Transfer Learning to Create and Understand Modular Content

Joshua Haley, Alejandro Carbonara & Jeremiah T. Folsom-Kovarik Ph.D.
Soar Technology, Inc.
{joshua.haley, alejandro.carbonara, jeremiah}@soartech.com

ABSTRACT

Machine Learning (ML) has provided innovative insights to empower automated content analysis and discovery. In the commercial space, sophisticated Deep Learning (DL) neural models enable better search queries as systems start to understand more about the content being searched. Recently, an AI system was able to comb through a large dataset of scientific literature to discover a novel antibiotic. While it might seem that understanding content is a solved problem, these breakthroughs have come at a cost: the large amount of data required for training the neural networks to correctly process technical terms and specialized language. As is often the case for military applications, however, the amount of data available is far more limited, leading to this question: How can we leverage recent advances to understand content with sparse data and train the warfighter?

The authors present a system that uses key advances in textual Deep Neural Embeddings to leverage transfer learning from a larger corpus in order to automatically understand specialized training content. This automation of understanding allows enhanced automated meta-data to be annotated with fine granularity without increased bookkeeping for content developers. Training content can be automatically indexed, annotated, and modularized for presentation in different forms, aiding in training content reuse, maintenance, and adaptive delivery without additional workload on instructional content designers. Individual varied instruction is envisioned in the Department of Defense future warfighter training systems, and creating and using fine-grained meta-data for instructional content is necessary to support this goal. Using fine-grained meta-data helps answer how content can be modularized, what learners can be expected to know after using the content, and which content should be presented to optimally teach and train learners with different contexts, backgrounds, and performance.

ABOUT THE AUTHORS

Joshua Haley has a Master of Science in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida. His expertise is in conducting applied Machine Learning research within Department of Defense use cases in support of the warfighter with a particular focus on computational understanding. Josh is currently a Doctoral student at the University of Central Florida in the Modeling and Simulation program.

Alejandro U. Carbonara is an Artificial Intelligence Scientist at Soar Technology, Inc. He has a Master of Science in Computer Science from Carnegie Mellon, with a focus on data privacy in machine learning models. His current research focuses on combining traditional and modern techniques to better allow systems to parse natural language in human understandable ways.

Jeremiah T Folsom-Kovarik, Ph.D. is the Senior Scientist for Artificial Intelligence Exploration research within the Intelligent Systems business area at Soar Technology, Inc. His research combines modern data science and machine learning approaches with SoarTech's 20+ years of experience in modeling expert knowledge and human reasoning in order to make machine learning effective and efficient under real-world conditions. He earned his Ph.D. in Computer Science from the University of Central Florida.

Transfer Learning to Create and Understand Modular Content

Joshua Haley, Alejandro Carbonara, & Jeremiah T. Folsom-Kovarik, Ph.D.
Soar Technology, Inc.
{joshua.haley, alejandro.carbonara, jeremiah}@soartech.com

INTRODUCTION

Intelligent Tutoring Systems (ITS) are defined as automated learning systems in which pedagogical principles have been encoded into a curriculum for instruction and feedback to learners (Anderson, Boyle, & Reiser, 1985). In many cases, the ITS user interaction presents merely as a changing of modality of instructional delivery, from a classroom collective setting to an individual online setting. While online delivery increases the flexibility and timing of instruction, without learner-driven or system-driven adaptation the instruction does not fully utilize the potential of one-on-one interaction between the student and content.

This one-on-one interaction is maximally effective in adaptive instructional systems (AIS), which can be defined as "artificially-intelligent, computer based systems that guide learning experiences by tailoring instruction and recommendations based on the goals, needs, and preferences of each individual learner or team in the context of domain learning objective" (Sottilare & Brawner, 2018). AIS enhances ITS by providing an intelligent and dynamic interaction facilitated by Artificial Intelligence (AI), in contrast to interactions based only on instructional content. An ITS decouples students from a group setting and allows them different speeds of progression through coursework. In addition to personalized progression, an AIS also can choose among the individually varied paths to mastery that are possible with modular, variable training content.

In order to inform the intelligent recommender systems that tailor instruction to individual needs (Sottilare, Stensrud, & Hampton, 2019), instructional content requires a greater amount of meta-data. In particular, at minimum, a mapping between content and its related instructional competency is required. Additional meta-data such as subtopics, crosscutting topics and material level (i.e., introductory vs. advanced) unlock additional dimensions for AISs to leverage. This meta-data represents yet another burden on instructional designers to create and maintain if manual annotation is required. Additionally, this meta-data may be outside the instructional designer's capability to determine. While a diving instructor can develop a lesson to train a warfighter in diving, it is an additional burden to also expect them to fully understand pedagogical science or software data requirements on that lesson introduced by the AIS.

Additionally, large corpora of training material created by instructional designers over the years exist for traditional classroom settings. Simply digitizing the presentation of learning material and encoding it into a new ITS may change the modality of content presentation, but it does not enable additional benefits of increased training transfer (Folsom-Kovarik, Mostafavi, et al., 2019), individualized adaptation (Bell & Sottilare, 2019), and appropriate presentation (Folsom-Kovarik, Chen, Mostafavi, & Brawner, 2019) for more effective learning. Instead, the learning material needs additional augmentation of meta-data to let learners, instructors, and AIS recommenders understand what and how the available content can teach or assess. As COVID-19 has led many traditional institutions (i.e., K-12, colleges) to expand their use of online instruction, an opportunity exists to exceed the status quo education capability by ensuring training content meets AIS requirements.

While traditional instructional delivery has large volumes of training data, meta-data is lacking; local content such as classroom lessons or exercises are built, but not annotated, and may change often. Additionally, content may consist of intractably indivisible and uniform monoliths that are hard to modularize or deliver as smaller, stand-alone, variable pieces that combine into a coherent learning experience. Our solution is to use ML to understand, segment (draw module boundaries around) and annotate materials. The ML generates enhanced content meta-data using a Natural Language Processing (NLP) computational understanding system. The system indexes content from authoritative and secondary sources across many topics, and maps them to the competencies within the domain with minimal instructor overhead. This automated extraction of meta-data enables the search and reuse of content, and makes new content modules maximally exploitable by AIS. Enhanced meta-data at a level of topical granularity beyond that of a traditional competency

module will enable advanced levels of individual tailoring not previously feasible due to meta-data limitations. When expert knowledge adds structure to machine learning algorithms, the combination can enable machine learning under real-world constraints: when available data is small, concepts evolve over time, or nontechnical users need to understand and control the machine learning.

User Story

The two primary users of generated meta-data are instructional designers and recommendation systems. A trainer who's been tasked to develop content for a course acts as an instructional designer. For example, this could be a Diver tasked with teaching new divers basic "out of air procedures." In a traditional classroom setting, the trainer might develop a slide-based presentation to lead the students through the content, and practice examples for reinforcement. However, if the trainer has access to the existing manuals and resources that have been processed by the automated meta-data extraction system, he is able to search the authoritative sources for pertinent parts of training manuals about "out of air procedures," retrieve the relevant modular content, and construct his lesson reusing existing doctrine as a starting point and filling in gaps as needed. Additionally, any new content that the trainer develops can use the same computational understanding models to automatically generate the applicable meta-data on the new modules without manual effort. Computational content understanding has reduced his workload in two ways. First, it has enabled easier search of the official materials to find reusable content, minimizing the amount of new content he needs to create. Second, it has allowed the trainer to create new content and focus on his area of expertise, not on pedagogical bookkeeping.

The same trainer later might be tasked with developing a specific course for a new learner group that might not ordinarily receive individualized training, such as divers preparing to capture a 3D image inside a shipwreck. Rather than presenting generic content or developing new content from scratch, he can search the existing content for the overlap between "out of air procedures", a defined competency, and "3D scan," which is not a defined competency but a topic discovered by ML. Even if there are no simple overlaps that contain both keywords verbatim, topics relevant to both entries such as "enclosed space" can be found to focus and individualize training. This automated exploration extends beyond domain-specific content; it can index additional content from the National Oceanographic and Atmospheric Administration (Joiner, 2001) and recommend supplementary content for additional examples that match the trainer's needs.

The second envisioned user of the generated meta-data is an intelligent recommendation engine or recommender integrated into an AIS. Intelligent recommendation systems within an AIS can only be as effective as their meta-data around instructional content allows (Folsom-Kovarik, Chen, et al., 2019). For example, a recommender might need to teach algebra. If all we have is a curriculum of lessons, the recommender must follow the curriculum and provide the lessons in sequential order. If, instead, we have a collection of modules each teaching a topic and a competency framework, now the recommender can offer a choice among several topics based on learner mastery, interest, or job relevance. Additional dimensions of meta-data such as introductory, advanced, and remedial material for the same topic provide the recommender with options on how to teach with adaptation to user proficiency. Finally, if ML has discovered fine-grained meta-data about background, context, and how topics are presented, the recommender can offer content framed in ways that are familiar to the learner or are tailored to student interests, e.g., recommend the quadratics lesson that includes an air pressure example because it is relevant to the student as a diver. Learning is promoted when it is connected with existing knowledge and skill and when it is relevant to the learner (Merrill, 2002).

METHODOLOGY

Natural Language Processing is a set of tools and algorithms that leverage the knowledge of language in their design and use (Jurafsky & Martin, 2008). Over the past decade, an increasing number of state-of-the-art NLP algorithms have begun to incorporate more data-driven approaches, as opposed to purely ontological approaches, into their algorithms. More recently, a great deal of research has been done to bring NLP into the scope of Deep Learning. The primary advances in the last several years have been semi-supervised representational design to embed words into a numeric vector space, a method that came to prominence with Google's word2vec (Yin & Schütze, 2013). Several common NLP tasks include speech recognition,

dialog management, information extraction, and content characterization. For this work, we are primarily interested in the tasks of information extraction and characterization as a way to produce pedagogically useful meta-data.

In order to ingest Portable Document Format (PDF) content into our models, a parser was developed using Google's Tesseract optical character recognition (OCR) library (Smith, 2007) and a custom image extraction technique using DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) to detect likely diagrams. The importance of extracting the figures and images before preforming recognition is necessitated by text content within figures adding noise to the parsed text. While our prototype system was limited to PDFs, additional content such as Moodle modules, a popular learning management system (Dougiamas & Taylor, 2003), could be similarly parsed to generate automated meta-data.

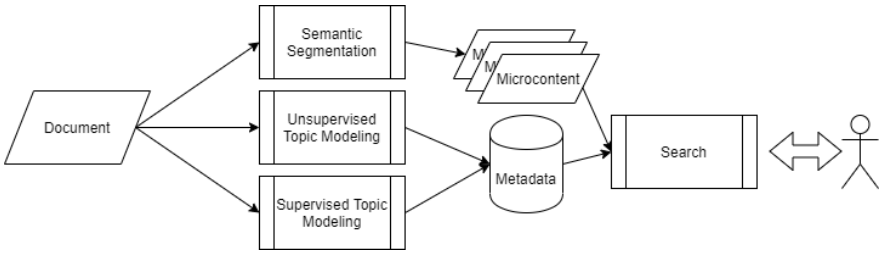

Figure 1: Automated Understanding Process

Overall, our system is a collection of NLP algorithms and models that infer a set of meta-data, including competencies exercised, difficulty, and topical content as depicted in figure 1 to facilitate search and utilization.

Word Embeddings

Word embeddings, popularized by google's word2vec, are numeric vector representations of words. They are derived by constructing a semi-supervised task, such as predicting a fifth word given the presence of four words of text. Such a task is considered semi-supervised because the task can be accomplished using a corpus of text and self-generation of prediction labels without human intervention. The underlying prediction neural network's intermediate output is then taken as a representation of the word within a numeric vector space. Current NLP benchmark tasks are dominated by models built on top of embedded representations, out-performing both traditional and statistical NLP approaches such as hidden Markov models.

The particular embedding underpinning our system is the BERT embedding model. Similar to word2vec, it uses a semi-supervised prediction task to train an underlying neural network and takes the intermediate layer output as the numeric representation (Devlin, Chang, Lee, & Toutanova, 2018). Unlike word2vec, BERT uses a transformer neural network architecture and operates on word pieces rather than whole words (Wu et al., 2016). This adds a level of resilience to words outside the training corpus as most complex words are comprised of syntactical "chunks" that retain meaning. Additionally, BERT's sequence-based architecture takes into account the context of words within a sentence, so the same word will yield different representations if its meaning changes (e.g., homographs or heteronyms), such as the word "lead" in the phrase "the sailor leads the team to the lead door."

Transfer Learning

In order to capture both the idiosyncratic syntactical patterns and semantic meaning unique to a technical domain, a form of computational transfer learning was required. Similar to the pedagogical transfer learning, ML transfer learning is a process by which model performance on one domain or task can be transferred and adapted to a new domain rather than training from scratch. In order to leverage the commercial-off-the-shelf BERT model, we started with the "bert_uncased_L-24_H-1024_A-16" that was trained with the Wikipedia and BooksCorpus corpa of 3.2 billon words (Wu et al., 2016), far more than the restricted training data available to our domain. This pre-trained model provides a representation of general English language, but it needed to be transferred to our technical domain in a two-step process. First, the corpus was parsed to

detect any words not covered by BERT's built-in vocabulary model. These words were then ranked based upon their occurrence count, and the top one thousand were added to the BERT vocabulary model. In order to adapt the model to both the new terms and semantics, we trained the semi-supervised BERT learning task with the pre-loaded weights against our corpus for several epochs.

Downstream clustering tasks showed an increase in silhouette score (Rousseeuw, 1987), a measure of cluster cohesion, indicating that the BERT model with transfer learning was more effective within the domain than the "vanilla" BERT Model. This process was required to allow the machine learning methods to understand the semantics of domain jargon as well as the syntax and idiosyncrasies of the domain.

Unsupervised Topic Modeling

In order to facilitate better searching of the content, an unsupervised topic modeling approach was devised with the goal of identifying topics from the content of the text itself. A naive method would entail a simple keyword search, but this method has several issues that must be overcome. First, not all parts of the sentence are relevant for topical search. The principal subject of the sentence is far more likely to appear within the sentence's noun phrases. Thus, we used the spaCy parser's part-of-speech tagger to extract the noun phrases to represent the candidate subjects of the sentence (Honnibal & Johnson, 2015). The second issue to overcome from a naive keyword search arises due to differences in syntax that do not yield semantic differences. Take for example the phrases "diving instructor" and "scuba instructor" appearing in two differing text snippets; when we are searching for content related to the instructor who teaches the activity, both text snippets should be returned as relevant content. While some terms can be associated via traditional knowledge-bases such as WordNet, those corpi are general purpose and may not incorporate the specific semantics of a technical domain.

To mitigate these limitations of a naive NLP approach, a machine learning semantic association procedure was derived based upon neural embeddings representing semantic relatedness. Candidate subject noun phrases were extracted via spaCy and then embedded via the BERT model into a numeric vector space. Words within this space are considered semantically related when the distance between the two vectors is small. However, this relatedness is not synonymous with synonyms, but also incorporates related concepts and even antonyms. Using this "closeness" metric, we then employed a clustering technique to see which of the words should be grouped together into a single concept. Without knowing the correct number of clusters apriori, a method named HDBSCAN was used in preference to more simple methods such as K-means, which requires the cluster count to be specified (McInnes, Healy, & Astels, 2017). The outcome of this clustering is usually hundreds of topics within a single instructional document, a level of annotation that would be too burdensome to place on a content curator.

With this topic model, we can add an additional dimension to the specificity by which training content can be selected. Additionally, we can select related content with similar topics to the content currently being presented to provide reinforcing examples.

Semantic Graph Segmentation

Segmentation is the NLP process of segmenting text into meaningful units along topical boundaries (Reynar, 1998). While this problem may seem trivial to describe, it is a difficult problem in practice due to the ambiguity of language. For our use case, we needed to identify the smallest piece of content that could be meaningfully presented while maintaining the necessary context for the content to make sense. Quite simply, a page of a training document might be too large to incorporate usefully, but a single sentence is too little to provide coherent meaning. In order to accomplish this goal, we wanted to detect when the semantics had sufficiently shifted to a new topic. We adopted the semantic graph segmentation algorithm outlined in (Glavaš, Nanni, & Ponzetto, 2016). This method works by first embedding each sentence using a word2vec embedding approach. These sentences are then used to generate a relatedness graph with edges being determined via a defined threshold and spatial proximity. A rendering of the semantic graph with each node representing a sentence and the edges representing semantic connection shows strong local semantic relatedness as expected, and several intersecting semantic paths through the document as can be seen in figure 2.

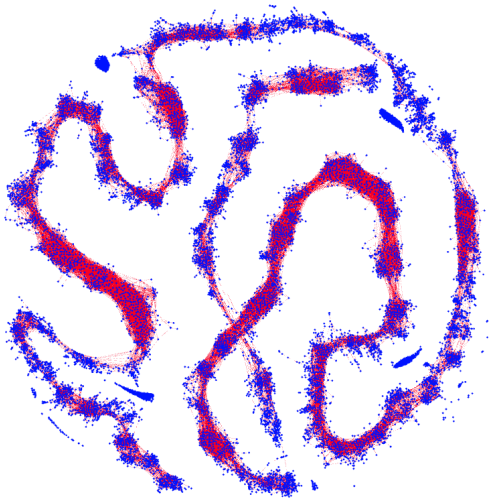

Figure 2: Document Semantic Relatedness Graph

Small sequentially connected neighborhoods within the graph, or cliques, indicate segments of high relatedness to be extracted as a meaningful piece of content. This bite-sized piece of content enables the instructor to find and reuse fine-grained content, without giving trainees more than they can chew.

Supervised Topic Modeling

To ensure that we can select the content related to the competencies being taught, a model must annotate content as related to those competencies. Regardless of the AIS's underlying competency model, this is a supervised text tagging problem related to the characteristics of the text in contrast to just the content of the text. A certain amount of labelled content can be used to characterize unlabeled content and provide suggestions for additional meta-data labels. The underlying competency structure can be used to inform the underlying supervised topic model's representation. "Is a" and "parent child" competency relations can define a hierarchical neural network decision tree. As these labels are deemed correct or incorrect by instructional designers as they use suggested content, the underlying model can be continuously refined as more data becomes available (Haley et al., 2019).

At this time, this functionality is not fully implemented; however, we envision this meta-data will enable selecting from multiple content choices to achieve the training for the specified competency.

Introductory Content and Difficulty Metrics

Competencies and supporting content often reflect the level of learner experience required. Introductory content introduces new topics to the learner for the first time and is characterized by the presence of more definitions due to a lack of assumed knowledge about the competency being taught (Waxman, 1987). Thus, in order to detect if content is introductory, we must detect introductions. To facilitate this goal, we used an encoding of regular expression-based Hearst patterns combined with the spaCy parser (Honnibal & Johnson, 2015) in order to detect introductions. A Hearst pattern is a lexical syntactical pattern that indicates a hyponym relationship (Hearst, 1992). Hyponyms are relations that suggest an "is a" type of definition. An example pattern might be "Noun Phrase such as Noun Phrase;" thus the phrase "fruit, such as an apple, is nutritious" would yield the introduction that an apple = fruit. Beyond just the Hearst patterns, we additionally added several patterns to detect acronym introduction. These introductions, then, indicate both the difficulty level of the content and the primary topics being taught.

In addition to the introductions, a series of reading level difficulty measures were evaluated against content. Introductory content and reading level are both indicators of whether the content is introductory and geared toward beginners or more advanced learners. Thus, when an instructional designer is developing a lesson,

they can tailor both content and delivery to the audience's prior experience, or when presented with several options, an AIS can select an appropriate level of difficulty.

Content Generation Process

The generation or extraction of content is a relatively straightforward search process after the generation of meta-data. The current system facilities two modalities of search, in-text and topical. The in-text search is a simple string-matching search, seeking out all segments of text that contain the search phrase. The topics act as a filter to select only segments containing an unsupervised topic. The in-text search and topic filtering can be combined to further increase the specificity of searched content. For example, an instructor could search for "out of air" and be presented with candidate content and all related topics. They then can select the topic of "caves" to further reduce the results to only the fine-grained content about out of air procedures in caves. An additional benefit of the semantic segmentation approach is that the underlying relatedness graph can be reused to recommend pieces of content related to the content identified via the search process. The identified content can then be post-processed into the required instructional format for presentation, whether that be a presentation or an learning management system module.

RESULTS AND DISCUSSION

No singular algorithm can perform computational content understanding. This system is a collection of models that each contributes to different aspects of content understanding. Herein, we discuss several of the observations for various methods and models. Two methods in particular will need to be addressed as future work: Topic Generation and Content Segmentation.

The unsupervised topic generation yielded 3271 distinct topics, well beyond the verbosity expected of an instructional designer. However, it was discovered that while the topics being generated were technically correct (the topic "ship" comes up often in nautical contexts), they were not at all useful for the search process to generate content. We discovered a new requirement that the topics found needed to not only be relevant, but to discriminate the portion of the text most relevant to the topic. Future work is addressing this concern by using a term frequency inverse document frequency (tf/idf) metric (Ramos et al., 2003). The approach will extend the tf/idf measure to take into account not only the specific term but the topic, which can be a collection of terms as determined by our unsupervised topic modeling approach. This approach will mitigate the non-discriminating search issue uncovered by this work.

The semantic content segmentation approach yielded 1920 different segments from the NOAA Diving Manual, a 520 page text resource. While evaluating the efficacy of content segmentation, we noticed inadequacies in the segments produced with several root causes. In the first case, text was missing from segments. Upon conducting error analysis and verification, we found that this deficiency was a result of errors in the OCR process combined with a sentence-filtering approach to remove noise. A second issue was the splitting of bulleted and numbered lists between segments. The underlying semantic segmentation does not consider document structure such as lists or even paragraph breaks. Finally, some of the segments were split at areas of too fine granularity. If the introduction of a paragraph provided context, it could be split from the latter half, which would then be missing required context. The approach must be augmented to take into account document structure and reference disambiguation to mitigate these issues.

Several of the underlying methods worked as expected. The BERT embeddings and transfer learning produced expected results: with the inclusion of technical terms and machine transfer learning, the downstream clustering tasks yielded higher quantitative metrics of cluster coherence as evidenced by an increased silhouette score from 0.5 with the general English model to 0.71 using the domain tuned model. Introduction detection missed several introductory definitions, but the inclusion of additional hearse patterns to detect statements such as acronym definitions yielded expected results extracting approximately 30 definitions from the NOAA Diving Manual, a non-introductory source.

CONCLUSIONS AND FUTURE WORK

The automation of meta-data generation represents an opportunity to reduce instructional content designers' bookkeeping, allow instructors to construct higher quality and more personalized lessons, and enable AIS recommenders to make finer-grained recommendations. The challenge in doing this based upon the content itself lies in the inherent ambiguities in language that result in ambiguous results for NLP processes. Our interpretation of the results have yielded several opportunities for enhancement and future work, a process being currently executed to enhance system utility. Despite the current limitations, this automated meta-data creation process is already enabling intelligent instructional content creation and utilization with reduced workload.

As we have struggled to retool our institutions of learning, from K-12 through advanced post-secondary programs, to an online environment as necessitated by the COVID-19 Pandemic, we have an opportunity to not only leverage the online reach of ITS systems, but to enhance outcomes via AIS facilitated by computational content understanding.

ACKNOWLEDGEMENTS

This material is partially based upon work supported by the Office of Naval Research (ONR) under contract N00014-19-C-2019. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of ONR or the Department of Defense.

REFERENCES

Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring system. *Science*, *228*(4698), 456-462.

Bell, B., & Sottilare, R. (2019). Adaptation vectors for instructional agents. In *International conference on human-computer interaction* (pp. 3–14).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (Mlm). Retrieved 2020-06-01, from `http://arxiv.org/abs/1810.04805`

Dougiamas, M., & Taylor, P. (2003). Moodle: Using learning communities to create an open source course management system. In D. Lassner & C. McNaught (Eds.), *Proceedings of edmedia + innovate learning 2003* (pp. 171–178). Honolulu, Hawaii, USA: Association for the Advancement of Computing in Education (AACE). Retrieved 2020-06-01, from `https://www.learntechlib.org/p/13739`

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). Density-based spatial clustering of applications with noise. In *Int. conf. knowledge discovery and data mining* (Vol. 240, p. 6).

Folsom-Kovarik, J. T., Chen, D.-W., Mostafavi, B., & Brawner, K. (2019). Measuring the complexity of learning content to enable automated comparison, recommendation, and generation. In *International conference on human-computer interaction* (pp. 188–203).

Folsom-Kovarik, J. T., Mostafavi, B., Sottilare, R. A., Davidson, I., Perez, R., & Walker, L. P. B. (2019). Approaches to enhancing transfer of training using adaptive instructional systems. In *2019 international conference on software, telecommunications and computer networks (softcom)* (pp. 1–6).

Glavaš, G., Nanni, F., & Ponzetto, S. P. (2016, August). Unsupervised text segmentation using semantic relatedness graphs. In *Proceedings of the fifth joint conference on lexical and computational semantics* (pp. 125–130). Berlin, Germany: Association for Computational Linguistics. Retrieved 2020-06-01, from `http://anthology.aclweb.org/S16-2016`

Haley, J., Dettmering, C., Hoehn, R., Barret, R., Mizan, A., Tanaka, A., & Stensrud, B. (2019). Persistent machine learning for government applications. In *Interservice/industry training, simulation, and education conference (iitsec).*

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. (July), 539. doi: 10.3115/992133.992154

Honnibal, M., & Johnson, M. (2015, September). An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1373–1378). Lisbon, Portugal: Association for Computational Linguistics. Retrieved 2020-06-01, from `https://aclweb.org/anthology/D/D15/D15-1162`

Joiner, J. T. (2001). *Noaa diving manual: Diving for science and technology.* National Oceanic and Atmospheric Administration.

Jurafsky, D., & Martin, J. H. (2008). Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing. *Upper Saddle River, NJ: Prentice Hall*.

McInnes, L., Healy, J., & Astels, S. (2017). hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, *2*(11), 205.

Merrill, M. (2002). First principles of instruction. *Educational Technology Research and Development*, *50*(3), 43-59.

Ramos, J., et al. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning* (Vol. 242, pp. 133–142).

Reynar, J. C. (1998). Topic segmentation: Algorithms and applications.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Smith, R. (2007). An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (icdar 2007)* (Vol. 2, pp. 629–633).

Sottilare, R., & Brawner, K. (2018). Component interaction within the generalized intelligent framework for tutoring (gift) as a model for adaptive instructional system standards. In *The adaptive instructional system (ais) standards workshop of the 14th international conference of the intelligent tutoring systems (its) conference, montreal, quebec, canada.*

Sottilare, R., Stensrud, B., & Hampton, A. J. (2019). Examining elements of an adaptive instructional system (ais) conceptual model. In *International conference on human-computer interaction* (pp. 239–250).

Waxman, H. C. (1987). Effective lesson introductions and preinstructional activities: A review of recent research. *The Journal of Classroom Interaction*, 5–7.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. , 1–23. Retrieved 2020-06-01, from `http://arxiv.org/abs/1609.08144`

Yin, W., & Schütze, H. (2013). Deep learning embeddings for discontinuous linguistic units. *arXiv preprint arXiv:1312.5129*.