
Alana: Social Dialogue using an Ensemble Model and a Ranker trained on User Feedback

**Ioannis Papaioannou, Amanda Cercas Curry, Jose L. Part
Igor Shalyminov, Xinnuo Xu, Yanchao Yu
Ondřej Dušek, Verena Rieser, Oliver Lemon**
The Interaction Lab, Department of Computer Science
Heriot-Watt University, Edinburgh, EH14 4AS, UK
{i.papaioannou, o.lemon}@hw.ac.uk

Abstract

We describe our Alexa prize system (called ‘Alana’) which consists of an ensemble of bots, combining rule-based and machine learning systems, and using a contextual ranking mechanism to choose system responses. This paper reports on the version of the system developed and evaluated in the semi-finals of the competition (i.e. up to 15 August 2017), but not on subsequent enhancements. The ranker for this system was trained on real user feedback received during the competition, where we address the problem of how to train on the noisy and sparse feedback obtained during the competition. In order to avoid initial problems of inappropriate and boring utterances coming from big datasets such as Reddit and Twitter, we later focussed on ‘clean’ data sources such as news and facts. We report on experiments with different ranking functions and versions of our NewsBot. We find that a multi-turn news strategy is beneficial, and that a ranker trained on the ratings feedback from users is also effective. Our system continuously improved using the data gathered over the course over the competition (1 July – 15 August). Our final user score (averaged user rating over the whole semi-finals period) was 3.12, and we achieved 3.3 for the averaged user rating over the last week of the semi-finals (8-15 August 2017). We were also able to achieve long dialogues (average 10.7 turns) during the competition period. In subsequent weeks, after the end of the semi-final competition, we have achieved our highest scores of 3.52 (daily average, 18th October), 3.45 (weekly average on 23 and 24 October), and average dialogue lengths of 14.6 turns (1 October), and median dialogue length of 2.25 minutes (average for 7 days on 10th October).

1 Introduction

Early systems for social chat, such as ELIZA (Weizenbaum, 1966), were based on carefully handwritten rules, but recent systems are often trained using a variety of (deep) learning techniques over large public data sets, such as OpenSubtitles or Twitter, e.g. (Vinyals & Le, 2015; Sordoni et al., 2015; Li et al., 2016). However, learning directly from data also has its pitfalls when deploying a system to real customers, as recent examples such Microsoft’s Tay bot demonstrate. We present a hybrid model, incorporating hand-crafted rules (validated and developed through customer feedback) and machine learning models trained on carefully chosen datasets.

Following previous hybrid systems, e.g. (Yu et al., 2016), we apply a ranker model to select the most relevant reply from a pool of replies generated by an ensemble of different agents/bots. It is still an open question how to best define this ranking function. Previous work has manually defined an evaluation function based on hand-selected turn-level features (Yu et al., 2016; Li et al., 2016). Other work has experimented with learning from crowdsourced user ratings (Lowe et al., 2017). One major

drawback of such previous work is that it only evaluates a possible response locally, i.e. per turn, rather than considering its contribution to the overall dialogue outcome, e.g. to engage the user. As such, these ranking functions often favour safe, but dull responses (Lowe et al., 2017).

We experimented with a variety of ranking functions and datasets as described below. This resulted in one of the top bots in the competition.

2 Overall Vision for Our Bot

Our overall vision was to create a humorous and engaging social chatbot that aims to keep users interested and enjoying a spoken interaction on topics of their choice for as long as possible. Our overarching inspiration for this vision is the kind of conversation that two new acquaintances might have in a pub – a mixture of topic-related chat, finding out about each other, and sharing amusing facts, jokes, stories, and items of news.

As such, our social chatbot was designed to have the following behaviour:

1. It should be able to engage in **open-domain topic-based** conversations, to minimise responses such as “I don’t know what you mean” or “I can’t answer that”.
2. The replies should sound **natural** and **non-repetitive**.
3. The replies should be **engaging** and **informative**, i.e. stimulate further conversation.

For learning open-domain topic-based conversation, we built several retrieval bots and also built on sequence-to-sequence (Seq2Seq) models (Sordoni et al., 2015; Vinyals & Le, 2015). We train vector-based models on large data sets such as Twitter or OpenSubtitles. To handle the specific topic-related aspects of this year’s Alexa challenge (e.g chat about baseball, celebrities, etc.), we also investigated combining these models with the large volumes of topic-tagged data, such as Reddit topics / sub-Reddits, and automatic summaries of online news.

3 System Design and Architecture

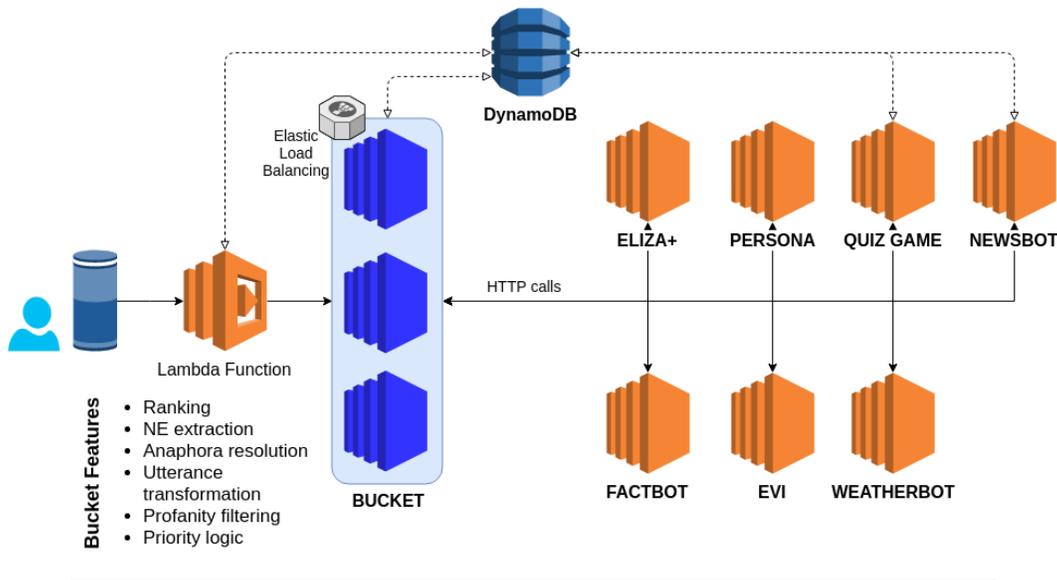


Figure 1: Alana is a hybrid hierarchical architecture with ranking

The system architecture is shown in Fig. 1. The user utterance is transformed into a JSON object, called *request*, using an AWS lambda function. The lambda function also analyses the user utterance to extract keywords in order to classify the user goal into an *intent*. We only use 2 AWS intents in the skill schema: *StopIntent* (which captures any form of stop-related phrases) and *GetAnswer* (which captures the full text of the user utterance). The *request* object also contains several other

metadata, such as the timestamp and the tokenized ASR confidence scores, which are used further down the pipeline. All this information is also stored in a DynamoDB table and forwarded to a load-balanced Amazon EC2 instance, henceforward called the *Bucket* (see Section 3.1).

In the *Bucket*, additional information is extracted from the sentence and forwarded along with the preprocessed utterance to an ensemble of bots (see Section 4). Each of these bots runs on a different EC2 instance, following a modular architecture for better performance as well as easier and faster maintenance of any component. The communication between each response bot and the bucket is done via simple HTTP GET calls. Each bot then returns one or more candidate responses to the *Bucket*. The *Bucket* then selects (and post-processes) one of the responses for output via the ranker.

3.1 The Bucket

The *Bucket* is an EC2 instance which runs the main logic of Alana. As explained above, every user utterance is sent to the *Bucket* in the form of a JSON request, which contains the full user utterance and selected metadata (session ID, confidence scores, etc.). Utterances with low ASR confidence result in an apology, confirmation of what was heard, and request for repetition.¹

All utterances with sufficient ASR confidence are preprocessed as follows:

- **Yes/No/Maybe utterances are transformed to full sentences.** We resolve elliptical yes/no user replies (e.g. *Yes.*, *No*, *I don't* etc.) by transforming their utterance into a full sentence in context (e.g. if in the last turn the system asked “*Do you like tea*”, and the user responds with a “*Yes*”, the response will be transformed into “*Yes I do like tea*”). The individual bots (see Section 4) are then queried using the preprocessed full-sentence utterances.
- **Indirect sentences are transformed to direct.** Indirect user questions such as “*I don't know who X is*” are transformed into direct questions such as “*Who is X*”. This helps the factual information retrieval bot (EVI, see Section 4) to answer these queries.

We also directly extract some additional intents from the user utterance using regular expressions, such as *StopIntent* (requests to stop the conversation), *TimeIntent* (requests for current time) and *RepeatIntent* (request to repeat the previous system utterance).²

The user utterance is then further annotated using Stanford part-of-speech tagger (Toutanova et al., 2003) and named entity recognizer (NER) (Finkel et al., 2005). The annotation is saved into DynamoDB along with dialogue history. Furthermore, basic *anaphora resolution* is performed using the last *person* or *location* named entity mentioned in the dialogue (see examples in Section 6).

The preprocessed and annotated utterance, as well as the last 3 turns of dialogue history, are then forwarded to all bots (see Section 4) using simple HTTP GET calls. Each response returned by the bots is added to the list of possible response candidates in the *Bucket*.

All the returned candidates are postprocessed and normalized. Profanity, single-word and repetitive (news only) candidates are filtered out. The response is selected in three steps:

1. **Bot priority list.** Some of the deployed bots are prioritized, i.e. if they produce a response, it is always selected. The priority order is the following: *Quiz game*, *Factbot*, *Weatherbot*, *Persona*, *Evi* (see Section 4).
2. **Contextual priority.** The NewsBot's response is prioritized if it stays on the topic of a previously mentioned news story.
3. **Ranking function.** If none of the priority bots produced an answer, the rest of the deployed bots' responses populate the list of candidates and the best response is selected via a ranking function (see Section 5).

In the extreme case where none of the bots produced an answer (or all of them were filtered out due to postprocessing rules), the system returns a random fun fact, produced by the *Factbot*.

¹We average the tokenized confidence scores included in the *requests* metadata.

²The *StopIntent* handled here covers more varied expressions and allows more flexibility than the Amazon built-in intent *StopIntent*, which is handled directly by the AWS lambda function.

4 Bots in the Ensemble

The bots in the ensemble can roughly be divided into two main categories:

1. **Data-driven Bots:** We experimented with retrieval based bots as well as generative Sequence-to-Sequence models (Seq2Seq, see section 4.1.2) While the former always produce well-formed sentences (as retrieved from the data set), the latter can generate new and possibly more contextually appropriate replies, however at the expense on needing larger data sets to learn from. We follow previous work by combing both paradigms into an ensemble-based approach (Song et al., 2016).
2. **Rule-based bots** are used to respond to the specific user queries in a controlled and consistent way, e.g. to queries about the personality of our bot (e.g. favourite things etc.) or the weather, using a combination of in-house developed bots and extended versions of 3rd party bots.

These two categories include the following bots:

Persona: A rule-based system implemented in AIML³ whose main purpose is to maintain personality-based responses consistent across turns, such as music tastes or other preferences. *Persona* also includes replies to other topics, where we want to guarantee an appropriate response to inappropriate user utterances and topics such as sex, as per the competition rules.

Eliza: We extended an existing Eliza-style chatbot called *Rosie*.⁴ Since the initial *Rosie* bot was designed for mobile devices, we heavily altered it for the Challenge.

NewsBot: An information retrieval bot based on an open-source framework Lucene.⁵ We build and continuously populate a search index of selected news sources provided via NewsAPI <https://newsapi.org>. For indexing as well as for the bot's responses, we use summaries of the news articles extracted with an open-source library called *Sumy*.⁶ In order to select a relevant piece of news for a user's query, we create 1, 2, and 3-grams over the query and dialogue context. We employ the BM25 algorithm to score news relevance, with named entities and noun phrases from the user query boosted using a set of weights adjusted empirically. A re-ranking step is then applied for the top 10 candidates based on the articles' recency. A multi-turn variant was also developed (see Section 7.1).

Factbot – Fun facts, Jokes, and Stories: A collection of facts, jokes and stories that get triggered whenever the user specifically asks for them or as a deflection strategy when no suitable response is found. For the fun facts, the user can also specify a named entity (“*Tell me a fact about X*”). Otherwise, a fact is chosen randomly. The data was collected from a multitude of online resources.

Quiz Game: A hand-crafted system developed using a VoiceXML. The user has to guess the right answer to topic-specific questions (e.g. 80's music, science, history, sport and geography). The user can end the game at any point. This quiz game has been removed for the Alexa finals.

EVI: A third party bot retrieving factual information (if applicable) about the user utterance, powered by the EVI question answering engine API, <https://www.evi.com/>) This bot returns only one candidate. Some EVI answers which would not be appropriate in a dialogue are filtered out.

Weatherbot: A simple rule-based bot that provides the user with weather-related information, if asked for, querying the *OpenWeatherMap API* (<https://openweathermap.org/>) on the fly.

4.1 Other Bots and Data

We also experimented with other data-driven bots, which were not included in the semi-final system.

4.1.1 Data Sets for Information Retrieval Bots

- **OpenSubtitles**(Lison & Tiedemann, 2016), with the automatic turn segmentation provided by Lison & Meena (2016). We used all dialogues of two or more turns and filtered the data as described below.

³<http://www.alicebot.org/aiml.html>

⁴<https://github.com/pandorabots/rosie>

⁵<https://lucene.apache.org>

⁶<https://pypi.python.org/pypi/sumy>

- **Cornell Movies, Jabberwacky, CNN:** these datasets proved to be too small for our purposes: Cornell Movie Dataset (Danescu-Niculescu-Mizil & Lee, 2011), Jabberwacky chatbot chat logs⁷, and CNN chat show transcripts from (Yu et al., 2016, 2017).

In order to comply with the competition rules, we first filtered the data for profanities. However, profanities are often context-dependent and hard to capture by a purely lexicon-driven approach (see Section 9 for more details). As such, we experimented with restricting the OpenSubtitles data set using age ratings of the movies. We obtained movie ratings from IMDb and only included in our dataset the movies with a U.S. “G” or U.K. “U” ratings (“general”, “universal”). Another problem from OpenSubtitles data was the occurrence of many personal names and other named entities that would appear out-of-context in a dialogue. We used Stanford NER (Finkel et al., 2005) to detect named entities and filtered out all context-response pairs containing named entities in the response. However, the downside of this approach is that we ended up with much smaller data sets which made data-driven approaches, such as the generative Seq2Seq approach less feasible.

4.1.2 Seq2Seq

Throughout system development, we experimented with a sequence-to-sequence dialogue model (Vinyals & Le, 2015), training it on several datasets. The first promising behaviour was obtained with Twitter data⁸: it was interesting and mostly grammatical yet often offensive and politically related. We then switched to a subset of Reddit logs – over 21,000 conversation snippets in the form of question-answer pairs cleaned from profanity and filtered to only contain small-talk conversation (thanks to Dr. Zhuoran Wang). In order to exclude ungrammatical responses, we disregarded all answers with a low confidence score (defined as the sum of the logits at the decoder’s output). We adjusted the confidence threshold empirically on a separate development set of 100 sample user utterances both collected from WoChat⁹ transcripts and paraphrased from a daily list of popular topics provided by Amazon. The experiment thus resulted in a casual conversation bot: its answers are supposed to be given at times when the user is following up on the previous system’s answer or just hesitating. Due to time constraints, the final version of the seq2seq bot was not deployed into production, and so its possible contribution to the users’ ratings is left for future work.

5 Ranking Functions

The responses proposed by each bot are ranked according to a set of features. We have experimented with several ranking functions.

5.1 Hand-engineered Ranker function

The hand-engineered ranking function uses the following features:

- **Coherence:** Following (Li et al., 2016), we reward semantic similarity between the user’s utterance and the candidates using Word2Vec (Mikolov et al., 2013)
- **Flow:** Also similar to (Li et al., 2016), we penalise similarity between consecutive system utterances in order to prevent repetition. Here, we use both Word2Vec and METEOR word n-gram overlap as measures of similarity.
- **Questions:** By promoting questions, we aim to incite the user to continue the conversation.
- **Named Entities:** We strongly reward utterances containing the same named entities as the user’s reply to promote candidates relating to the same topic.
- **Noun Phrases:** Similarly, we reward matching noun phrases between the user’s and the system’s utterances. Noun phrases are identified based on part-of-speech tagging (see Section 3.1).
- **Dullness:** We compare each response to a list of dull responses such as “I don’t know” and penalise Word2Vec similarity between them, since we would like the bot’s utterances to be engaging, similarly to (Li et al., 2016).
- **Topic Divergence:** We trained a Latent Dirichlet Allocation (LDA) model on a weighted combination of preprocessed versions of the OpenSubtitles and the WashingtonPost datasets. We set the

⁷<http://www.jabberwacky.com/>

⁸https://github.com/Marsan-Ma/chat_corpus

⁹<http://workshop.colips.org/wochat/>

vocabulary size to $20k$ and the number of topics to 200, and we used a tailored stop-words list. For every proposed answer in the bucket, we compute the topic divergence from the user utterance.

- **Sentiment Polarity:** We use the VADER sentiment analyser (Gilbert & Hutto, 2014) from the NLTK toolkit,¹⁰ which provides a floating point value indicating sentence sentiment.

These features are calculated using the last two system turns in order to maintain dialogue context. The final score is a weighted sum of these features:

$$\begin{aligned} score = & 0.25 * turn_0 + 0.25 * turn_1 + 0.25 * turn_2 + 0.25 * noun_phrases \\ & + 3 * named_entities - 0.25 * topic_divergence \end{aligned} \quad (1)$$

where $turn_i$ is computed using the i -th utterance counting from the end of the dialogue history:

$$\begin{aligned} turn_i = & -0.2 * flow_{sem_similarity} - 3 * flow_{METEOR} + 0.1 * coherence_{sem_similarity} \\ & - 0.24 * dullness + 0.2 * question + 0.1 * sentiment_polarity \end{aligned} \quad (2)$$

5.2 Linear Classifier Ranker

In order to use the feedback ratings obtained from real users in the competition, we also trained the VowpalWabbit linear classifier (Langford et al., 2007) to rank Bucket responses based on the following features:

- bag-of-n-grams from the context (preceding 3 utterances) and the response (unigrams, bigrams, and trigrams)
- position-specific n-grams at the beginning of the context and the response (first 5 positions)
- dialogue flow features, same as for the hand-engineered ranker (see Section 5.1)
- bot name.

The ranker is trained as a binary classifier, but it outputs a floating-point score in practice. At runtime, the highest-scoring response is selected for the output.

We initially trained the ranker on Cornell movies, Twitter, and Jabberwacky datasets (see Section 4.1.1), with positive examples from the real dialogues and negative ones randomly sampled from the rest of the set, but the ranker only learned to prefer responses similar to data from these datasets; its performance in real dialogues was lacking in our tests. Therefore, after collecting enough live dialogues during the Alexa Prize competition, we retrained the ranker on *real dialogues collected during the competition*. The rating target function is an approximation of human ratings – we use all context-response pairs from successful dialogues (human rating 4 or 5) as positive examples (value +1) and all pairs from unsuccessful dialogues (rating 1 or 2) as negative (value -1) and train the ranker to mimic this rating.

We collected 60k dialogue instances over one month for training and 7k dialogue instances over 4 days as a development set. We did not perform any large-scale parameter optimization, but based on performance on the development data, we selected the following VowpalWabbit parameters:

- logistic loss function (logistic regression),
- feature concatenations (context + response n-grams, pairs of n-grams from responses, bot name + response n-grams, bot name + context n-grams, bot name + dialogue flow, bot name + context n-grams + response n-grams),
- 16-bit feature hash table,
- 1 pass over the training data.

This setup reached 69.40% accuracy in classifying the development data items as positive or negative. The results of deploying this Linear Ranker are presented in section 7.2.

6 An Example Dialogue

Note: The dialogue presented here does not come from real customer data, but was recreated by interacting with our system (running a text-based version on Telegram). The same structure of

¹⁰<http://www.nltk.org/api/nltk.sentiment.html>

interaction and Named Entities are used as occurred in a real dialogue. We also indicate which bot in our ensemble generated the answer.

A: Hi, [...] What would you like to talk about?
U: music
A: (<i>Persona</i>) Great! Who is your favorite singer?
U: Bob Dylan
A: (<i>Evi + driver</i>) Bob Dylan is an American songwriter, singer, painter, and writer. What are your opinions on Bob Dylan?
U: When was he born?
A: (<i>Evi + anaphora resolution</i>) Bob Dylan was born on Saturday May 24th 1941.
U: What's happening with him ?
A: (<i>News-multi-turn + anaphora resolution</i>) I heard this on the radio – Was Dylan too freewheelin' in borrowing for Nobel lecture? The whiff of plagiarism is blowin' in the wind for Bob Dylan. Want to know more?
U: sure
A: (<i>News-multi-turn</i>) It seems that you are interested in this topic. I'm so glad. Here's more. Here's Dylan: Some men who receive injuries are led to God, others are led to bitterness The cases Pitzer found are not blatant or explicit – there are no verbatim sentences, only identical phrases and similar phrasing. What are your opinions on Bob Dylan?
U: I love him!
A: (<i>Persona</i>) Great! What else do you like?
U: cooking
A: (<i>Facts</i>) I heard that every day, Americans eat an estimated 18 acres of pizza.

7 Comparative Experiments and Evaluation

We analysed the logs of our system every day, which provided inspiration for new features as well as issues for debugging. User feedback was used to train a Linear Ranker (see section 5.2).

We ran several experiments during the Challenge, focussing on ranking and multi-turn news. It was not possible to run controlled experiments within the Challenge setup, due to many factors such as changing user population, a changing news index, and mixture of within- and between- subjects design. Nevertheless, we present data below that indicates performance of several system features.

7.1 Improving Coherence: Multi-turn News Results

In order to test our hypothesis that user ratings can be improved by having more coherent multi-turn dialogues, we deployed two multi-turn variants of the NewsBot: (1) Alana v1.1, which presents summary headline information to users, and then determines whether to keep or change the topic based on user feedback, and (2) Alana v1.2, which produces additional news relevant to utterances from the Factbot. We found that v1.1 resulted in an improvement of the overall bot ratings. However, users rated v1.2 slightly lower, possibly because news stories were being overproduced.

To assess the extent to which ratings had been affected by the multi-turn news, we calculated the relative score for each version, by dividing the average score of dialogues containing multi-turn news by the average score of all dialogues. The relative score for each system is shown in Figure 2. We can see that even though the overall score for v1.2 decreased, the relative score of dialogues containing multi-turn news increased. This shows that the decrease in overall score was caused by dialogues without multi-turn news. We attribute this to improved coherence in dialogues where the multi-turn news were triggered compared to dialogues with only single-turn news.

Figure 3 shows that the average number of turns of consecutive news also increased from around 1 to 2.5. Both of these metrics demonstrate that users accepted multi-turn dialogues.

7.2 Trained Linear Ranker

The Linear Ranker, trained on the user feedback received during the competition (see Section 5.2), was deployed on top of Alana v1.1, and evaluated in comparison to the hand-crafted ranking function (see Section 5.1). The results are shown in Table 1. This shows that we can continuously improve

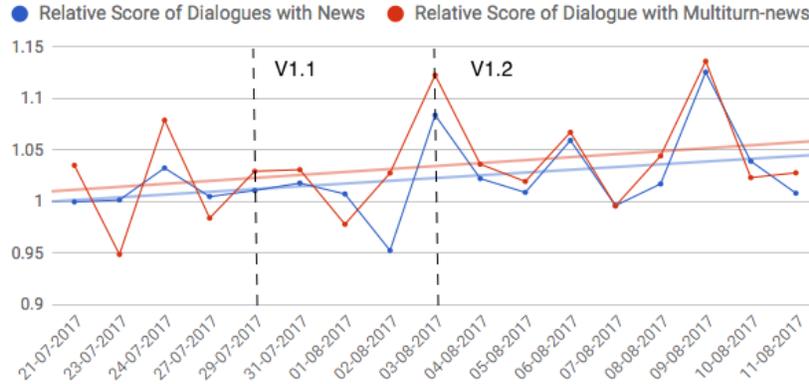


Figure 2: Relative Score of Dialogues with News/Multiturn-News. Y-axis denotes the relative score of dialogues with news/Multiturn-News, calculated by the average score of dialogues with news/multi-turn news divided by the average score of all dialogues on that date. X-axis is the date.

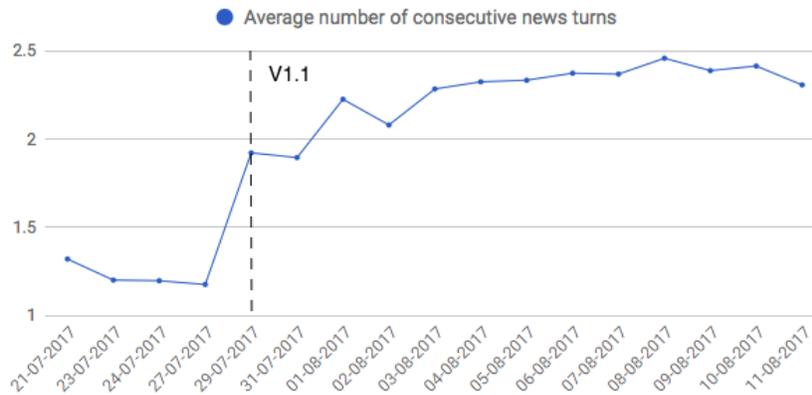


Figure 3: The average number of turns of consecutive News. The Y-axis denotes the average number of turns of consecutive news, and the X-axis is the date.

system performance by training on real customer feedback from the competition, even though it is noisy and sparse (ratings are only available for whole dialogues, and not each dialogue turn).

8 Overall Results: Alexa Challenge Leaderboard

Over the whole semi-finals competition, from 1 July, our bot obtained the scores shown in Table 2 and Figure 4 on the Challenge Leaderboard. This shows our system to be very competitive, being ranked 3rd overall in terms of average score over the whole semi-finals period. At the end of the competition, we were also 3rd in terms of average score over the final week (8-15 august, average score of 3.3), and we had the 2nd highest rating on the final day (15th august, score 3.59).

In subsequent weeks, after the end of the semi-final competition, we have achieved our high scores of 3.45 (weekly average on 23 and 24 October), average dialogue lengths of 14.6 turns (1 October), and median dialogue length of 2.25 minutes (average for 7 days on 10th October).

9 Further Lessons and Observations

Politics can be a very polarising topic, which needs careful manual handling in *Persona*. For example, we observed that customers gave us a low rating not because the conversation was bad, but they felt that they did not agree with a political view that our bot was perceived to have, even when it was simply reporting news stories. We finally settled on not providing any opinion on this

System	average user rating	number of dialogues
Alana v1.1 : Hand-engineered Ranker	3.26	191
Alana v1.1 : Trained Linear Ranker	3.28	272

Table 1: Results: Trained Linear Ranker

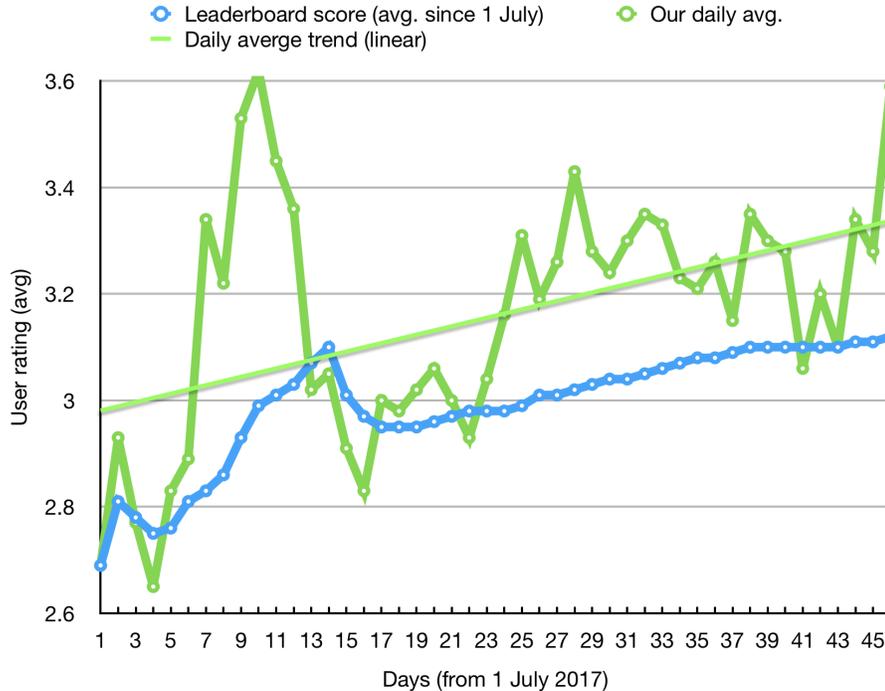


Figure 4: Semi-final results: Heriot-Watt Alana system

topic (“*Sorry, I wasn’t programmed to have an opinion on Trump. What do you think of him?*”).

Problems with Data-driven Systems: One major drawback of training on data was the lack of control regarding the system’s response, which often conflicted with the competition rules, such as giving financial advice. For example, an early version of our system would tell a customer to “*sell, sell, sell*” after mentioning his/her stocks. As described above, we filtered out offensive language, such as swearing, sexual comments, etc. However, offensive remarks cannot always be identified using a lexicon-based approach, but its meaning often manifests itself through contextual usage of language. For example, our system said it could “*sleep with any women I want to*”, where the phrase “*sleep with*” can also be used in many different and more innocent contexts. As such, we could only use restrictive, but small data sets (see 4.1.1) for the competition, where we were able to guarantee that such language would not be used.

Conversation Drivers: After delivering news, fun facts, etc., we then prompt the user in order to drive the conversation forward e.g. (“*What do you think about that?*”). We experimented with different versions for more social dialogue. However, in all cases, we observed that the users rarely engaged with the question. We concluded that the primary user interest is to be entertained (rather than challenged in a discussion) and moved towards actively promoting fun facts or delivering news.

10 Future work

This paper reports on the version of the system developed and evaluated in the semi-finals of the competition (i.e. up to 15 August 2017), but not subsequent enhancements. In the Amazon Alexa Challenge finals, we have replaced the linear ranker with a neural model, trained on the increased number of user ratings we were able to collect from July-October 2017. We also deployed a version of this socialbot on the Pepper robot platform, using a mixture of task-based and social chat functions

	avg. user rating (whole competition)	avg. duration (whole competition)	avg. number of turns (whole competition)	Last 7 days avg. (L7D)
Alana system	3.12	2.02	10.3	3.3

Table 2: Semi-final Results: Heriot-Watt Alana system: 1 July – 15 August 2017

(Papaioannou & Lemon, 2017; Papaioannou et al., 2017), to both help and entertain users in shopping situations. In September 2017 this system was deployed in a supermarket in Edinburgh for 8 days, interacting with real customers, and was filmed for a BBC documentary. We also plan to enter further upgrades of Alana, for example using an upgraded Neural ranker and seq2seq bot, into future competitions such as DSTC and other conversational agent challenges.

References

- Danescu-Niculescu-Mizil, Cristian and Lee, Lillian. Chameleons in Imagined Conversations: A New Approach to Understanding Coordination of Linguistic Style in Dialogs. In *Proc. CMCL*, pp. 76–87, 2011.
- Finkel, Jenny Rose, Grenager, Trond, and Manning, Christopher. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. ACL*, pp. 363–370, 2005.
- Gilbert, C. J. and Hutto, Erric. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *AAAI Conference on Weblogs and Social Media*, pp. 216–225, Ann Arbor, MI, USA, 2014.
- Langford, John, Li, Lihong, and Strehl, Alex. Vowpal wabbit online learning project, 2007.
- Li, Jiwei, Monroe, Will, Ritter, Alan, Galley, Michel, Gao, Jianfeng, and Jurafsky, Dan. Deep Reinforcement Learning for Dialogue Generation. In *Proc. EMNLP*, 2016.
- Lison, Pierre and Meena, Raveesh. Automatic Turn Segmentation for Movie & TV Subtitles. In *2016 IEEE Workshop on Spoken Language Technology*, 2016.
- Lison, Pierre and Tiedemann, Jörg. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proc. LREC*, Portorož, Slovenia, 2016.
- Lowe, Ryan, Noseworthy, Michael, Serban, Iulian, Angelard-Gontier, Nicolas, Bengio, Yoshua, and Pineau, Joelle. Towards an automatic turing test: Learning to evaluate dialogue responses. In *ACL*, 2017.
- Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
- Papaioannou, Ioannis and Lemon, Oliver. Combining chat and task-based multimodal dialogue for more engaging hri: A scalable method using reinforcement learning. In *Proc. HRI '17 Companion*, 2017.
- Papaioannou, Ioannis, Dondrup, Christian, Novikova, Jekaterina, and Lemon, Oliver. Hybrid chat and task dialogue for more engaging hri using reinforcement learning. In *Proc. RO-MAN*. IEEE, 2017.
- Song, Yiping, Yan, Rui, Li, Xiang, Zhao, Dongyan, and Zhang, Ming. Two are better than one: An ensemble of retrieval- and generation-based dialog systems. *CoRR*, abs/1610.07149, 2016.
- Sordani, Alessandro, Galley, Michel, Auli, Michael, Brockett, Chris, Ji, Yangfeng, Mitchell, Margaret, Nie, Jian-Yun, Gao, Jianfeng, and Dolan, Bill. A neural network approach to context-sensitive generation of conversational responses. In *Proc. NAACL-HLT*, 2015.
- Toutanova, Kristina, Klein, Dan, Manning, Christopher D., and Singer, Yoram. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc NAACL-HLT*, pp. 173–180, 2003.
- Vinyals, Oriol and Le, Quoc. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.
- Weizenbaum, Joseph. ELIZA – A Computer Program For the Study of Natural Language Communication Between Man and Machine. *Communications of the ACM*, 9(1):36–35, 1966.
- Yu, Zhou, Xu, Ziyu, Black, Alan W, and Rudnicky, Alex I. Strategy and Policy Learning for Non-Task-Oriented Conversational Systems. In *Proc. SIGDIAL*, Los Angeles, CA, USA, 2016.
- Yu, Zhou, Black, Alan W., and Rudnicky, Alexander I. Learning Conversational Systems that Interleave Task and Non-Task Content. In *Proc. IJCAI*, Melbourne, Australia, 2017. arXiv:1703.00099.