Word2vec based natural language simplification tool

Ayazhan Zhakhan
Georgia Institute of Technologies
Medellin, Colombia
ayazhan@gatech.edu

ABSTRACT
In this paper, a new language simplification (NLS) tool is introduced. At its core is a word2vec algorithm called GloVe, which is used to create vector representations of words and identify similarity distances between them. During the development stage, two most popular natural language simplification methods were explored: lexical simplification and explanation generation. Benefits and challenges of these methods were analyzed. The final tool, however, was built using semantic maps method, which combines features of both lexical simplification and explanation generation.

Author Keywords
Natural language simplification, GloVe, word2vector, contracted natural languages, lexical simplification, syntactic simplification, explanation generation, semantic maps.

INTRODUCTION AND MOTIVATION

Problem

English is a rich and complex natural language which hosts about 44% of all the information produced (Lobachev 2008). English is also one of the most information dense languages (Pellegrino, Coupé 2011). These characteristics along with its rich history made English the language of choice for science and technology. However, mastering English is challenging. Vocabulary is the primary obstacle for many non-native speakers since the text coverage is not proportional to the vocabulary size (Nation, Waring 1997). With the first 1000 basic words, one can understand 72% of English text. However, every additional 1000 words only give moderate increase of text coverage. At the same time, studies have shown that English proficiency is not an effective predictor of academic success, especially for STEM related fields. To assist non-native speakers wishing to pursue a career in science and to make knowledge more accessible, natural language simplification tools are needed.

Existing solutions

Controlled natural languages (CNLs) are engineered subsets of natural languages with restricted grammar and vocabulary. These restrictions reduce ambiguity and complexity.

There are two main groups of CNLs: human oriented CNLs and machine-oriented CNLs. Human oriented CNLs are designed to improve the readability and comprehensibility of text. Machine-oriented CNLs are designed to improve translatability and make it easier to analyze text with machines. This paper is focused on human oriented CNLs (Schwitter 2010).

There are three main approaches used for text simplification, which are lexical simplification, syntactic simplification, and explanation generation (Shardlow 2014).

Lexical simplification identifies complex words and replaces them with simpler versions. It does not modify grammar. The main challenge with lexical simplification is the preservation of meaning. Because many words in English have multiple meanings, understanding context is crucial (Deschacht, Moens 2009).

Syntactic simplification is a technique that identifies grammatical complexities in a text and rewrites it using simpler structure. For example, long sentences are broken down into smaller components and sentences where passive voice is replaced with active voice.

Explanation generation technique augments a difficult concept in a text by adding more information. This technique is especially popular in the medical field. However, explanation generation is domain specific and predominantly used in medical field. There were also attempts to use this method to help second language learners (Kauchak 2013).

For this project, lexical simplification and explanation generation techniques were utilized.

PROPOSED SOLUTION

Idea

The objective of this project was to simplify the language used in computer science (CS) field. Lexical simplification was chosen as a base method for the NLS tool. To find substitute words for complex words, two text corpuses were compiled: one with Common English (such as Wikipedia English), and one with Technical English (such as CS textbooks). Next, word2vec algorithm, called GloVe, was trained on both datasets to create a vector representation of each word in the corpus. The goal was to identify a

substitute word for every complex term in Technical English from Common English.

Key Components

Word2Vector representation

Vector representation is a leading technique for measuring syntactic and semantic similarities of words. Each word is represented in a vector form by its distances to other words in multiple dimensions. The algorithm performing vector representation groups conceptually related words such as numbers and weekdays close to each other. These representations find relationships that go beyond syntactic similarities. Using simple algebraic operations on word vectors, deeper understanding of the relationship between words can be extracted (Fuchs, Kaljurand 2008).

GloVe

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. It considers both global and local context of words for vectorization. It uses co-occurrence matrix to find global context and skip-gram model for local context. It learns vector representations for words so that the dot product for a pair of words is proportional to their co-occurrence count. GloVe model is an open source and collections of pre-trained word vectors are available online (Pennington 2013).

To measure similarity distances between words, k-nearest neighbors algorithm with cosine distance metric was used.

Text Corpus

To generate vector representations of words a large structured set of texts is required. For this project, only English text was analyzed. Two distinct corpuses were compiled: Common English and Technical English. For Technical English, only computer science related text was considered.

DEVELOPMENT

Data Gathering

Common English

For Common English, the Wikicorpus dataset provided by the Department of Computer Science of the Universitat Politècnica de Catalunya was used. The corpus contained around 600 million words.

Technical English

Technical English corpus was created by combining 29,555 research papers from www.arxiv.org repository, 206 computer science related textbooks and 4,232 lecture transcripts from Online Master of Science in Computer Science (OMSCS) program. The final corpus size was 51,249,895 words.

Data preprocessing

The GloVe algorithm required the text to be in UTF-8 format with no punctuations. To prepare the data, first, non-English words were removed. Next, numbers, punctuations and other non-alphabetical characters were removed. For the final step, ASCII encoded text was converted to UTF-8 format and special characters used for encoding were removed.

Model training

The original GloVe code published by Stanford University was used to create vector representations of words for both Common English and Technical English. Word similarity distances were measured using KNN algorithm from scikit-learn library. Distance metric parameter of the KNN was set to “cosine”.

RESULTS

First interesting observation was the difference between clustering of words from the two corpuses. For the same word, the model trained on Common English (Common Model) found different neighboring words than the model trained on Technical English (Technical Model). For example, for the word “Space”, Common Model returned “Area” and “Volume” as neighboring words, whereas Technical Model returned “Manifold” and “Dimensional”. These results were expected because GloVe uses co-occurrence count to construct vector matrices.

Furthermore, GloVe algorithm grouped conceptually similar words together. Such groups not only included synonyms, but also words with different part-of-speech (POS) tags, variations of the base word (such as plural version) and other conceptually related but lexically different words.

As interesting as these results were, unfortunately they could not be used to develop a lexical based NLS tool. There were multiple challenges which had to be addressed:

First, trained models did not always return synonyms. Often, the closest neighboring words were conceptually similar, but had different lexical meaning and could not be used as a direct substitute. Therefore, automatic identification of the best substitute word posed a challenge. This task was further complicated when words with multiple meanings were analyzed. The true meaning of such words could only be inferred through the context, which is missing in vector representation.

Another obstacle faced was the fact that the returned substitute words were not always simpler than the original word, which defeated the purpose of a language simplification.

Finally, because different models produced different groups, mapping words from different model diluted true meaning of the original word even more. For example, for the word “Gravity”, Technical Model returned “two-dimensional” as the closest neighbor, whereas Common Model returned “Velocity”. Both are relevant in their fields but identifying which one is a better substitute is challenging. If a substitute word is chosen based on Technical Model’s prediction, then there is no guarantee that the substitute word would be a simpler word. If, on the other hand, the substitute word is chosen from Common Model, then the true meaning of the word might get lost.

ALTERNATIVE SOLUTION

Due to the challenges mentioned earlier, lexical simplification technique could not be utilized with vector word representations. However, developed model had its own set of benefits that could be used for the NLS tool. For example, sets of words generated by the model represented conceptually similar words, also known as semantic maps.

According to the study by Sökmen, semantic mapping is a powerful technique used to identify, recall and understand the meaning of words (Sökmen 1997). Today, this technique is mainly used in schools to help students with disabilities to enrich their vocabulary. Current ways of creating these maps are brainstorming and manual search for synonyms.

A tool (GloVeNLS) developed for this project performed these tasks automatically. It found both synonyms and conceptually similar words. For example, for a word “Car”, GloVeNLS returned synonym “Vehicle”, a type of car - “Truck” and conceptually similar word “Driver”. Furthermore, semantic mapping aids semantic memorization, which is a form of general knowledge that people accumulate throughout their lives. Semantic form of memory has been proven to be more robust because meanings are linked to ideas and concepts.

APPLICATION

For the proof of concept, a mobile application that produces semantic mapping was developed. For a queried word, it generates 4 conceptually similar words. These words do not explain the meaning of the word directly, but provide with enough information to convey the meaning.

Due to the size constraints and slow querying speed of KNN model, online querying proved to be inefficient for a mobile application. Instead, 10,000 most common English words were pre-queried, and results were uploaded to a MySQL database. Next, a web application was developed using HTML, CSS and JavaScript languages. Queries were processed using AJAX in real time. Next, designed web application was converted into a mobile application using PhoneGap development framework (see image 1 for a screenshot).

Online demo of the application is available here:

http://zhakhan.com/glove/

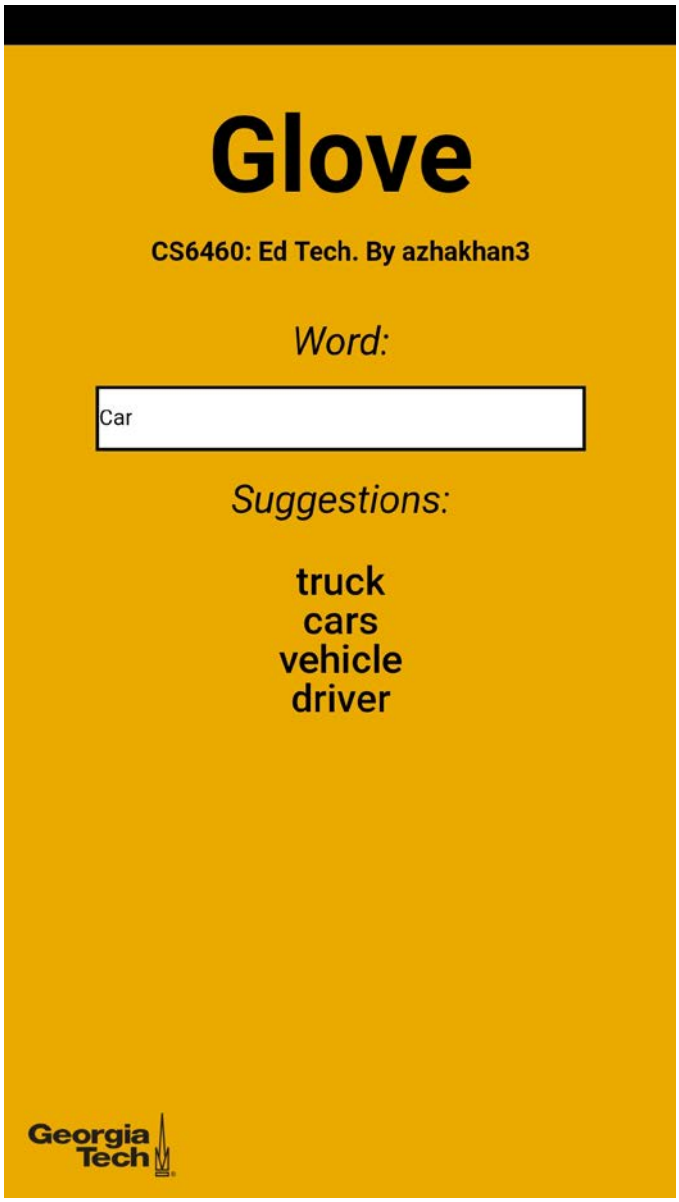

Figure 1 - Application Screenshot

Table 1 demonstrates semantic maps produced by GloVeNLS tool.

Table 1 - Sample Results

School: • college • schools • campus • graduate	**Car:** • truck • cars • vehicle • driver
Sweet: • taste • delicious • honey • flavor	**Computer:** • computers • software • technology • electronic
Pen: • ballpoint • pencil • pens • duick	**Phone:** • telephone • phones • cellphone • internet
Girl: • boy • woman • mother • girls	**Sad:** • awful • sorry • terrible • sadly

FUTURE IMPROVEMENTS

To improve performance of GloVeNLS, the model could be trained on a larger corpus. It would be also useful to explore domain specific corpuses to train models that meet the needs of specific fields, such as medicine, physics, computer science, etc. These models would provide more relevant maps due to the nature of word2vec algorithm.

Next, recommended words could be listed based on their popularity and similarity. For this, each word in the dictionary must be tagged with its popularity score. When computing ranking for the maps, the following scoring formula could be used:

$$\boldsymbol{Relevance = Distance * Popularity + POS}$$

** Where POS is 1 if it matches the POS of the queried word and 0 otherwise.*

There are still some questions which must be addressed, such as:

How to automatically evaluate true relevance of neighboring words?

What is the balance between simplified and original word?

OTHER POSSIBLE APPLICATIONS

Language learning is another interesting application for GloVeNLS. As mentioned earlier, semantic mapping is a powerful tool used to help students memorize words in a more natural way. GloVeNLS could be used to automatically generate semantic maps. Furthermore, because GloVe algorithm only uses co-occurrence count, GloVeNLS could be easily extended to other languages.

CONCLUSION

In this paper, natural language simplification techniques were analyzed, such as lexical simplification and explanation generation. Unsupervised learning algorithm GloVe was used to create vector representations of words. These vectors were then used to develop a language simplification tool. Challenges of this approach were explored and analyzed. Furthermore, the alternative tool was proposed, which used vector representations to produce semantic maps. Finally, an application was developed as the proof of the concept.

REFERENCES

1. Lobachev, S. (2008). Top languages in global information production. *Partnership: The Canadian Journal of Library and Information Practice and Research*, *3*(2).
2. Pellegrino, F., Coupé, C., & Marsico, E. (2011). Across-language perspective on speech information rate. *Language*, *87*(3), 539-558.
3. Nation, P., & Waring, R. (1997). Vocabulary size, text coverage and word lists. *Vocabulary: Description, acquisition and pedagogy*, *14*, 6-19.
4. Schwitter, R. (2010, August). Controlled natural languages for knowledge representation. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (pp. 1113-1121). Association for Computational Linguistics.
5. Hoard, J. E., Wojcik, R., & Holzhauser, K. (1992). An automated grammar and style checker for writers of Simplified English. In *Computers and Writing* (pp. 278-296). Springer, Dordrecht.
6. Fuchs, N. E., Kaljurand, K., & Kuhn, T. (2008). Attempto Controlled English for Knowledge Representation, Reasoning Web: 4th International Summer School 2008, Venice, Italy, September 7-11, 2008, Tutorial Lectures.
7. D. Kauchak, "Improving text simplification language modeling using unsimplified text data," in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 1537–1546.
8. Pellegrino, F., Coupé, C., & Marsico, E. (2011). Across-language perspective on speech information rate. *Language*, *87*(3), 539-558.

9. Trauzettel-Klosinski, S., & Dietz, K. (2012). Standardized assessment of reading performance: the new International Reading Speed Texts IReST. *Investigative ophthalmology & visual science*, *53*(9), 5452-5461.
10. Sökmen, Anita J. "Current trends in teaching second language vocabulary." *Readings in Methodology* 152 (1997).
11. Bartusiak, Roman, et al. "WordNet2Vec: Corpora agnostic word vectorization method." *Neurocomputing* (2017).