτικής προέλευσης.
- Για έναν χρήστη με δυσανεξία στη γλουτένη, θα φιλτράρει αυτόματα τα τρόφιμα που περιέχουν γλουτένη από τις προτάσεις.
- Και πάλι, το περιεχόμενο δεν προέρχεται από ένα τεράστιο υπερσύνολο δεδομένων γευμάτων που εφαρμόζεται σε όλους τους χρήστες, αλλά συντίθεται από μια βάση γνώσεων που περιέχει προσαρμόσιμες πληροφορίες με βάση την συγκεκριμένη κατάσταση και τους περιορισμούς του χρήστη.
- Για παράδειγμα, οι συνταγές δημιουργούνται με προδιαγραφές συστατικών που ταιριάζουν με τις συνεχώς μεταβαλλόμενες θερμιδικές ανάγκες του χρήστη καθώς εξελίσσεται το επίπεδο φυσικής κατάστασης και τα σωματικά στατιστικά του.

5. Στοιχεία Παρακίνησης:

- Το περιεχόμενο παρακίνησης και οι ειδοποιήσεις της εφαρμογής εξατομικεύονται με βάση τον τύπο προσωπικότητας του χρήστη και την ανταπόκρισή του σε διαφορετικές στρατηγικές παρακίνησης.
- Κάποιοι χρήστες μπορεί να λαμβάνουν ενθαρρυντικά μηνύματα, ενώ άλλοι λαμβάνουν περισσότερη ανατροφοδότηση βασισμένη σε δεδομένα.

Σε αυτό το παράδειγμα, το GenUI επιτρέπει στο FitAI να δημιουργήσει μια εξαιρετικά εξατομικευμένη εμπειρία για κάθε χρήστη, αυξάνοντας πιθανώς τη δέσμευση, την ικανοποίηση και την πιθανότητα επίτευξης των στόχων φυσικής κατάστασης. Τα στοιχεία διεπαφής, το περιεχόμενο, ακόμη και η “προσωπικότητα” της εφαρμογής προσαρμόζονται για να εξυπηρετούν καλύτερα τις ανάγκες και τις προτιμήσεις κάθε μεμονωμένου χρήστη.

Η Μετάβαση στον Σχεδιασμό με Προσανατολισμό στο Αποτέλεσμα

Το GenUI αντιπροσωπεύει μια θεμελιώδη αλλαγή στην προσέγγιση του σχεδιασμού διεπαφής χρήστη!, μεταβαίνοντας από την εστίαση στη δημιουργία συγκεκριμένων στοιχείων διεπαφής σε μια πιο ολιστική προσέγγιση προσανατολισμένη στο αποτέλεσμα. Αυτή η μετάβαση έχει αρκετές σημαντικές επιπτώσεις:

1. Εστίαση στους Στόχους του Χρήστη:

- Οι σχεδιαστές θα χρειαστεί να σκέφτονται πιο βαθιά σχετικά με τους στόχους του χρήστη και τα επιθυμητά αποτελέσματα αντί για συγκεκριμένα στοιχεία διεπαφής.
- Η έμφαση θα δοθεί στη δημιουργία συστημάτων που μπορούν να παράγουν διεπαφές που βοηθούν τους χρήστες να επιτύχουν τους στόχους τους αποτελεσματικά και αποδοτικά.
- Θα αναδυθούν νέα πλαίσια διεπαφής χρήστη που θα δίνουν στους σχεδιαστές βασισμένους σε τεχνητή νοημοσύνη τα εργαλεία που χρειάζονται για να μπορούν να δημιουργούν εμπειρίες χρήστη *εν κινήσει* και *από το μηδέν* αντί να βασίζονται σε προκαθορισμένες προδιαγραφές οθόνης.

2. Μεταβαλλόμενος Ρόλος των Σχεδιαστών:

- Οι σχεδιαστές θα μεταβούν από τη δημιουργία σταθερών διατάξεων στον καθορισμό κανόνων, περιορισμών και κατευθυντήριων γραμμών που θα ακολουθούν τα συστήματα AI κατά τη δημιουργία διεπαφών.
- Θα χρειαστεί να αναπτύξουν δεξιότητες σε τομείς όπως η ανάλυση δεδομένων, η μηχανική καθοδήγησης AI και η συστημική σκέψη για την αποτελεσματική καθοδήγηση των συστημάτων GenUI.

3. Σημασία της Έρευνας Χρηστών:

- Η έρευνα χρηστών γίνεται ακόμη πιο κρίσιμη στο πλαίσιο του GenUI, καθώς οι σχεδιαστές πρέπει να κατανοούν όχι μόνο τις προτιμήσεις των χρηστών, αλλά και πώς αυτές οι προτιμήσεις και ανάγκες αλλάζουν σε διαφορετικά πλαίσια.
- Η συνεχής δοκιμή χρηστών και οι κύκλοι ανατροφοδότησης θα είναι απαραίτητοι για τη βελτίωση της ικανότητας της AI να δημιουργεί αποτελεσματικές διεπαφές.

4. Σχεδιασμός για Μεταβλητότητα:

- Αντί να δημιουργούν μια μοναδική “τέλεια” διεπαφή, οι σχεδιαστές θα πρέπει να εξετάζουν πολλαπλές πιθανές παραλλαγές και να διασφαλίζουν ότι το σύστημα μπορεί να δημιουργεί κατάλληλες διεπαφές για διαφορετικές ανάγκες χρηστών.
- Αυτό περιλαμβάνει το σχεδιασμό για ακραίες περιπτώσεις και τη διασφάλιση ότι οι παραγόμενες διεπαφές διατηρούν τη χρηστικότητα και την προσβασιμότητα σε διαφορετικές διαμορφώσεις.
- Η διαφοροποίηση προϊόντων αποκτά νέες διαστάσεις που περιλαμβάνουν διαφορετικές προοπτικές στην ψυχολογία χρήστη και την αξιοποίηση μοναδικών συνόλων δεδομένων και βάσεων γνώσεων που δεν είναι διαθέσιμες στους ανταγωνιστές.

Προκλήσεις και Ζητήματα προς Εξέταση

Ενώ το GenUI προσφέρει συναρπαστικές δυνατότητες, παρουσιάζει επίσης διάφορες προκλήσεις και ζητήματα προς εξέταση:

1. Τεχνικοί Περιορισμοί:

- Η τρέχουσα τεχνολογία AI, αν και προηγμένη, εξακολουθεί να έχει περιορισμούς στην κατανόηση σύνθετων προθέσεων χρηστών και στη δημιουργία πραγματικά προσαρμοσμένων διεπαφών στο περιβάλλον.
- Ζητήματα απόδοσης σχετικά με την παραγωγή στοιχείων διεπαφής σε πραγματικό χρόνο, ειδικά σε λιγότερο ισχυρές συσκευές.

2. Απαιτήσεις Δεδομένων:

- Ανάλογα με την περίπτωση χρήσης, τα αποτελεσματικά συστήματα GenUI ενδέχεται να απαιτούν σημαντικές ποσότητες δεδομένων χρηστών για τη δημιουργία εξατομικευμένων διεπαφών.
- Οι προκλήσεις στην ηθική συλλογή αυθεντικών δεδομένων χρηστών εγείρουν ανησυχίες σχετικά με την προστασία και την ασφάλεια των δεδομένων, καθώς και πιθανές προκαταλήψεις στα δεδομένα που χρησιμοποιούνται για την εκπαίδευση μοντέλων GenUI.

3. Χρηστικότητα και Συνέπεια:

- Τουλάχιστον μέχρι η πρακτική να γίνει ευρέως διαδεδομένη, μια εφαρμογή με συνεχώς μεταβαλλόμενες διεπαφές θα μπορούσε να οδηγήσει σε προβλήματα χρηστικότητας, καθώς οι χρήστες ενδέχεται να δυσκολεύονται να βρουν οικεία στοιχεία ή να πλοηγηθούν αποτελεσματικά.
- Η εξισορρόπηση μεταξύ εξατομίκευσης και διατήρησης μιας συνεπούς, εύκολα κατανοητής διεπαφής θα είναι κρίσιμη.

4. Υπερβολική Εξάρτηση από την ΑΙ:

- Υπάρχει κίνδυνος υπερβολικής ανάθεσης σχεδιαστικών αποφάσεων σε συστήματα ΑΙ, που θα μπορούσε ενδεχομένως να οδηγήσει σε μη εμπνευσμένες, προβληματικές ή απλώς ελαττωματικές επιλογές διεπαφής.
- Η ανθρώπινη εποπτεία και η δυνατότητα παράκαμψης των σχεδίων που παράγονται από την ΑΙ θα παραμείνουν σημαντικές στο προβλέψιμο μέλλον.

5. Ζητήματα Προσβασιμότητας:

- Η διασφάλιση ότι οι δυναμικά παραγόμενες διεπαφές παραμένουν προσβάσιμες σε χρήστες με αναπηρίες παρουσιάζει εντελώς νέες προκλήσεις, γεγονός που προκαλεί ανησυχία δεδομένου του χαμηλού επιπέδου συμμόρφωσης προσβασιμότητας που επιδεικνύουν τα τυπικά συστήματα.
- Από την άλλη πλευρά, οι σχεδιαστές τεχνητής νοημοσύνης μπορεί να υλοποιηθούν με *ενσωματωμένη* μέριμνα για την προσβασιμότητα και δυνατότητες για τη δημιουργία προσβάσιμων διεπαφών άμεσα, όπως ακριβώς δημιουργούν διεπαφές χρήστη για μη-εμποδιζόμενους χρήστες.
- Σε κάθε περίπτωση, τα συστήματα GenUI θα πρέπει να σχεδιάζονται με ισχυρές κατευθυντήριες γραμμές προσβασιμότητας και διαδικασίες ελέγχου.

6. Εμπιστοσύνη και Διαφάνεια Χρηστών:

- Οι χρήστες μπορεί να αισθάνονται άβολα με διεπαφές που φαίνεται να “γνωρίζουν υπερβολικά πολλά” για αυτούς ή αλλάζουν με τρόπους που δεν κατανοούν.
- Η παροχή διαφάνειας σχετικά με το πώς και γιατί εξατομικεύονται οι διεπαφές θα είναι σημαντική για την οικοδόμηση της εμπιστοσύνης των χρηστών.

Μελλοντικές Προοπτικές και Ευκαιρίες

Το μέλλον της Παραγωγικής Διεπαφής Χρήστη (GenUI) υπόσχεται τεράστιες δυνατότητες για την επανάσταση στον τρόπο με τον οποίο αλληλεπιδρούμε με ψηφιακά προϊόντα και υπηρεσίες. Καθώς αυτή η τεχνολογία συνεχίζει να εξελίσσεται, μπορούμε να προβλέψουμε μια σεισμική αλλαγή στον τρόπο με τον οποίο σχεδιάζονται, υλοποιούνται και βιώνονται οι διεπαφές χρήστη. Πιστεύω ότι το GenUI είναι το φαινόμενο που τελικά θα ωθήσει το λογισμικό μας στη σφαίρα αυτού που τώρα θεωρείται επιστημονική φαντασία.

Μία από τις πιο συναρπαστικές προοπτικές του GenUI είναι η δυνατότητά του να ενισχύσει την προσβασιμότητα σε μια κλίμακα που ξεπερνά την απλή διασφάλιση ότι τα άτομα με σοβαρές αναπηρίες δεν αποκλείονται εντελώς από τη χρήση του λογισμικού σας. Προσαρμόζοντας αυτόματα τις διεπαφές στις ατομικές ανάγκες των χρηστών, το GenUI θα μπορούσε να κάνει τις ψηφιακές εμπειρίες πιο συμπεριληπτικές από ποτέ. Φανταστείτε διεπαφές που προσαρμόζονται αβίαστα για να παρέχουν μεγαλύτερο κείμενο για νεότερους ή οπτικά εμποδιζόμενους χρήστες ή απλοποιημένες διατάξεις για άτομα με γνωστικές αναπηρίες, όλα χωρίς να απαιτείται χειροκίνητη διαμόρφωση ή ξεχωριστές “προσβάσιμες” εκδόσεις εφαρμογών.

Οι δυνατότητες εξατομίκευσης του GenUI είναι πιθανό να οδηγήσουν σε αυξημένη εμπλοκή των χρηστών, ικανοποίηση και πιστότητα σε ένα ευρύ φάσμα ψηφιακών προϊόντων. Καθώς οι διεπαφές γίνονται πιο συντονισμένες με τις ατομικές προτιμήσεις και συμπεριφορές, οι χρήστες θα βρίσκουν τις ψηφιακές εμπειρίες πιο διαισθητικές και ευχάριστες, οδηγώντας πιθανώς σε βαθύτερες και πιο ουσιαστικές αλληλεπιδράσεις με την τεχνολογία.

Το GenUI έχει επίσης τη δυνατότητα να μεταμορφώσει τη διαδικασία ένταξης νέων χρηστών. Δημιουργώντας διαισθητικές, εξατομικευμένες εμπειρίες πρώτης χρήσης που προσαρμόζονται γρήγορα στο επίπεδο εμπειρίας κάθε χρήστη, το GenUI θα

μπορούσε να μειώσει σημαντικά την καμπύλη εκμάθησης που σχετίζεται με νέες εφαρμογές. Αυτό θα μπορούσε να οδηγήσει σε ταχύτερους ρυθμούς υιοθέτησης και αυξημένη αυτοπεποίθηση των χρηστών στην εξερεύνηση νέων χαρακτηριστικών και λειτουργιών.

Μια ακόμη συναρπαστική δυνατότητα είναι η ικανότητα της Γενετικής Διεπαφής Χρήστη να διατηρεί μια συνεπή εμπειρία χρήστη σε διαφορετικές συσκευές και πλατφόρμες, βελτιστοποιώντας παράλληλα για κάθε συγκεκριμένο πλαίσιο χρήσης. Αυτό θα μπορούσε να επιλύσει τη μακροχρόνια πρόκληση της παροχής συνεκτικών εμπειριών σε ένα ολοένα και πιο κατακερματισμένο τοπίο συσκευών, από έξυπνα τηλέφωνα και ταμπλέτες μέχρι επιτραπέζιους υπολογιστές και αναδυόμενες τεχνολογίες όπως τα γυαλιά επαυξημένης πραγματικότητας.

Η φύση της Γενετικής Διεπαφής Χρήστη που βασίζεται στα δεδομένα ανοίγει ευκαιρίες για γρήγορη επανάληψη και βελτίωση στο σχεδιασμό διεπαφών χρήστη. Συλλέγοντας δεδομένα σε πραγματικό χρόνο για το πώς οι χρήστες αλληλεπιδρούν με τις παραγόμενες διεπαφές, οι σχεδιαστές και οι προγραμματιστές μπορούν να αποκτήσουν πρωτοφανείς γνώσεις σχετικά με τη συμπεριφορά και τις προτιμήσεις των χρηστών. Αυτός ο κύκλος ανατροφοδότησης θα μπορούσε να οδηγήσει σε συνεχείς βελτιώσεις στο σχεδιασμό διεπαφών χρήστη, καθοδηγούμενες από πραγματικά πρότυπα χρήσης αντί για υποθέσεις ή περιορισμένες δοκιμές χρηστών.

Για να προετοιμαστούν για αυτή τη μετάβαση, οι σχεδιαστές θα πρέπει να εξελίσσουν τις δεξιότητες και τη νοοτροπία τους. Η εστίαση θα μετατοπιστεί από τη δημιουργία σταθερών διατάξεων στην ανάπτυξη ολοκληρωμένων συστημάτων σχεδίασης και κατευθυντήριων γραμμών που μπορούν να καθοδηγήσουν τη δημιουργία διεπαφών με τη βοήθεια της τεχνητής νοημοσύνης. Οι σχεδιαστές θα πρέπει να καλλιεργήσουν μια βαθιά κατανόηση της ανάλυσης δεδομένων, των τεχνολογιών τεχνητής νοημοσύνης και της συστημικής σκέψης για να καθοδηγούν αποτελεσματικά τα συστήματα Γενετικής Διεπαφής Χρήστη.

Επιπλέον, καθώς η Γενετική Διεπαφή Χρήστη θολώνει τα όρια μεταξύ σχεδιασμού

και τεχνολογίας, οι σχεδιαστές θα πρέπει να συνεργάζονται στενότερα με προγραμματιστές και επιστήμονες δεδομένων. Αυτή η διεπιστημονική προσέγγιση θα είναι καθοριστική στη δημιουργία συστημάτων Γενετικής Διεπαφής Χρήστη που δεν είναι μόνο οπτικά ελκυστικά και φιλικά προς το χρήστη, αλλά και τεχνικά εύρωστα και ηθικά ορθά.

Οι ηθικές επιπτώσεις της Γενετικής Διεπαφής Χρήστη θα έρθουν επίσης στο προσκήνιο καθώς η τεχνολογία ωριμάζει. Οι σχεδιαστές θα παίξουν καθοριστικό ρόλο στην ανάπτυξη πλαισίων για την υπεύθυνη χρήση της τεχνητής νοημοσύνης στο σχεδιασμό διεπαφών, διασφαλίζοντας ότι η εξατομίκευση βελτιώνει τις εμπειρίες των χρηστών χωρίς να διακυβεύει την ιδιωτικότητα ή να χειραγωγεί τη συμπεριφορά των χρηστών με ανήθικους τρόπους.

Καθώς κοιτάζουμε προς το μέλλον, η Γενετική Διεπαφή Χρήστη παρουσιάζει τόσο συναρπαστικές ευκαιρίες όσο και σημαντικές προκλήσεις. Έχει τη δυνατότητα να δημιουργήσει πιο διαισθητικές, αποτελεσματικές και ικανοποιητικές ψηφιακές εμπειρίες για τους χρήστες σε όλο τον κόσμο. Παρόλο που θα απαιτήσει από τους σχεδιαστές να προσαρμοστούν και να αποκτήσουν νέες δεξιότητες, προσφέρει επίσης μια πρωτοφανή ευκαιρία να διαμορφώσουν το μέλλον της διάδρασης ανθρώπου-υπολογιστή με βαθύ και ουσιαστικό τρόπο. Το ταξίδι προς τα πλήρως υλοποιημένα συστήματα Γενετικής Διεπαφής Χρήστη θα είναι αναμφίβολα πολύπλοκο, αλλά τα πιθανά οφέλη όσον αφορά τη βελτίωση των εμπειριών χρήστη και της ψηφιακής προσβασιμότητας το καθιστούν ένα μέλλον για το οποίο αξίζει να προσπαθήσουμε.

Ευφυής Ενορχήστρωση Ροών Εργασίας



Στον τομέα της ανάπτυξης εφαρμογών, οι ροές εργασίας παίζουν καθοριστικό ρόλο στον ορισμό του τρόπου με τον οποίο δομούνται και εκτελούνται οι εργασίες, οι διαδικασίες και οι αλληλεπιδράσεις χρηστών. Καθώς οι εφαρμογές γίνονται πιο περίπλοκες και οι προσδοκίες των χρηστών συνεχίζουν να αυξάνονται, η ανάγκη για ευφυή και προσαρμοστική ενορχήστρωση ροών εργασίας γίνεται όλο και πιο εμφανής.

Η προσέγγιση της “Ευφυούς Ενορχήστρωσης Ροών Εργασίας” επικεντρώνεται στην αξιοποίηση των στοιχείων τεχνητής νοημοσύνης για τη δυναμική ενορχήστρωση και βελτιστοποίηση σύνθετων ροών εργασίας εντός των εφαρμογών. Στόχος είναι η δημιουργία εφαρμογών που είναι πιο αποδοτικές, ανταποκρίσιμες και προσαρμόσιμες σε δεδομένα και περιεχόμενο πραγματικού χρόνου.

Σε αυτό το κεφάλαιο, θα εξερευνήσουμε τις βασικές αρχές και τα πρότυπα που στηρίζουν την προσέγγιση της ευφυούς ενορχήστρωσης ροών εργασίας. Θα εξετάσουμε πώς η τεχνητή νοημοσύνη μπορεί να χρησιμοποιηθεί για την ευφυή δρομολόγηση εργασιών, την αυτοματοποίηση λήψης αποφάσεων και τη δυναμική προσαρμογή ροών εργασίας με βάση διάφορους παράγοντες όπως η συμπεριφορά χρήστη, η απόδοση του συστήματος και οι επιχειρησιακοί κανόνες. Μέσα από πρακτικά παραδείγματα και σενάρια του πραγματικού κόσμου, θα καταδείξουμε το μεταμορφωτικό δυναμικό της τεχνητής νοημοσύνης στην απλοποίηση και βελτιστοποίηση των ροών εργασίας των εφαρμογών.

Είτε αναπτύσσετε εταιρικές εφαρμογές με περίπλοκες επιχειρηματικές διαδικασίες είτε εφαρμογές που απευθύνονται σε καταναλωτές με δυναμικές διαδρομές χρήστη, τα πρότυπα και οι τεχνικές που συζητούνται σε αυτό το κεφάλαιο θα σας εφοδιάσουν με τη γνώση και τα εργαλεία για τη δημιουργία ευφών και αποδοτικών ροών εργασίας που βελτιώνουν τη συνολική εμπειρία χρήστη και προωθούν την επιχειρηματική αξία.

Επιχειρηματική Ανάγκη

Οι παραδοσιακές προσεγγίσεις στη διαχείριση ροών εργασίας συχνά βασίζονται σε προκαθορισμένους κανόνες και στατικά δέντρα αποφάσεων, τα οποία μπορεί να είναι άκαμπτα, ανελαστικά και ανίκανα να αντιμετωπίσουν τη δυναμική φύση των σύγχρονων εφαρμογών.

Σκεφτείτε ένα σενάριο όπου μια εφαρμογή ηλεκτρονικού εμπορίου πρέπει να διαχειριστεί μια περίπλοκη διαδικασία εκπλήρωσης παραγγελιών. Η ροή εργασίας μπορεί να περιλαμβάνει πολλαπλά βήματα όπως επικύρωση παραγγελίας, έλεγχο αποθέματος, επεξεργασία πληρωμών, αποστολή και ειδοποιήσεις πελατών. Κάθε βήμα μπορεί να έχει το δικό του σύνολο κανόνων, εξαρτήσεων, εξωτερικών ενσωματώσεων και μηχανισμών χειρισμού εξαιρέσεων. Η διαχείριση μιας τέτοιας

ροής εργασίας χειροκίνητα ή μέσω σκληρά κωδικοποιημένης λογικής μπορεί γρήγορα να γίνει δύσχρηστη, επιρρεπής σε σφάλματα και δύσκολη στη συντήρηση.

Επιπλέον, καθώς η εφαρμογή κλιμακώνεται και ο αριθμός των ταυτόχρονων χρηστών αυξάνεται, η ροή εργασίας ενδέχεται να χρειαστεί να προσαρμοστεί και να βελτιστοποιηθεί με βάση τα δεδομένα πραγματικού χρόνου και την απόδοση του συστήματος. Για παράδειγμα, κατά τη διάρκεια περιόδων αιχμής της κίνησης, η εφαρμογή μπορεί να χρειαστεί να προσαρμόσει δυναμικά τη ροή εργασίας για να δώσει προτεραιότητα σε συγκεκριμένες εργασίες, να καταναίμει αποτελεσματικά τους πόρους και να διασφαλίσει μια ομαλή εμπειρία χρήστη.

Εδώ είναι που εφαρμόζεται η προσέγγιση “Έξυπνης Ενορχήστρωσης Ροής Εργασίας”. Αξιοποιώντας τα στοιχεία TN, οι προγραμματιστές μπορούν να δημιουργήσουν ροές εργασίας που είναι έξυπνες, προσαρμοστικές και αυτοβελτιστοποιούμενες. Η TN μπορεί να αναλύσει τεράστιες ποσότητες δεδομένων, να μάθει από προηγούμενες εμπειρίες και να λάβει τεκμηριωμένες αποφάσεις σε πραγματικό χρόνο για την αποτελεσματική ενορχήστρωση της ροής εργασίας.

Βασικά Οφέλη

- 1. Αυξημένη Αποδοτικότητα:** Η TN μπορεί να βελτιστοποιήσει την κατανομή εργασιών, την αξιοποίηση πόρων και την εκτέλεση της ροής εργασίας, οδηγώντας σε ταχύτερους χρόνους επεξεργασίας και βελτιωμένη συνολική αποδοτικότητα.
- 2. Προσαρμοστικότητα:** Οι ροές εργασίας που καθοδηγούνται από TN μπορούν να προσαρμόζονται δυναμικά στις μεταβαλλόμενες συνθήκες, όπως διακυμάνσεις στη ζήτηση χρηστών, την απόδοση του συστήματος ή τις επιχειρηματικές απαιτήσεις, διασφαλίζοντας ότι η εφαρμογή παραμένει ανταποκρίσιμη και ανθεκτική.
- 3. Αυτοματοποιημένη Λήψη Αποφάσεων:** Η TN μπορεί να αυτοματοποιήσει πολύπλοκες διαδικασίες λήψης αποφάσεων εντός της ροής εργασίας,

μειώνοντας τη χειροκίνητη παρέμβαση και ελαχιστοποιώντας τον κίνδυνο ανθρώπινων λαθών.

4. **Εξατομίκευση:** Η ΤΝ μπορεί να αναλύσει τη συμπεριφορά, τις προτιμήσεις και το πλαίσιο του χρήστη για να εξατομικεύσει τη ροή εργασίας και να προσφέρει προσαρμοσμένες εμπειρίες σε μεμονωμένους χρήστες.
5. **Κλιμάκωση:** Οι ροές εργασίας που τροφοδοτούνται από ΤΝ μπορούν να κλιμακώνονται απρόσκοπτα για να χειριστούν αυξανόμενους όγκους δεδομένων και αλληλεπιδράσεων χρηστών, χωρίς να διακυβεύεται η απόδοση ή η αξιοπιστία.

Στις επόμενες ενότητες, θα εξερευνήσουμε τα βασικά πρότυπα και τις τεχνικές που επιτρέπουν την υλοποίηση έξυπνων ροών εργασίας και θα παρουσιάσουμε παραδείγματα από τον πραγματικό κόσμο για το πώς η ΤΝ μεταμορφώνει τη διαχείριση ροών εργασίας σε σύγχρονες εφαρμογές.

Βασικά Πρότυπα

Για την υλοποίηση της έξυπνης ενορχήστρωσης ροής εργασίας σε εφαρμογές, οι προγραμματιστές μπορούν να αξιοποιήσουν διάφορα βασικά πρότυπα που αξιοποιούν τη δύναμη της ΤΝ. Αυτά τα πρότυπα παρέχουν μια δομημένη προσέγγιση στο σχεδιασμό και τη διαχείριση ροών εργασίας, επιτρέποντας στις εφαρμογές να προσαρμόζονται, να βελτιστοποιούνται και να αυτοματοποιούν διαδικασίες με βάση τα δεδομένα πραγματικού χρόνου και το πλαίσιο. Ας εξερευνήσουμε μερικά από τα θεμελιώδη πρότυπα στην έξυπνη ενορχήστρωση ροής εργασίας.

Δυναμική Δρομολόγηση Εργασιών

Αυτό το πρότυπο περιλαμβάνει τη χρήση ΤΝ για την έξυπνη δρομολόγηση εργασιών εντός μιας ροής εργασίας με βάση διάφορους παράγοντες όπως η προτεραιότητα

εργασίας, η διαθεσιμότητα πόρων και η απόδοση του συστήματος. Οι αλγόριθμοι ΤΝ μπορούν να αναλύσουν τα χαρακτηριστικά κάθε εργασίας, να λάβουν υπόψη την τρέχουσα κατάσταση του συστήματος και να λάβουν τεκμηριωμένες αποφάσεις για την ανάθεση εργασιών στους καταλληλότερους πόρους ή διαδρομές επεξεργασίας. Η δυναμική δρομολόγηση εργασιών διασφαλίζει ότι οι εργασίες κατανέμονται και εκτελούνται αποτελεσματικά, βελτιστοποιώντας τη συνολική απόδοση της ροής εργασίας.

```
1 class TaskRouter
2     include Raix::ChatCompletion
3     include Raix::FunctionDispatch
4
5     attr_accessor :task
6
7     # list of functions that can be called by the AI entirely at its
8     # discretion depending on the task received
9
10    function :analyze_task_priority do
11        TaskPriorityAnalyzer.perform(task)
12    end
13
14    function :check_resource_availability, # ...
15    function :assess_system_performance, # ...
16    function :assign_task_to_resource, # ...
17
18    DIRECTIVE = "You are a task router, responsible for intelligently
19        assigning tasks to available resources based on priority, resource
20        availability, and system performance..."
21
22    def initialize(task)
23        self.task = task
24        transcript << { system: DIRECTIVE }
25        transcript << { user: task.to_json }
26    end
27
28    def perform
29        while task.unassigned?
30            chat_completion
31        end
32    end
33 end
```

```
32     # todo: add max loop counter and break
33     end
34
35     # capture the transcript for later analysis
36     task.update(routing_transcript: transcript)
37     end
38 end
```

Σημειώστε τον βρόχο που δημιουργείται από την έκφραση `while` στη γραμμή 29, η οποία συνεχίζει να προτρέπει την TN μέχρι να ανατεθεί η εργασία. Στη γραμμή 35, αποθηκεύουμε το αντίγραφο της εργασίας για μεταγενέστερη ανάλυση και αποσφαλμάτωση, εάν καταστεί απαραίτητο.

Λήψη Αποφάσεων Βάσει Περιεχομένου

Μπορείτε να χρησιμοποιήσετε παρόμοιο κώδικα για να λάβετε αποφάσεις με επίγνωση του περιεχομένου εντός μιας ροής εργασίας. Αναλύοντας σχετικά δεδομένα όπως τις προτιμήσεις χρηστών, τα ιστορικά μοτίβα και τις εισόδους πραγματικού χρόνου, τα στοιχεία TN μπορούν να καθορίσουν την καταλληλότερη πορεία δράσης σε κάθε σημείο απόφασης στη ροή εργασίας. Προσαρμόστε τη συμπεριφορά της ροής εργασίας σας με βάση το συγκεκριμένο πλαίσιο κάθε χρήστη ή σεναρίου, παρέχοντας εξατομικευμένες και βελτιστοποιημένες εμπειρίες.

Προσαρμοστική Σύνθεση Ροής Εργασίας

Αυτό το μοτίβο επικεντρώνεται στη δυναμική σύνθεση και προσαρμογή των ροών εργασίας με βάση τις μεταβαλλόμενες απαιτήσεις ή συνθήκες. Η TN μπορεί να αναλύσει την τρέχουσα κατάσταση της ροής εργασίας, να εντοπίσει σημεία συμφόρησης ή ανεπάρκειες, και να τροποποιήσει αυτόματα τη δομή της ροής εργασίας για βελτιστοποίηση της απόδοσης. Η προσαρμοστική σύνθεση ροής εργασίας επιτρέπει στις εφαρμογές να εξελίσσονται και να βελτιώνουν συνεχώς τις διαδικασίες τους χωρίς να απαιτείται χειροκίνητη παρέμβαση.

Χειρισμός και Ανάκαμψη από Εξαιρέσεις

Ο χειρισμός και η ανάκαμψη από εξαιρέσεις είναι κρίσιμες πτυχές της ευφυούς ενορχήστρωσης ροής εργασίας. Όταν εργάζεστε με στοιχεία TN και πολύπλοκες ροές εργασίας, είναι απαραίτητο να προβλέπετε και να χειρίζεστε τις εξαιρέσεις με κομψό τρόπο για να διασφαλίσετε τη σταθερότητα και την αξιοπιστία του συστήματος.

Ακολουθούν ορισμένα βασικά σημεία και τεχνικές για τον χειρισμό και την ανάκαμψη από εξαιρέσεις σε ευφυείς ροές εργασίας:

1. **Διάδοση Εξαιρέσεων:** Εφαρμόστε μια συνεπή προσέγγιση για τη διάδοση εξαιρέσεων σε όλα τα στοιχεία της ροής εργασίας. Όταν προκύπτει μια εξαίρεση εντός ενός στοιχείου, θα πρέπει να συλλαμβάνεται, να καταγράφεται και να διαδίδεται στον ενορχηστρωτή ή σε ένα διακριτό στοιχείο υπεύθυνο για τον χειρισμό εξαιρέσεων. Η ιδέα είναι να κεντριοποιηθεί ο χειρισμός εξαιρέσεων και να αποτραπεί η σιωπηλή κατάποση εξαιρέσεων, καθώς και να ανοίξουν δυνατότητες για [Ευφυή Χειρισμό Σφαλμάτων](#).
2. **Μηχανισμοί Επανάληψης:** Οι μηχανισμοί επανάληψης βοηθούν στη βελτίωση της ανθεκτικότητας της ροής εργασίας και στον κομψό χειρισμό παροδικών αποτυχιών. Σίγουρα προσπαθήστε να εφαρμόσετε μηχανισμούς επανάληψης για παροδικές ή ανακτήσιμες εξαιρέσεις, όπως η συνδεσιμότητα δικτύου ή η μη διαθεσιμότητα πόρων που μπορούν να επαναληφθούν αυτόματα μετά από καθορισμένη καθυστέρηση. Έχοντας έναν ενορχηστρωτή ή χειριστή εξαιρέσεων που βασίζεται στην TN σημαίνει ότι οι στρατηγικές επανάληψής σας δεν χρειάζεται να είναι μηχανικής φύσης, βασιζόμενες σε σταθερούς αλγόριθμους όπως η εκθετική υποχώρηση. Μπορείτε να αφήσετε τον χειρισμό της επανάληψης στη “διακριτική ευχέρεια” του στοιχείου TN που είναι υπεύθυνο για την απόφαση του τρόπου χειρισμού της εξαίρεσης.
3. **Εναλλακτικές Στρατηγικές:** Εάν ένα στοιχείο AI αποτύχει να παρέχει

μια έγκυρη απόκριση ή αντιμετώπιση σφάλμα—κάτι σύνηθες δεδομένης της πρωτοποριακής φύσης του—πρέπει να υπάρχει ένας εναλλακτικός μηχανισμός για να διασφαλιστεί ότι η ροή εργασίας μπορεί να συνεχιστεί. Αυτό θα μπορούσε να περιλαμβάνει τη χρήση προεπιλεγμένων τιμών, εναλλακτικών αλγορίθμων, ή έναν **Άνθρωπο στον Βρόχο** για τη λήψη αποφάσεων και τη διατήρηση της προώθησης της ροής εργασίας.

4. **Αντισταθμιστικές Ενέργειες:** Οι οδηγίες του ενορχηστρωτή θα πρέπει να περιλαμβάνουν οδηγίες σχετικά με τις αντισταθμιστικές ενέργειες για τον χειρισμό εξαιρέσεων που δεν μπορούν να επιλυθούν αυτόματα. Οι αντισταθμιστικές ενέργειες είναι βήματα που λαμβάνονται για την αναίρεση ή τον μετριασμό των επιπτώσεων μιας αποτυχημένης λειτουργίας. Για παράδειγμα, εάν αποτύχει ένα βήμα επεξεργασίας πληρωμής, μια αντισταθμιστική ενέργεια θα μπορούσε να είναι η αναστροφή της συναλλαγής και η ειδοποίηση του χρήστη. Οι αντισταθμιστικές ενέργειες βοηθούν στη διατήρηση της συνέπειας και της ακεραιότητας των δεδομένων όταν προκύπτουν εξαιρέσεις.
5. **Παρακολούθηση και Ειδοποίηση Εξαιρέσεων:** Δημιουργήστε μηχανισμούς παρακολούθησης και ειδοποίησης για τον εντοπισμό και την ενημέρωση των σχετικών ενδιαφερόμενων μερών σχετικά με κρίσιμες εξαιρέσεις. Ο ενορχηστρωτής μπορεί να γνωρίζει τα όρια και τους κανόνες για την ενεργοποίηση ειδοποιήσεων όταν οι εξαιρέσεις υπερβαίνουν ορισμένα όρια ή όταν προκύπτουν συγκεκριμένοι τύποι εξαιρέσεων. Αυτό επιτρέπει την προληπτική αναγνώριση και επίλυση προβλημάτων πριν επηρεάσουν το συνολικό σύστημα.

Ακολουθεί ένα παράδειγμα χειρισμού εξαιρέσεων και ανάκτησης σε ένα στοιχείο ροής εργασίας Ruby:

```
1 class InventoryManager
2   def check_availability(order)
3     begin
4       # Perform inventory check logic
5       inventory = Inventory.find_by(product_id: order.product_id)
6       if inventory.available_quantity >= order.quantity
7         return true
8       else
9         raise InsufficientInventoryError,
10            "Insufficient inventory for product #{order.product_id}"
11       end
12     rescue InsufficientInventoryError => e
13       # Log the exception
14       logger.error("Inventory check failed: #{e.message}")
15
16       # Retry the operation after a delay
17       retry_count ||= 0
18       if retry_count < MAX_RETRIES
19         retry_count += 1
20         sleep(RETRY_DELAY)
21         retry
22       else
23         # Fallback to manual intervention
24         NotificationService.admin("Inventory check failed: Order #{order.id}")
25         return false
26       end
27     end
28   end
29 end
```

Σε αυτό το παράδειγμα, το στοιχείο `InventoryManager` ελέγχει τη διαθεσιμότητα ενός προϊόντος για μια δεδομένη παραγγελία. Εάν η διαθέσιμη ποσότητα είναι ανεπαρκής, εγείρει ένα `InsufficientInventoryError`. Η εξαίρεση συλλαμβάνεται, καταγράφεται και εφαρμόζεται ένας μηχανισμός επανάληψης. Εάν ξεπεραστεί το όριο επαναλήψεων, το στοιχείο καταφεύγει σε χειροκίνητη παρέμβαση ειδοποιώντας έναν διαχειριστή.

Με την εφαρμογή ισχυρών μηχανισμών χειρισμού και ανάκαμψης από εξαιρέσεις, μπορείτε να διασφαλίσετε ότι οι έξυπνες ροές εργασίας σας είναι ανθεκτικές,

συντηρήσιμες και ικανές να χειρίζονται απρόσμενες καταστάσεις με κομψό τρόπο.

Αυτά τα πρότυπα αποτελούν το θεμέλιο της έξυπνης ενορχήστρωσης ροής εργασίας και μπορούν να συνδυαστούν και να προσαρμοστούν ώστε να ανταποκρίνονται στις συγκεκριμένες απαιτήσεις διαφορετικών εφαρμογών. Αξιοποιώντας αυτά τα πρότυπα, οι προγραμματιστές μπορούν να δημιουργήσουν ροές εργασίας που είναι ευέλικτες, ανθεκτικές και βελτιστοποιημένες για απόδοση και εμπειρία χρήστη.

Στην επόμενη ενότητα, θα εξερευνήσουμε πώς αυτά τα πρότυπα μπορούν να υλοποιηθούν στην πράξη, χρησιμοποιώντας παραδείγματα από τον πραγματικό κόσμο και αποσπάσματα κώδικα για να απεικονίσουμε την ενσωμάτωση των στοιχείων AI στη διαχείριση ροής εργασίας.

Υλοποίηση Έξυπνης Ενορχήστρωσης Ροής Εργασίας στην Πράξη

Τώρα που έχουμε εξερευνήσει τα βασικά πρότυπα στην έξυπνη ενορχήστρωση ροής εργασίας, ας εμβαθύνουμε στο πώς αυτά τα πρότυπα μπορούν να υλοποιηθούν σε εφαρμογές του πραγματικού κόσμου. Θα παρέχουμε πρακτικά παραδείγματα και αποσπάσματα κώδικα για να απεικονίσουμε την ενσωμάτωση των στοιχείων AI στη διαχείριση ροής εργασίας.

Έξυπνος Επεξεργαστής Παραγγελιών

Ας εμβαθύνουμε σε ένα πρακτικό παράδειγμα υλοποίησης έξυπνης ενορχήστρωσης ροής εργασίας χρησιμοποιώντας ένα στοιχείο `OrderProcessor` με υποστήριξη AI σε μια εφαρμογή ηλεκτρονικού εμπορίου Ruby on Rails. Ο `OrderProcessor` υλοποιεί την έννοια του [Διαχειριστή Διαδικασιών Enterprise Integration](#) που

συναντήσαμε για πρώτη φορά στο Κεφάλαιο 3 όταν συζητούσαμε για το [Πλήθος Εργαζομένων](#). Το στοιχείο θα είναι υπεύθυνο για τη διαχείριση της ροής εργασίας εκπλήρωσης παραγγελιών, λαμβάνοντας αποφάσεις δρομολόγησης με βάση ενδιάμεσα αποτελέσματα και ενορχηστρώνοντας την εκτέλεση διαφόρων βημάτων επεξεργασίας.

Η διαδικασία εκπλήρωσης παραγγελίας περιλαμβάνει πολλαπλά βήματα όπως επικύρωση παραγγελίας, έλεγχο αποθέματος, επεξεργασία πληρωμής και αποστολή. Κάθε βήμα υλοποιείται ως ξεχωριστή διεργασία εργασίας που εκτελεί μια συγκεκριμένη εργασία και επιστρέφει το αποτέλεσμα στον OrderProcessor. Τα βήματα δεν είναι υποχρεωτικά και δεν χρειάζεται καν να εκτελούνται με συγκεκριμένη σειρά.

Ακολουθεί ένα παράδειγμα υλοποίησης του OrderProcessor. Διαθέτει δύο mixins από το [Raix](#). Το πρώτο (ChatCompletion) του δίνει τη δυνατότητα να κάνει ολοκλήρωση συνομιλίας, που είναι αυτό που το καθιστά στοιχείο AI. Το δεύτερο (FunctionDispatch) ενεργοποιεί την κλήση συναρτήσεων από το AI, επιτρέποντάς του να απαντά σε ένα prompt με μια κλήση συνάρτησης αντί για ένα μήνυμα κειμένου.

Οι λειτουργίες εργασίας (validate_order, check_inventory, κ.λπ.) αναθέτουν στις αντίστοιχες κλάσεις εργασίας τους, οι οποίες μπορεί να είναι στοιχεία τεχνητής νοημοσύνης ή μη, με τη μόνη απαίτηση να επιστρέφουν τα αποτελέσματα της εργασίας τους σε μια μορφή που μπορεί να αναπαρασταθεί ως συμβολοσειρά.



Όπως και με όλα τα άλλα παραδείγματα σε αυτό το μέρος του βιβλίου, αυτός ο κώδικας είναι ουσιαστικά ψευδοκώδικας και προορίζεται μόνο για να μεταδώσει το νόημα του μοτίβου και να εμπνεύσει τις δικές σας δημιουργίες. Πλήρεις περιγραφές των μοτίβων και ολοκληρωμένα παραδείγματα κώδικα περιλαμβάνονται στο Μέρος 2.

```
1  class OrderProcessor
2    include Raix::ChatCompletion
3    include Raix::FunctionDispatch
4
5    SYSTEM_DIRECTIVE = "You are an order processor, tasked with..."
6
7    def initialize(order)
8      self.order = order
9      transcript << { system: SYSTEM_DIRECTIVE }
10     transcript << { user: order.to_json }
11   end
12
13   def perform
14     # will continue looping until `stop_looping!` is called
15     chat_completion(loop: true)
16   end
17
18   # list of functions available to be called by the AI
19   # truncated for brevity
20
21   def functions
22     [
23       {
24         name: "validate_order",
25         description: "Invoke to check validity of order",
26         parameters: {
27           ...
28         },
29         ...
30     ]
31   end
32
33   # implementation of functions that can be called by the AI
34   # entirely at its discretion, depending on the needs of the order
35
36   def validate_order
37     OrderValidationWorker.perform(@order)
38   end
39
40   def check_inventory
41     InventoryCheckWorker.perform(@order)
42   end
```

```
43
44  def process_payment
45      PaymentProcessingWorker.perform(@order)
46  end
47
48  def schedule_shipping
49      ShippingSchedulerWorker.perform(@order)
50  end
51
52  def send_confirmation
53      OrderConfirmationWorker.perform(@order)
54  end
55
56  def finished_processing
57      @order.update!(transcript:, processed_at: Time.current)
58      stop_looping!
59  end
60 end
```

Στο παράδειγμα, ο `OrderProcessor` αρχικοποιείται με ένα αντικείμενο παραγγελίας και διατηρεί μια καταγραφή της εκτέλεσης της ροής εργασίας, στη συνηθισμένη μορφή καταγραφής συνομιλίας που είναι εγγενής στα μοντέλα μεγάλης γλώσσας. Δίνεται πλήρης έλεγχος στην τεχνητή νοημοσύνη για να ενορχηστρώσει την εκτέλεση διαφόρων βημάτων επεξεργασίας, όπως η επικύρωση παραγγελίας, ο έλεγχος αποθέματος, η επεξεργασία πληρωμής και η αποστολή.

Κάθε φορά που καλείται η μέθοδος `chat_completion`, η καταγραφή αποστέλλεται στην τεχνητή νοημοσύνη για να παρέχει μια ολοκλήρωση ως κλήση συνάρτησης. Εξαρτάται εξ ολοκλήρου από την τεχνητή νοημοσύνη να αναλύσει το αποτέλεσμα του προηγούμενου βήματος και να καθορίσει την κατάλληλη ενέργεια. Για παράδειγμα, εάν ο έλεγχος αποθέματος αποκαλύψει χαμηλά επίπεδα αποθέματος, ο `OrderProcessor` μπορεί να προγραμματίσει μια εργασία αναπλήρωσης. Εάν η επεξεργασία πληρωμής αποτύχει, μπορεί να ξεκινήσει μια επανάληψη ή να ειδοποιήσει την υποστήριξη πελατών.

Το παραπάνω παράδειγμα δεν έχει καθορισμένες συναρτήσεις για αναπλήρωση ή ειδοποίηση της υποστήριξης πελατών, αλλά θα μπορούσε σίγουρα να έχει.

Η καταγραφή μεγαλώνει κάθε φορά που καλείται μια συνάρτηση και χρησιμεύει ως αρχείο της εκτέλεσης της ροής εργασίας, συμπεριλαμβανομένων των αποτελεσμάτων κάθε βήματος και των οδηγιών που δημιουργούνται από την τεχνητή νοημοσύνη για τα επόμενα βήματα. Αυτή η καταγραφή μπορεί να χρησιμοποιηθεί για αποσφαλμάτωση, έλεγχο και παροχή ορατότητας στη διαδικασία εκπλήρωσης παραγγελιών.

Αξιοποιώντας την τεχνητή νοημοσύνη στον OrderProcessor, η εφαρμογή ηλεκτρονικού εμπορίου μπορεί να προσαρμόσει δυναμικά τη ροή εργασίας με βάση δεδομένα πραγματικού χρόνου και να χειριστεί εξαιρέσεις με έξυπνο τρόπο. Το στοιχείο τεχνητής νοημοσύνης μπορεί να λάβει τεκμηριωμένες αποφάσεις, να βελτιστοποιήσει τη ροή εργασίας και να διασφαλίσει την ομαλή επεξεργασία παραγγελιών ακόμη και σε πολύπλοκα σενάρια.

Το γεγονός ότι η μόνη απαίτηση για τις διεργασίες εργασίας είναι να επιστρέψουν κάποια κατανοητή έξοδο για να εξετάσει η τεχνητή νοημοσύνη όταν αποφασίζει τι να κάνει στη συνέχεια, μπορεί να αρχίσει να σας φανερώνει πώς αυτή η προσέγγιση μπορεί να μειώσει την εργασία αντιστοίχισης εισόδου/εξόδου που συνήθως απαιτείται κατά την ενσωμάτωση διαφορετικών συστημάτων μεταξύ τους.

Έξυπνος Συντονιστής Περιεχομένου

Οι εφαρμογές κοινωνικών μέσων γενικά απαιτούν τουλάχιστον ελάχιστο συντονισμό περιεχομένου για να διασφαλίσουν μια ασφαλή και υγιή κοινότητα. Αυτό το παράδειγμα του στοιχείου ContentModerator αξιοποιεί την τεχνητή νοημοσύνη για να ενορχηστρώσει έξυπνα τη ροή εργασίας συντονισμού, λαμβάνοντας αποφάσεις

με βάση τα χαρακτηριστικά του περιεχομένου και τα αποτελέσματα διαφόρων βημάτων συντονισμού.

Η διαδικασία συντονισμού περιλαμβάνει πολλαπλά βήματα όπως ανάλυση κειμένου, αναγνώριση εικόνας, αξιολόγηση φήμης χρήστη και χειροκίνητη επισκόπηση. Κάθε βήμα υλοποιείται ως ξεχωριστή διεργασία εργασίας που εκτελεί μια συγκεκριμένη εργασία και επιστρέφει το αποτέλεσμα στον ContentModerator.

Ακολουθεί ένα παράδειγμα υλοποίησης του ContentModerator:

```
1 class ContentModerator
2   include Raix::ChatCompletion
3   include Raix::FunctionDispatch
4
5   SYSTEM_DIRECTIVE = "You are a content moderator process manager,
6     tasked with the workflow involved in moderating user-generated content..."
7
8   def initialize(content)
9     @content = content
10    @transcript = [
11      { system: SYSTEM_DIRECTIVE },
12      { user: content.to_json }
13    ]
14  end
15
16  def perform
17    complete(@transcript)
18  end
19
20  def model
21    "openai/gpt-4"
22  end
23
24  # list of functions available to be called by the AI
25  # truncated for brevity
26
27  def functions
28    [
29      {
30        name: "analyze_text",
```

```
31         # ...
32     },
33     {
34         name: "recognize_image",
35         description: "Invoke to describe images...",
36         # ...
37     },
38     {
39         name: "assess_user_reputation",
40         # ...
41     },
42     {
43         name: "escalate_to_manual_review",
44         # ...
45     },
46     {
47         name: "approve_content",
48         # ...
49     },
50     {
51         name: "reject_content",
52         # ...
53     }
54 ]
55 end
56
57 # implementation of functions that can be called by the AI
58 # entirely at its discretion, depending on the needs of the order
59
60 def analyze_text
61     result = TextAnalysisWorker.perform(@content)
62     continue_with(result)
63 end
64
65 def recognize_image
66     result = ImageRecognitionWorker.perform(@content)
67     continue_with(result)
68 end
69
70 def assess_user_reputation
71     result = UserReputationWorker.perform(@content.user)
72     continue_with(result)
```

```
73     end
74
75     def escalate_to_manual_review
76         ManualReviewWorker.perform(@content)
77         @content.update!(status: 'pending', transcript: @transcript)
78     end
79
80     def approve_content
81         @content.update!(status: 'approved', transcript: @transcript)
82     end
83
84     def reject_content
85         @content.update!(status: 'rejected', transcript: @transcript)
86     end
87
88     private
89
90     def continue_with(result)
91         @transcript << { function: result }
92         complete(@transcript)
93     end
94 end
```

Σε αυτό το παράδειγμα, ο ContentModerator αρχικοποιείται με ένα αντικείμενο περιεχομένου και διατηρεί μια καταγραφή συντονισμού σε μορφή συνομιλίας. Το στοιχείο AI έχει πλήρη έλεγχο της ροής συντονισμού, αποφασίζοντας ποια βήματα θα εκτελέσει με βάση τα χαρακτηριστικά του περιεχομένου και τα αποτελέσματα κάθε βήματος.

Οι διαθέσιμες λειτουργίες εργασίας που μπορεί να καλέσει το AI περιλαμβάνουν τις `analyze_text`, `recognize_image`, `assess_user_reputation`, και `escalate_to_manual_review`. Κάθε λειτουργία αναθέτει την εργασία στην αντίστοιχη διαδικασία εργασίας (TextAnalysisWorker, ImageRecognitionWorker, κλπ.) και προσθέτει το αποτέλεσμα στην καταγραφή συντονισμού, με εξαίρεση τη λειτουργία κλιμάκωσης, η οποία λειτουργεί ως τελική κατάσταση. Τέλος, οι λειτουργίες `approve_content` και `reject_content` λειτουργούν επίσης ως τελικές καταστάσεις.

Το στοιχείο AI αναλύει το περιεχόμενο και καθορίζει την κατάλληλη ενέργεια.

Εάν το περιεχόμενο περιέχει αναφορές εικόνων, μπορεί να καλέσει τον εργάτη `recognize_image` για βοήθεια με οπτική αξιολόγηση. Εάν κάποιος εργάτης προειδοποιήσει για πιθανώς επιβλαβές περιεχόμενο, το ΑΙ μπορεί να αποφασίσει να κλιμακώσει το περιεχόμενο για χειροκίνητη αξιολόγηση ή απλά να το απορρίψει άμεσα. Ωστόσο, ανάλογα με τη σοβαρότητα της προειδοποίησης, το ΑΙ μπορεί να επιλέξει να χρησιμοποιήσει τα αποτελέσματα της αξιολόγησης φήμης του χρήστη για να αποφασίσει πώς να χειριστεί περιεχόμενο για το οποίο δεν είναι σίγουρο. Ανάλογα με την περίπτωση χρήσης, ίσως οι αξιόπιστοι χρήστες να έχουν περισσότερη ελευθερία στο τι μπορούν να δημοσιεύσουν. Και ούτω καθεξής...

Όπως και στο προηγούμενο παράδειγμα διαχειριστή διαδικασιών, η καταγραφή συντονισμού χρησιμεύει ως αρχείο της εκτέλεσης της ροής εργασίας, συμπεριλαμβανομένων των αποτελεσμάτων κάθε βήματος και των αποφάσεων που δημιουργήθηκαν από το ΑΙ. Αυτή η καταγραφή μπορεί να χρησιμοποιηθεί για έλεγχο, διαφάνεια και βελτίωση της διαδικασίας συντονισμού με την πάροδο του χρόνου.

Αξιοποιώντας το ΑΙ στον `ContentModerator`, η εφαρμογή κοινωνικής δικτύωσης μπορεί να προσαρμόζει δυναμικά τη ροή συντονισμού με βάση τα χαρακτηριστικά του περιεχομένου και να χειρίζεται έξυπνα πολύπλοκα σενάρια συντονισμού. Το στοιχείο ΑΙ μπορεί να λαμβάνει τεκμηριωμένες αποφάσεις, να βελτιστοποιεί τη ροή εργασίας και να διασφαλίζει μια ασφαλή και υγιή εμπειρία κοινότητας.

Ας εξερευνήσουμε δύο ακόμη παραδείγματα που επιδεικνύουν τον προγραμματισμό εργασιών πρόβλεψης και το χειρισμό εξαιρέσεων και ανάκτησης στο πλαίσιο της έξυπνης ενορχήστρωσης ροής εργασίας.

Προγραμματισμός Εργασιών Πρόβλεψης σε Σύστημα Υποστήριξης Πελατών

Σε μια εφαρμογή υποστήριξης πελατών που έχει κατασκευαστεί με `Ruby on Rails`, η αποτελεσματική διαχείριση και ιεράρχηση των αιτημάτων υποστήριξης

είναι κρίσιμη για την παροχή έγκαιρης βοήθειας στους πελάτες. Το στοιχείο `SupportTicketScheduler` αξιοποιεί το AI για να προγραμματίζει προβλεπτικά και να αναθέτει αιτήματα υποστήριξης σε διαθέσιμους πράκτορες με βάση διάφορους παράγοντες όπως ο επείγων χαρακτήρας του αιτήματος, η εξειδίκευση του πράκτορα και ο φόρτος εργασίας.

```
1 class SupportTicketScheduler
2     include Raix::ChatCompletion
3     include Raix::FunctionDispatch
4
5     SYSTEM_DIRECTIVE = "You are a support ticket scheduler,
6         tasked with intelligently assigning tickets to available agents..."
7
8     def initialize(ticket)
9         @ticket = ticket
10        @transcript = [
11            { system: SYSTEM_DIRECTIVE },
12            { user: ticket.to_json }
13        ]
14    end
15
16    def perform
17        complete(@transcript)
18    end
19
20    def model
21        "openai/gpt-4"
22    end
23
24    def functions
25        [
26            {
27                name: "analyze_ticket_urgency",
28                # ...
29            },
30            {
31                name: "list_available_agents",
32                description: "Includes expertise of available agents",
33                # ...
34            },
```

```
35     {
36         name: "predict_agent_workload",
37         description: "Uses historical data to predict upcoming workloads",
38         # ...
39     },
40     {
41         name: "assign_ticket_to_agent",
42         # ...
43     },
44     {
45         name: "reschedule_ticket",
46         # ...
47     }
48 ]
49 end
50
51 # implementation of functions that can be called by the AI
52 # entirely at its discretion, depending on the needs of the order
53
54 def analyze_ticket_urgency
55     result = TicketUrgencyAnalyzer.perform(@ticket)
56     continue_with(result)
57 end
58
59 def list_available_agents
60     result = ListAvailableAgents.perform
61     continue_with(result)
62 end
63
64 def predict_agent_workload
65     result = AgentWorkloadPredictor.perform
66     continue_with(result)
67 end
68
69 def assign_ticket_to_agent
70     TicketAssigner.perform(@ticket, @transcript)
71 end
72
73 def delay_assignment(until)
74     until = DateTimeStandardizer.process(until)
75     SupportTicketScheduler.delay(@ticket, @transcript, until)
76 end
```

```
77
78 private
79
80 def continue_with(result)
81     @transcript << { function: result }
82     complete(@transcript)
83 end
84 end
```

Σε αυτό το παράδειγμα, ο `SupportTicketScheduler` αρχικοποιείται με ένα αντικείμενο εισιτηρίου υποστήριξης και διατηρεί μια καταγραφή προγραμματισμού. Το στοιχείο AI αναλύει τις λεπτομέρειες του εισιτηρίου και προγραμματίζει προγνωστικά την ανάθεση του εισιτηρίου με βάση παράγοντες όπως ο επείγων χαρακτήρας του εισιτηρίου, η εξειδίκευση του χειριστή και ο προβλεπόμενος φόρτος εργασίας του χειριστή.

Οι διαθέσιμες λειτουργίες που μπορεί να επικαλεστεί το AI περιλαμβάνουν τις `analyze_ticket_urgency`, `list_available_agents`, `predict_agent_workload`, και `assign_ticket_to_agent`. Κάθε λειτουργία αναθέτει την εργασία σε ένα αντίστοιχο στοιχείο ανάλυσης ή πρόβλεψης και προσθέτει το αποτέλεσμα στην καταγραφή προγραμματισμού. Το AI έχει επίσης τη δυνατότητα να καθυστερήσει την ανάθεση χρησιμοποιώντας τη λειτουργία `delay_assignment`.

Το στοιχείο AI εξετάζει την καταγραφή προγραμματισμού και λαμβάνει τεκμηριωμένες αποφάσεις σχετικά με την ανάθεση εισιτηρίων. Λαμβάνει υπόψη τον επείγοντα χαρακτήρα του εισιτηρίου, την εξειδίκευση των διαθέσιμων χειριστών και τον προβλεπόμενο φόρτο εργασίας κάθε χειριστή για να καθορίσει τον καταλληλότερο χειριστή για τη διαχείριση του εισιτηρίου.

Αξιοποιώντας τον προγνωστικό προγραμματισμό εργασιών, η εφαρμογή υποστήριξης πελατών μπορεί να βελτιστοποιήσει την ανάθεση εισιτηρίων, να μειώσει τους χρόνους απόκρισης και να βελτιώσει τη συνολική ικανοποίηση των πελατών. Η προληπτική και αποτελεσματική διαχείριση των εισιτηρίων υποστήριξης διασφαλίζει ότι τα σωστά εισιτήρια ανατίθενται στους σωστούς

χειριστές την κατάλληλη στιγμή.

Χειρισμός Εξαιρέσεων και Ανάκτηση σε μια Διαδικασία Επεξεργασίας Δεδομένων

Ο χειρισμός εξαιρέσεων και η ανάκτηση από αποτυχίες είναι απαραίτητα για τη διασφάλιση της ακεραιότητας των δεδομένων και την αποτροπή απώλειας δεδομένων.. Το στοιχείο `DataProcessingOrchestrator` χρησιμοποιεί AI για τον έξυπνο χειρισμό εξαιρέσεων και την ενορχήστρωση της διαδικασίας ανάκτησης σε μια διαδικασία επεξεργασίας δεδομένων

```
1 class DataProcessingOrchestrator
2     include Raix::ChatCompletion
3     include Raix::FunctionDispatch
4
5     SYSTEM_DIRECTIVE = "You are a data processing orchestrator..."
6
7     def initialize(data_batch)
8         @data_batch = data_batch
9         @transcript = [
10             { system: SYSTEM_DIRECTIVE },
11             { user: data_batch.to_json }
12         ]
13     end
14
15     def perform
16         complete(@transcript)
17     end
18
19     def model
20         "openai/gpt-4"
21     end
22
23     def functions
24         [
25             {
26                 name: "validate_data",
27                 # ...
```

```
28     },
29     {
30         name: "process_data",
31         # ...
32     },
33     {
34         name: "request_fix",
35         # ...
36     },
37     {
38         name: "retry_processing",
39         # ...
40     },
41     {
42         name: "mark_data_as_failed",
43         # ...
44     },
45     {
46         name: "finished",
47         # ...
48     }
49 ]
50 end
51
52 # implementation of functions that can be called by the AI
53 # entirely at its discretion, depending on the needs of the order
54
55 def validate_data
56     result = DataValidator.perform(@data_batch)
57     continue_with(result)
58 rescue ValidationException => e
59     handle_validation_exception(e)
60 end
61
62 def process_data
63     result = DataProcessor.perform(@data_batch)
64     continue_with(result)
65 rescue ProcessingException => e
66     handle_processing_exception(e)
67 end
68
69 def request_fix(description_of_fix)
```

```
70     result = SmartDataFixer.new(description_of_fix, @data_batch)
71     continue_with(result)
72 end
73
74 def retry_processing(timeout_in_seconds)
75     wait(timeout_in_seconds)
76     process_data
77 end
78
79 def mark_data_as_failed
80     @data_batch.update!(status: 'failed', transcript: @transcript)
81 end
82
83 def finished
84     @data_batch.update!(status: 'finished', transcript: @transcript)
85 end
86
87 private
88
89 def continue_with(result)
90     @transcript << { function: result }
91     complete(@transcript)
92 end
93
94 def handle_validation_exception(exception)
95     @transcript << { exception: exception.message }
96     complete(@transcript)
97 end
98
99 def handle_processing_exception(exception)
100     @transcript << { exception: exception.message }
101     complete(@transcript)
102 end
103 end
```

Σε αυτό το παράδειγμα, ο `DataProcessingOrchestrator` αρχικοποιείται με ένα αντικείμενο παρτίδας δεδομένων και διατηρεί ένα αρχείο καταγραφής επεξεργασίας. Το στοιχείο AI ενορχηστρώνει τη διαδικασία επεξεργασίας δεδομένων, χειριζόμενο εξαιρέσεις και ανακάμπτοντας από αποτυχίες όταν χρειάζεται.

Οι διαθέσιμες συναρτήσεις που μπορεί να καλέσει η AI περιλαμβάνουν τις `validate_data`, `process_data`, `request_fix`, `retry_processing`, και `mark_data_as_failed`. Κάθε συνάρτηση αναθέτει την εργασία σε ένα αντίστοιχο στοιχείο επεξεργασίας δεδομένων και προσθέτει το αποτέλεσμα ή τις λεπτομέρειες της εξαίρεσης στο αρχείο καταγραφής επεξεργασίας.

Εάν προκύψει μια εξαίρεση επικύρωσης κατά τη διάρκεια του βήματος `validate_data`, η συνάρτηση `handle_validation_exception` προσθέτει τα δεδομένα της εξαίρεσης στο αρχείο καταγραφής και επιστρέφει τον έλεγχο στην AI. Παρομοίως, εάν προκύψει μια εξαίρεση επεξεργασίας κατά τη διάρκεια του βήματος `process_data`, η AI μπορεί να αποφασίσει για τη στρατηγική ανάκαμψης.

Ανάλογα με τη φύση της εξαίρεσης που συναντάται, η AI μπορεί κατά την κρίση της να αποφασίσει να καλέσει το `request_fix`, το οποίο αναθέτει σε ένα στοιχείο `SmartDataFixer` που λειτουργεί με AI (δείτε το κεφάλαιο Αυτοθεραπευόμενα Δεδομένα). Ο διορθωτής δεδομένων λαμβάνει μια περιγραφή σε απλά αγγλικά για το πώς θα πρέπει να τροποποιήσει το `@data_batch` ώστε να μπορεί να επαναληφθεί η επεξεργασία. Ίσως μια επιτυχημένη επανάληψη θα συνεπαγόταν την αφαίρεση εγγραφών από την παρτίδα δεδομένων που απέτυχαν στην επικύρωση ή/και την αντιγραφή τους σε διαφορετική διαδικασία επεξεργασίας για ανθρώπινη επισκόπηση; Οι δυνατότητες είναι σχεδόν ατελείωτες.

Με την ενσωμάτωση του χειρισμού εξαιρέσεων και της ανάκαμψης που καθοδηγείται από AI, η εφαρμογή επεξεργασίας δεδομένων γίνεται πιο ανθεκτική και ανεκτική σε σφάλματα. Ο `DataProcessingOrchestrator` διαχειρίζεται έξυπνα τις εξαιρέσεις, ελαχιστοποιεί την απώλεια δεδομένων και διασφαλίζει την ομαλή εκτέλεση της ροής εργασίας επεξεργασίας δεδομένων.

Παρακολούθηση και Καταγραφή

Η παρακολούθηση και η καταγραφή παρέχουν ορατότητα στην πρόοδο, την απόδοση και την υγεία των στοιχείων ροής εργασίας που λειτουργούν με AI,

επιτρέποντας στους προγραμματιστές να παρακολουθούν και να αναλύουν τη συμπεριφορά του συστήματος. Η εφαρμογή αποτελεσματικών μηχανισμών παρακολούθησης και καταγραφής είναι απαραίτητη για την αποσφαλμάτωση, τον έλεγχο και τη συνεχή βελτίωση των έξυπνων ροών εργασίας.

Παρακολούθηση Προόδου και Απόδοσης Ροής Εργασίας

Για να διασφαλιστεί η ομαλή εκτέλεση των έξυπνων ροών εργασίας, είναι σημαντικό να παρακολουθείται η πρόοδος και η απόδοση κάθε στοιχείου ροής εργασίας. Αυτό περιλαμβάνει την παρακολούθηση βασικών μετρικών και συμβάντων καθ' όλη τη διάρκεια του κύκλου ζωής της ροής εργασίας.

Μερικές σημαντικές πτυχές για παρακολούθηση περιλαμβάνουν:

- 1. Χρόνος Εκτέλεσης Ροής Εργασίας:** Μέτρηση του χρόνου που χρειάζεται κάθε στοιχείο ροής εργασίας για να ολοκληρώσει την εργασία του. Αυτό βοηθά στον εντοπισμό σημείων συμφόρησης απόδοσης και στη βελτιστοποίηση της συνολικής αποδοτικότητας της ροής εργασίας.
- 2. Χρήση Πόρων:** Παρακολούθηση της χρήσης των πόρων του συστήματος, όπως CPU, μνήμη και αποθηκευτικός χώρος, από κάθε στοιχείο ροής εργασίας. Αυτό βοηθά στη διασφάλιση ότι το σύστημα λειτουργεί εντός της χωρητικότητάς του και μπορεί να διαχειριστεί αποτελεσματικά το φόρτο εργασίας.
- 3. Ποσοστά Σφαλμάτων και Εξαιρέσεις:** Παρακολουθήστε την εμφάνιση σφαλμάτων και εξαιρέσεων εντός των στοιχείων της ροής εργασίας. Αυτό βοηθά στον εντοπισμό πιθανών προβλημάτων και επιτρέπει τον προληπτικό χειρισμό και την αποκατάσταση σφαλμάτων.
- 4. Σημεία Απόφασης και Αποτελέσματα:** Παρακολουθήστε τα σημεία απόφασης εντός της ροής εργασίας και τα αποτελέσματα των αποφάσεων που λαμβάνονται από την τεχνητή νοημοσύνη. Αυτό παρέχει πληροφορίες σχετικά με τη συμπεριφορά και την αποτελεσματικότητα των στοιχείων τεχνητής νοημοσύνης.

Τα δεδομένα που συλλέγονται από τις διαδικασίες παρακολούθησης μπορούν να εμφανίζονται σε πίνακες ελέγχου ή να χρησιμοποιούνται ως είσοδοι σε προγραμματισμένες αναφορές που ενημερώνουν τους διαχειριστές συστήματος σχετικά με την υγεία του συστήματος.



Τα δεδομένα παρακολούθησης μπορούν να τροφοδοτηθούν σε μια διαδικασία διαχειριστή συστήματος με τεχνητή νοημοσύνη για επισκόπηση και πιθανή δράση!

Καταγραφή Βασικών Συμβάντων και Αποφάσεων

Η καταγραφή είναι μια απαραίτητη πρακτική που περιλαμβάνει τη σύλληψη και αποθήκευση σχετικών πληροφοριών για βασικά συμβάντα, αποφάσεις και εξαιρέσεις που συμβαίνουν κατά την εκτέλεση της ροής εργασίας.

Μερικές σημαντικές πτυχές προς καταγραφή περιλαμβάνουν:

- 1. Έναρξη και Ολοκλήρωση Ροής Εργασίας:** Καταγράψτε τους χρόνους έναρξης και λήξης κάθε στιγμιότυπου ροής εργασίας, μαζί με τυχόν σχετικά μεταδεδομένα όπως τα δεδομένα εισόδου και το πλαίσιο χρήστη.
- 2. Εκτέλεση Στοιχείων:** Καταγράψτε τις λεπτομέρειες εκτέλεσης κάθε στοιχείου της ροής εργασίας, συμπεριλαμβανομένων των παραμέτρων εισόδου, των αποτελεσμάτων εξόδου και τυχόν ενδιάμεσων δεδομένων που δημιουργούνται.
- 3. Αποφάσεις και Συλλογισμός Τεχνητής Νοημοσύνης:** Καταγράψτε τις αποφάσεις που λαμβάνονται από τα στοιχεία τεχνητής νοημοσύνης, μαζί με τον υποκείμενο συλλογισμό ή τις βαθμολογίες εμπιστοσύνης. Αυτό παρέχει διαφάνεια και επιτρέπει τον έλεγχο των αποφάσεων που λαμβάνονται από την τεχνητή νοημοσύνη.

4. Εξαιρέσεις και Μηνύματα Σφάλματος: Καταγράψτε τυχόν εξαιρέσεις ή μηνύματα σφάλματος που παρουσιάζονται κατά την εκτέλεση της ροής εργασίας, συμπεριλαμβανομένου του ίχνους στοίβας και των σχετικών πληροφοριών πλαισίου.

Η καταγραφή μπορεί να υλοποιηθεί χρησιμοποιώντας διάφορες τεχνικές, όπως η εγγραφή σε αρχεία καταγραφής, η αποθήκευση καταγραφών σε βάση δεδομένων ή η αποστολή καταγραφών σε μια κεντρική υπηρεσία καταγραφής. Είναι σημαντικό να επιλέξετε ένα πλαίσιο καταγραφής που παρέχει ευελιξία, κλιμάκωση και εύκολη ενσωμάτωση με την αρχιτεκτονική της εφαρμογής.

Εδώ είναι ένα παράδειγμα για το πώς μπορεί να υλοποιηθεί η καταγραφή σε μια εφαρμογή Ruby on Rails χρησιμοποιώντας την κλάση ActiveSupport::Logger:

```
1 class WorkflowLogger
2   def self.log(message, severity = :info)
3     @logger ||= ActiveSupport::Logger.new('workflow.log')
4     @logger.formatter ||= proc do |severity, datetime, progname, msg|
5       "#{datetime} [{severity}] #{msg}\n"
6     end
7     @logger.send(severity, message)
8   end
9 end
10
11 # Usage example
12 WorkflowLogger.log("Workflow initiated for order #{@order.id}")
13 WorkflowLogger.log("Payment processing completed successfully")
14 WorkflowLogger.log("Inventory check failed for item #{item.id}", :error)
```

Με τη στρατηγική τοποθέτηση δηλώσεων καταγραφής σε όλα τα στοιχεία της ροής εργασίας και τα σημεία λήψης αποφάσεων της TN, οι προγραμματιστές μπορούν να συλλέξουν πολύτιμες πληροφορίες για αποσφαλμάτωση, έλεγχο και ανάλυση.

Οφέλη της Παρακολούθησης και Καταγραφής

Η εφαρμογή παρακολούθησης και καταγραφής στην ευφυή ενορχήστρωση ροής εργασίας προσφέρει αρκετά οφέλη:

1. Αποσφαλμάτωση και Αντιμετώπιση Προβλημάτων: Τα λεπτομερή αρχεία καταγραφής και τα δεδομένα παρακολούθησης βοηθούν τους προγραμματιστές να εντοπίσουν και να διαγνώσουν προβλήματα γρήγορα. Παρέχουν πληροφορίες σχετικά με τη ροή εκτέλεσης της εργασίας, τις αλληλεπιδράσεις των στοιχείων και τυχόν σφάλματα ή εξαιρέσεις που παρουσιάστηκαν.

2. Βελτιστοποίηση Απόδοσης: Η παρακολούθηση των μετρήσεων απόδοσης επιτρέπει στους προγραμματιστές να εντοπίσουν σημεία συμφόρησης και να βελτιστοποιήσουν τα στοιχεία της ροής εργασίας για καλύτερη αποδοτικότητα. Αναλύοντας τους χρόνους εκτέλεσης, τη χρήση πόρων και άλλες μετρήσεις, οι προγραμματιστές μπορούν να λάβουν τεκμηριωμένες αποφάσεις για τη βελτίωση της συνολικής απόδοσης του συστήματος.

3. Έλεγχος και Συμμόρφωση: Η καταγραφή βασικών συμβάντων και αποφάσεων παρέχει μια διαδρομή ελέγχου για κανονιστική συμμόρφωση και λογοδοσία. Επιτρέπει στους οργανισμούς να παρακολουθούν και να επαληθεύουν τις ενέργειες που εκτελούνται από τα στοιχεία TN και να διασφαλίζουν τη συμμόρφωση με τους επιχειρηματικούς κανόνες και τις νομικές απαιτήσεις.

4. Συνεχής Βελτίωση: Τα δεδομένα παρακολούθησης και καταγραφής χρησιμεύουν ως πολύτιμες εισροές για τη συνεχή βελτίωση των ευφυών ροών εργασίας. Αναλύοντας ιστορικά δεδομένα, εντοπίζοντας μοτίβα και μετρώντας την αποτελεσματικότητα των αποφάσεων TN, οι προγραμματιστές μπορούν να βελτιώνουν και να ενισχύουν επαναληπτικά τη λογική ενορχήστρωσης της ροής εργασίας.

Παράμετροι και Βέλτιστες Πρακτικές

Κατά την εφαρμογή παρακολούθησης και καταγραφής στην ευφυή ενορχήστρωση ροής εργασίας, εξετάστε τις ακόλουθες βέλτιστες πρακτικές:

1. Καθορισμός Σαφών Μετρήσεων Παρακολούθησης: Προσδιορίστε τις βασικές μετρήσεις και τα συμβάντα που πρέπει να παρακολουθούνται με βάση τις

συγκεκριμένες απαιτήσεις της ροής εργασίας. Εστιάστε σε μετρήσεις που παρέχουν ουσιαστικές πληροφορίες σχετικά με την απόδοση, την υγεία και τη συμπεριφορά του συστήματος.

2. Εφαρμογή Λεπτομερούς Καταγραφής: Βεβαιωθείτε ότι οι δηλώσεις καταγραφής τοποθετούνται σε κατάλληλα σημεία εντός των στοιχείων της ροής εργασίας και των σημείων λήψης αποφάσεων TN. Καταγράψτε σχετικές πληροφορίες πλαισίου, όπως παραμέτρους εισόδου, αποτελέσματα εξόδου και τυχόν ενδιάμεσα δεδομένα που δημιουργούνται.

3. Χρήση Δομημένης Καταγραφής: Υιοθετήστε μια δομημένη μορφή καταγραφής για να διευκολύνετε την εύκολη ανάλυση και επεξεργασία των δεδομένων καταγραφής. Η δομημένη καταγραφή επιτρέπει καλύτερη αναζήτηση, φιλτράρισμα και συγκέντρωση των καταχωρήσεων καταγραφής.

4. Διαχείριση Διατήρησης και Εναλλαγής Αρχείων Καταγραφής: Εφαρμόστε πολιτικές διατήρησης και εναλλαγής αρχείων καταγραφής για τη διαχείριση της αποθήκευσης και του κύκλου ζωής των αρχείων καταγραφής. Καθορίστε την κατάλληλη περίοδο διατήρησης με βάση τις νομικές απαιτήσεις, τους περιορισμούς αποθήκευσης και τις ανάγκες ανάλυσης. Εάν είναι δυνατόν, μεταφέρετε την καταγραφή σε μια υπηρεσία τρίτου μέρους όπως το [Papertrail](#).

5. Ασφάλεια Ευαίσθητων Πληροφοριών: Να είστε προσεκτικοί κατά την καταγραφή ευαίσθητων πληροφοριών, όπως προσωπικά αναγνωρίσιμων πληροφοριών (PII) ή εμπιστευτικών επιχειρηματικών δεδομένων. Εφαρμόστε κατάλληλα μέτρα ασφαλείας, όπως απόκρυψη δεδομένων ή κρυπτογράφηση, για την προστασία ευαίσθητων πληροφοριών στα αρχεία καταγραφής.

6. Ενσωμάτωση με Εργαλεία Παρακολούθησης και Ειδοποιήσεων: Αξιοποιήστε εργαλεία παρακολούθησης και ειδοποιήσεων για να κεντριοποιήσετε τη συλλογή, ανάλυση και οπτικοποίηση δεδομένων παρακολούθησης και καταγραφής. Αυτά τα εργαλεία μπορούν να παρέχουν πληροφορίες σε πραγματικό χρόνο, να δημιουργούν ειδοποιήσεις βάσει προκαθορισμένων ορίων και να διευκολύνουν την προληπτική

ανίχνευση και επίλυση προβλημάτων. Το αγαπημένο μου από αυτά τα εργαλεία είναι το [Datadog](#).

Με την εφαρμογή ολοκληρωμένων μηχανισμών παρακολούθησης και καταγραφής, οι προγραμματιστές μπορούν να αποκτήσουν πολύτιμες πληροφορίες σχετικά με τη συμπεριφορά και την απόδοση των έξυπνων ροών εργασίας. Αυτές οι πληροφορίες επιτρέπουν την αποτελεσματική αποσφαλμάτωση, βελτιστοποίηση και συνεχή βελτίωση των συστημάτων ενορχήστρωσης ροών εργασίας που βασίζονται στην τεχνητή νοημοσύνη.

Ζητήματα Κλιμάκωσης και Απόδοσης

Η κλιμάκωση και η απόδοση είναι κρίσιμες πτυχές που πρέπει να ληφθούν υπόψη κατά το σχεδιασμό και την υλοποίηση συστημάτων ενορχήστρωσης έξυπνων ροών εργασίας. Καθώς ο όγκος των ταυτόχρονων ροών εργασίας και η πολυπλοκότητα των στοιχείων που βασίζονται στην τεχνητή νοημοσύνη αυξάνονται, γίνεται απαραίτητο να διασφαλιστεί ότι το σύστημα μπορεί να διαχειριστεί αποτελεσματικά το φόρτο εργασίας και να κλιμακωθεί ομαλά για να ανταποκριθεί στις αυξανόμενες απαιτήσεις.

Διαχείριση Μεγάλου Όγκου Ταυτόχρονων Ροών Εργασίας

Τα συστήματα ενορχήστρωσης έξυπνων ροών εργασίας συχνά χρειάζεται να διαχειριστούν μεγάλο αριθμό ταυτόχρονων ροών εργασίας. Για να διασφαλιστεί η κλιμάκωση, εξετάστε τις ακόλουθες στρατηγικές:

1. Ασύγχρονη Επεξεργασία: Εφαρμόστε μηχανισμούς ασύγχρονης επεξεργασίας για να αποσυνδέσετε την εκτέλεση των στοιχείων ροής εργασίας. Αυτό επιτρέπει στο σύστημα να χειρίζεται πολλαπλές ροές εργασίας ταυτόχρονα χωρίς να μπλοκάρει ή να περιμένει την ολοκλήρωση κάθε στοιχείου. Η ασύγχρονη επεξεργασία μπορεί

να επιτευχθεί χρησιμοποιώντας ουρές μηνυμάτων, αρχιτεκτονικές βασισμένες σε συμβάντα ή πλαίσια επεξεργασίας εργασιών παρασκηνίου όπως το Sidekiq.

2. Κατανεμημένη Αρχιτεκτονική: Σχεδιάστε την αρχιτεκτονική του συστήματος ώστε να χρησιμοποιεί στοιχεία χωρίς διακομιστή (όπως το AWS Lambda) ή απλά να κατανέμει το φόρτο εργασίας σε πολλαπλούς κόμβους ή διακομιστές παράλληλα με τον κύριο διακομιστή εφαρμογών σας. Αυτό επιτρέπει την οριζόντια κλιμάκωση, όπου μπορούν να προστεθούν επιπλέον κόμβοι για τη διαχείριση αυξημένων όγκων ροών εργασίας.

3. Παράλληλη Εκτέλεση: Εντοπίστε ευκαιρίες για παράλληλη εκτέλεση εντός των ροών εργασίας. Ορισμένα στοιχεία ροής εργασίας μπορεί να είναι ανεξάρτητα μεταξύ τους και να μπορούν να εκτελεστούν ταυτόχρονα. Αξιοποιώντας τεχνικές παράλληλης επεξεργασίας, όπως πολυνηματισμό ή κατανεμημένες ουρές εργασιών, το σύστημα μπορεί να βελτιστοποιήσει τη χρήση πόρων και να μειώσει το συνολικό χρόνο εκτέλεσης της ροής εργασίας.

Βελτιστοποίηση Απόδοσης Στοιχείων Βασισμένων στην Τεχνητή Νοημοσύνη

Τα στοιχεία που βασίζονται στην τεχνητή νοημοσύνη, όπως τα μοντέλα μηχανικής μάθησης ή οι μηχανές επεξεργασίας φυσικής γλώσσας, μπορεί να είναι υπολογιστικά απαιτητικά και να επηρεάσουν τη συνολική απόδοση του συστήματος ενορχήστρωσης ροής εργασίας. Για τη βελτιστοποίηση της απόδοσης των στοιχείων τεχνητής νοημοσύνης, εξετάστε τις ακόλουθες τεχνικές:

1. Προσωρινή αποθήκευση: Εάν η επεξεργασία τεχνητής νοημοσύνης είναι καθαρά παραγωγική και δεν περιλαμβάνει αναζητήσεις πληροφοριών σε πραγματικό χρόνο ή εξωτερικές ενσωματώσεις για τη δημιουργία των chat completions, τότε μπορείτε να εξετάσετε μηχανισμούς προσωρινής αποθήκευσης για την αποθήκευση και επαναχρησιμοποίηση των αποτελεσμάτων συχνά προσπελάσιμων ή υπολογιστικά ακριβών λειτουργιών.

2. Βελτιστοποίηση Μοντέλου: Βελτιστοποιήστε συνεχώς τον τρόπο που χρησιμοποιείτε τα μοντέλα τεχνητής νοημοσύνης στα στοιχεία ροής εργασίας. Αυτό μπορεί να περιλαμβάνει τεχνικές όπως η *Απόσταξη Προτροπών* ή μπορεί απλά να είναι θέμα δοκιμής νέων μοντέλων καθώς γίνονται διαθέσιμα.

3. Ομαδική Επεξεργασία: Εάν εργάζεστε με μοντέλα κατηγορίας GPT-4, ίσως μπορείτε να αξιοποιήσετε τεχνικές ομαδικής επεξεργασίας για να επεξεργαστείτε πολλαπλά σημεία δεδομένων ή αιτήματα σε μία ενιαία παρτίδα, αντί να τα επεξεργάζεστε μεμονωμένα. Με την επεξεργασία δεδομένων σε παρτίδες, το σύστημα μπορεί να βελτιστοποιήσει τη χρήση πόρων και να μειώσει την επιβάρυνση των επαναλαμβανόμενων αιτημάτων μοντέλου.

Παρακολούθηση και Προφίλ Απόδοσης

Για τον εντοπισμό σημείων συμφόρησης της απόδοσης και τη βελτιστοποίηση της κλιμάκωσης του έξυπνου συστήματος ενορχήστρωσης ροής εργασίας, είναι κρίσιμο να εφαρμοστούν μηχανισμοί παρακολούθησης και προφίλ. Εξετάστε τις ακόλουθες προσεγγίσεις:

1. Μετρήσεις Απόδοσης: Καθορίστε και παρακολουθήστε βασικές μετρήσεις απόδοσης, όπως χρόνο απόκρισης, διεκπεραιωτική ικανότητα, χρήση πόρων και καθυστέρηση. Αυτές οι μετρήσεις παρέχουν πληροφορίες για την απόδοση του συστήματος και βοηθούν στον εντοπισμό περιοχών για βελτιστοποίηση. Ο δημοφιλής συναθροιστής μοντέλων τεχνητής νοημοσύνης [OpenRouter](#) περιλαμβάνει μετρήσεις $Host^1$ και $Speed^2$ σε κάθε απόκριση API, καθιστώντας εύκολη την παρακολούθηση αυτών των βασικών μετρήσεων.

2. Εργαλεία Προφίλ: Χρησιμοποιήστε εργαλεία προφίλ για να αναλύσετε την απόδοση μεμονωμένων στοιχείων ροής εργασίας και λειτουργιών τεχνητής

¹Το Host είναι ο χρόνος που χρειάστηκε για να ληφθεί το πρώτο byte της ροής παραγωγής από τον host του μοντέλου, γνωστό και ως "χρόνος μέχρι το πρώτο byte."

²Η ταχύτητα υπολογίζεται ως ο αριθμός των tokens ολοκλήρωσης διαιρεμένος με το συνολικό χρόνο παραγωγής. Για μη-ροές αιτημάτων, η καθυστέρηση θεωρείται μέρος του χρόνου παραγωγής.

νοημοσύνης. Τα εργαλεία προφίλ μπορούν να βοηθήσουν στον εντοπισμό σημείων υψηλής απόδοσης, αναποτελεσματικών διαδρομών κώδικα ή λειτουργιών που καταναλώνουν πολλούς πόρους. Δημοφιλή εργαλεία προφίλ περιλαμβάνουν το New Relic, το Scout, ή ενσωματωμένους profilers που παρέχονται από τη γλώσσα προγραμματισμού ή το πλαίσιο εργασίας.

3. Δοκιμή Φόρτου: Διεξάγετε δοκιμές φόρτου για να αξιολογήσετε την απόδοση του συστήματος υπό διαφορετικά επίπεδα ταυτόχρονων φορτίων εργασίας. Η δοκιμή φόρτου βοηθά στον εντοπισμό των ορίων κλιμάκωσης του συστήματος, στην ανίχνευση υποβάθμισης της απόδοσης και στη διασφάλιση ότι το σύστημα μπορεί να διαχειριστεί την αναμενόμενη κίνηση χωρίς να διακυβεύεται η απόδοση.

4. Συνεχής Παρακολούθηση: Εφαρμόστε μηχανισμούς συνεχούς παρακολούθησης και ειδοποιήσεων για την προληπτική ανίχνευση ζητημάτων απόδοσης και σημείων συμφόρησης. Δημιουργήστε πίνακες ελέγχου παρακολούθησης και ειδοποιήσεις για την παρακολούθηση βασικών δεικτών απόδοσης (KPIs) και τη λήψη ειδοποιήσεων όταν παραβιάζονται προκαθορισμένα όρια. Αυτό επιτρέπει την άμεση αναγνώριση και επίλυση προβλημάτων απόδοσης.

Στρατηγικές Κλιμάκωσης

Για τη διαχείριση αυξανόμενων φορτίων εργασίας και τη διασφάλιση της κλιμάκωσης του συστήματος ευφυούς ενορχήστρωσης ροών εργασίας, εξετάστε τις ακόλουθες στρατηγικές κλιμάκωσης:

1. Κάθετη Κλιμάκωση: Η κάθετη κλιμάκωση περιλαμβάνει την αύξηση των πόρων (π.χ., CPU, μνήμη) μεμονωμένων κόμβων ή διακομιστών για τη διαχείριση υψηλότερων φορτίων εργασίας. Αυτή η προσέγγιση είναι κατάλληλη όταν το σύστημα απαιτεί περισσότερη υπολογιστική ισχύ ή μνήμη για τη διαχείριση σύνθετων ροών εργασίας ή λειτουργιών AI.

2. Οριζόντια Κλιμάκωση: Η οριζόντια κλιμάκωση περιλαμβάνει την προσθήκη περισσότερων κόμβων ή διακομιστών στο σύστημα για την κατανομή του φόρτου

εργασίας. Αυτή η προσέγγιση είναι αποτελεσματική όταν το σύστημα πρέπει να διαχειριστεί μεγάλο αριθμό ταυτόχρονων ροών εργασίας ή όταν ο φόρτος εργασίας μπορεί εύκολα να κατανεμηθεί σε πολλαπλούς κόμβους. Η οριζόντια κλιμάκωση απαιτεί κατανεμημένη αρχιτεκτονική και μηχανισμούς εξισορρόπησης φορτίου για τη διασφάλιση ομοιόμορφης κατανομής της κίνησης.

3. Αυτόματη Κλιμάκωση: Εφαρμόστε μηχανισμούς αυτόματης κλιμάκωσης για την αυτόματη προσαρμογή του αριθμού των κόμβων ή των πόρων με βάση τη ζήτηση φόρτου εργασίας. Η αυτόματη κλιμάκωση επιτρέπει στο σύστημα να κλιμακώνεται δυναμικά προς τα πάνω ή προς τα κάτω ανάλογα με την εισερχόμενη κίνηση, διασφαλίζοντας τη βέλτιστη χρήση πόρων και οικονομική αποδοτικότητα. Πλατφόρμες cloud όπως το Amazon Web Services (AWS) ή το Google Cloud Platform (GCP) παρέχουν δυνατότητες αυτόματης κλιμάκωσης που μπορούν να αξιοποιηθούν για συστήματα ευφυούς ενορχήστρωσης ροών εργασίας.

Τεχνικές Βελτιστοποίησης Απόδοσης

Εκτός από τις στρατηγικές κλιμάκωσης, εξετάστε τις ακόλουθες τεχνικές βελτιστοποίησης απόδοσης για την ενίσχυση της αποδοτικότητας του συστήματος ευφυούς ενορχήστρωσης ροών εργασίας:

1. Αποδοτική Αποθήκευση και Ανάκτηση Δεδομένων: Βελτιστοποιήστε τους μηχανισμούς αποθήκευσης και ανάκτησης δεδομένων που χρησιμοποιούνται από τα στοιχεία ροής εργασίας. Χρησιμοποιήστε αποδοτική ευρετηρίαση βάσεων δεδομένων, τεχνικές βελτιστοποίησης ερωτημάτων και προσωρινή αποθήκευση δεδομένων για την ελαχιστοποίηση της καθυστέρησης και τη βελτίωση της απόδοσης των λειτουργιών εντάσεως δεδομένων.

2. Ασύγχρονη Είσοδος/Εξόδος: Χρησιμοποιήστε ασύγχρονες λειτουργίες εισόδου/εξόδου για την αποφυγή μπλοκαρίσματος και τη βελτίωση της ανταπόκρισης του συστήματος. Η ασύγχρονη είσοδος/έξοδος επιτρέπει στο σύστημα να διαχειρίζεται πολλαπλά αιτήματα ταυτόχρονα χωρίς να περιμένει την

ολοκλήρωση των λειτουργιών εισόδου/εξόδου, μεγιστοποιώντας έτσι τη χρήση των πόρων.

3. Αποδοτική Σειριοποίηση και Αποσειριοποίηση: Βελτιστοποιήστε τις διαδικασίες σειριοποίησης και αποσειριοποίησης που χρησιμοποιούνται για την ανταλλαγή δεδομένων μεταξύ των στοιχείων ροής εργασίας. Χρησιμοποιήστε αποδοτικές μορφές σειριοποίησης, όπως τα Protocol Buffers ή το MessagePack, για να μειώσετε την επιβάρυνση της σειριοποίησης δεδομένων και να βελτιώσετε την απόδοση της επικοινωνίας μεταξύ των στοιχείων.



Για εφαρμογές βασισμένες σε Ruby, εξετάστε τη χρήση του [Universal ID](#). Το Universal ID αξιοποιεί τόσο το MessagePack όσο και το Brotli (ένας συνδυασμός σχεδιασμένος για ταχύτητα και κορυφαία συμπίεση δεδομένων). Όταν συνδυάζονται, αυτές οι βιβλιοθήκες είναι έως και 30% ταχύτερες και με ποσοστά συμπίεσης εντός 2-5% σε σύγκριση με τα Protocol Buffers.

4. Συμπίεση και Κωδικοποίηση: Εφαρμόστε τεχνικές συμπίεσης και κωδικοποίησης για να μειώσετε το μέγεθος των δεδομένων που μεταφέρονται μεταξύ των στοιχείων ροής εργασίας. Οι αλγόριθμοι συμπίεσης, όπως το gzip ή το Brotli, μπορούν να μειώσουν σημαντικά τη χρήση του εύρους ζώνης δικτύου και να βελτιώσουν τη συνολική απόδοση του συστήματος.

Λαμβάνοντας υπόψη τις πτυχές κλιμάκωσης και απόδοσης κατά το σχεδιασμό και την υλοποίηση συστημάτων ενορχήστρωσης ευφών ροών εργασίας, μπορείτε να διασφαλίσετε ότι το σύστημά σας μπορεί να χειριστεί μεγάλους όγκους παράλληλων ροών εργασίας, να βελτιστοποιήσει την απόδοση των στοιχείων που βασίζονται στην TN και να κλιμακωθεί απρόσκοπτα για να καλύψει τις αυξανόμενες απαιτήσεις. Η συνεχής παρακολούθηση, η δημιουργία προφίλ και οι προσπάθειες βελτιστοποίησης είναι απαραίτητες για τη διατήρηση της απόδοσης και της ανταπόκρισης του συστήματος καθώς ο φόρτος εργασίας και η πολυπλοκότητα αυξάνονται με την πάροδο του χρόνου.

Έλεγχος και Επικύρωση Ροών Εργασίας

Ο έλεγχος και η επικύρωση αποτελούν κρίσιμες πτυχές της ανάπτυξης και συντήρησης συστημάτων ενορχήστρωσης ευφύων ροών εργασίας. Δεδομένης της πολύπλοκης φύσης των ροών εργασίας που βασίζονται στην TN, είναι απαραίτητο να διασφαλιστεί ότι κάθε στοιχείο λειτουργεί όπως αναμένεται, η συνολική ροή εργασίας συμπεριφέρεται σωστά και οι αποφάσεις της TN είναι ακριβείς και αξιόπιστες. Σε αυτή την ενότητα, θα εξερευνήσουμε διάφορες τεχνικές και παραμέτρους για τον έλεγχο και την επικύρωση ευφύων ροών εργασίας.

Μοναδιαίος Έλεγχος Στοιχείων Ροής Εργασίας

Ο μοναδιαίος έλεγχος περιλαμβάνει τον έλεγχο μεμονωμένων στοιχείων ροής εργασίας σε απομόνωση για την επαλήθευση της ορθότητας και της αξιοπιστίας τους. Κατά τον μοναδιαίο έλεγχο στοιχείων ροής εργασίας που βασίζονται στην TN, λάβετε υπόψη τα εξής:

- 1. Επικύρωση Εισόδου:** Ελέγξτε την ικανότητα του στοιχείου να χειρίζεται διαφορετικούς τύπους εισόδων, συμπεριλαμβανομένων έγκυρων και μη έγκυρων δεδομένων. Επαληθεύστε ότι το στοιχείο χειρίζεται ομαλά τις οριακές περιπτώσεις και παρέχει κατάλληλα μηνύματα σφάλματος ή εξαιρέσεις.
- 2. Επαλήθευση Εξόδου:** Επιβεβαιώστε ότι το στοιχείο παράγει την αναμενόμενη έξοδο για ένα δεδομένο σύνολο εισόδων. Συγκρίνετε την πραγματική έξοδο με τα αναμενόμενα αποτελέσματα για να διασφαλίσετε την ορθότητα.
- 3. Χειρισμός Σφαλμάτων:** Δοκιμάστε τους μηχανισμούς χειρισμού σφαλμάτων του δομικού στοιχείου προσομοιώνοντας διάφορα σενάρια σφαλμάτων, όπως μη έγκυρη είσοδο, μη διαθεσιμότητα πόρων ή απρόσμενες εξαιρέσεις. Επαληθεύστε ότι το δομικό στοιχείο συλλαμβάνει και χειρίζεται τα σφάλματα κατάλληλα.
- 4. Οριακές Συνθήκες:** Δοκιμάστε τη συμπεριφορά του δομικού στοιχείου υπό οριακές συνθήκες, όπως κενή είσοδος, μέγιστο μέγεθος εισόδου ή ακραίες τιμές.

Βεβαιωθείτε ότι το δομικό στοιχείο χειρίζεται αυτές τις συνθήκες ομαλά χωρίς να καταρρέει ή να παράγει εσφαλμένα αποτελέσματα.

Ακολουθεί ένα παράδειγμα ενός ελέγχου μονάδας για ένα δομικό στοιχείο ροής εργασίας σε Ruby χρησιμοποιώντας το πλαίσιο δοκιμών RSpec:

```
1 RSpec.describe OrderValidator do
2   describe '#validate' do
3     context 'when order is valid' do
4       let(:order) { build(:order) }
5
6       it 'returns true' do
7         expect(subject.validate(order)).to be true
8       end
9     end
10
11    context 'when order is invalid' do
12      let(:order) { build(:order, total_amount: -100) }
13
14      it 'returns false' do
15        expect(subject.validate(order)).to be false
16      end
17    end
18  end
19 end
```

Σε αυτό το παράδειγμα, το συστατικό OrderValidator ελέγχεται χρησιμοποιώντας δύο περιπτώσεις δοκιμών: μία για μια έγκυρη παραγγελία και μία άλλη για μια μη έγκυρη παραγγελία. Οι περιπτώσεις δοκιμών επαληθεύουν ότι η μέθοδος validate επιστρέφει την αναμενόμενη λογική τιμή με βάση την εγκυρότητα της παραγγελίας.

Έλεγχος Ενοποίησης Αλληλεπιδράσεων Ροής Εργασίας

Ο έλεγχος ενοποίησης επικεντρώνεται στην επαλήθευση των αλληλεπιδράσεων και της ροής δεδομένων μεταξύ διαφορετικών συστατικών της ροής εργασίας. Διασφαλίζει ότι τα συστατικά συνεργάζονται απρόσκοπτα και παράγουν τα

αναμενόμενα αποτελέσματα. Κατά τον έλεγχο ενοποίησης των ευφυών ροών εργασίας, λάβετε υπόψη τα εξής:

1. Αλληλεπίδραση Συστατικών: Ελέγξτε την επικοινωνία και την ανταλλαγή δεδομένων μεταξύ των συστατικών της ροής εργασίας. Επαληθεύστε ότι η έξοδος ενός συστατικού περνάει σωστά ως είσοδος στο επόμενο συστατικό της ροής εργασίας.

2. Συνέπεια Δεδομένων: Διασφαλίστε ότι τα δεδομένα παραμένουν συνεπή και ακριβή καθώς ρέουν μέσα από τη ροή εργασίας. Επαληθεύστε ότι οι μετατροπές δεδομένων, οι υπολογισμοί και οι συγκεντρώσεις εκτελούνται σωστά.

3. Διάδοση Εξαιρέσεων: Ελέγξτε πώς οι εξαιρέσεις και τα σφάλματα διαδίδονται και χειρίζονται μεταξύ των συστατικών της ροής εργασίας. Επαληθεύστε ότι οι εξαιρέσεις συλλαμβάνονται, καταγράφονται και αντιμετωπίζονται κατάλληλα για την αποφυγή διακοπής της ροής εργασίας.

4. Ασύγχρονη Συμπεριφορά: Εάν η ροή εργασίας περιλαμβάνει ασύγχρονα συστατικά ή παράλληλη εκτέλεση, ελέγξτε τους μηχανισμούς συντονισμού και συγχρονισμού. Διασφαλίστε ότι η ροή εργασίας συμπεριφέρεται σωστά σε ταυτόχρονα και ασύγχρονα σενάρια.

Ακολουθεί ένα παράδειγμα ενός ελέγχου ενοποίησης για μια ροή εργασίας σε Ruby χρησιμοποιώντας το πλαίσιο δοκιμών RSpec:

```
1 RSpec.describe OrderProcessingWorkflow do
2
3   let(:order) { build(:order) }
4
5   it 'processes the order successfully' do
6     expect(OrderValidator).to receive(:validate).and_return(true)
7     expect(InventoryManager).to receive(:check_availability).and_return(true)
8     expect(PaymentProcessor).to receive(:process_payment).and_return(true)
9     expect(ShippingService).to receive(:schedule_shipping).and_return(true)
10
11     workflow = OrderProcessingWorkflow.new(order)
12     result = workflow.process
13
14     expect(result).to be true
15     expect(order.status).to eq('processed')
16   end
17
18 end
```

Σε αυτό το παράδειγμα, το OrderProcessingWorkflow δοκιμάζεται επαληθεύοντας τις αλληλεπιδράσεις μεταξύ διαφορετικών στοιχείων της ροής εργασίας. Η περίπτωση δοκιμής καθορίζει τις προσδοκίες για τη συμπεριφορά κάθε στοιχείου και διασφαλίζει ότι η ροή εργασίας επεξεργάζεται την παραγγελία επιτυχώς, ενημερώνοντας αντίστοιχα την κατάσταση της παραγγελίας.

Δοκιμή Σημείων Απόφασης TN

Η δοκιμή των σημείων απόφασης TN είναι κρίσιμη για τη διασφάλιση της ακρίβειας και της αξιοπιστίας των ροών εργασίας που βασίζονται στην TN. Κατά τη δοκιμή των σημείων απόφασης TN, λάβετε υπόψη τα εξής:

- 1. Ακρίβεια Αποφάσεων:** Επαληθεύστε ότι το στοιχείο TN λαμβάνει ακριβείς αποφάσεις με βάση τα δεδομένα εισόδου και το εκπαιδευμένο μοντέλο. Συγκρίνετε τις αποφάσεις TN με τα αναμενόμενα αποτελέσματα ή τα δεδομένα αναφοράς.
- 2. Οριακές Περιπτώσεις:** Δοκιμάστε τη συμπεριφορά του στοιχείου TN σε οριακές

περιπτώσεις και ασυνήθιστα σενάρια. Επαληθεύστε ότι το στοιχείο TN χειρίζεται αυτές τις περιπτώσεις με χάρη και λαμβάνει λογικές αποφάσεις.

3. Μεροληψία και Δικαιοσύνη: Αξιολογήστε το στοιχείο TN για πιθανές μεροληψίες και διασφαλίστε ότι λαμβάνει δίκαιες και αμερόληπτες αποφάσεις. Δοκιμάστε το στοιχείο με διαφορετικά δεδομένα εισόδου και αναλύστε τα αποτελέσματα για τυχόν μεροληπτικά μοτίβα.

4. Επεξηγησιμότητα: Εάν το στοιχείο TN παρέχει εξηγήσεις ή συλλογισμούς για τις αποφάσεις του, επαληθεύστε την ορθότητα και τη σαφήνεια των εξηγήσεων. Βεβαιωθείτε ότι οι εξηγήσεις ευθυγραμμίζονται με την υποκείμενη διαδικασία λήψης αποφάσεων.

Ακολουθεί ένα παράδειγμα δοκιμής ενός σημείου απόφασης TN σε Ruby χρησιμοποιώντας το πλαίσιο δοκιμών RSpec:

```
1 RSpec.describe FraudDetector do
2   describe '#detect_fraud' do
3     context 'when transaction is fraudulent' do
4       let(:tx) do
5         build(:transaction, amount: 10_000, location: 'High-Risk Country')
6       end
7
8       it 'returns true' do
9         expect(subject.detect_fraud(tx)).to be true
10      end
11    end
12
13    context 'when transaction is legitimate' do
14      let(:tx) do
15        build(:transaction, amount: 100, location: 'Low-Risk Country')
16      end
17
18      it 'returns false' do
19        expect(subject.detect_fraud(tx)).to be false
20      end
21    end
22  end
23 end
```

Σε αυτό το παράδειγμα, το στοιχείο AI FraudDetector δοκιμάζεται με δύο περιπτώσεις δοκιμών: μία για δόλια συναλλαγή και μία άλλη για νόμιμη συναλλαγή. Οι περιπτώσεις δοκιμών επαληθεύουν ότι η μέθοδος detect_fraud επιστρέφει την αναμενόμενη λογική τιμή με βάση τα χαρακτηριστικά της συναλλαγής.

Έλεγχος από Άκρο σε Άκρο

Ο έλεγχος από άκρο σε άκρο περιλαμβάνει τη δοκιμή ολόκληρης της ροής εργασίας από την αρχή μέχρι το τέλος, προσομοιώνοντας πραγματικά σενάρια και αλληλεπιδράσεις χρηστών. Διασφαλίζει ότι η ροή εργασίας συμπεριφέρεται σωστά και παράγει τα επιθυμητά αποτελέσματα. Κατά την εκτέλεση ελέγχων από άκρο σε άκρο για ευφυείς ροές εργασίας, λάβετε υπόψη τα εξής:

- 1. Σενάρια Χρήστη:** Προσδιορίστε τα συνήθη σενάρια χρήστη και δοκιμάστε τη συμπεριφορά της ροής εργασίας σε αυτά τα σενάρια. Επαληθεύστε ότι η ροή εργασίας χειρίζεται σωστά τις εισόδους των χρηστών, λαμβάνει κατάλληλες αποφάσεις και παράγει τα αναμενόμενα αποτελέσματα.
- 2. Επικύρωση Δεδομένων:** Βεβαιωθείτε ότι η ροή εργασίας επικυρώνει και καθαρίζει τις εισόδους των χρηστών για την αποφυγή ασυνεπειών στα δεδομένα ή ευπαθειών ασφαλείας. Δοκιμάστε τη ροή εργασίας με διάφορους τύπους δεδομένων εισόδου, συμπεριλαμβανομένων έγκυρων και μη έγκυρων δεδομένων.
- 3. Ανάκτηση από Σφάλματα:** Δοκιμάστε την ικανότητα της ροής εργασίας να ανακάμπτει από σφάλματα και εξαιρέσεις. Προσομοιώστε σενάρια σφαλμάτων και επαληθεύστε ότι η ροή εργασίας τα χειρίζεται ομαλά, καταγράφει τα σφάλματα και λαμβάνει κατάλληλες ενέργειες ανάκαμψης.
- 4. Απόδοση και Κλιμάκωση:** Αξιολογήστε την απόδοση και την κλιμάκωση της ροής εργασίας υπό διαφορετικές συνθήκες φόρτου. Δοκιμάστε τη ροή εργασίας με μεγάλο όγκο ταυτόχρονων αιτημάτων και μετρήστε τους χρόνους απόκρισης, τη χρήση πόρων και τη συνολική σταθερότητα του συστήματος.

Ακολουθεί ένα παράδειγμα δοκιμής από άκρο σε άκρο για μια ροή εργασίας σε Ruby χρησιμοποιώντας το πλαίσιο δοκιμών RSpec και τη βιβλιοθήκη Capybara για την προσομοίωση αλληλεπιδράσεων χρήστη:

```
1 RSpec.describe 'Order Processing Workflow' do
2   scenario 'User places an order successfully' do
3     visit '/orders/new'
4     fill_in 'Product', with: 'Sample Product'
5     fill_in 'Quantity', with: '2'
6     fill_in 'Shipping Address', with: '123 Main St'
7     click_button 'Place Order'
8
9     expect(page).to have_content('Order Placed Successfully')
10    expect(Order.count).to eq(1)
11    expect(Order.last.status).to eq('processed')
12  end
13 end
```

Σε αυτό το παράδειγμα, η δοκιμή από άκρο σε άκρο προσομοιώνει έναν χρήστη που κάνει μια παραγγελία μέσω της διεπαφής ιστού. Συμπληρώνει τα απαιτούμενα πεδία της φόρμας, υποβάλλει την παραγγελία και επιβεβαιώνει ότι η παραγγελία υποβάλλεται επιτυχώς σε επεξεργασία, εμφανίζοντας το κατάλληλο μήνυμα επιβεβαίωσης και ενημερώνοντας την κατάσταση της παραγγελίας στη βάση δεδομένων.

Συνεχής Ενσωμάτωση και Ανάπτυξη

Για να διασφαλιστεί η αξιοπιστία και η συντηρησιμότητα των ευφυών ροών εργασίας, συνιστάται η ενσωμάτωση των δοκιμών και της επικύρωσης στη διοχέτευση συνεχούς ενσωμάτωσης και ανάπτυξης (CI/CD). Αυτό επιτρέπει την αυτοματοποιημένη δοκιμή και επικύρωση των αλλαγών ροής εργασίας πριν από την ανάπτυξή τους στο περιβάλλον παραγωγής. Εξετάστε τις ακόλουθες πρακτικές:

1. Αυτοματοποιημένη Εκτέλεση Δοκιμών: Ρυθμίστε τη διοχέτευση CI/CD ώστε να εκτελεί αυτόματα τη σουίτα δοκιμών όποτε γίνονται αλλαγές στον κώδικα της ροής

εργασίας. Αυτό διασφαλίζει ότι τυχόν παλινδρομήσεις ή αποτυχίες εντοπίζονται νωρίς στη διαδικασία ανάπτυξης.

2. Παρακολούθηση Κάλυψης Δοκιμών: Μετρήστε και παρακολουθήστε την κάλυψη δοκιμών των στοιχείων ροής εργασίας και των σημείων λήψης αποφάσεων AI. Στοχεύστε σε υψηλή κάλυψη δοκιμών για να διασφαλίσετε ότι οι κρίσιμες διαδρομές και τα σενάρια ελέγχονται διεξοδικά.

3. Συνεχής Ανατροφοδότηση: Ενσωματώστε τα αποτελέσματα των δοκιμών και τις μετρήσεις ποιότητας κώδικα στη ροή εργασίας ανάπτυξης. Παρέχετε συνεχή ανατροφοδότηση στους προγραμματιστές σχετικά με την κατάσταση των δοκιμών, την ποιότητα του κώδικα και τυχόν ζητήματα που εντοπίστηκαν κατά τη διαδικασία CI/CD.

4. Περιβάλλοντα Δοκιμών: Αναπτύξτε τη ροή εργασίας σε περιβάλλοντα δοκιμών που αντικατοπτρίζουν στενά το περιβάλλον παραγωγής. Εκτελέστε πρόσθετες δοκιμές και επικύρωση στο περιβάλλον δοκιμών για να εντοπίσετε τυχόν ζητήματα που σχετίζονται με την υποδομή, τη διαμόρφωση ή την ενσωμάτωση δεδομένων.

5. Μηχανισμοί Επαναφοράς: Υλοποιήστε μηχανισμούς επαναφοράς σε περίπτωση αποτυχιών ανάπτυξης ή κρίσιμων ζητημάτων που εντοπίζονται στην παραγωγή. Βεβαιωθείτε ότι η ροή εργασίας μπορεί να επανέλθει γρήγορα σε μια προηγούμενη σταθερή έκδοση για την ελαχιστοποίηση του χρόνου διακοπής λειτουργίας και των επιπτώσεων στους χρήστες.

Με την ενσωμάτωση δοκιμών και επικύρωσης σε όλο τον κύκλο ζωής ανάπτυξης των ευφυνών ροών εργασίας, οι οργανισμοί μπορούν να διασφαλίσουν την αξιοπιστία, την ακρίβεια και τη συντηρησιμότητα των συστημάτων τους που βασίζονται στην τεχνητή νοημοσύνη. Οι τακτικές δοκιμές και η επικύρωση βοηθούν στον εντοπισμό σφαλμάτων, την πρόληψη παλινδρομήσεων και την οικοδόμηση εμπιστοσύνης στη συμπεριφορά και τα αποτελέσματα της ροής εργασίας.

Μέρος 2: Τα Πρότυπα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Μηχανική Προτροπών

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Αλυσίδα Σκέψης

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παραδείγματα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παραγωγή Περιεχομένου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Δημιουργία Δομημένων Οντοτήτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Καθοδήγηση Πράκτορα LLM

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Οφέλη και Παράμετροι

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Εναλλαγή Λειτουργίας

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πότε να το Χρησιμοποιήσετε

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ανάθεση Ρόλου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πότε να το Χρησιμοποιήσετε

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παραδείγματα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Αντικείμενο Προτροπής

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πρότυπο Προτροπής

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Οφέλη και Παράμετροι προς Εξέταση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πότε να το Χρησιμοποιήσετε:

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Δομημένη Είσοδος/Εξοδος

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Κλιμάκωση Δομημένης Εισόδου/Εξόδου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Οφέλη και Παράμετροι προς Εξέταση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Αλυσίδωση Προτροπών

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πότε να τη Χρησιμοποιήσετε

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα: Διαδικασία Ένταξης της Olympia

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Επανασυντάκτης Προτροπών

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Οριοθέτηση Απόκρισης

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Οφέλη και Παράμετροι προς Εξέταση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Χειρισμός Σφαλμάτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Αναλυτής Ερωτημάτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Υλοποίηση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Επισήμανση Μερών του Λόγου (POS) και Αναγνώριση Ονομαστικών Οντοτήτων (NER)

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ταξινόμηση Πρόθεσης

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Εξαγωγή Λέξεων-Κλειδιών

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Οφέλη

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Επανασυντάκτης Ερωτημάτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Οφέλη

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ventriloquist

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πότε να το Χρησιμοποιήσετε

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Διακριτά Συστατικά

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Κατηγορήμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πότε να το Χρησιμοποιήσετε

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πρόσοψη API

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Βασικά Πλεονεκτήματα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πότε να το Χρησιμοποιήσετε

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ταυτοποίηση και Εξουσιοδότηση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Χειρισμός Αιτημάτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Μορφοποίηση Απαντήσεων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Χειρισμός Σφαλμάτων και Οριακές Περιπτώσεις

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ζητήματα Κλιμάκωσης και Απόδοσης

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Σύγκριση με Άλλα Πρότυπα Σχεδίασης

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Result Interpreter

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πότε να το Χρησιμοποιήσετε

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Εικονική Μηχανή

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πότε να το Χρησιμοποιήσετε

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πίσω από τη Μαγεία

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προδιαγραφή και Δοκιμές

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προσδιορισμός της Συμπεριφοράς

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Σύνταξη Περιπτώσεων Δοκιμών

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα: Δοκιμή του Στοιχείου Μεταφραστή

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Επανάληψη Αλληλεπιδράσεων HTTP

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Άνθρωπος στον Βρόχο (HITL)

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Μοτίβα Υψηλού Επιπέδου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Υβριδική Νοημοσύνη

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προσαρμοστική Ανταπόκριση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Εναλλαγή Ρόλων Ανθρώπου-TN

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Κλιμάκωση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Βασικά Οφέλη

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πρακτική Εφαρμογή: Υγειονομική Περίθαλψη

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Βρόχος Ανατροφοδότησης

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Εφαρμογές και Παραδείγματα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προηγμένες Τεχνικές Ενσωμάτωσης Ανθρώπινης Ανατροφοδότησης

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παθητική Ακτινοβολία Πληροφοριών

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προβολή Πληροφοριών Βάσει Περιεχομένου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προληπτικές Ειδοποιήσεις

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Επεξηγηματικές Γνώσεις

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Διαδραστική Εξερεύνηση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Βασικά Οφέλη

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Εφαρμογές και Παραδείγματα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Συνεργατική Λήψη Αποφάσεων (CDM)

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Συνεχής Μάθηση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Εφαρμογές και Παραδείγματα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ηθικά Ζητήματα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ο Ρόλος του HITL στον Μετριάσμό των Κινδύνων AI

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Τεχνολογικές Εξελίξεις και Μελλοντικές Προοπτικές

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προκλήσεις και Περιορισμοί των Συστημάτων HITL

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ευφυής Χειρισμός Σφαλμάτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παραδοσιακές Προσεγγίσεις Χειρισμού Σφαλμάτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Διάγνωση Σφαλμάτων Βάσει Περιβάλλοντος

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Μηχανική Προτροπών για Διάγνωση Σφαλμάτων Βάσει Περιεχομένου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παραγωγή Ενισχυμένη με Ανάκτηση για Διάγνωση Σφαλμάτων Βάσει Περιεχομένου

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ευφυής Αναφορά Σφαλμάτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προληπτική Αποτροπή Σφαλμάτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Έξυπνη Ανάκαμψη από Σφάλματα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Εξατομικευμένη Επικοινωνία Σφαλμάτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προσαρμοστική Ροή Εργασίας Διαχείρισης Σφαλμάτων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Έλεγχος Ποιότητας

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Eval

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πρόβλημα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Λύση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράμετροι προς εξέταση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Κατανόηση των Χρυσών Προτύπων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς λειτουργούν οι αξιολογήσεις χωρίς αναφορά

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προστατευτικό Πλαίσιο

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πρόβλημα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Λύση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Πώς Λειτουργεί

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Παράδειγμα

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Ζητήματα προς εξέταση

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Προστατευτικά πλαίσια και Αξιολογήσεις: Δύο όψεις του ίδιου νομίσματος

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Η Εναλλαξιμότητα των Προστατευτικών πλαισίων και των Αξιολογήσεων χωρίς αναφορά

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Εφαρμογή Διπλού Σκοπού Δικλίδων Ασφαλείας και Αξιολογήσεων

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Γλωσσάριο

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Γλωσσάριο

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

A

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

B

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

C

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

D

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

E

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

F

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

G

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

H

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

I

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

J

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

K

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

L

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

M

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

N

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

O

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Π

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Q

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

R

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

S

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

T

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

U

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

V

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

W

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Z

Αυτό το περιεχόμενο δεν είναι διαθέσιμο στο δείγμα του βιβλίου. Το βιβλίο μπορεί να αγοραστεί στο Leanpub στο <http://leanpub.com/patterns-of-application-development-using-ai-el>.

Index

- AI, 84, 112, 152, 161
 - conversational, 236
 - model, 112
 - εφαρμογές, 142
 - μοντέλο, 175, 177
- Alpaca, 15
- Altman, Sam, 20
- Amazon Web Services, 280
- Anthropic, 25, 45, 84, 146, 155
- APIs, 82, 140, 172
- BERT, 15, 27
- Brotli, 281
- Byte Pair Encoding (BPE), 16
- C (Γλώσσα Προγραμματισμού), 132
- Capybara library, 288
- chatbots εξυπηρέτησης πελατών, 38
- ChatGPT, 33, 60
- Claude, 9, 49, 88
- Claude 3, 56, 144, 146, 153, 155
- Claude 3 Opus, 85
- Claude v1, 19
- Claude v2, 19
- Cohere (Πάροχος LLM), 25, 38
- concurrent workflows, 281
- Customer Sentiment Analysis, 113
- Datadog, 276
- decision
 - making capabilities, 113
- Dohan, et al., 49
- errors
 - Intelligent Error Handling, 161
- F#, 105
- Facebook, 27
- FitAI, 236
- Gemma 7B, 12
- Generative UI (GenUI), 229, 230
- GitLab, 105
- Google, 25
 - API, 72, 74
 - Cloud Platform, 280
 - Gemini, 24
 - Gemini 1.5 Pro, 15, 19, 21
 - PaLM (Pathways Language Model), 19
 - PaLM (Μοντέλο Γλώσσας Pathways), 27
 - T5, 15
 - Πλατφόρμα Cloud AI, 27
- GPT-3, 14, 19
- GPT-4, 7, 14, 19, 24, 35, 49, 56, 72, 119, 133, 136, 144, 151, 228, 229, 278

- Graham, Paul, 21
- GraphQL, 122
- Groq, 29, 136
- gzip, 281
- Hohpe, Gregor, 118
- Honeybadger, 107
- HTTP, 169
- input
 - prompts, 64
 - validation, 282
- intelligent workflow orchestration, 281
- iterative refinement, 162
- JSON (JavaScript Object Notation), 144,
 - 148, 153, 166, 186
- K-μέσων, 138
- Large Language Model (LLM), 162
- Llama, 15
- Llama 2-70B, 57
- Llama 3 70B, 12
- Llama 3 8B, 12
- Louvre, 48
- Managed Streaming for Apache Kafka, 46
- Markdown, 165
- Memorial Sloan Kettering Cancer Center,
 - 46
- MessagePack, 281
- Meta, 27
- Metropolitan Museum of Art, 48
- Mistral, 28
 - 7B, 12
 - 7B Instruct, 19, 229
- Mixtral
 - 8x22B, 12
 - 8x7B, 64
- New Relic, 279
- Ollama, 28
- Olympia, 38, 72, 145, 161, 170, 186
- OpenAI, 4, 25, 45, 84
- OpenRouter, 30, 32, 170, 278
- output verification, 282
- Perplexity (Πάροχος), 13
- prompts
 - engineering, 64
 - Prompt Template, 229
- Protocol Buffers, 281
- PyTorch, 27
- Qwen2 70B, 12
- Rails, 219
- Railway Oriented Programming (ROP), 107
- Raix, 256
 - βιβλιοθήκη, 110
- Response Fencing, 229
- RSpec, 283, 284, 288
- Ruby, 105, 106, 128, 182, 288
- Ruby on Rails, 1, 126, 255, 263
- Rudall, Alex, 26
- Rust (Γλώσσα Προγραμματισμού), 105, 132

- Scout, 279
- Stripe, 146
- Structured IO, 229
- system directive, 112
- T5, 27
- Together.ai, 29
- Unicode-encodable language, 16
- Universal ID, 281
- Wall, Larry, 3
- Wisper, 107, 121, 170, 177
- Wooley, Chad, 105
- XML, 152
- Yi-34B, 57
- Άνθρωπος στον Βρόχο (HITL), 200
- Έξυπνος Συντονιστής Περιεχομένου, 259
- Αλυσίδα Σκέψης (Chain of Thought), 51
- Αλυσίδα Σκέψης (CoT), 156
- Αξιολόγηση και Διαστρωμάτωση
 Συμπτωμάτων, 115
- Αρχιτεκτονική Μικροϋπηρεσιών, 101
- Αυτοθεραπευόμενα Δεδομένα, 183, 270
- Αυτόματη Συνέχιση, 179
- Αφελής Μπέιζ, 137
- Γεννητική Διεπαφή Χρήστη (GenUI), 234
- Γεννητικός Προ-εκπαιδευμένος
 Μετασχηματιστής (GPT), 77
- Δειγματοληψία Top-k, 54
- Δειγματοληψία Top-p (πυρήνα), 54
- Διαστρωμάτωση Κινδύνου, 116
- Διαχειριστής Διαδικασιών, 121
 Enterprise Integration, 255
- Διαχειριστής Διεργασιών, 118
- Διεπαφή Χρήστη
 σχεδιασμός, 244
- Διεπαφή Χρήστη (UI)
 διεπαφές, 222, 239
 πλαίσια, 239
 τεχνολογίες, 234
- Διερμηνέας Αποτελεσμάτων, 160
- Δρώντα, 36
- Δυναμική Δρομολόγηση Εργασιών, 250
- Δυναμική Επιλογή Εργαλείων, 148
- Εγγαστρίμυθος, 197
- Εξαγωγή συμπερασμάτων, 6
- Εξαναγκασμένη Επιλογή Εργαλείων, 149
- Επαλήθευση Ασφάλισης, 115
- Ερμής (Ρωμαίος θεός), 50
- Ερμής (πλανήτης), 50
- Ερμής (χημικό στοιχείο), 50
- Εφαρμογές Ηλεκτρονικού Εμπορίου, 104
- Θερμοκρασία, 61
- Καθαρισμός Κειμένου, 127
- Καθολικό Κλειδίωμα Διερμηνέα (GIL), 131
- Κωδικοποίηση Ζευγών Byte (BPE), 14
- Λανθάνουσα Κατανομή Ντίριχλε, 138
- Μάθηση Μίας Εκμάθησης, 69
- Μεγάλα Γλωσσικά Μοντέλα (LLM), 222
- Μεγάλα Γλωσσικά Μοντέλα (MGM), 1
- Μεγάλα Μοντέλα Γλώσσας (LLM), 183
- Μεγάλο Γλωσσικό Μοντέλο (LLM), 86,

- 142, 234
- Μεγάλο Γλωσσικό Μοντέλο (ΜΓΜ), 20,
33, 77, 78, 136, 163, 209
τοπίο, 30
- Μεγάλο Μοντέλο Γλώσσας (LLM), 140,
152, 158, 228
- Μεγάλο Μοντέλο Γλώσσας (ΜΜΓ), 18
- Μηχανές Διαनुσμάτων Υποστήριξης
(ΜΔΥ), 137
- Μοντέλο Μεγάλης Γλώσσας (LLM), 125,
165, 186, 258
- Μοντέλο Μεγάλης Γλώσσας (ΜΜΓ), 4, 82,
88, 99
- Οριοθέτηση Απόκρισης, 197
- Παραγωγή Ενισχυμένη με Ανάκτηση
(RAG), 35, 90, 142
- Παραγωγή Επαυξημένη με Ανάκτηση
(RAG), 43
- Παραγωγή με Ενίσχυση Ανάκτησης
(RAG), 52
- Παραγωγική Διεπαφή Χρήστη (GenUI),
222, 239, 243
- Παραγωγικότητα, 213
- Πλήθος Εργατών, 186
- Πλήθος Πρακτόρων, 135
- Ποινή Παρουσίας, 55
- Πολλαπλοί Πράκτορες
Επιλυτές Προβλημάτων, 35
- Πολυτροπικά
γλωσσικά μοντέλα, 23
μοντέλα, 22
- Ποσοτικοποίηση, 32
- Προ-εκπαιδευμένος Μετασχηματιστής
Γέννησης (GPT), 9
- Προτάσεις Προϊόντων, 104
- Πρότυπα Ενσωμάτωσης Επιχειρήσεων,
118
- Συλλογή Ιατρικού Ιστορικού, 114
- Συνεχής Ενσωμάτωση και Ανάπτυξη
(CI/CD), 288
διοχέτευση, 288
- Συνεχής Παρακολούθηση Κινδύνου, 117
- TN, 74, 145, 169, 234
εφαρμογές, 156, 168, 181
μοντέλο, 101, 235
σημεία απόφασης, 285
συνομιλητική, 7, 35
σύνθετα συστήματα, 34, 38
- Τεχνητή Νοημοσύνη, 227
- Τεχνητή νοημοσύνη
μοντέλο, 174
- Υποστήριξη Κλινικών Αποφάσεων, 117
- Χρόνος έως το Πρώτο Σύμβολο (ΧΕΠΣ),
31
έλεγχος από άκρο σε άκρο, 287, 288
έλεγχος ενοποίησης, 283
έλεγχος και συμμόρφωση, 274
έξυπνα τηλέφωνα, 244
έξυπνη ενορχήστρωση ροής εργασίας, 278
αισιόδοξο κλείδωμα, 124
αλληλεπιδράσεις τύπου παιχνιδιού
ρόλων, 7
αλυσίδωση εργατών ΑΙ, 126
ανάθεση εισιτηρίων, 266

- ανάλυση συναισθήματος, 19, 113, 127–129,
133, 134, 152, 163
- ανάπτυξη εφαρμογών, 246
- ανίχνευση απάτης
σύστημα, 110
- αναγνώριση θεμάτων, 137
- ανατροφοδότηση
Κύκλος Ανατροφοδότησης, 67
- ανθρωπομορφισμός, 78
- αντιστοίχιση προτύπων, 171
- απάντηση κλειστών και ανοιχτών
ερωτήσεων, 60
- απαισιόδοξο κλειδίωμα, 124
- αποδοτικότητα, 248
- αποσφαλμάτωση, 251
και αντιμετώπιση προβλημάτων, 274
και δοκιμές, 149
- αποτυχία κλήσης συνάρτησης, 151
- αποφάσεις
δέντρα, 247
- απόδοση
βελτιστοποίηση, 150, 274
προβλήματα, 279
συμβιβασμοί, 6
- αρθρωτότητα, 100
- αρχή ελάχιστων προνομίων, 82
- αρχιτεκτονική transformer, 7
- αρχιτεκτονική βασισμένη σε συμβάντα,
123
- αρχιτεκτονική εταιρικών εφαρμογών, 43
- αρχιτεκτονική λογισμικού, 2
- ασύγχρονη επεξεργασία, 277
- αυτοπαλινδρομική μοντελοποίηση, 49
- αυτόματη κλιμάκωση, 280
- βάσεις γνώσεων, 8
- βάσεις δεδομένων, 140
-αντικείμενο υποστηριζόμενο, 119
στρατηγικές κλειδώματος, 124
- βάση γνώσεων της Olympia, 103
- βασικά μοντέλα, 61
- βασικά πρότυπα, 249
- βελτιστοποίηση
απόδοσης, 220
- γλωσσικά
μοντέλα, 75, 83
- γλώσσα
Ανίχνευση Γλώσσας, 127
εργασίες, 5
μοντέλα, 48
- γραμμή εντολών
Διεπαφή Γραμμής Εντολών (CLI), 29
- γραμματικοί κανόνες, 5
- γραμμική άλγεβρα, 48
- γραμμική παλινδρόμηση, 49
- γραφικά μοντέλα, 49
- γυαλιά επαυξημένης πραγματικότητας,
244
- δεδομένα
Ανάκτηση Δεδομένων, 124
Επικύρωση Δεδομένων, 287
Συγχρονισμός Δεδομένων, 124
ακεραιότητα, 267
ανάλυση, 38, 165
διαδικασία επεξεργασίας, 267

- διατήρηση, 124
- εργασίες επεξεργασίας, 143
- προετοιμασία, 123
- προστασία, 30, 241
- ροή, 125
- δεδομένα εκπαίδευσης, 48
- δημιουργική γραφή, 38, 60
- διαδικασία απόσταξης, 87
- διαδικτυακοί έμποροι λιανικής, 229
- διατήρηση και εναλλαγή αρχείων
 - καταγραφής, 275
- διατροφική παραγωγή, 25
- διαχείριση γνώσης, 36
- διαχείριση κυκλοφορίας, 37
- διεθνοποίηση, 218
- διεκπεραιωτική ικανότητα, 31
- διεπαφές φωνητικού ελέγχου, 38
- δοκιμές χρηστών και ανατροφοδότηση,
 - 221
- δομημένα δεδομένα, 152
- δομημένη καταγραφή, 275
- δυναμική παραγωγή διεπαφής χρήστη,
 - 210
- δόμηση αφήγησης, 22
- εικονικοί βοηθοί, 38
- εισαγωγές SQL, 81
- εκπαίδευση με οδηγίες
 - μοντέλα εκπαιδευμένα με οδηγίες, 59
- εκπαιδευτικές εφαρμογές, 36
- εμπειρία χρήστη, 218
- εμπιστοσύνη χρήστη, 242
- εννοιολογικές και πρακτικές προκλήσεις,
 - 223
- ενορχήστρωση έξυπνης ροής εργασίας,
 - 255
- ενσωμάτωση ΜΓΜ, 210
- εξατομίκευση, 211, 243, 249
 - Εξατομικευμένες Φόρμες, 224
 - Εξατομικευμένο Μικροκείμενο, 230
- εξατομικευμένες προτάσεις προϊόντων,
 - 104
- εξωτερικές υπηρεσίες ή API, 143
- επαναληπτική βελτίωση, 86
- επεξεργασία ροής, 169, 175, 181
 - λογική, 177
- επεξηγησιμότητα, 286
- επιστήμη των υπολογιστών, 83
- επιστήμη υπολογιστών, 80
- επιτραπέζιοι υπολογιστές, 244
- επιχειρησιακοί κανόνες, 247
- ευελιξία και δημιουργικότητα, 220
- ευφυής ενορχήστρωση ροών εργασίας,
 - 246
- εφαρμογή chatbot, 135
- εφοδιαστική αλυσίδα
 - βελτιστοποίηση, 37
- ηθική
 - επιπτώσεις, 223
- ηλεκτρονικό εμπόριο, 215, 247
- θεωρία του νου, 45
- ιατρικές ανακαλύψεις, 114
- ιδιότητες ACID, 124
- ιστορικά μοτίβα, 251
- καθοδήγηση

- μηχανική, 240
- καθυστέρηση, 31
- κατάτμηση σε σύμβολα, 14
- καταγραφή ελέγχου, 121
- κατανεμημένη αρχιτεκτονική, 277
- κλήση εργαλείου, 172
- κλιμάκωση, 249, 276
- λήψη
 - αποφάσεων περιπτώσεις, 151
- λανθάνων χώρος, 45, 48
- λεκτικές μονάδες, 6
- λεξικά, 148
- λεπτομερής καταγραφή, 275
- λογαριασμός, 103
- λογική διακόπτη κυκλώματος, 181
- μάθηση με λίγα παραδείγματα, 71
- μάθηση μηδενικής εκμάθησης, 68
- μάθηση μηδενικού δείγματος, 67
- μέθοδος finalize, 177, 178
- μέθοδος οριστικοποίησης, 175
- μήνυμα ενεργοποίησης, 118
- μεροληψία
 - και δικαιοσύνη στην TN, 286
- μετάφραση, 19, 219
- μη επιβλεπόμενη μάθηση, 5
- μηχανισμοί επανάληψης, 125
- μηχανισμοί επαναφοράς, 289
- μικροπροσαρμογή, 91
- μοντέλα βασισμένα στην ανάκτηση, 8
- μοντέλο OPT, 27
- νευρωνικά δίκτυα, 4, 7
- ντετερμινιστική συμπεριφορά, 66
- οδηγία συστήματος, 145
- οικοσύστημα, 166
- ολοκλήρωση υψηλής απόδοσης, 29
- ομαδική επεξεργασία, 278
- ομαδοποίηση εγγράφων, 137
- οπτική διεπαφή, 233
- οριακές περιπτώσεις, 67
- οριακές συνθήκες, 283
- πάροχοι φιλοξενίας μοντέλων ανοιχτού κώδικα, 229
- πίνακας κατακερματισμού, 171
- πίνακες, 148
- παράγοντες κινδύνου, 108, 109
- παράλληλη εκτέλεση, 277
- παράμετροι
 - εισόδου, 145
- παράμετρος
 - Αριθμός Παραμέτρων, 31
 - επιδράσεις, 146
 - εύρος, 12
- παραφραση, 61
- παραγωγή συνθετικών δεδομένων, 60
- παρακολούθηση
 - και ειδοποίηση, 253
 - και καταγραφή, 125, 273
 - μετρήσεις, 275
- παρακολούθηση βασικών μετρικών, 271
- πειραματισμός
 - πλαίσιο, 217
- περίληψη, 60
- περιβάλλον

- Δημιουργία Περιεχομένου βάσει
 - Περιβάλλοντος, 215, 216
- Παραγωγή Περιεχομένου βάσει
 - Περιβάλλοντος, 209
 - άπειρα μεγάλες είσοδοι, 18
 - παράθυρο, 17
- περιβάλλοντα δοκιμών, 289
- περιεχόμενο
 - Επαύξηση, 52
 - Κατηγοριοποίηση Περιεχομένου, 127
 - λήψη αποφάσεων βάσει
 - περιεχομένου, 251
 - παράθυρο, 251
 - φιλτράρισμα, 30
- περιεχόμενο που δημιουργείται από
 - χρήστες, 126
- πιθανοτικά μοντέλα, 48
- πλαίσια ανάπτυξης, 167
- πλαίσιο
 - Παραγωγή Περιεχομένου Βάσει
 - Συμφραζομένων, 223
- πληροφορίες
 - ανάκτηση, 8, 143
 - εξαγωγή, 60
- ποινές επανάληψης, 58
- προβλέψεις, 6
- προβλήματα χρηστικότητας, 241
- προοδευτική αποκάλυψη, 231
- προσαρμογή, 30
- προσαρμοστική διεπαφή χρήστη, 232
- προσαρμοστική ροή εργασίας
 - Προσαρμοστική Σύνθεση Ροής
 - Εργασίας, 251
- προσβασιμότητα, 242, 243
- προσωρινή αποθήκευση, 277
- προτροπές
 - Αντικείμενο Προτροπής, 85
 - Απόσταξη Προτροπών, 52, 83, 89, 278
 - Πρότυπο Προτροπών, 68
 - αλυσίδωση, 68, 82
 - βελτίωση, 78
 - μηχανική, 46, 51, 52, 68, 74, 76
 - σχεδιασμός, 66, 78
- προτροπή με λίγα παραδείγματα, 72
- ροή δεδομένων, 171
- ροή εργασίας πολλαπλών βημάτων, 126
- σήμανση τύπου markup, 81
- σημεία
 - απόφασης, 271
- σημεία συμφόρησης, 251
- στένεψε το μονοπάτι, 44
- στένωση της διαδρομής, 43
- στοίβα ELK, 125
- στρατηγικές εφεδρείας, 125
- στρατηγικές κατάτμησης και στόχευσης,
 - 217
- στρατηγικές παρακίνησης, 238
- συμβάντα αποστελλόμενα από
 - διακομιστή (SSE), 169
- συμπεριληπτικές διεπαφές, 224
- συμφραζόμενα
 - Παραγωγή Περιεχομένου βάσει
 - Συμφραζομένων, 224
 - Προτάσεις Πεδίων βάσει

- Συμφραζομένων, 225
- συνάρτηση
 - ιστορικό κλήσεων, 175
 - κλήση, 140, 141
 - ονόματα, 173
- συνάφεια
 - Συναφής Δημιουργία Περιεχομένου, 214
- συνέπεια
 - και αναπαραγωγιμότητα, 150
- συναισθηματικός τόνος, 163
- συναρτησιακός προγραμματισμός, 104
- συνδεσιμότητα δικτύου, 252
- συνεργατικό φιλτράρισμα, 104
- συνομιλία
 - βρόχος, 178
 - καταγραφή, 176, 178
- συντακτικά σφάλματα, 149
- συντονισμός οδηγίων, 11
 - μοντέλα εκπαιδευμένα σε οδηγίες, 56
- συστήματα δημοσίευσης-συνδρομής, 123
- συστήματα ερωταποκρίσεων, 8
- σφάλματα
 - ανάκαμψη, 287
 - διαχείριση, 124
 - ποσοστά, 125
 - χειρισμός, 121, 160, 282
- σχεδιασμός αντιμετώπισης έκτακτων αναγκών, 37
- σχεδιασμός εφαρμογών και πλαίσια εργασίας, 223
- σύγχρονες εφαρμογές, 249
- σύμβολα, 14
- σύνθετες εργασίες, 164
- σύνολα, 134
 - σύνολο πρακτόρων, 134
- σύνολα μοντέλων, 133
- ταμπλέτες, 244
- ταξινομητές κατάταξης, 40
- ταξινόμηση, 60, 137
- τοπικά περιβάλλοντα ανάπτυξης, 174
- τραγωδία των κοινών, 214
- υλικό, 31
- υπάλληλοι Databricks, 60
- υπερπαράμετρος, 53
- υποστήριξη πελατών, 36
- φιλτράρισμα βάσει περιεχομένου, 104
- φυσική γλώσσα
 - Επεξεργασία Φυσικής Γλώσσας (NLP), 114
 - Επεξεργασία Φυσικής Γλώσσας (ΕΦΓ), 136
- χειρισμός εξαιρέσεων, 252, 254
- χειριστές ροής, 170
- χειροκίνητη παρέμβαση, 254
- χρήση εργαλείων, 140, 141, 168
- χρόνος επεξεργασίας, 125
- χωρίς κατάσταση, 176
- ψηφιακό τοπίο, 217
- ψηφοφορία πλειοψηφίας, 133
- ψυχολογία χρήστη, 240