Improving Context Based Thesaurus

Jae Il Lee, Nathaniel Schub, Aniruddh Nautiyal

December 2018

Abstract:

With the established use of machine learning in today's language modeling, capturing semantic similarity between words is a key challenge. We attempt to improve the similarity results given a target word and the context around it by using a context-aware biLSTM model with adaptive word embeddings and evaluate the results in comparison to more traditional language models.

Keywords: NLP, thesaurus, ELMo

1. Introduction

Word embeddings are an effective way to capture the distributional semantics in text. However, they take into account the local context (window size) and not the global context of the sentence. Any word sense perturbations are averaged out, and the sense of a particular word isn't encoded. Word sense also, on the other hand is difficult to capture in modeling in the first place, is ambiguous and often has no straightforward techniques or satisfactory criteria of correctness. Manually organized thesauruses are usually organized into a multi-level hierarchy based on human perception of word senses, whereas automatic thesaurus generation techniques have fewer levels of organization and are simpler. This makes using word senses a problematic approach for building a thesaurus. If two words can be substituted in place of one another without changing the truth value of the sentence, that offers little to no information, if one word is more appropriate in the place of another. Thus, it is extremely difficult to model word meanings with machines algorithmically. Differences aside, it would be useful to consider both the approaches as they complement each other, to generate and evaluate a thesaurus. In this project, we focus on using distributional similarity as the organizing principle and use vector similarity metrics for the evaluation of the models.

2. Datasets

We used the 1 Billion Word Benchmark Corpus (Chelba, 2015) that is derived from WMT 2011 News Crawl data, for training our models. Apart from having sufficient depth and variety, we selected this to allow for direct benchmarking with a state-of-art model (ELMo) which is this trained on the 1 Billion Word Corpus.

For the similarity evaluation, we used three standard datasets that have a list of word pairs with their similarity scored by human annotators: SimLex-999 (Hill et al., 2015), WordSim-353 (Finkelstein et al., 2001), and MEN (Bruni et al., 2014). The WordSim and MEN datasets tend to mix the concepts of semantic similarity and semantic relatedness in their scoring scheme. Due to this, the SimLex-999 dataset was developed to differentiate genuine similarity with association. Hence, it is the most challenging of the three datasets for a model to score. An example is the word pair (hard, difficult) which returned a similarity score of 8.77/10, or (weird, normal) which returned a 0.77/10 (Hill et al., 2015). We use ranking base method, on strength of similarity of each word pair to evaluate the models.

3. Approach

The word embeddings learned by a model are a generalized representation of the relationships between the words as seen by the model. We first use Spearman Rank correlation coefficient (or Spearman's Rho, for short) to evaluate how far apart the rankings of the word pairs are, as scored by each by each model. We evaluated the performance of the *Word2Vec* and *GloVe* embeddings against each of three evaluation datasets (SimLex, WordSim, and MEN), and selected SimLex as the benchmark to compare with the rest of the models because it was the most demanding evaluation dataset. We also planned to employ Facebook's *FastText* (Joulin et al., 2016) that does a character level modeling, and

overcomes some issues with word-level modeling, such as better handling of out-of-vocabulary words; however, the state-of-art model *Deep contextualized word representations* (ELMo) (Peters et al. 2018) that we chose for final evaluation, employs the character level modeling similar to FastText.

We generate an initial list of candidate words from *Word2Vec* word embeddings and then re-rank the words based on the *ELMo* model, which should take the context of the given sentence into account and return an filtered list of candidates - the basis of a context-aware thesaurus.

Sample Sentences for Evaluation:
To be able to validate the performance of the context-based language models (*RNN, Ngram* etc.) we generated a list of 10 sentences per word listed in our word pair datasets (e.g. 2 words x 999 word pairs x 10 sentences). We have used the APIs of Naver's English Dictionary (Kim) and generated 19,980 total sentences).

4. Models

Word Embeddings:
We first analyzed the word similarities from the pre-trained word embeddings of the *Word2Vec* trained on the Google News Corpus and *GloVe* trained on Twitter 2B tweets. While they performed closely, we quickly saw that they had issues in capturing the specific meaning of the words, and hence assessing the similarity among them. The Spearman Rank correlation for the two was weak. In our initial experiments, the embeddings failed to recognize difference between clear antonyms (i.e., major and minor).

We then trained the *Word2Vec* embeddings on the 1 Billion Word Corpus, to serve as a baseline score. We generated an embedding for each word in the SimLex word pairs, then calculated the cosine similarity between them. We then ranked the pairs by their cosine similarity scores, the ones with higher score were ranked higher, indicating the words in the pair had strong synonyms relationship. We then compared those ranks with the SimLex ranks and calculated Spearman's Rho. The process is shown below in Figure 1.

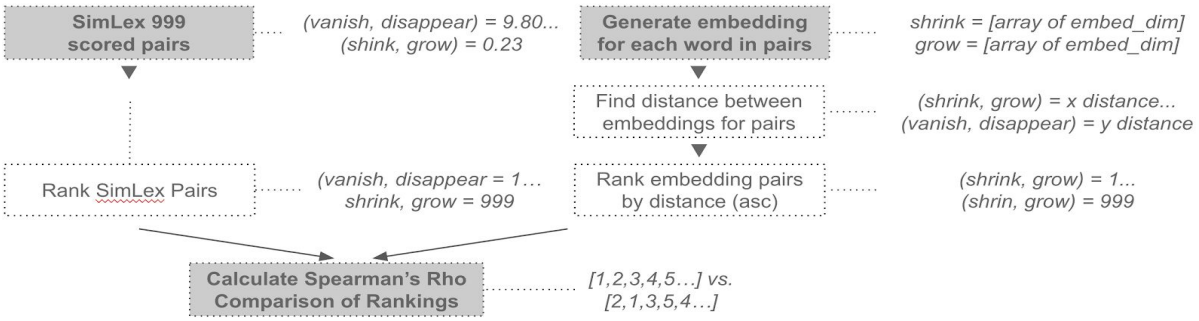

Figure 1: Calculating baseline Spearman's Rho for the trained Word2Vec embeddings

RNN:
With the sample sentences for each word pair in our SimLex dataset, we ran them through an RNN model with parameters (vocabulary : 10,000, hidden state size : 200, number of layers : 2, SoftMax sample size : 200) chosen to be able to effectively run on the same 1 Billion Word Corpus (one hundredth of the full dataset). With this RNN model we were able to get the probability score and rank the word pairs for us to compare them with the baseline set with the *Word2Vec*.

Trigram:
Sample sentences generated for each of the SimLex word pairs presented an opportunity to compare the similarities between the words in each pair using an n-gram model. We trained a trigram model on one hundredth of the full 1 Billion Word Corpus, as the time to generate the contexts and counts across the full corpus was computationally too expensive with the resources at hand. The trigram model yields the

probability that each word would appear given the context. We used using add-k smoothing with k=1. We compared the probabilities that each word would appear in its own sentences with the probability that it would appear in the other word in the pair's sentences. Higher differences in probability would suggest that the words were not readily interchangeable across a range of contexts. We then ranked the word pairs by the difference in probabilities, and compared the trigram-generated rankings against SimLex rankings. The process is shown in Figure 2.

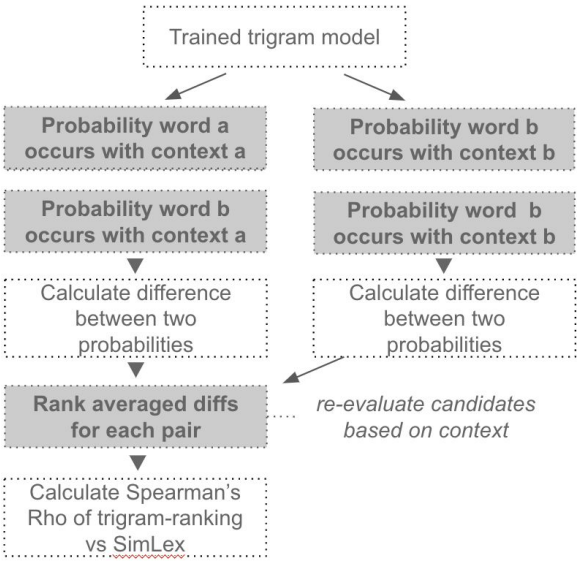

Figure 2: Calculating Spearman's Rho for each SimLex pair (word a, word b) to evaluate trigram model vs. SimLex ranks

ELMo:

ELMo (Embeddings from Language Models) is a deep contextualized word embedding generator that is trained on the 1 Billion Word Corpus and uses bidirectional LSTMs to understand a word's context within its sentence (Peters et. al., 2018).

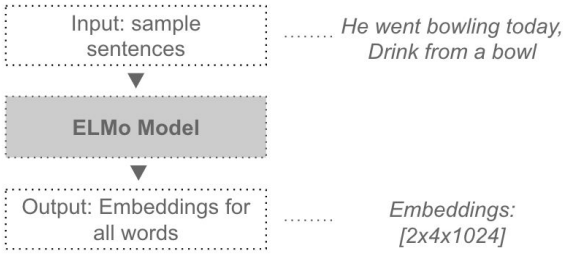

Figure 3: ELMO inputs and/outputs

As an example, we created a TSNE plot to examine that *ELMo* is context-aware. In one sentence, bowl is used as in "fill the bowl", and in the other, bowl is used as in the sport of bowling. There is clear separation between the two words, showing *ELMo* recognizes based on their contexts that they are completely unrelated.

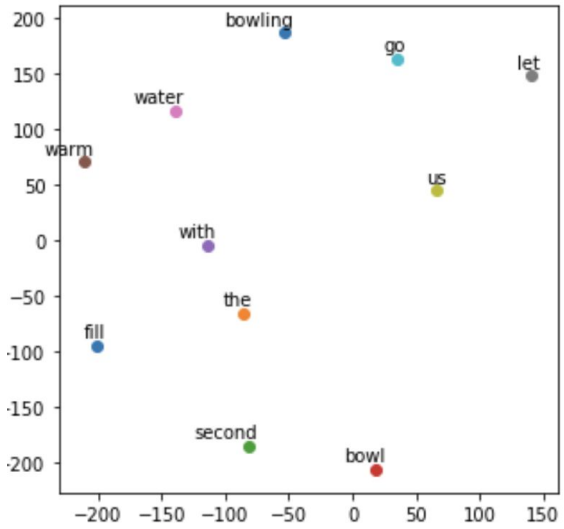

Figure 4: TSNE plot showing ELMo understands semantic differences in word usage.

To evaluate *ELMo*'s strength in understanding context and its potential use as a context-aware thesaurus, we fed each of the 10 sentences for each of the SimLex word pairs into *ELMo* and extracted each target word's embeddings. We then averaged out embeddings of each word, which averaged the range of contexts from the generated sentences, then calculated the cosine distance between the two embeddings for each word in the word pair. These word pairs were ranked based on the cosine similarity scores. Words pairs with small distances (high similarities) between the two averaged embeddings were ranked higher than words with high distances between the two embeddings. We then compared the *ELMo*-generated rankings of the word pairs with the SimLex rankings of the word pairs, and calculated Spearman's Rho. The process is shown in Figure 5 below.

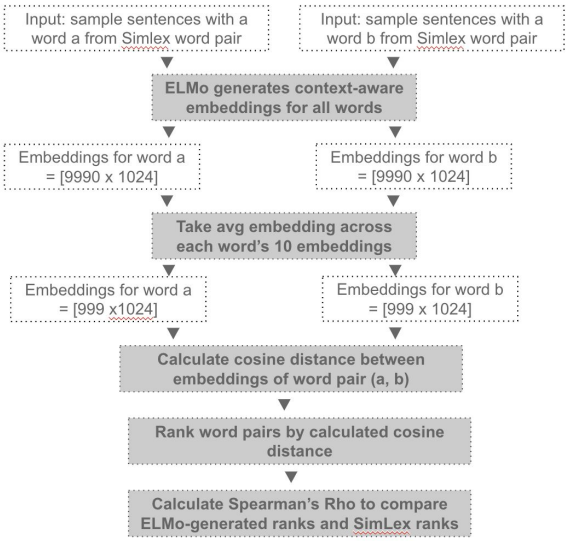

Figure 5: Process to generate ranks from ELMo model to compare to SimLex

5. Results

RNN:

The RNN model was able to evaluate the probability of the full sentence occuring with the candidate words put in place, but there was a critical limitation to the evaluation. The scores given by the RNN model would reflect the probability of the full context of the sentence but this was not appropriate to explore the details of the candidates and thus was not able to produce a meaningful score to compare against the SimLex ranking.

Trigram:

The trigram model did a poor job of accurately ranking the word pairs against the reference SimLex rankings. The calculated Spearman's Rho was ~0.1, suggesting little correlation between trigram and SimLex rankings. There were several reasons for this, first being the computing challenges trigram poses. Second, similarly to the RNN, the delta of probabilities is not an ideal metric to rank against the SimLex. Third, many combinations of words and their frequencies were missing from the trigram model or had such low counts that the resulting probabilities were not informative of actual language patterns. Fourth, the trigram fails to capture context from the whole sentence. This all means that the ranking of probabilities we compared to the SimLex rankings for each word pair was compromised. Ngram models might be more useful across a narrower range of contexts, not the wide reaching 1 Billion Word Corpus and the wide range of words and topics covered by the SimLex word pairs and the generated sentences we used to evaluate them.

ELMo:

The evaluation the *ELMo* model showed somewhat mixed results, as shown in Table 1. While the *ELMo* model scored much better than *Word2Vec* on SimLex, the performance on the other two datasets was marginally lower. We suspected that some of the word pairs might have higher disagreement between the human annotators themselves and this could be reflected in the standard deviation of the word pair's score. However, a rank delta between SimLex and *ELMo*'s rankings versus standard deviation of each word pair's score, showed no correlation ($\rho = -0.07$).

Dataset	Word2Vec	ELMo	Change(%)
SimLex	0.367	0.434	18.136
WordSim	0.591	0.570	-3.567
MEN	0.676	0.643	-4.961

Table 1: Spearman's Rho for Word2Vec and ELMo trained on the 1 Billion word corpus.

We selected the SimLex word pairs and similarity rankings to evaluate the other models because we know *ELMo* to be more context aware than *Word2Vec,* but this was only demonstrated by evaluation against SimLex, not the other evaluation datasets (Peters et. al. 2018).

6. Prospective Final Model and Conclusions

Based on the findings thus far, we designed a final context-aware thesaurus model using the two most successful approaches we evaluated: *ELMo* and *Word2Vec*. With a given target word and a target context sentence, we used the *Word2Vec* embeddings to generate a preliminary list of candidates of the target word based on cosine similarity. We then replaced the target word with each of the candidates, one at a time, within the sentence, and filtered out candidates that did not fit into the sentence using part-of-speech tagging. We then ran the target context sentence with each of the filtered word candidate list through *ELMo*, and provide a final

list of ordered synonyms based the same principle of minimizing the distance with the target word's own embedding. We used the part-of-speech tagger because *Word2Vec* often suggests candidate words with different parts of speech than the target word, as potential synonyms, and we found that *ELMo* sometimes fails to put adequate distance between different parts of speech.

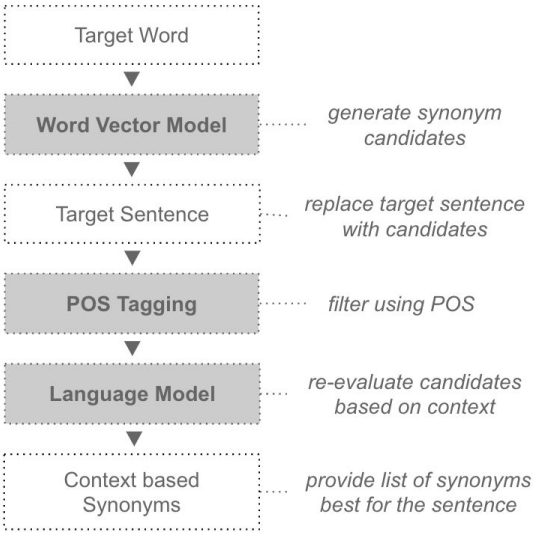

Figure 6: Flow of Prospective Final Model

In the examples of the final model we created, several of the embedding-generated synonyms were retreads of the same word (common typos from the corpus, plurals, gerunds, etc.). These unhelpful synonyms often outnumbered the useful synonyms, which seems to be a difficult-to-avoid consequence of attempting to use word embeddings and contexts to generate a thesaurus. More traditional thesauruses that do not necessarily attempt to be context-aware but rely entirely on human-generated similarities between words may give more practical results.

7. References

[Chelba et al., 2015] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In INTERSPEECH.

[Hill et al., 2015] Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. Computational Linguistics, 41(4):665–695.

[Finkelstein et al., 2001] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing Search in Context: The Concept Revisited. In Proceedings of the 10th international conference on World Wide Web.

[Bruni et al., 2014] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal Distributional Semantics. Journal of Artificial Intelligence Research, 49:1–47.

[Peters et al., 2018] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. 2018. Deep contextualized word representations. In Proceedings of the 2018 Conference of the North American Chapter of the ACL.

[Mikolov et al., 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS.*

[Pennington et al., 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In EMNLP.

[Joulin et al., 2016] Armand Joulin, Edouard Grave, Piotr Bojanowski Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification.

[Kim] Taehoon Kim, NAVER Korean English Dictionary. At endic.naver.com.