Context Injection: A Novel Approach to Text Simplification

Josh Miller, Aniruddha Tapas, David Lowell
CS 6120 NLP Spring 2018
lowell.d@husky.neu.edu
miller.josh@husky.neu.edu
tapas.a@husky.neu.edu

Abstract

Academic texts are often dense and carefully phrased. This creates a barrier to entry for novice readers to understand the material while also preventing current text simplification and summarization techniques from being effective. In this paper, we explore a new method of text simplification which we call *context injection*. This method explores the feasibility of finding *jargon terms* and defining them, adding background information to the text from external sources. Notably, we find these domain-specific jargon terms without knowledge of the document's domain. Finally, we evaluate this algorithm's effectiveness on the level of end-user perception. We conclude that this method is comparably effective to other related works but adds the novelty of a new text simplification approach and eliminates the need for a domain-specific background corpus during training.

1 Introduction

Academic and technical works are typically rich in jargon terms: words or phrases which have a specific meaning within the domain, and which are not typically known to a lay audience. Readers of such documents are often assumed to be experts (Pinker, 2014), and thus already familiar with the technical language of the field. As a result, jargon is generally employed casually and without definition.

This creates a barrier to entry for novice students attempting to learn this material, given the rich literature on the difference in learning needs between novices and experts (Rey and Buchwald, 2011; Qiao et al., 2014; Leppink and van den Heuvel, 2015). The prevalence of jargon can also act as an impediment to cross-disciplinary collaboration, as even experts are typically unfamiliar with the jargon of domains outside their area of expertise. Thus, our primary problem is that academic texts do not include sufficient background information for readers without a specialized background, and this creates a barrier to entry in academic fields.

Current text simplification and text summarization techniques (see Related Work) are ineffective to solve this problem. Because each word of an academic text is chosen intentionally and with brevity in mind, these texts are often dense and irreplaceable. Moreover, the background information needed by the novice reader is simply not present in the text, and so any amount of NLP processing on the document alone would not be able to recover the contextual information that the novice reader lacks.

To address this problem, we propose a novel method of text simplification which we call *context injection*. The purpose of this approach is to add this absent background information to a text. Specifically, we identify words and phrases from the text which we deem jargon. These are domain terms with which a novice reader is unlikely to be familiar, but whose meaning is critical for understanding the text. We then output a novice readable definition for each such identified jargon term. This generated list of definitions serves as a tool to aid the reader's comprehension of the text, without requiring them to resort to outside resources themselves.

2 Related Work

Jargon term identification is the first component of context injection. We must locate jargon terms in the text before defining them. This subproblem is studied by Meyers (2014) in his development of the Termolator algorithm. We build from his work to identify candidate sites for context injection.

The second component of context injection is definition generation. We frame this problem as a variant of the general text summarization problem. Specifically, we locate Wikipedia articles that correspond to our identified jargon terms. A summary of such an article corresponds well with a definition for the jargon term. There is a wealth of research on summarization algorithms (Allahyari et al., 2017), of which we primarily rely upon the SumBasic Alogrithm (Vanderwende et al., 2007) (see Methodology for details).

3 Methodology

In order to achieve context injection, we developed a two-phase algorithm that performs:

1. Identifying jargon terms from the input text
2. Generating easily readable definitions of the recognized jargon terms

We used a combination of two approaches, using TFIDF and using The Termolator (Meyers et al., 2015) to identify jargons. TFIDF, short for term frequency

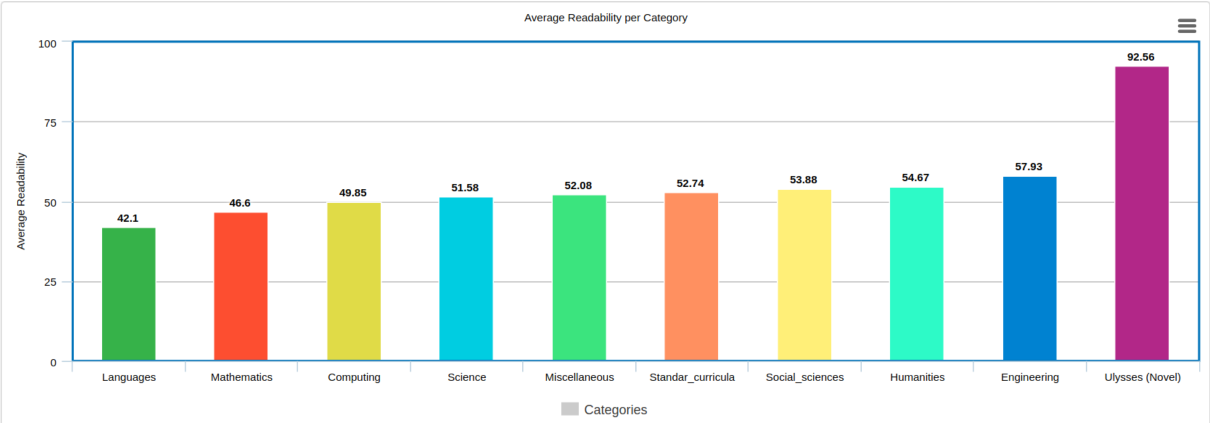

Figure 1: **Readability metrics for the trained corpus.** This figure shows our readability scores for each category of the Wikibooks database. Notably, Languages is the easiest to read, and Engineering is the most difficult. For comparison, we include the test document "Ulysses", by James Joyce, acquired from Project Gutenberg (Joyce, 1939).

- inverse document frequency, is a measure to reflect the importance of a word to a document in a corpus. We observed that using TFIDF resulted in identifying rare words but not necessarily domain-specific jargons. The Termolator identifies jargons using chunking to extract out-of-vocabulary words, nominalizations, technical adjectives, and other specialized word classes. It finds chunks that are relatively more frequent in a "foreground" corpus about a single topic than they are in a "background" or multi-topic corpus. Using the Termolator resulted in an effective extraction of specific jargon terms. Once the jargons were extracted, we implemented a variant of the SumBasic algorithm(Vanderwende et al., 2007) to summarize Wikipedia articles corresponding to the extracted jargons. Using these summaries, we built a list of definitions so as to solve our problem of aiding the reader's comprehension of the document.

3.1 Data Collection and Pre-Processing

For this project, we used a database consisting of Wikibooks (Wikipedia, 2003), Simple English Wikibooks (Wikipedia, 2004) and Simple English Tweets (Profiling, 2013). We used textbooks since they are a fair representation of academic texts. Additionally, there is little research on using text simplification on longer works, so our project is an extension of current text simplification techniques. We separate the data into two categories: low-readability Wikibooks containing many jargon phrases, and high-readability Simple-English Wikibooks and Tweets which are expected to have very few jargon phrases.

We collected Wikibooks from the Wikimedia dumps. The dumps are in the form of XML files containing multiple articles. We removed HTML tags and extracted clean texts for each article from these XMLs. Then we divided our data into comparable categories of academia, such as Science, Langauge, Mathematics etc. In order to determine the category for each cleaned article, we implemented a crawler to crawl Wikipedia; and then created a "book" by concatenating articles of the same topic. A similar procedure was applied to Simple Wikibooks after they were downloaded from the Simple Wikibooks dumps. In total, our database was comprised of 2747 Wikibooks and 30 Simple Wikibooks with an average text size of 13.5 MB per book.

We downloaded the raw dataset of tweets in the XML form provided by PAN (Profiling, 2013). PAN fosters digital text forensics research by organizing shared task evaluations. The raw datset contained IDs and URLs of easily readable tweets by males and females of different age groups. We used the Tweepy API (Roesslein, 2009) to scrape tweets using the IDs from the raw dataset and stored them as separate text files in the database. Lastly we gathered and cleaned 9303 tweets to be added to our Simple-English data.

3.2 Readability

Many metrics have previously been created to determine the readability of a document. These include, for example, the Flesch-Kincaid formulas (Flesch, 1948), Gunning's FOG Index (Gunning, 1952), the Dale-Chall formula (Chall and Dale, 1995), McLaughlin's SMOG Index (Laughlin, 1969), and the Coleman-Liau Index (Coleman and Liau, 1975). Yet, these metrics fail to take into account a number of variables which contribute to the readability of a document, opting instead for an easily-interpretable final score (Gray and Leary, 1935). Therefore, the current readability model expands on these indices and includes additional variables, such as the percentage of unique words, as well as an account of "hedge" words, "weasel" words, and "filler" words. The details of feature engineering may be found in the codebase attached with this article. Finally, we use a ridge regression model to fit the weights of these parameters. The model is trained on two sets

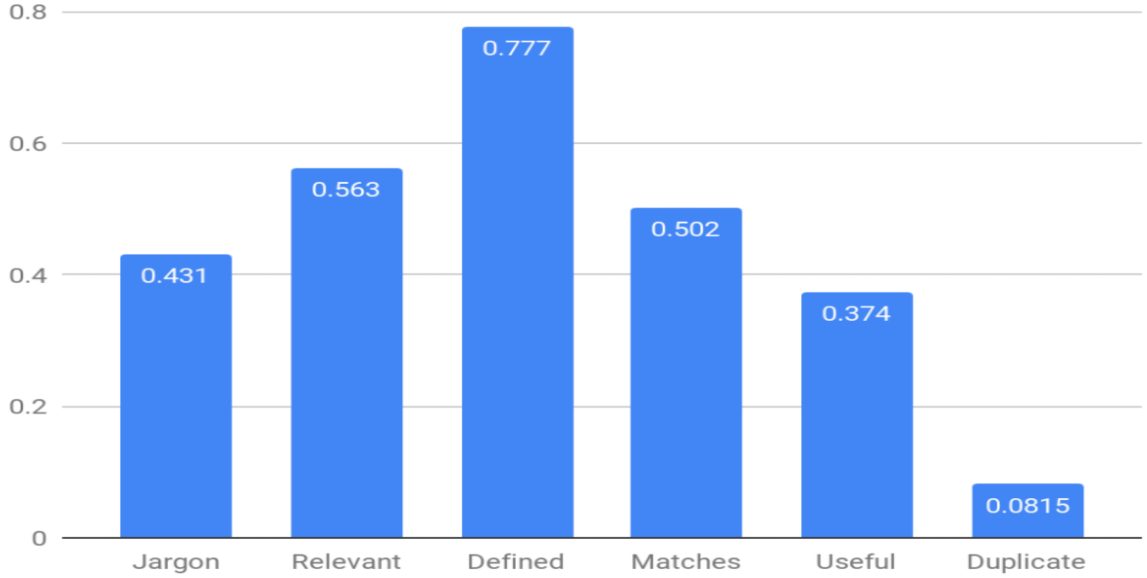

Figure 2: **Average Results of the Pilot Experiment.** These data show participant responses as a domain-wise average. The vertical axis represents the percent of participants who responded positively to a given jargon. Notably, participants believed that nearly half of the phrases identified by The Termolator were jargon, and more than half of the phrases identified were relevant to the domain.

of data: the "easy" data (for which all data are given a label score of 1) include our Twitter data and "simple English" from the SimpleWiki. The "difficult" data (for which all data are given a label score of 100) include the remainder of our database, namely the Wikibooks on various topics.

3.3 Identifying Jargon

To identify jargon, we compare two methods: a naive approach using TF-IDF, and a more sophisticated approach using The Termolator (Meyers et al., 2015). Both algorithms are trained on our database of Wikibooks and then tested on several sample Wikibook documents randomly selected. Notably, these documents were not contained in the database during training.

3.4 Defining Jargon

To generate jargon term definitions, we locate and summarize wikipedia articles corresponding to those terms. To locate an appropriate article, we treat the queried phrase as a search term and summarize the results.

Our summarization algorithm is a variant of the SumBasic algorithm(Vanderwende et al., 2007). We modify traditional SumBasic by enforcing the condition that the first sentence selected for inclusion in the summary is always the first sentence in the wikipedia article. This is motivated by the Wikipedia Style Guide (2018), which mandates that the first sentence's grammatical subject should be the article's title and that it ”should tell the nonspecialist reader what, or who, the subject is”. As a result, the first sentence both introduces the topic at hand in a human comprehensible manner and begins a definition, making it an excellent choice for the first sentence of our summarization.

3.5 Pilot Experiment: Reader Analysis

Because jargon is relative to the background knowledge of the reader, we test our jargon definitions on novice participants. Eight subjects volunteered to participate in this pilot study. Each participant answered six questions for each item (i.e. jargon phrase). Items were split across six randomly chosen domains: Horticulture, Parkour, Conlang, Stone Masonry, Card Games, and Chess Strategy. Each domain had at most 30 items. For each item, participants answered yes or no to these questions:

1. Do you consider this phrase jargon?
2. Is this phrase relevant to the domain?
3. Have we provided a definition for this phrase?
4. Does the definition seem correct? (i.e. The definition describes the phrase and not another concept.)
5. Do you believe this definition is useful for learning about the domain?
6. Is this phrase a duplicate of a previous phrase?

Participants were not timed and encouraged to take breaks as needed. The task took 30 minutes on average to complete.

4 Results

4.1 Pilot Experiment Results

Averaging over all domains (Fig. 2), participants responded positively to the items being jargon for 43.1%

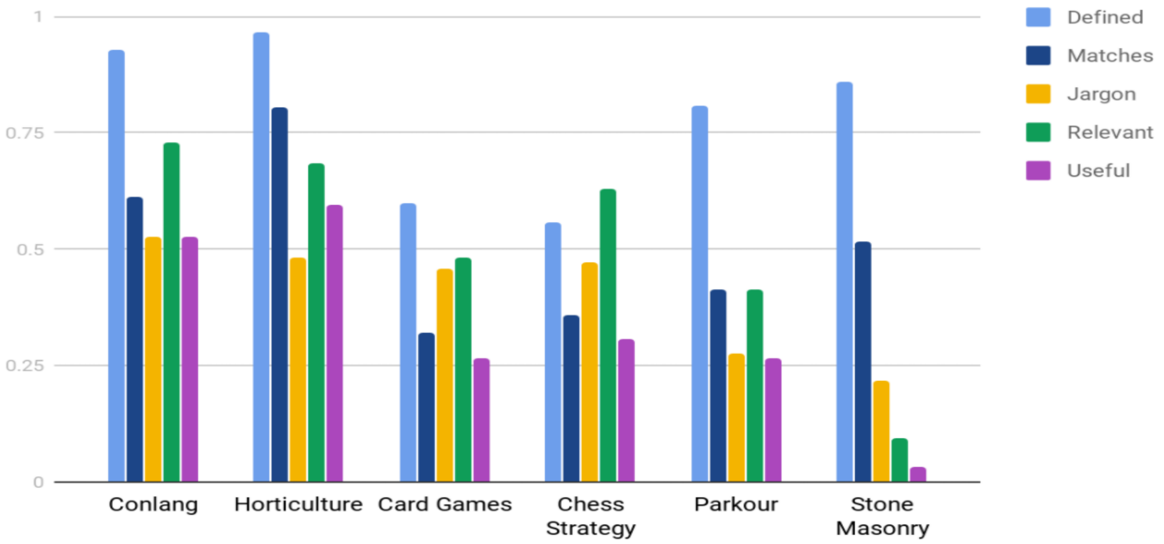

Figure 3: **Pilot Experiment Results by Domain.** These data show participant responses as an average per domain. The vertical axis represents the percent of participants who responded positively to a given jargon. Notably, more jargon was found for Conlang and Horticulture. This may reflect the participants' own domain knowledge on these subjects.

of all items (SD=.475). Items were 56.25% relevant (SD=.445), 77.69% matching (SD=.360), 50.24% useful (SD=.417), and duplicates only 8.1% of the time (SD=.209).

Inter-rater reliability was calculated in this way: for each pair of raters, for each item, count the number of agreements, and divide this by the total. Across all questions, this IRR was 0.790. Per question, the IRR was .610, .703, .940, .776, .799, and .914 respectively.

4.2 Jargon Identification Results

We calculate the precision of our algorithm for the task of identifying jargon phrases. We base our understanding of the ground truth on the responses our survey participants provided. Specifically, we categorize a term as being accurately identified as jargon if over half of participants reported it as being jargon, being relevant, and not being a duplicate. All other identified terms are categorized as having been misidentified as jargon. By this metric, our precision is 41.77%. We do not report our recall for this task, as we do not have access to the full set of jargon in the analyzed texts.

Over terms successfully identified as jargon, we report a 71.21% success rate for our definition generation task. We consider a definition to have been successfully generated if over half of survey participants consider the term to be defined, the definition to match the term, and the definition to be useful.

The total percent of our output which consists of successfully identified and defined jargon terms is therefore 29.75%.

5 Discussion

Preliminary attempts to find and define jargon without knowing the domain appear promising, given the difficult nature of the problem. Overall, our results boast nearly 30% precision. This statistic is comparable to the current state-of-the-art publications in automatic keyphrase extraction (Hasan and Ng, 2014; Kim et al., 2013). Consider, for example, that when the Termolator was originally used to identify jargon, even when trained on a specialized background corpus it was able to achieve only between 70% and 85% accuracy (Meyers et al., 2015). We demonstrate here that it is feasible to identify and define jargon without a specialized background corpus at a cost to performance. This approach may yet prove to be extremely advantageous to novice readers when utilized in future work. Lastly, we highlight the lower agreement between participants on what is jargon and what is relevant. Of course, participants agreed strongly on whether jargon was defined or duplicated a previous phrase, but when they were asked to decide whether a phrase was relevant jargon, the responses were mixed. This is likely due to several factors. First, participants may have different background knowledge, and what one considers jargon may not be jargon to another. Second, we hypothesize that individuals have different ”thresholds” of what they believe to be relevant to a subject. By the nature of human differences, and because jargon is intrinsically reflexive of background knowledge, these results are unsurprising. Truly, this work is motivated by the fact that a known vocabulary by one may be jargon to another, which is why the language used by experts writing in their subfield is incomprehensible to novice readers.

5.1 Limitations of this Study

First, it should be noted that this pilot research is not statistically significant considering our small sample size (n=8). As exploratory research, this much is expected. Moreover, our approach did not test in a way that had ground truth jargon phrases, which is why we could not report the recall of our algorithm, since we do not know how many false negatives there were. However, these limitations do not undermine the comprehensive result of this study. Namely, that jargon can be identified and defined without the need for a specialized background corpus.

6 Conclusions and Future Directions

Academic texts are often dense and carefully phrased. This creates a barrier to entry for novice readers to understand the material while also preventing current text simplification and summarization techniques from being effective. In this paper, we explored a new method of text simplification which we called *context injection*. This method explored the feasibility of finding *jargon terms* and defining them, adding background information to the text from external sources. Notably, we found these domain-specific jargon terms without knowledge of the document's domain. Finally, we evaluated this algorithm's effectiveness on the level of end-user perception. Our overall precision was 29.75%, and we were successful in defining 71.21% of identified jargon phrases. We conclude that this method is comparably effective to other related works but adds the novelty of a new text simplification approach and eliminates the need for a domain-specific background corpus during training.

There are several avenues for future work. One approach is to apply this work directly as a benefit to the reader by creating software which interactively defines jargon as needed by the reader. Another approach is to explore new methods for finding jargon, such as by using a reinforcement learning approach to identify which methods (e.g. TF-IDF, KL-Divergence) are better than others at identifying reader-perceived jargon phrases. Finally, we could consider defining only jargon which is not defined in the text. These jargon phrases could be identified, for example, by counting the number of co-references to a potential jargon phrase, and only defining jargon phrases with a low number of co-references.

7 Resources

The authors publicly provide all project code at: https://github.com/joshmiller17/context-injection.

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. Text summarization techniques: A brief survey. *arXiv preprint arXiv:1707.02268* .

Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology* 60(2):283.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology* 32(3):221.

William Scott Gray and Bernice Elizabeth Leary. 1935. What makes a book readable. .

R. Gunning. 1952. *The Technique of Clear Writing*. McGraw-Hill, New York.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1262–1273.

J. Joyce. 1939. *Ulysses*. v. 1. Odyssey Press. https://books.google.com/books?id=fxWfE1JLUIMC.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2013. Automatic keyphrase extraction from scientific articles. *Language resources and evaluation* 47(3):723–742.

G. Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of Reading* 12(8):639–646. http://www.jstor.org/stable/40011226.

J. Leppink and A. van den Heuvel. 2015. The evolution of cognitive load theory and its application to medical education. *Perspect Med Educ* 4(3):119–127.

Adam Meyers, Zachary Glass, Angus Grieve-Smith, Yifan He, Shasha Liao, and Ralph Grishman. 2014. Jargon-term extraction by chunking. In *Proceedings of the COLING Workshop on Synchronic and Diachronic Approaches to Analyzing Technical Language*. pages 11–20.

Adam Meyers, Yifan He, Zachary Glass, and Olga Babko-Malaya. 2015. The termolator: Terminology recognition based on chunking, statistical and search-based scores. In *CLBib@ ISSI*. pages 34–43.

Steven Pinker. 2014. Why academics stink at writing. https://www.chronicle.com/article/Why-Academics-Writing-Stinks/148989.

PAN Author Profiling. 2013. PAN-Author-Profiling. http://pan.webis.de/clef17/pan17-web/author-profiling.html.

Y. Q. Qiao, J. Shen, X. Liang, S. Ding, F. Y. Chen, L. Shao, Q. Zheng, and Z. H. Ran. 2014. Using cognitive theory to facilitate medical education. *BMC Med Educ* 14:79.

G. D. Rey and F. Buchwald. 2011. The expertise reversal effect: cognitive load and motivational explanations. *J Exp Psychol Appl* 17(1):33–48.

Joshua Roesslein. 2009. tweepy.api. http://www.tweepy.org/.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management* 43(6):1606–1618.

Wikipedia. 2003. Wikibooks. https://en.wikibooks.org/.

Wikipedia. 2004. Simple-English-Wikibooks. https://simple.wikibooks.org/.

Wikipedia. 2018. Wikipedia:manual of style/lead section. https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Lead_section.