BetterDef: Improving the Embedding Definition Model

Philip Meyers and **Sebastian Pretzer**
{meyers,pretzer}@u.northwestern.edu

Abstract

Previous research has shown that word embeddings can capture semantics. Recently, research has shown that through recurrent neural networks (RNNs), one can generate a definition from a word's embedding. Such an approach offers impressive results despite ignoring that a word's homographs may have different meanings and definition structures. Our work improves upon this model by distinguishing among different definitions for the same word through a given part-of-speech (POS) and a given context sentence.

1 Introduction

Word embeddings, or continuous representations of words, are instrumental in many machine learning tasks including information retrieval and natural language processing. They are a more efficient representation of words, and they have contextual similarity, allowing for improved modeling of data.

(Mikolov et al., 2013) first found word embedding semantic relationships through a vector offset method, also known as parallelogram similarity. More work has been done to improve embeddings for the analogy tasks, but until recently, none try to extract information solely from a single embedding itself. (Noraset et al., 2016) found that using RNN models, the researchers could generate definitions for a given word and its embedding. Our work is based off the latter work, and we will describe the original and improved models in detail later in Section 2.

An open problem with word embeddings is the issue of homographs, or words that are spelled the same but have multiple meanings. When word embeddings are trained, the model does not inherently know which definition, or sense, that the target word is using in a given context. This causes the word embedding to attempt to capture all senses of the word in one representation.

An approach to solve this problem is word-sense disambiguation. Given a context sentence and a list of potential senses, the system selects the sense that has the highest probability of being the correct sense. The context can be represented with two main features: collocation features and bag-of-words features. Collocation features are features of words within a small window around the context. This could include what the word is, its distance from the target word, its POS, etc. Bag-of-words features counts the number of times words occur near the target inside a given window, where order and distance do not matter.

Recent research by (Sun et al., 2017) have come up with a simple approach to providing these context vectors without much computation. The researchers built a normalized coocurrence matrix from a Wikipedia dump. They then used a weighted sum of the embeddings of all the words in a specific context. This approach has worked well against state of the art models for unsupervised polysemous word representation learning. We integrate this work into our model to isolate different senses of a word.

An example sentence is a good way to differentiate different sense of a target word by putting a constraint on the number of senses that can be used. *Taking a check to the bank teller* is significantly different from *sitting on the bank of the river*. *Check* and *teller* will occur around one sense of *bank* much more often than *river*. Another way to differentiate is the target word senses is the POS for the target word. By integrating POS, the model can differentiate *the bird flies* from *those are flies*.

Our contributions are as follows:

Word	POS	Generated Definition
thud	noun	a sharp noise
terrorism	noun	a state of extreme and violent fear
mononucleosis	noun	inflammation of the brain
ode	noun	a short poem of songlike quality
Jerusalem	noun	the capital and largest city of Israel
nazis	noun	a person who is regarded as eccentric or mad

Figure 1: A selection of generated definitions.

1. We recreate the definition modeling experiment in (Noraset et al., 2016) using definitions from WordNet and Oxford English Dictionary (OED), showing the reproducibility in their experiments.

2. We integrate POS into the model to differentiate between senses of the defined words.

3. We highlight a lack of data for definition contexts.

4. We show that by separating out the senses of homographs, we can increase the quality of definitions by up to 1 BLEU score.

2 Previous Work

2.1 Definition Modeling

(Noraset et al., 2016) built a RNN language model that took in a word embedding and a definition, and was able to output high quality definitions for unseen word embeddings. The model is an encoder-decoder LSTM, but it has a modified update gate influenced by the GRU structure proposed by (Cho et al., 2014). The modification allows the word being defined to be fed in at every time step, but only when it is necessary. The model passes over words that carry semantic importance, instead of structural or stop words.

In addition to the embedding of the word being defined, the characters of the word are passed through a CNN Before passing the embedding through the RNN, the researchers concatenated a CNN-trained vector that passed over the characters of the word. This was done to capture affixes to words. For example, *reappear* has a semantic relatedness to *appear* and the CNN captured that relatedness. The CNN passed of a sequence that consisted of each character in the target word represented with one-hot encoding. Using multiple kernels, the researchers used max pooling to create the final representation that was then concatenated to the word embedding.

2.2 Multisense

In Multisense contextual vectors, (Sun et al., 2017) create a cooccurence matrix using data from a 2016 Wikipedia Dump. They then normalize the matrix by dividing each of the matrix values by frequency of the two words. They then take a context window for a target word, which in this case had a window size of 5. They then sum up the weighted embedding vectors for every word in the context window and divide by the total number of words to get the weighted average. Using this approach, (Sun et al., 2017) match the performance of, or outperform, previous papers that looked at embeddings for homographs.

3 Our Approach

Our first goal for this project was reproducing the results presented in Definition Modeling. The codebase was fairly extensive so it took some time to understand all of the moving pieces. It was especially difficult when we first started out, because we looked at the torch version. Once we had the TensorFlow model in place, we added POS as a one-hot encoding vector.

3.1 POS Implementation

Each definition was tagged with a POS. In WordNet, the POS tags included *n, v, a, s,* and *r*, which corresponded to noun, verb, adjective, adjective satellite, and adverb, respectively. In the Oxford English Dictionary, parts of speech are *noun*, *adjective*, *verb*, *adverb*, *residual*, *interjection*, *preposition*, *pronoun*, *numeral*, *conjunction*, *determiner*, *contraction*, *predeterminer*, and *other*. Adjective satellites are particular to the WordNet dataset. Adjectives are arranged in clusters of either head synsets or satellite synsets. To elaborate, the head synset represents a general-

ized meaning of a group of words, and each satellite synset represents are more specific sense of that meaning. These POS tags are encoded in 5- and 14-dimensional one-hot vectors, respectively (per model). This POS vector was concatenated with the CNN affix vector to the word embedding, which is then fed into the encoder.

3.2 Context Implementation

The next step was adding context vectors to the model. One of the main issues with adding context sentences to embeddings was that the WordNet database does not include context sentences. It would be impractical to manually add context sentences to almost 100,000 definitions. It is impossible to scrape the web for context sentences for a given word and part of speech because there generally multiple senses for a single (word, POS) pair. Instead, we pulled data from the Oxford English Dictionary (OED) because it contains example sentences for some of definitions.

Once we had acquired context sentences, the next step was to implement the context vectors. Our goal was to follow the implementation of (Sun et al., 2017). Due to time constraints, we were only able to implement an unweighted average of the context word embeddings to make the overall context embedding. Instead of using a specific window size like in the multisense embeddings, we used every word in the context sentence. The context vector was then concatenated along with the POS vector to the previous model built in Definition Modeling.

4 Results

We now present our results from our experiments. We evaluate 8 total architectures under 8 schemas. The achitectures are all possible permutations of models using GoogleNews and FastText embeddings with and without POS features trained on WordNet and OED definitions. Results are summarized in Table /reffig:results. An additional model was trained an a subset of OED entries containing example usages, however, as discussed in Section 4.1, the model failed to produce meaningful results.

4.1 Context

Our approach for context failed due to lack of training data. The WordNet training data split contains 94,794 entries (with 11,385 and 11,386 in the test and validation splits, respectively for a a total of 117,565 total entries). In contrast, the entire OED dataset contains 67,767 entries, of which only 30,837 have example sentences. Splitting the 30,837 entries into an 80/10/10 training/test/validation scheme (as used in the original work) left only 24,670 entries for the training dataset. Despite many training runs and hyperparameter adjustment, the context language model failed to converge, much less produce meaningful definitions. We removed when context and POS vectors to reduce the number of trainable parameters but the vanilla model still failed to converge. After 50 iterations the model was only able to achieve a perplexity of 157.7 on the held out validation set. We did not attempt to evaluate the model's decoded samples under BLEU score given the extremely poor quality (Figure 2).

Our baseline architecture does not include either POS or context. Since the WordNet dataset does not contain any context sentences, we compare the baseline and POS-added architectures between OED and WordNet.

4.2 Embeddings

Our models used both GoogleNews and FastText embeddings to see if changing the embeddings from the implementation by (Noraset et al., 2016) would improve the results. Table 1 shows that for the WordNet definitions, GoogleNews almost always performs better than FastText. For the OED definitions, the results were more mixed, with no significantly better embedding set. We argue the reason there was no clear outperformer with the OED definitions is due to the fact that OED is lower quality overall. It does not have to do with the differences in the embeddings. The GoogleNews embeddings performed better than FastText with the WordNet dataset. We hypothesize this is because GoogleNews embeddings are trained as full words whereas FastText embeddings are trained on character ngrams. But we cannot say for certain why this would impact the results.

4.3 POS

Adding a POS one-hot vector to the embedding gave us a bittersweet improvement. Table 1 shows that adding in the POS vector gave a 0.5-1 BLEU score increase across the board. This increase occurred regardless of the dictionary or embedding set, or decoding temperature that we used. The models with the lowest perplexity and the high-

Word	POS	Generated Definition
consecutive	adjective	have a recording or (often used for each of three one , in the number equivalent to the same rough - surfaced untrue) in four circumstances , equal ten ; half
shining	adjective	cheese , or brilliant , given a relations , especially with any chain or punishment on property)
misfortune	noun	a misfortune or misfortune that misfortune misfortune misfortune so misfortune $\times 31$
sex	verb	efficient , especially collectively
offset	verb	(of a person) pass or by the killing which someone has a hard deal of all

Figure 2: Sample of definitions generated by model with additional context and POS features. The model failed to learn a basic distribution of the English language. It also does not appear to incorporate any sense of the word in the definition.

est BLEU scores all had POS included. While we could see an increase in BLEU scores with POS, it is not statistically significant. The decrease in the amount of mappings from a concatenated embedding to a group of definitions was not enough.

One interesting thing to note is that while WordNet only have 5 different parts of speech, OED had 14. This increase in the number of different embeddings did not seem to impact the results, as the increase between WordNet and OED BLEU scores were similar when POS was added. There are a variety of factors that could have led to this, so we can only say that POS encoding cannot improve the model significantly by itself.

4.4 Dictionaries

A quick review of the results indicates that the two dictionaries performed wildly differently. Across all scored sampling temperates and greedy, the model trained and evaluated on WordNet definitions achieved an average BLEU score of 16.9. The model trained and evaluated on the Oxford English Dictionary definitions only achieved an 11.8. In one sense, this is a clear victory for the WordNet model. However, this victory is not without caveats. First, as discussed previously, the WordNet dataset is almost double the size of the OED dataset (117,565 entries versus 67,767 entries). Model evaluation performance is certainly not linear with the size of dataset and is generally asymptotic after a certain point. Furthermore, the non-context OED model did demonstrate convergence (unlike the context model). Still, the disparity in training data can be at least partially attributed to the difference in performance.

The greatest source of disparity between the model performances is likely the definitions themselves. The definitions that OED provides are on average far longer and often include an example to contextualize the word. Consider the definitions for *apple*:

- **Oxford English Dictionary**: the round fruit of a tree of the rose family, which typically has thin red or green skin and crisp flesh. Many varieties have been developed as dessert or cooking fruit or for making cider.
- **WordNet**: fruit with red or yellow or green skin and sweet to tart crisp whitish flesh.

While the OED prides itself on these kinds of definitions, such length and context are far more difficult to recreate. It is unsuprising that the WordNet model has much higher evaluation scores.

4.5 Sampling and Scoring

The trained language model learns a distribution over the provided definitions and hence can output the likelihood of a definition. Using this likelihood along with sampling decoding, we were able to achieve much better performance than the baseline greedy decoding results. Using a fixed sample size of 40 (the number of samples to generate per testing entry), we evaluated across 3 sampling temperatues and 3 scoring schemas. For temperatures, we sampled with $t = \{0.05, 0.005, 0.0005\}$. For scoring schemas, we tested how scoring only the most likely 1, 5, and 10 (of the 40 total) per entry effected the BLEU score evaluation. First, we found that choosing between 1, 5, and 10 samples to score did not offer any significant differences

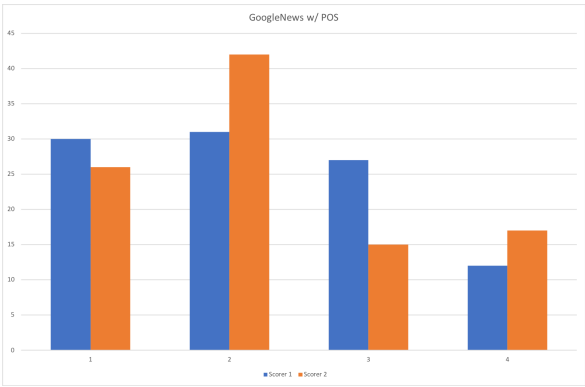

Figure 3: Distributions of subjective ratings of definitions from wn_googlenews_pos model. Blue and orange bars are for 2 different scorers. Scores range from 1 (left) to 4 (right), where 1 is non-sensical or grammatically incorrect and 4 is dictionary-quality.

across BLEU scores. We ultimately chose to only score the most likely decoded sample to most accurately mirror a dictionary which contains a single entry per word (ignoring homographs).

Table 1 presents the results across all models and sampling temperatures, scoring only the most likely generated sample where applicable. Quantitatively, a decoding temperature of $t = 0.0005$ offered the best scored results, while a temperature of $t = 0.00005$ offered the best unscored results (marginally better than the baseline greedy decoding). Not too much can actually be drawn from these differences as 1-2 BLEU points is relatively insignificant, so a difference of less than 1 BLEU point is unlikely to be perceptible.

5 Lessons Learned

We were very excited about this project. It was our first implementation of academic-level research. While we have read research papers and looked at implementations of research in a theoretical context, it is very different working on it first hand. It is also the first time we have seriously worked with TensorFlow. Both of us had made toy projects in TensorFlow but nothing as in depth and extensive as this.

We have worked on many projects that have used large amounts of data, but that data has always come preprocessed, or processing is done by a library. Now that we have processed data ourselves, we have really started to see the full benefits of making memory efficient processing algorithms. If we had not thought about how to efficiently process our data, our time to completion would have taken days instead of minutes or hours.

One large oversight on our part was that we did not acquire the OED dataset as early as we should have. The WordNet dataset contained 111,000 definitions and we thought we would get the same out of OED. We only were able to get 67,000 definitions from OED and only 30,000 of those definitions had a context sentence. If we had known earlier that the lack of data would have hindered our results, we could have looked for alternative methods to deal with context or found alternative datasets that were larger.

It was unfortunate the codebase we had worked with was fairly difficult to understand. This was mainly due to a combination of lack of commenting and partial functions. While in theory, the code should be so easy to understand comments would not be necessary, it does not work in practice as much as we would like. Partial functions only seem exacerbate this problem.

6 Future Work

In the future, we would hope to get a context dataset that had significantly more instances. This would allow us to see if context could play a part in differentiating words with multiple senses when we use the Definition Model. Another addition to that would be to weight the word embeddings that are summed to the context vector. But that is contingent on the larger dataset.

The results published by (Noraset et al., 2016) showed that jointly training on both WordNet and GCIDE significantly improved the results. If there was a different dictionary dataset that also contained context sentences, we could merge OED with that dataset to hopefully get enough definitions to lower perplexity as well as boost our overall BLEU scores.

Acknowledgments

This work was supported the Web Artificial Intelligence Labratory (WebSAIL) at Northwestern University. We want to also thank Thanapon Noraset for helping us when we got stuck. Last but not least, we thank Prof. Doug Downey for putting up with us and guiding us throughout this quarter.

	Perplexity	BLEU Score						
		Greedy	t = 0.05		t = 0.005		t = 0.0005	
			Unscored	Scored	Unscored	Scored	Unscored	Scored
wn_googlenews	45.73	17.1	12.4	15.7	17.0	17.5	17.1	17.2
wn_googlenews_pos	44.30	17.4	12.9	16.1	17.5	**17.9**	17.5	17.6
wn_fasttext	54.63	16.6	12.1	15.2	16.5	16.9	16.5	16.6
wn_fasttext_pos	**43.47**	17.3	12.6	15.9	17.3	17.6	17.3	17.3
oed_googlenews	62.44	11.9	9.7	10.0	11.9	12.3	11.9	11.9
oed_googlenews_pos	60.05	12.3	10.1	10.4	12.4	12.8	12.7	12.4
oed_fasttext	60.61	11.8	9.7	10.4	11.8	12.2	11.8	11.8
oed_fasttext_pos	**58.54**	12.5	10.2	11.2	12.4	**12.9**	12.5	12.5

Table 1: Results of all trained models. Bolded scores are the best scores per evaluation (Perplexity and BLEU) for each dictionary model (WordNet and OED). *Scored* columns are calculated from the highest likelihood sample from each sample group.

References

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR* abs/1406.1078. http://arxiv.org/abs/1406.1078.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2016. Definition modeling: Learning to define word embeddings in natural language.

Yifan Sun, Nikhil Rao, and Weicong Ding. 2017. A simple approach to learn polysemous word embeddings.