Learning Topic-Sensitive Word Representations

Fadaee, M.; Bisazza, A.; Monz, C.

DOI
10.18653/v1/P17-2070

Publication date
2017

Document Version
Final published version

Published in
The 55th Annual Meeting of the Association for Computational Linguistics

License
CC BY

Citation for published version (APA):
Learning Topic-Sensitive Word Representations

Marzieh Fadaee Arianna Bisazza Christof Monz
Informatics Institute, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, The Netherlands
{m.fadaee,a.bisazza,c.monz}@uva.nl

Abstract

Distributed word representations are widely used for modeling words in NLP tasks. Most of the existing models generate one representation per word and do not consider different meanings of a word. We present two approaches to learn multiple topic-sensitive representations per word by using Hierarchical Dirichlet Process. We observe that by modeling topics and integrating topic distributions for each document we obtain representations that are able to distinguish between different meanings of a given word. Our models yield statistically significant improvements for the lexical substitution task indicating that commonly used single word representations, even when combined with contextual information, are insufficient for this task.

1 Introduction

Word representations in the form of dense vectors, or word embeddings, capture semantic and syntactic information (Mikolov et al., 2013a; Pennington et al., 2014) and are widely used in many NLP tasks (Zou et al., 2013; Levy and Goldberg, 2014; Tang et al., 2014; Gharbieh et al., 2016). Most of the existing models generate one representation per word and do not distinguish between different meanings of a word. However, many tasks can benefit from using multiple representations per word to capture polysemy (Reisinger and Mooney, 2010). There have been several attempts to build repositories for word senses (Miller, 1995;Navigli and Ponzetto, 2010), but this is laborious and limited to few languages. Moreover, defining a universal set of word senses is challenging as polysemous words can exist at many levels of granularity (Kilgarriff, 1997;Navigli, 2012).

In this paper, we introduce a model that uses a nonparametric Bayesian model, Hierarchical Dirichlet Process (HDP), to learn multiple topic-sensitive representations per word. Yao and van Durme (2011) show that HDP is effective in learning topics yielding state-of-the-art performance for sense induction. However, they assume that topics and senses are interchangeable, and train individual models per word making it difficult to scale to large data. Our approach enables us to use HDP to model senses effectively using large unannotated training data. We investigate to what extent distributions over word senses can be approximated by distributions over topics without assuming these concepts to be identical. The contributions of this paper are: (i) We propose three unsupervised, language-independent approaches to approximate senses with topics and learn multiple topic-sensitive embeddings per word. (ii) We show that in the Lexical Substitution ranking task our models outperform two competitive baselines.

2 Topic-Sensitive Representations

In this section we describe the proposed models. To learn topics from a corpus we use HDP (Teh et al., 2006; Lau et al., 2014). The main advantage of this model compared to non-hierarchical methods like the Chinese Restaurant Process (CRP) is that each document in the corpus is modeled using a mixture model with topics shared between all documents (Teh et al., 2005; Brody and Lапата, 2009). HDP yields two sets of distributions that we use in our methods: distributions over topics for words in the vocabulary, and distributions over topics for documents in the corpus. Similarly to Neelakantan et al. (2014), we use neighboring words to detect the meaning of the context, however, we also use the two HDP dis-
Figure 1: Illustration of our topic-sensitive representation models: (a) hard-topic labeled representations (HTLE), (b) hard topic-labeled representations plus generic word representation (HTLEadd), (c) soft topic-labeled representations (STLE).

Contributions. By doing so, we take advantage of the topic of the document beyond the scope of the neighboring words, which is helpful when the immediate context of the target word is not sufficiently informative. We modify the Skipgram model (Mikolov et al., 2013a) to obtain multiple topic-sensitive representations per word type using topic distributions. In addition, we do not cluster context windows and train for different senses of the words individually. This reduces the sparsity problem and provides a better representation estimation for rare words. We assume that meanings of words can be determined by their contextual information and use the distribution over topics to differentiate between occurrences of a word in different contexts, i.e., documents with different topics. We propose three different approaches (see Figure 1): two methods with hard topic labeling of words and one with soft labeling.

2.1 Hard Topic-Labeled Representations

The trained HDP model can be used to hard-label a new corpus with one topic per word through sampling. Our first model variant (Figure 1(a)) relies on hard labeling by simply considering each word-topic pair as a separate vocabulary entry. To reduce sparsity on the context side and share the word-level information between similar contexts, we use topic-sensitive representations for target words (input to the network) and standard, i.e., unlabeled, word representations for context words (output). Note that this results in different input and output vocabularies. The training objective is then to maximize the log-likelihood of context words $w_{i+j}$ given the target word-topic pair $w_i^T$:

$$\mathcal{L}_{\text{HardT-SG}} = \frac{1}{T} \sum_{t=1}^{T} \sum_{i \in \mathcal{I}} \log p(w_{i+j} | w_i^T)$$

where $I$ is the number of words in the training corpus, $c$ is the context size and $\tau$ is the topic assigned to $w_i$ by HDP sampling. The embedding of a word in context $h(w_i)$ is obtained by simply extracting the row of the input lookup table ($r$) corresponding to the HDP-labeled word-topic pair:

$$h_{\text{HTLE}}(w_i) = r(w_i^T)$$

(1)

A possible shortcoming of the HTLE model is that the representations are trained separately and information is not shared between different topic-sensitive representations of the same word. To address this issue, we introduce a model variant that learns multiple topic-sensitive word representations and generic word representations simultaneously (Figure 1(b)). In this variant (HTLEadd), the target word embedding is obtained by adding the word-topic pair representation ($r'$) to the generic representation of the corresponding word ($r_0$):

$$h_{\text{HTLEadd}}(w_i) = r'(w_i^T) + r_0(w_i)$$

(2)

2.2 Soft Topic-Labeled Representations

The model variants above rely on the hard labeling resulting from HDP sampling. As a soft alternative to this, we can directly include the topic distributions estimated by HDP for each document (Figure 1(c)). Specifically, for each update, we use the topic distribution to compute a weighted sum over the word-topic representations ($r''$):

$$h_{\text{STLE}}(w_i) = \sum_{k=1}^{K} p(\tau_k | d_i) \cdot r''(w_i^T \tau_k)$$

(3)

where $T$ is the total number of topics, $d_i$ the document containing $w_i$, and $p(\tau_k | d_i)$ the probability assigned to topic $\tau_k$ by HDP in document $d_i$. The training objective for this model is:

$$\mathcal{L}_{\text{SoftT-SG}} = \frac{1}{T} \sum_{i=1}^{I} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{i+j} | w_i, \tau)$$

442
<table>
<thead>
<tr>
<th>word</th>
<th>bat</th>
<th>Pre-trained w2v</th>
<th>bats, batting, Pinch_hitter_Brayan_Pena, batsman, batted, Hawaiian_hoary, Batting bats, batting, Bat, catcher, fielder, hitter, outfield, hitting, batted, catchers, balls uroderma, magnirostrum, sorenseni, miniopterus, promops, luctus, micronycteris</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Pre-trained GloVe</td>
<td>SGE on Wikipedia</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TSE on Wikipedia</td>
<td>$\tau_1$, $\tau_2$ ball, pitchout, batter, toss-foul, umpire, batting, bowes, straightened, fielder, flies vespertilionidae, heran, hipposideros, sorenseni, luctus, coxi, kerivoula, natterer</td>
</tr>
<tr>
<td>word</td>
<td>jaguar</td>
<td>Pre-trained w2v</td>
<td>jaguars, Macho_B, panther, lynx, rhino, lizzard, tapir, tiger, leopard, Florida_panther</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pre-trained GloVe</td>
<td>SGE on Wikipedia</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TSE on Wikipedia</td>
<td>$\tau_1$, $\tau_2$ ford, bmw, chevrolet, honda, porsche, monza, nissan, dodge, marauder, peugeot, opel, wiedii, puma, margay, tapirus, jaguarundi, yagourouandi, vison, concolor, tajacu</td>
</tr>
<tr>
<td></td>
<td>appeal</td>
<td>Pre-trained w2v</td>
<td>appeals, appealing, appealed, Appeal, rehearing, apezl, Appealing, ruling, Appeals</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pre-trained GloVe</td>
<td>SGE on Wikipedia</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TSE on Wikipedia</td>
<td>$\tau_1$, $\tau_2$ court, case, appeals, appealed, decision, proceedings, disapproves, ruling, sfa, steadfast, lackadaisical, assertions, lack, symbolize, fans, attempt, unenthusiastic</td>
</tr>
</tbody>
</table>

Table 1: Nearest neighbors of three examples in different representation spaces using cosine similarity. w2v and GloVe are pre-trained embeddings from (Mikolov et al., 2013a) and (Pennington et al., 2014) respectively. SGE is the Skipgram baseline and TSE is our topic-sensitive Skipgram (cf. Equation (1)), both trained on Wikipedia. $\tau_k$ stands for HDP-inferred topic $k$.

where $\tau$ is the topic of document $d_i$ learned by HDP. The STLE model has the advantage of directly applying the distribution over topics in the Skipgram model. In addition, for each instance, we update all topic representations of a given word with non-zero probabilities, which has the potential to reduce the sparsity problem.

### 2.3 Embeddings for Polyseamous Words

The representations obtained from our models are expected to capture the meaning of a word in different topics. We now investigate whether these representations can distinguish between different word senses. Table 1 provides examples of nearest neighbors. For comparison we include our own baseline, i.e., embeddings learned with Skipgram on our corpus, as well as Word2Vec (Mikolov et al., 2013b) and GloVe embeddings (Pennington et al., 2014) pre-trained on large data.

In the first example, the word *bat* has two different meanings: animal or sports device. One can see that the nearest neighbors of the baseline and pre-trained word representations either center around one primary, i.e., most frequent, meaning of the word, or it is a mixture of different meanings. The topic-sensitive representations, on the other hand, correctly distinguish between the two different meanings. A similar pattern is observed for the word *jaguar* and its two meanings: car or animal. The last example, *appeal*, illustrates a case where topic-sensitive embeddings are not clearly detecting different meanings of the word, despite having some correct words in the lists. Here, the meaning *attract* does not seem to be captured by any embedding set.

These observations suggest that topic-sensitive representations capture different word senses to some extent. To provide a systematic validation of our approach, we now investigate whether topic-sensitive representations can improve tasks where polysemy is a known issue.

### 3 Evaluation

In this section we present the setup for our experiments and empirically evaluate our approach on the context-aware word similarity and lexical substitution tasks.

#### 3.1 Experimental setup

All word representations are learned on the English Wikipedia corpus containing 4.8M documents (1B tokens). The topics are learned on a 100K-document subset of this corpus using the HDP implementation of Teh et al. (2006). Once the topics have been learned, we run HDP on the whole corpus to obtain the word-topic labeling (see Section 2.1) and the document-level topic distributions (Section 2.2). We train each model vari-
ant with window size $c = 10$ and different embedding sizes (100, 300, 600) initialized randomly.

We compare our models to several baselines: Skipgram (SGE) and the best-performing multi-sense embeddings model per word type (MSSG) (Neelakantan et al., 2014). All model variants are trained on the same training data with the same settings, following suggestions by Mikolov et al. (2013a) and Levy et al. (2015). For MSSG we use the best performing similarity measure (avgSimC) as proposed by Neelakantan et al. (2014).

### 3.2 Context-Aware Word Similarity Task

Despite its shortcomings (Faruqui et al., 2016), word similarity remains the most frequently used method of evaluation in the literature. There are multiple test sets available but in almost all of them word pairs are considered out of context. To the best of our knowledge, the only word similarity data set providing word context is SCWS (Huang et al., 2012). To evaluate our models on SCWS, we run HDP on the data treating each word’s context as a separate document. We compute the similarity of each word pair as follows:

$$Sim(w_1, w_2) = \cos(h(w_1), h(w_2))$$

where $h(w_i)$ refers to any of the topic-sensitive representations defined in Section 2. Note that $w_1$ and $w_2$ can refer to the same word.

Table 2 provides the Spearman’s correlation scores for different models against the human ranking. We see that with dimensions 100 and 300, two of our models obtain improvements over the baseline. The MSSG model of Neelakantan et al. (2014) performs only slightly better than our HLTE model by requiring considerably more parameters (600 vs. 100 embedding size).

### 3.3 Lexical Substitution Task

This task requires one to identify the best replacements for a word in a sentential context. The presence of many polysemous target words makes this task more suitable for evaluating sense embedding. Following Melamud et al. (2015) we pool substitutions from different instances and rank them by the number of annotators that selected them for a given context. We use two evaluation sets: LS-SE07 (McCarthy andNavigli, 2007), and LS-CIC (Kremer et al., 2014).

Unlike previous work (Szarvas et al., 2013; Kremer et al., 2014; Melamud et al., 2015) we do not use any syntactic information, motivated by the fact that high-quality parsers are not available for most languages. The evaluation is performed by computing the Generalized Average Precision (GAP) score (Kishida, 2005). We run HDP on the evaluation set and compute the similarity between target word $w_t$ and each substitution $w_s$ using two different inference methods in line with how we incorporate topics during training:

**Sampled (Smp):**

$$Sim_{\text{Smp}}(w_s, w_t) = \frac{\cos(h(w_s^\tau), h(w_t^\tau)) + \sum_{c} \cos(h(w_s^c), o(w_c))}{C}$$

**Expected (Exp):**

$$Sim_{\text{Exp}}(w_s, w_t) = \frac{\sum_{\tau} \frac{p(\tau)p(\tau^\tau)}{p(\tau)}}{C} \frac{\sum_{\tau, c} \cos(h(w_s^\tau), o(w_c))p(\tau)}{C}$$

where $h(w_s^\tau)$ and $h(w_t^\tau)$ are the representations for substitution word $s$ with topic $\tau$ and target word $t$ with topic $\tau^\tau$ respectively (cf. Section 2), $w_c$ are context words of $w_t$ taken from a sliding window of the same size as the embeddings, $o(w_c)$ is the context (i.e., output) representation of $w_c$, and $C$ is the total number of context words. Note
that these two methods are consistent with how we train HTLE and STLE.

The sampled method, similar to HTLE, uses the HDP model to assign topics to word occurrences during testing. The expected method, similar to STLE, uses the HDP model to learn the probability distribution of topics of the context sentence and uses the entire distribution to compute the similarity. For the Skipgram baseline we compute the similarity $\text{Sim}_{\text{SGE}+\text{C}}(w_s, w_t)$ as follows:

$$\cos(h(w_s), h(w_t)) + \frac{\sum \cos(h(w_s), o(w_i))}{C}$$

which uses the similarity between the substitution word and all words in the context, as well as the similarity of target and substitution words.

Table 3 shows the GAP scores of our models and baselines.\(^1\) One can see that all models using multiple embeddings per word perform better than SGE. Our proposed models outperform both SGE and MSSG in both evaluation sets, with more pronounced improvements for LS-CIC. We further observe that our expected method is more robust and performs better for all embedding sizes.

Table 4 shows the GAP scores broken down by the main word classes: noun, verb, adjective, and adverb. With 100 dimensions our best model (HTLE) yields improvements across all POS tags, with the largest improvements for adverbs and smallest improvements for adjectives. When increasing the dimension size of embeddings the improvements hold up for all POS tags apart from adverbs. It can be inferred that larger dimension sizes capture semantic similarities for adverbs and context words better than other parts-of-speech categories. Additionally, we observe for both evaluation sets that the improvements are preserved when increasing the embedding size.

### 4 Related Work

While the most commonly used approaches learn one embedding per word type (Mikolov et al., 2013a; Pennington et al., 2014), recent studies have focused on learning multiple embeddings per word due to the ambiguous nature of language (Qiu et al., 2016). Huang et al. (2012) cluster word contexts and use the average embedding of each cluster as word sense embeddings, which yields improvements on a word similarity task. Nee-lakantan et al. (2014) propose two approaches, both based on clustering word contexts: In the first, they fix the number of senses manually, and in the second, they use an ad-hoc greedy procedure that allocates a new representation to a word if existing representations explain the context below a certain threshold. Li and Jurafsky (2015) used a CRP model to distinguish between senses of words and train vectors for senses, where the number of senses is not fixed. They use two heuristic approaches for assigning senses in a context: ‘greedy’ which assigns the locally optimum sense label to each word, and ‘expectation’ which computes the expected value for a word in a given context with probabilities for each possible sense.

### 5 Conclusions

We have introduced an approach to learn topic-sensitive word representations that exploits the document-level context of words and does not require annotated data or linguistic resources. Our evaluation on the lexical substitution task suggests that topic distributions capture word senses to some extent. Moreover, we obtain statistically significant improvements in the lexical substitution task while not using any syntactic information. The best results are achieved by our hard topic-labeled model which learns topic-sensitive representations by assigning topics to target words.

### Acknowledgments

This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project numbers 639.022.213 and 639.021.646, and a Google Faculty Research Award. We thank the anonymous reviewers for their helpful comments.

---

\(^1\)We use the nonparametric rank-based Mann-Whitney-Wilcoxon test (Sprent and Smeeton, 2016) to check for statistically significant differences between runs.
References


Arvind Neelakantan, Jeevan Shankar, Alexandre Pas-sos, and Andrew McCallum. 2014. Efficient non-
parametric estimation of multiple embeddings per
word in vector space. In Proceedings of the 2014
Conference on Empirical Methods in Natural Lan-
guage Processing (EMNLP). Association for Compu-
tational Linguistics, Doha, Qatar, pages 1059–

Jeffrey Pennington, Richard Socher, and Christopher
Manning. 2014. Glove: Global vectors for word
representation. In Proceedings of the 2014 Con-
ference on Empirical Methods in Natural Language
Processing (EMNLP). Association for Computa-
tional Linguistics, Doha, Qatar, pages 1532–1543.

Lin Qiu, Kewei Tu, and Yong Yu. 2016. Context-
dependent sense embedding. In Proceedings of the
2016 Conference on Empirical Methods in Natu-
ral Language Processing. Association for Computa-
tional Linguistics, Austin, Texas, pages 183–191.

Multi-prototype vector-space models of word mean-
ing. In Human Language Technologies: The
2010 Annual Conference of the North American
Chapter of the Association for Computational Lin-
guistics. Association for Computational Linguis-
tics, Los Angeles, California, pages 109–117.

Peter Sprent and Nigel C Smeeton. 2016. Applied non-
parametric statistical methods. CRC Press.

György Szarvas, Róbert Busa-Fekete, and Eyke
Hüllermeier. 2013. Learning to rank lexical sub-
stitutions. In Proceedings of the 2013 Confer-
ence on Empirical Methods in Natural Language
Processing. Association for Computational Linguis-

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting
Liu, and Bing Qin. 2014. Learning sentiment-
specific word embedding for twitter sentiment clas-
sification. In Proceedings of the 52nd Annual Meet-
ing of the Association for Computational Linguis-
tics (Volume 1: Long Papers). Association for Compu-
tational Linguistics, Baltimore, Maryland, pages
1555–1565. http://www.aclweb.org/anthology/P14-
1146.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal,
and David M. Blei. 2005. Sharing clusters
among related groups: Hierarchical dirichlet pro-
cesses. In L. K. Saul, Y. Weiss, and L. Bottou,
editors, Advances in Neural Information Process-
ing Systems 17, MIT Press, pages 1385–1392.
http://papers.nips.cc/paper/2698-sharing-clusters-
among-related-groups-hierarchical-dirichlet-
processes.pdf.

Yee W. Teh, Michael I. Jordan, Matthew J. Beal,
and David M. Blei. 2006. Hierarchical dirichlet pro-
cesses. Journal of the American Statistical Asso-

Xuchen Yao and Benjamin van Durme. 2011. Non-
parametric bayesian word sense induction. In
Graph-based Methods for Natural Language Pro-
cessing. The Association for Computer Linguistics,
pages 10–14.

Will Y. Zou, Richard Socher, Daniel Cer, and
Christopher D. Manning. 2013. Bilingual word
embeddings for phrase-based machine transla-
tion. In Proceedings of the 2013 Conference on
Empirical Methods in Natural Language Pro-
cessing. Association for Computational Linguis-

447