2018 12th IEEE International Conference on Semantic Computing

Context-Based Few-Shot Word Representation Learning

Chenxing Li
IIIS, Tsinghua University
Beijing, China
Email: li-cx17@mails.tsinghua.edu.cn

Gerui Wang
University of Illinois at Urbana-Champaign
Urbana, IL, USA
Email: geruiw2@illinois.edu

Gerard de Melo*
Rutgers University
Piscataway, NJ, USA
Contact: http://gerard.demelo.org

***Abstract*—Word representation learning methods have mostly been designed for and evaluated on frequent words. However, in real-world settings, deep neural architectures are often expected to accept a large vocabulary of possible input words. In this paper, we investigate context-based techniques for few-shot learning of representations for infrequent words. We first adapt word2vec to account for expanded contexts and subsequently introduce an additional smoothing procedure. Experiments on similarity benchmarks show significant improvements for rare words.**

I. Introduction

Unlike many traditional semantic analysis methods, deep learning architectures for text generally assume a fixed input vocabulary, mapped to a set of word vectors. However, end users often have the expectation that a given natural language processing tool will be able to operate on virtually any valid input word, rather than outputting `<UNK>` for unknown words it has not been trained on. Thus, it is important to gracefully handle out-of-vocabulary words not in the training data.

Embeddings pretrained on large amounts of external unlabeled text, to some extent, help us to cope with infrequent words. However, word2vec [1] is based on the idea of repeated gradient updates for each word vector and normally uses a frequency cut-off such that words occurring less often than a threshold (5 by default) are excluded from the vocabulary. Several recent papers propose to address this using external data such as dictionaries, which provide word definitions [2], [3], [4], [5] and semantic relationships [6]. In certain cases, word forms arise from regular processes and hence can be interpreted via morphological or character-based architectures [7], [8], [9]. Such models may be able to guess the meaning of a word such as *trilateralism*, for instance.

In the absence of such internal cues as well as of external information, when faced with a word observed only once or a few times, we have to pay more attention to its context to obtain a reasonable representation. This paper explores how to achieve this, by more effectively exploiting the little contextual information that is available. Hearing a new word used in context, humans are remarkably adept at inferring a basic notion of a its meaning. To illustrate this point, consider the fictional word *wampimuk* in the sentence "*We found a cute, hairy wampimuk sleeping behind the tree.*" [10], [11].

*Gerard de Melo's research is supported by the DARPA SocialSim program.

This paper investigates how to instantiate these ideas in context-based methods for improved word representations, geared towards few-shot settings. Section II first introduces our technique for expanded context modeling and finally proposes an additional smoothing procedure.

II. Context Modeling

Our model extends the well-known word2vec skip-gram with negative sampling [1] approach. Rather than just using unigram words as context, we also capture bigrams in the neighborhood, while accounting for their relative position. Clearly, unigrams and bigrams together are linguistically more informative than just unigrams (consider e.g. *information retrieval* vs. just *information* and *retrieval*). The relative position as well, is informative (for the target word *card*, consider e.g. *card manufacturer* vs. *manufacturer card*).

Unfortunately, modeling such position-specific context bigrams as opposed to just unigrams leads to a quadratic explosion in the number of parameters, and hence would not easily scale to large corpora. We shall resolve this by treating contextual units as hash bin-based features.

In our model, words $w \in V$, individual context words $c \in C$, and features $f \in F$, are represented via vectors. We consider the softmax conditional probability

$$p_\theta\left((w,f) \in E_1 \cup E_2 \mid w, f\right) = \frac{1}{1+e^{-v_f \cdot v_w}}$$

for word vectors v_w and vectors v_f for features as well as context words f, where $E_1 \subseteq V \times C$ contains word-context word pairs (skip-grams) from the input and $E_2 \subseteq V \times F$ contains word-feature pairs. E_1' and E_2' ($E_i \subseteq E_i' \subseteq V \times F, i \in \{1,2\}$) additionally contain pairs $(w,f) \in V \times (C \cup F)$ used in negative sampling. Let $E = E_1 \cup E_2, E' = E_1' \cup E_2'$. We then define a loss function as follows:

$$L(\theta) = -\frac{1}{|E'|}\left(\sum_{(w,f)\in E'} \log p_\theta\left((w,f) \in E \mid w, f\right) - \sum_{(w,c)\in E'} \log p_\theta\left((w,f) \notin E \mid w, f\right)\right)$$

0-7695-6360-0/18/$31.00 ©2018 IEEE
DOI 10.1109/ICSC.2018.00041

A. Neighbor and Bigram Cues

In addition to regular context words as in the standard skip-gram model, we consider immediate neighbor and bigram cues for the 1 or 2 words, respectively, that immediately precede or follow the target word w_i, captured as tuples $(w_{i-1}, \text{“}-\text{”})$, $(w_{i-2}, w_{i-1}, \text{“}-\text{”})$, $(w_{i+1}, \text{“}+\text{”})$, or $(w_{i+1}, w_{i+2}, \text{“}+\text{”})$ in a set F_0. The label serves to distinguish whether the context words appear before (“−”) or after (“+”) the target word.

B. Hashing

Straightforwardly using F_0 as prediction targets for the skip-gram model would lead to a quadratic increase of the output vocabulary size. For scalability, we instead fix the size of F in advance and define binned features

$$f_i = \{f_0 \mid h(f_0) \bmod |F| = i\}$$

for i in $0, \ldots, |F| - 1$, and a suitable hash function h. Thus, instead of learning to predict individual neighbors or bigram cues, our loss function encourages the model to learn word vectors that are predictive of features that bin together *sets* of neighbor and bigram cues (in addition to also predicting individual context words as in the regular word2vec skip-gram model). Although the model now no longer needs to predict individual neighbors or bigrams within each binned set, it still is trained to select the most likely such binned sets out of a large number (typically millions) of potential candidate bins.

C. Neighborhood-Based Smoothing

While the contextual features above provide more detailed information about a word's occurrence contexts, few-shot or even one-shot learning remains difficult, since the features may turn out to provide too sparse information. Previous work showed that distributional vectors can be used to better initialize embedding methods [12]. Inspired by transductive learning methods, we instead adopt a form of relational smoothing specifically targeting rare words by using vector representations of similar words.

Given clusters $C_i \in C$ of words, as well as two integer thresholds T_1, T_2 as hyperparameters, we define

$$V' = \{w \in V : \mathcal{F}(w) \leq T_1\}$$
$$C'_i = \{w \in C_i : \mathcal{F}(w) > T_2\}$$

where $\mathcal{F}(w)$ is the frequency of w in the training corpus, V' is the set of rare words, while C'_i is the set of non-rare words in a word cluster C_i. Given initial word embeddings $v_0(w)$ for every word $w \in V$, we first define $v_i = \frac{1}{|C'_i|} \sum_{w \in C'_i} v_0(w)$. Then, each word $w \in V$ is assigned a new vector

$$v_\alpha(w) = \begin{cases} v_0(w) & \text{if } w \notin V' \text{ or } |C'_{i(w)}| = 0 \\ (1-\alpha)v_0(w) + \alpha v_{i(w)} & \text{otherwise} \end{cases}$$

where α is a hyperparameter and $i(w)$ denotes the index of the cluster C_i that includes w.

To induce the word clusters $C_i \in C$, we rely on Brown clustering [13], which greedily merges words into clusters in terms of their contextual similarity. Brown clustering is competitive with certain (non-state-of-the-art) vector-based methods [14], [15], [16]. In our experiments, we train the Brown clusters on the training corpus. In practice, a post-hoc form of few-shot learning can be supported as well, because words with the lowest frequencies are processed last by the algorithm, so any new word at test time can be appended to the vocabulary. Assuming previous clusters remain unchanged, one just runs an extra loop iteration to observe with which previous cluster the newly added word will be merged.

III. Experiments

A. Corpus

All experiments are carried out on the filtered plaintext version (“fil9”) of the 1GB enwik9 Wikipedia corpus[1]. Its small size enables us to obtain a greater ratio of infrequent words in available word relatedness benchmarks.

B. Benchmarks

The task of word similarity evaluation involves computing the cosine similarity of word vectors between pairs of words, and then computing Spearman's rank correlation compared to gold standard human ratings. For this evaluation, we created subsets $\{\text{RWt}i\_j\}_i$ of the Stanford Rare Word Similarity Dataset[2], containing pairs for which both words appear at least i times, but at most one appears more than j times in the training corpus. We also created extra subsets $\{\text{RWs}i\_j\}_i$, containing pairs for which both appear at most j times, but at most one appears less than i times. These subsets were all binned so as to ensure that each subset had at least 150 pairs. The resulting datasets, together with RW01, which contains all pairs occurring in our corpus, are listed in Table I.

Dataset	Word Pairs
RWt1_2	191
RWt3_4	168
RWt5_7	185
RWt8_12	184
RWt13_22	192
RWt23_40	194
RWt41_80	180
RWs1_50	162
RWs51_150	155
RWt151_300	151
RW01	1928

TABLE I: Subsets of RW01

C. Training

We use skip-grams with negative-sampling, training for 20 iterations. For lack of an alternative, we tuned the hyper-parameters on WS353t1_2000, a subset of the well-known WS353 word similarity benchmark created analogously to the $\{\text{RWt}i\_j\}_i$ sets.[3] The optimal parameters on this set were then

[1] http://mattmahoney.net/dc/textdata

[2] http://stanford.edu/~lmthang/morphoNLM/

[3] For WS353, word frequencies are generally high. Hence, j is increased to 2000 to obtain sufficient word pairs.

applied to the test set. In particular, we use a dimensionality of 300, a max. neighbor distance of 5, and 5-fold negative sampling. For the feature-based context modeling, the size of feature set F ranged from 10^5 to 2×10^7. For the smoothing procedure, we use $|C| = 2000$ classes and fix $\alpha = 0.5$.

D. Results

In our main experiment, we combined the feature-based context modeling and the neighborhood-based smoothing by applying the smoothing method on embeddings trained using our feature-based context modeling. We retain all bigram features and rely on hashing to reduce them to a fixed feature set F. We tuned the size of F in feature-based context modeling first and obtained $|F| = 2 \times 10^7$. Then, we tuned the smoothing and obtained $T_1 = 22$, $T_2 = 3200$.

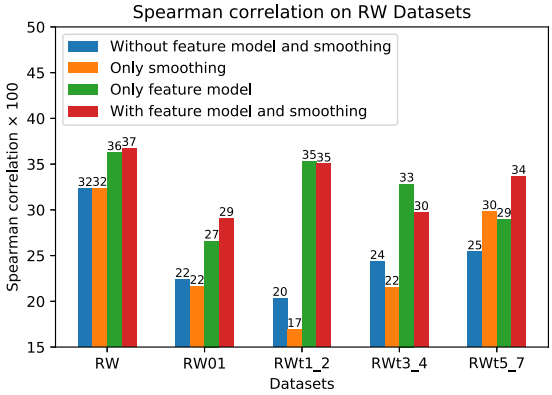
We compared the Spearman correlation between the word2vec baseline, embeddings trained with feature-based context modeling, and final embeddings after additional smoothing. Figure 1a shows that our methods provide substantial gains on rare words, particularly for very infrequent ones. Figures 1b/1c show that our methods do not harm the vector quality for frequent words. Since $T_1 = 22$, the smoothing model does not affect words with frequency larger than 22, especially for WS353 and SimLex999, where the word frequencies are generally very high.

E. Analysis of Feature Hashing

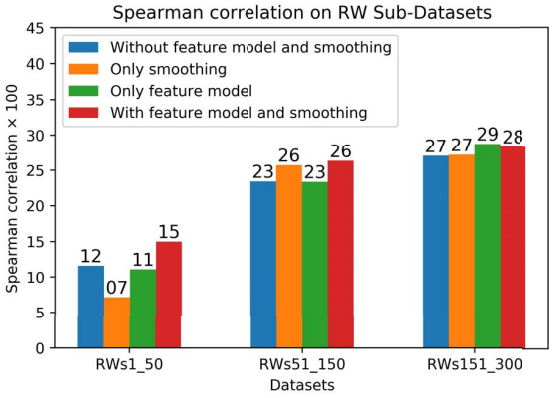
We further analysed the contribution of the features without further smoothing. The size of feature set F ranged from 10^5 to 2×10^7. Figure 2a shows that the Spearman correlation for rare words increases with increasing $|F|$, while Figure 2b shows that the choice of $|F|$ does not exert any substantial influence on the Spearman correlation on frequent words.

For further analysis, we also compare the Spearman correlation between feature-based modeling and the regular word2vec skip-gram model under the same parameters for more specific test subsets. Figure 3 shows that the feature-based model has apparent advantages on rare words.

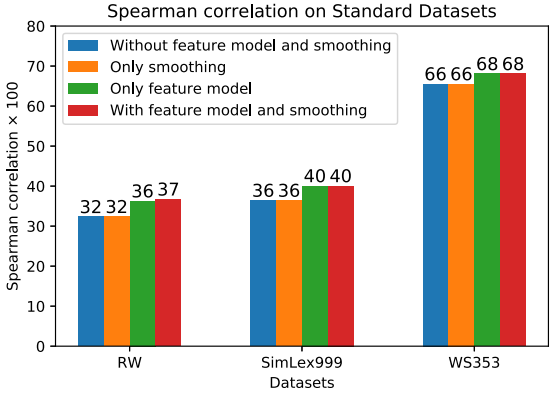
As an example, consider the rare word *inexpert*, which is semantically similar to *unprofessional*. It turns out that with feature-based context modeling (again forgoing the additional context-based smoothing step), the cosine angle between their respective vectors improves from 0.25 to 0.52. The word *inexpert* occurs only twice in our corpus, and one of the sentences is: *The story's technique still seems somewhat inexpert, with passages of local color description occasionally interrupting the flow of the narrative.* The bigram *seems somewhat* has 14 occurrences. Examples of words with the same cue ("`seems`", "`somewhat`", "–") are listed in Table II. The last column shows the similarity between a word w and the word *unprofessional*. The second and third columns show the improvement of similarities between w and *inexpert*. Thus, a higher similarity between *inexpert* and *unprofessional* results from increasing the similarity between *inexpert* and words similar to *unprofessional*.



(a) Spearman correlations on RW and RWt subsets



(b) Spearman correlations on RWs subsets

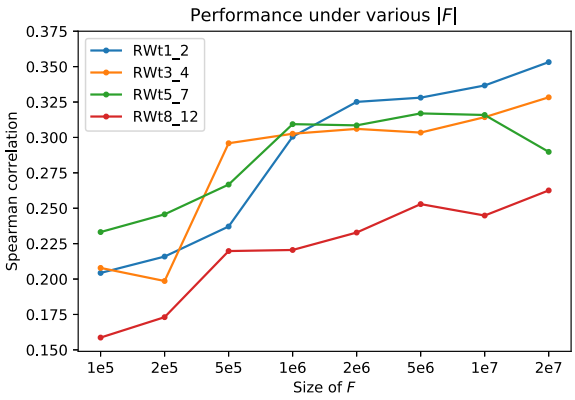


(c) Spearman correlations on standard datasets

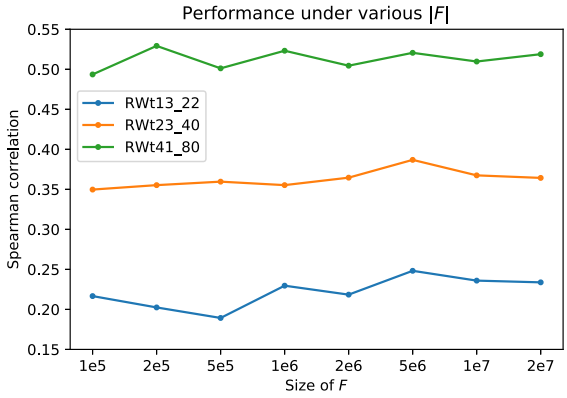
Fig. 1: Main results

F. Analysis of Smoothing Procedure

Figure 3 evaluates smoothing alone, without the feature-based modeling, in comparison with the full model. Recall that the smoothing method is applied only to rare words, for which the threshold T_1 is relatively small. Hence, it has almost no



(a) Dataset RWt with rare words



(b) Dataset RWt with common words

Fig. 2: Spearman correlation for various feature set sizes

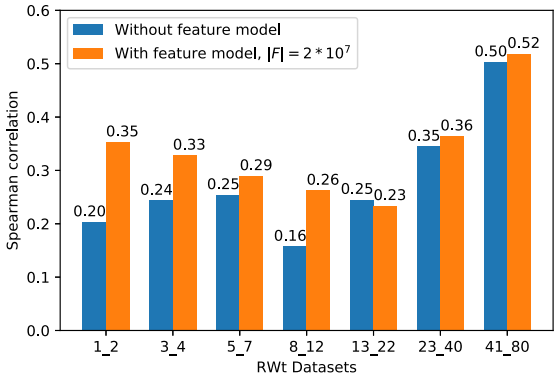


Fig. 3: Model with and without our features

impact on the results for WS353 and SimLex999, implying that it can safely be applied without distorting the results of frequent words. For the RW data set, we do not obtain statistically significant improvements with smoothing alone, yet we obtain even further gains over feature hashing when it is combined with smoothing. This is because smoothing hinges on having sufficiently accurate vectors to start with. This is achieved when both techniques are combined.

Word w	$v_w \cdot v_i$	$v'_w \cdot v'_i$	$v_w \cdot v_u$
inappropriate	0.34	0.43	0.53
portly	0.36	0.59	0.35
flawed	0.33	0.41	0.34
naive	0.31	0.43	0.31
suspect	0.30	0.37	0.28
arbitrary	0.44	0.48	0.27

TABLE II: Case study, where v_w: normalized embedding vector of word w from the model without our features, v'_w: normalized embedding vector of w from the model with our features, u: the word *unprofessional*, i: the word *inexpert*.

IV. Conclusion

We have shown that significantly improved modeling of rare words is possible by 1) accounting for the context in a more fine-grained manner to make the best possible use of the limited information that we have in few-shot learning, and 2) applying smoothing to mitigate the effects of data sparsity. The method proves fairly safe to apply, as it improves vectors of rare words without distorting those of frequent words.

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv:1301.3781*, 2013.

[2] D. Bahdanau, T. Bosc, S. Jastrzębski, E. Grefenstette, P. Vincent, and Y. Bengio, "Learning to Compute Word Embeddings On the Fly," *arXiv:1706.00286*, 2017.

[3] M. T. Pilehvar and N. Collier, "Inducing embeddings for rare and unseen words by leveraging lexical resources," in *Proceedings of EACL*, 2017.

[4] V. Prokhorov, M. T. Pilehvar, D. Kartsaklis, P. Liò, and N. Collier, "Learning rare word representations using semantic bridging," *arXiv:1707.07554*, 2017.

[5] A. Herbelot and M. Baroni, "High-risk learning: acquiring new word vectors from tiny data," in *Proceedings of EMNLP*, 2017, pp. 304–309.

[6] G. de Melo, "Wiktionary-based word embeddings," in *Proceedings of MT Summit XV*, 2015.

[7] L. Wang, Z. Cao, Y. Xia, and G. de Melo, "Morphological segmentation with Window LSTM neural networks," in *Proceedings of AAAI*, 2016.

[8] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of ACL*, August 2016.

[9] C. D. Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging," in *Proceedings of ICML*, ser. Proceedings of Machine Learning Research, vol. 32, no. 2, 2014, pp. 1818–1826.

[10] S. McDonald and M. Ramscar, "Testing the distributional hypothesis: The influence of context on judgements of semantic similarity," in *Proceedings of CogSci*, 2001, pp. 611–616.

[11] A. Lazaridou, E. Bruni, and M. Baroni, "Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world," in *Proceedings of ACL*, 2014.

[12] I. Sergienya and H. Schütze, "Learning better embeddings for rare words using distributional representations," in *Proceedings of EMNLP*, 2015.

[13] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[14] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of ACL*, 2010, pp. 384–394.

[15] L. Derczynski and S. Chester, "Generalised brown clustering and roll-up feature generation," in *Proceedings of AAAI*, 2016.

[16] M. R. Ciosici, "Improving quality of hierarchical clustering for large data series," Ph.D. dissertation, Aarhus University, 2015.