Fast Training of word2vec Representations Using N-gram Corpora

Filip Ginter, Jenna Kanerva

Department of Information Technology
University of Turku, Finland
ginter@cs.utu.fi, jmnybl@utu.fi

Abstract

In this paper, we study methods to train the popular *word2vec* vector space representation of the lexicon using only n-gram collections. By using the n-grams rather than the full text corpus we gain a substantial speed-up in training, as well as get the opportunity to train from corpora which would not be available otherwise.

1. Introduction

Among the notable recent developments in the research on continuous vector space representations of the lexicon is the introduction of the *word2vec* method of Mikolov et al. (2013a). The *word2vec* method is sufficiently efficient to be trained on corpora in the billion-token range and several models are publicly available that have been trained on corpora in the hundreds of billion token category.

However, the vast majority of researchers in the academia are unlikely to have access to a hundreds of billion word corpus and not necessarily the computational resources to train *word2vec* models on corpora this large either. The availability of such corpora is also often prevented by copyright protection. Fortunately, it is often legally possible to release n-gram collections from these otherwise closed text corpora. For example, in 2006 Google released the first n-gram collection (LDC2006T13) and in 2009 and 2012 they released collections of n-grams by publication year from their book corpus. For Finnish a tri-gram collection was recently released, based on a 5 billion word corpus from journals and other periodicals dating back to 1820.

In this work, our primary objective is to investigate whether and how these n-gram collections can be utilized to induce a vector space representation of the lexicon using the *word2vec* method. Besides being able to learn models based on corpora we otherwise would not have access to, we will also be interested in whether models of matching performance in down-stream applications could be trained more efficiently, which would in turn allow us to experiment with various training techniques more freely.

2. *word2vec* skip-gram architecture

The *word2vec* skip-gram method is a simplified neural network model trained by sliding a context window along the text and learning word representations by training the network to predict a single context word from the focus word at the middle of the sliding window. The network is trained using back-propagation and therefore its learning rate parameter α as well as its gradual decrease over time need to be set appropriately. The choice to use the skip-gram training of *word2vec* was made because it lends itself very naturally to our purposes, as it only requires to be given a single focus-context word pair at a time and does not need the whole context to be available at once.

3. Training from n-gram collections

A single n-gram can be viewed as the left-hand half of a context of its right-most word, and the right-hand half of a context of its left-most word. Even though we irrecoverably lose the access to the complete context window, as we will meet the left-hand and right-hand halves as separate n-grams, this is in fact fully compatible with the skip-gram *word2vec* induction method, which considers only a single context word at a time. The skip-gram training focus-context word pairs for a 5-gram $w_1 \dots w_5$ would thus be the eight pairs $(w_1, w_2) \dots, (w_1, w_5), (w_5, w_1) \dots (w_5, w_4)$. One can also use the fact that for any pair (w_i, w_j), a sliding window method would ultimately also visit (w_j, w_i), and therefore, we can also extract the additional eight pairs $(w_2, w_1) \dots, (w_5, w_1), (w_1, w_5) \dots (w_4, w_5)$, thus totaling 16 training examples from a single 5-gram.

The most direct interpretation of the count C associated with each n-gram would be to repeat the forward and back-propagation steps of the network C times, which would be prohibitively inefficient. One could also use the count C to adjust the learning rate α of the back-propagation algorithm on a per-ngram basis, such that the smaller the count C, the smaller the update of the weights becomes for pairs from this particular n-gram. And finally, we could simply ignore the n-gram counts and treat all n-grams as equal. In what we think is one of the more surprising outcomes of this work, we could not find a way to implement the learning rate adjustment to give better results than simply ignoring the counts, which is the strategy we will use in the evaluation as well. We note, however, that the *word2vec* method downsamples common words, which to some extent achieves the same goal of decreasing the impact of extremely common words on training.

With or without the count-adjusted learning rate, the training then proceeds in the same manner as the *word2vec* skip-gram model, where the pairs of words extracted from each n-gram are used to train the network as if they were extracted from running text.

4. Evaluation

We evaluate the models on three different tasks on Finnish and English. The Finnish corpus comprises of 1.5B tokens extracted from the CommonCrawl Internet crawl dataset, totaling 264M unique 5-grams (Kanerva et al.,

Task	Finnish		English		
	base	n-gram	base1	base2	n-gram
Wordsim	22.95	19.28	45.72	75.71	27.32
SRL	63.81	66.29	66.5	64.83	65.96

Table 1: F_1-scores for the word similarity and semantic role labeling tasks.

	English		
	base1	base2	n-gram
FIN-base	41.2-34.4	46.6-44.9	41.2-33.8
FIN-n-gram	40.5-39.0	48.7-48.2	42.2-37.4

Table 2: Top-5 accuracy of Finnish→English (first number) and English→Finnish (second number) translation.

2014). For English, we use the 5-grams from the well-known Google Books n-gram collection (569M unique 5-grams) (Lin et al., 2012). However, as we do not have access to the underlying texts from which the n-grams were collected, we use two English baseline models instead: one induced on 5.7B words from the union of the English Gigaword and Wikipedia corpora (referred to as base1 in the text), and also the publicly available (at `https://code.google.com/p/word2vec`) model built on 100B words from the Google News corpus (base2).

In the *word similarity task*, we follow the setting of Mikolov et al. (2013a) who used an English dataset of 19,544 semantic and syntactic queries to test the ability to recover word similarities by a simple analogy technique. The English test set contains 17,232 queries. For Finnish, we imitate the English dataset when applicable, obtaining 13,840 queries.

In the *translation task* we test the property of *word2vec* representations lending themselves to a simple linear transformation from one language vector space to another, as shown by Mikolov et al. (2013b). Replicating their experimental setup, we use Finnish—English translation word pairs obtained by taking the most frequent words in the Finnish corpus and translating them with Google Translate into English. In total we use 7K word pairs, making sure that no word is present in any two of the training, development, and test sets.

Finally, we compare the models on the *semantic role labeling task*, using the SRL system of Kanerva and Ginter (2014) which applies word vectors as the primary source of information. The Finnish results are reported on the *Finnish PropBank*, a 205K word corpus of manually annotated predicate-argument structures (Haverinen et al., 2013). The English results are based on the SRL data from the CoNLL'09 Shared Task (Hajič et al., 2009).

4.1 Evaluation results

The results of the word similarity and semantic role labeling tasks are shown in Table 1, and the results of the translation task are shown in Table 2. The comparison of full-text vs. n-gram trained models is currently best seen for Finnish, where we use the same corpus throughout all experiments. On the SRL and translation tasks, the n-gram trained models match or even surpass the full text models. In the word similarity task, the Finnish scores are closely comparable as well. For English, on the SRL and translation tasks the n-gram based models are on a par with the full text baselines, while for the word similarity task, the baseline English models give a vastly superior performance. This we however think is likely due to the different underlying corpora (Google Books for n-grams verus English News and Wikipedia for the baselines). The possibility for substantially different scores even among models trained on full text is well demonstrated by the 30pp difference between the two English baselines. Further experiments are thus needed for English to establish comparable scores.

For the translation task, it is interesting to note that the top-5 accuracy results are well in line with those previously reported in (Mikolov et al., 2013b), who show top-5 accuracy ranging from 42% for the Czech→English pair to 52% for the Spanish→English pair. With models only trained on the n-gram collections, we obtain 42.2% for Finnish→English and 37.4% for the English→ Finnish pair.

One of the primary advantages of training the models on n-gram collections is their relative compactness. This is especially prominent on the Google Books n-gram corpus. With a context span of ± 4 words, the training on the n-gram corpus consists of 9.1B word pairs, while in the full text the corresponding count is a staggering 3.7T pairs. This amounts to over 400$\times$ speed-up in the training time, which is the difference between several hours and several months. In practical terms, even the largest English n-gram models could be trained in 12 hours on a single computer.

5. Conclusions and future work

Overall, we find — where comparable scores exist — that the performance of the models trained on n-grams approaches models trained on full text. Considering the enormous reduction in training time, especially with the large English corpora where the training is two orders of magnitude faster when carried out on n-grams, we see this as a viable technique that will allow further experimentation and parameter optimization which would be otherwise prohibitive due to the computational costs involved.

As the future work (some which has already been carried out but did not fit into the extended abstract page limit), we will expand the experiments to include also syntactically informed *word2vec* models, utilizing the recently released syntactic n-gram corpora for English and Finnish (Goldberg and Orwant, 2013; Kanerva et al., 2014). The evaluation on English will be extended to provide fully comparable scores based on the same underlying corpus.

All source code necessary for replicating our results and training new models will be made publicly available at `https://github.com/fginter/gensim`.

Acknowledgments

This work has been supported by the Emil Aaltonen Foundation and Kone Foundation. Computational resources were provided by CSC – IT Center for Science. We would also like to thank Radim Řehůřek for making the *gensim* implementation available.

References

Yoav Goldberg and Jon Orwant. 2013. A dataset of Syntactic-Ngrams over time from a very large corpus of English books. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 241–247. Association for Computational Linguistics.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics.

Katri Haverinen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Jenna Nyblom, Stina Ojala, Timo Viljanen, Tapio Salakoski, and Filip Ginter. 2013. Towards a dependency-based PropBank of general Finnish. In *Proceedings of the 19th Nordic Conference on Computational Linguistics (NoDaLiDa'13)*, pages 41–57.

Jenna Kanerva and Filip Ginter. 2014. Post-hoc manipulations of vector space models with application to Semantic Role Labeling. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 1–10.

Jenna Kanerva, Juhani Luotolahti, Veronika Laippala, and Filip Ginter. 2014. Syntactic n-gram collection from a large-scale corpus of Internet Finnish. In *Proceedings of the Sixth International Conference Baltic HLT 2014*, pages 184–191. IOS Press.

Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. 2012. Syntactic annotations for the Google Books ngram corpus. In *Proceedings of the ACL 2012 System Demonstrations*, pages 169–174. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Workshop Proceedings of International Conference on Learning Representations*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.