Multilingual Word Embeddings from Sentence Representations

Benno Kruit
10576223
benno.kruit@student.uva.nl

Sara Veldhoen
10545298
sara.veldhoen@student.uva.nl

Abstract

This is an abstract

1 Introduction

Distributional semantics is a fast developing field that concerns the establishment of a high-dimensional semantic space were words get a geometrical interpretation. We investigate how data in more than one language can be used to create a single semantic space where words in more languages get positioned. Such a semantic space may have a theoretical interest on its own right, but can also be used for tasks related to translation, such as cross-lingual information retrieval and machine translation.

We come up with an approach to induce such cross-lingual embeddings based on sentence-aligned data, based on sentence correlation. We show how this is fundamentally different from approaches that focus on local correlation in an n-gram context. Without word alignments, local correlation cannot be used in cross-lingual induction of word embeddings. Global (sentence-based) correlation yields word embeddings that are less informative of the function of a word in a sentence, but can be quite useful in IR-related tasks. We use such a task to evaluate the quality of our embeddings, namely cross-lingual document classification.

This document is structured as follows. In section **??**, we discuss several approaches to multilingual distributional semantics. We introduce two existing monolingual models to obtain sentence embeddings in section 3, and extend them for the multilingual case. We explain how word embeddings can be obtained in section 4. The tasks we use for evaluation of the resulting embeddings are described in section 5, followed by the experiments we conducted and emperical results in section 6. In section 7, we set forth some considerations and ideas for future investigation of this topic. We conclude with some final remarks in section 8.

2 Related Work

Some research has focused on the induction of multilingual word embeddings, using both different techniques to obtain word representations, and different approaches to the cross-lingual aspects. The evaluation methods applied also vary a lot.

2.1 Linear Mapping

According to Mikolov et al. (Mikolov et al., 2013b), the vector space of word representations in different languages are geometrically similar, because words in languages are grounded in real world concepts. It is therefore possible to find a linear mapping between these vector spaces.

The approach is to first train word embeddings on large monolingual data for both languages separately, using the `word2vec` implementation. In the reported experiments, the so-called CBOW architecture is used, that predicts a word given its context in both directions.

Using a relatively small set of gold standard word translations, in this case obtained from Google Translate, a transformation matrix W is searched. The training objective is to minimize the distance between words that are translations of one another.

The evaluation is performed on a test set of gold-standard word translations, again from Google Translate. The word representation in the source language is transformed using W, and a ranked list of the nearest words in the target language is the output. The precision at ranks 1 and 5 is reported.

2.2 Multitask Learning

Klementiev and Titov (Klementiev et al., 2012) induce distributed representations for a pair of languages jointly. By doing so, words in both languages are represented in a single vector space.

The induction is treated as a multitask learning problem where each task corresponds to a single word. The training influences other tasks depending on the task-relatedness. The latter is derived from co-occurrence statistics in bilingual parallel data: the number of alignment links between that word and its (supposed) translations.

The word representations are induced in a neural language model architecture. The n preceding words form the context, their representations are concatenated to form a context vector. The probability of the next word occuring is predicted from this vector. The training procedure aims to find the word representations that minimize the data (log) likelihood: $L(\theta) = \sum_{t=1}^{T} \log \hat{P}_{\theta}(w_t | w_{t-n+1:t-1})$.

The method is evaluated on a real-world task: crosslingual document classification. Topic annotations are available for documents in one of the languages, and the system predicts the topics of documents in the other language. The jointly induced word representation outperform two other approaches to the problem: glossing (where every word in the document is translated separately, based on word alignments) and Machine Translation.

2.3 Joint Learning from Sentence Embeddings

Unlike the previous approaches, Hermann and Blunsum (Hermann and Blunsom, 2013) start from sentence alignments, which share the same semantics. The assumption is that some function can describe the composition of word embeddings into a sentence embedding. For the sake of their argument, the authors use a simple bag-of-words additive interpretation of composition. The word embeddings are induced jointly for both languages from these sentence-embeddings, by minimizing the distance between both sums of word embeddings. In order to make sure the weights won't be reduced to zero, similarity between unaligned sentence embeddings is penalized.

The same evaluation as in (Klementiev et al., 2012) is applied, i.e. the document classification task. Furthermore, the authors present a graphical qualitative analysis. In (Hermann, 2014), this approach is expanded by evaluating on a larger number of language pairs.

2.4 An Autoencoder Approach

Recent work that is highly relevant constructs word embeddings using a sentence autoencoder (**?**). The autoencoder predicts which words are in a sentence given the (transformed) sum of their embeddings. Building on this, the authors learn joint bilingual word embeddings by using two decoders: one to predict which words are in the original sentence, and one to predict which words are in the parallel sentence. The error signal from both decoders is propagated to the words in both languages, and is therefore distributed over the words in the parallel sentences in the same manner. Additionally, the authors ensure that the word representations of both languges are correlated by adding a correlation term to the objective function.

As with the previous model, this model assumes a bag-of-words additive interpretation of composition. The model takes no complex composition into account on either the encoder or decoder side. As with (Hermann and Blunsom, 2013), it is not based on word alignments, and is also evaluated in the same manner as (Klementiev et al., 2012).

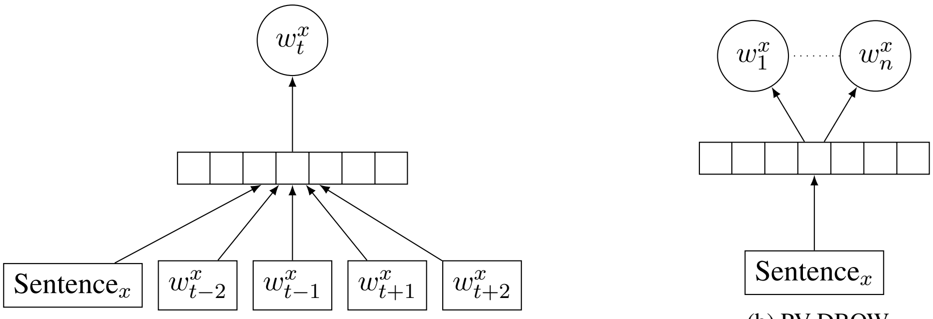
3 Sentence embeddings

Like most of the aforementioned approaches, we aim to induce multilingual word embeddings from parallel data. In order to make sure the semantic spaces for all languages are aligned, we rely solely upon the fact that sentences are aligned without using word alignments. We introduce the `paragraph2vec` from (Le and Mikolov, 2014). Next, we show how these models can be used in a multilingual context.

3.1 `paragraph2vec`

An efficient model to induce word embeddings from (monolingual) text is called `word2vec` and was introduced in (Mikolov et al., 2013a). It was extended to a version that can induce the same kind of embeddings for paragraphs: `paragraph2vec` (Le and Mikolov, 2014). A paragraph in this case can be any sequence of words, e.g. a sentence, paragraph or entire document. There are two different models to induce them, called PV-DM (dsitributed memorey) and PV-DBOW (distributed bag of words). The authors combine paragraphs obtained from both models in their experiments.

In the DM model, a *paragraph vector* is used as a part of the context of each word in the sequence



(a) PV-DM

(b) PV-DBOW

Figure 1: `paragraph2vec` models

(figure 1a). The hidden layer is formed by taking the average (or sum) of the sentence vector and word vectors of the context. The network tries to predict the index of the word that was left out of the context. This way, the paragraph vector influences the learned representations of those words in the same way that their context words do.

In the DBOW model, no word embeddings are trained. Rather, the sentence embedding is trained by trying to predict the indexes of all words that occur in the sentence (see figure 1b).

3.2 Embeddings for parallel sentences

Our first approach consisted of running `PV-DM` from (Le and Mikolov, 2014), but using the same paragraph vector when training word vectors from parallel sentences. This means the sentences embedding is used (together with the surrounding words) to predict every word that occurs in that sentence for both languages. Note that the softmax output ranges over the vocabularies of both languages. The error signal from that prediction determines the value of the sentence vector.

The second approach is using `PV-DBOW` from (Le and Mikolov, 2014) on the parallel sentences. From the embedding of a single parallel sentence representation, the network tries to predict all words that occur in the sentence in either language. Note that no word embeddings are trained, only word indexes are predicted from the sentence embedding. The error is propagated back to train the sentence embeddings. This model is depicted in figure 2. This extension is not restricted to bilingual training: in principle, sentences in any number of languages can be trained as long as they are parallel across all languages.

Both of these methods construct usable multilingual sentence embeddings. The multilingual `PV-DBOW` has the added benefit of being faster, because no word embeddings are trained.

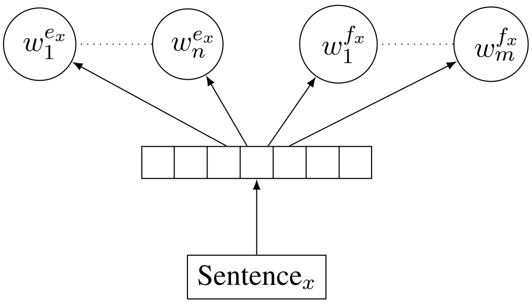


Figure 2: Bilingual PV-DBOW

4 Multilingual word embeddings

4.1 Words from PV-DM

Our first approach to obtaining joint multilingual word embeddings was to train the multilingual `PV-DM`. However, the error signal that is used to train the word representations does not lend itself for learning a joint multilingual word vector space. The model is predicting either words in one language or words in the other language, but never both. If both languages had similar embeddings, the model would make prediction errors by predicting the word index for the translation of the target as well. This way, the model the model is inadvertently trained to distinguish between the languages.

The model does learn sensible word embeddings in both languages independently. The word embeddings are located near related words in their own language, and far away from embeddings from the other language.

Additionally, if the languages used the features in the same way, subtracting a word from its literal translation could result in a 'translation vector'. This vector could then be added to words in one language to find their counterpart in the other.

However, it seems to be impossible to construct such a vector with this model. In this approach the vector dimensions therefore encode different information for both languages and also the sentences.

The second approach was to use two separate monolingual PV-DM that models independently predict words from contexts in either language. To enforce alignment between the languages, we used strong initialization of the paragraph vector. Namely, the sentence embeddings that resulted from dbow training were used and kept fixed. The word vectors were initialized to be as near as possible to the embeddings of the sentences that those words appear in. We therefore initialized every word vector as the average of all sentence embeddings that it occurs in. The idea was to then further refine the word embeddings using a smaller context in the two PV-DM models. However, the training occurs independently for both languages and the commonality of the semantic space relies solely on the sentence embeddings. The impact of this signal was not strong enough to construct a joint bilingual vector space in this way. Again, a coherent semantic space emerged for both languages individually, but was not shared between them.

4.2 Words as sentence averages

However, the word initialization discussed above is interesting in its own right. Inspired by the bag-of-words additive interpretation of composition, we flip the model upside down. Instead of assuming that a sentence representation is the average of its word embeddings, we assume a word embedding is the average of the sentences it occurs in. This way, a word embedding is as close as possible to the representations of all sentences in which it is used. Word embeddings are thus computed with the following equation:

$$[\![w]\!] = \frac{1}{freq(w, D)} \sum_{s \in D} freq(w, s)[\![s]\!]$$

These embeddings appear quite usefull for the document classification task we evaluate them on, as we will show in section 6.

In this setting, word embeddings are not trained on any information about their direct context. As with (Hermann and Blunsom, 2013) and (**?**), we assume a bag-of-words sentence representation and train word embeddings just from their sentence co-occurrence.

As described above, we were unable to convert these word embeddings from sentence co-occurrence into word embeddings from local co-occurrence using the multilingual PV-DM model. This shows that there is a fundamental difference between the two types of word embeddings. Word embeddings from sentence co-occurrence reflect the topic of the sentence they occur in, while word embeddings based on local co-occurrence also reflect information about the role of a word within a sentence. Without word alignments however, there is no way to determine parallel word contexts. Therefore, there must be a notion of word alignment in a model that aims to find multilingual word embeddings from local co-occurrence.

We imagine that the two types of word embeddings should also be evaluated differently and used for different tasks. Word embeddings from sentences are useful for information retrieval, but tasks that involve a finer-grained sense of word meaning benefit more from word embeddings based on local co-occurrence. Those tasks include Machine Translation and Semantic Parsing.

5 Evaluation

It is not trivial to measure the quality of the multilingual word embeddings. The semantic space should be reliable for each language in isolation, and consistent across languages. Even the former is not easy to assess. In (Mikolov et al., 2013a), an analogy task is introduced to this aim, which we apply to our English word embeddings as wel.

The latter is evaluated on a real-world task of cross-lingual document classification. The models we use rely on bag-of-word representations of sentences, as explained in section 4. Therefore, we do not expect a fine-grained semantic analysis of sentences and words but rather capture something like 'topicality'. It thus make sense to apply a document classification task, following the evaluation strategy of (Klementiev et al., 2012; **?**; Hermann and Blunsom, 2014).

5.1 Word analogy task

HIER MOET NOG IETS

5.2 Document classification - RCV

In (Klementiev et al., 2012) a cross-lingual document classification task is introduced. The task, that is also used in (Hermann and Blunsom, 2013), is based on Reuters corpora, which has topic-annotated documents. The evaluation data is available for English and German documents that belong to a single topic, and thus the gold standard can be represented by a one-hot vector.

A vector representation is obtained for each document in the dataset. In (Klementiev et al., 2012), the document vector is the average of the representations of its *tokens*, weighted by idf score. In (Hermann and Blunsom, 2013), the document vector is the average of the representations of its *sentences*. We use both approaches, depending on the experimental settings.

As a classifier, we use the implementation of an averaged perceptron algorithm from (Klementiev et al., 2012). It is trained to predict classes (topics) from document representations. In the cross-lingual setting, the perceptron is trained for document classification in one language, and tested on data in another resulting in a classification accuracy score. If the semantic space is coherent between languages, performance should not diverge much between monolingual and cross-lingual document classification.

The topics in the RCV evaluation sets belong to four topics: Corporate/Industrial, Economics, Government/Social, and Markets. For both languages, the documents are split into train sets with 100, 200, 500, 1000, 5000 and 100000 documents, and a test set of around 5000 documents. As a baseline, we compute chance accuracy for the majority class estimate. For both languages, the majority class was Markets, with around 46.8% of the documents.

5.3 TED document classification

The WIT TED corpus (Cettolo et al., 2012) contains short documents with transcriptions and translations of TED talks, with topic annotations. The original distribution was aimed at machine translation, but (Hermann and Blunsom, 2014) propose it for a multilingual document classification task. The major advantage of this task over the previous one, is the availability of documents in many languages. It has documents in English sentence-aligned with other languages, six of which are also in the Europarl data we use for obtaining our data: Spanish, French, German, Italian, Dutch, and Portuguese.

There are fifteen classification labels, i.e. topics, in this set. Note that contrary to the previous task, a document can have more than one topic annotation. A binary classifier is thus trained for each topic, using the same system as before. Performance is reported both as classification accuracy and F1 score. As the chance accuracy for majority class is quite high, since there are only few positive examples per class, F1 is more informative for comparing performance.

The majority class estimate is not usable as a baseline for F1 performance: as the majority of the documents are labeled negative, precision would be zero and thus F1 too (or, actually, undefined). As an alternative baseline, we compare to a stochastic classifier that predicts 'true' with probability $P = pos/total$. The expected number of True Positives is thus $P * pos = P^2 * |X|$, the expected False Positives and False Negatives are both $P * (1 - P) * |X|$. We can now compute expected F1:

$$\begin{aligned} F1 &= \frac{2 * TP}{2 * TP + FN + FP} \\ &= \frac{2 * P^2 * |X|}{2 * P^2 * |X| + 2((1 - P) * P * |X|)} \\ &= P \end{aligned}$$

Therefore, we use the ratio of positive examples as a baseline for the performance on TED data.

6 Experiments and results

We conducted several experiments using the multilingual `paragraph2vec` models described in 3. In this section, the training data and implementations we use are explained. We report emperical results for different experimental set-ups.

6.1 Data

For training the sentence and word embeddings, we use either of two subsets of the Europarl corpus. The default training set contains 500 000 lines of Europarl data. Another set has 50 000 lines that are sentence-aligned across English, German, Dutch, French, Spanish, Italian, and Portuguese. These cross-lingual sentence alignments were created by matching the English side of all pairwise aligned corpora. We use this dataset to train the multilingual dbow model in more than two languages, without making use of a pivot language.

All documents were tokenized and lowercased. No other preprocessing, such as stemming, was applied. All words (and sentences) are represented by vectors of length 256. In all experiments, words that occurred fewer than five times were excluded. The resulting vocabulary sizes in these datasets are presented in table 1.

	Europarl 50k	Europarl 500k
English	8377	24403
German	11578	47071
Dutch	10008	
French	11092	
Spanish	10865	
Italian	11503	
Portuguese	11101	

Table 1: Vocabulary size for Europarl data using a rare word cut-off of 5.

As a baseline, we use the majority class estimate for the RCV data, and the F1 baseline P explained in section 5. We also compare to the performance of vectors that results from the multitask-learning approach by (Klementiev et al., 2012) described in section 2. A distribution of their word embeddings in four language pairs (German-English, Czech-English, French-English, and Spanish-English) is available on `http://klementiev.org/data/distrib/`. The alignments used to populate the interaction matrix are obtained from the Europarl corpus. The word embeddings of the German-English part of the data are trained on the Reuters data that also make up the RCV evaluation set. Note that both the amount and nature of this training data differs from ours. Because of this, also the vocabularies differ. Their vocabulary size for English and German are 43612 and 50108, respectively. Since he I-Matrix vectors that we use are induced from RCV data, they fit better with that evaluation set. Also the length of these vectors is much smaller: 40 (instead of 256).

6.2 Implementation

HIER MOET NOG IETS

6.3 Sentence embeddings from multilingual dbow

Using our multilingual version of the paragraph2vec dbow architecture, we obtain sentence embeddings for parallel sentences: DE-EN are German and English paired. EN and DE are monolingually trained sentence embeddings trained with the original dbow model. In each case, the model is trained on Europarl data, for 10 epochs. We start with a learning rate (α) of 0.025, which is decreased with 0.002 after each epoch.

In order to evaluate the quality of the sentence embeddings, we obtain sentence representations for the (parallel) TED corpus. We use the trained model, keeping the softmax weights fixed and training the TED sentence representations for 10 iterations.

We apply the induced sentence embeddings to the TED document classification task. In this case, we take the document representation to simply be the average of its sentence embeddings. These representations are then used to train and test the two document classification tasks. The results are in the second column of table 2.

6.4 Word embeddings from sentence embeddings

As explained in section 4, we obtain word embeddings from the sentence embeddings by taking the average of all sentence embeddings the word occurs in. We will refer to this setting as ‘parawords’. Note that we use only the Europarl-trained sentences for this, not the TED sentences that we reported on above. Again, we use sentences trained on English and German monolingually, as well as paired. The results are reported in the four rightmost columns table 2.

Although the sentence representations trained on German perform somewhat better than those trained on English, the word representations induced from the latter appear much more useful for the document classification. Note that this is the case even for training/ testing on German document representations. In this case, the information from the English sentences is transferred to the German words. That means that the model might be trained monolingually on sentences in one language for which many data are available, and can be transferred to create word embeddings for other languages using a, possibly smaller, amount of parallel data.

The bilingually trained sentence representations yield better word representations than the monolingual ones in three of the four evaluation settings, which can be explained in there being more data available. However, the performance of sentence representations increases only slightly in the

Sentences		Classification [train]-[test]			
trained on:	paragraphs	EN-EN	DE-DE	DE-EN	EN-DE
EN	.340	.274	.286	.305	.284
DE	.354	.263	.190	.166	.270
DE-EN	.363	.319	.304	.264	.323

Table 2: F1 scores on TED classification task for sentence representations (paragraphs) and word representations (parawords).

bilingual case.

We can also infer from these numbers that, apparently, monolingual classification is not necessarily 'easier' than the cross-lingual case.

	Classification [train]-[test]			
	EN-EN	DE-DE	EN-DE	DE-EN
Majority	.468	.468	.468	.468
I-matrix	.817	.570	.524	.621
Paraword	.837	.694	.656	.748

Table 3: Accuracy score on RCV evaluation task with 1000 training documents, for word representations from I-matrix training and our own model trained on English-German parallel data.

In table 4, we report performance of the discussed experimental settings on monolingual (EN-EN) classification, to get an idea of how fit the different approaches are for the evaluation tasks. We report performance on the RCV task, again given 1000 training examples, TED, and the analogy task. The Google News vectors are provided in They are English embeddings of length 300 and trained on a huge corpus (Google News). We reckon their performance therefore establishes an upper limit.

	vector	RCV	TED	Analogy
Setting	length	acc	F1	acc
Baseline		.468	.118	
I-Matrix	40	.817	.149	.006
Paragraph e	256	-	.340	-
Paragraph e-d	256	-	.363	-
Paraword e	256	.836	.274	.025
Paraword e-d	256	.837	.319	.072
Google News	300	.915	.486	.613

Table 4: Monolingual (EN-EN) evaluation for various settings. RCV Accuracy: given 1000 training examples for classification.

6.5 Adding more languages

As mentioned before, the multilingual dbow model can be trained for a larger number of languages at the same time. The precondition however, is to have sentence-aligned text for all those languages at the same time. We train on such data in seven languages, and create the word embeddings for each language in the same manner as before. We run the evaluation in all combinations of train and test data (TED). The results are shown in table 5.

	de	en	nl	es	fr	it	pb
de	.368	.339	.403	.368	.282	.373	.319
en	.387	.404	.389	.321	.352	.374	.351
nl	.426	.397	.417	.393	.412	.428	.321
es	.399	.301	.428	.387	.354	.362	.355
fr	.398	.417	.545	.407	.385	.332	.406
it	.406	.405	.377	.436	.373	.4	.359
pb	.403	.333	.369	.354	.374	.405	.315

Table 5

We investigated how language relatedness could influence these results. We train the dbow system on either German, English and Dutch ('GERM') or French, Spanish and Italian ('LATIN').

We report the average over all evaluation settings with training/ testing on Germanic and Latin languages in table 6. There is no apparent correlation between the language families of dbow training and evaluation.

	Classification [train]-[test]			
	GERM-GERM	LATIN-LATIN	LATIN-GERM	GERM-LATIN
GERM	.338	.374	.383	.349
LATIN	.348	.371	.364	.347

Table 6

7 Discussion and future work

Our experiments are somewhat tentative, given the scope of this research. We discuss some considerations and directions for more thorough research.

7.1 Word senses

Many approaches to word embeddings assume that each word gets mapped to a single embedding. However, words often posses several senses: different meanings that may be more or less related. In order to improve the quality of semantic spaces, these word senses should be taken into account. Parallel data might actually help to induce word senses, since languages can actually have different words where another language uses the same word for different senses.

7.2 Issues of morpology

Of course, languages differ in what it means to be a 'word'. English has quite simple morphology, while Latin languages often use inflections and German and Dutch have a lot of compound words. Simple tokenization of text may thus not produce equally reliable vocabularies for different languages, and therefore distort performance. A possible solution is to apply stemming or other morphological decomposition. We have not looked into these possibilities, but it would sure be a good idea to take this in consideration in further research on crosslingual embeddings.

7.3 Models of composition

In our experiments, we used a bag-of-words representation of the sentences in the training of sentence embeddings. In the estimation of word embeddings, we assume a reversed additive compositional model: we take the word to be the average of all sentences it occurs in. A lot of research has focused at more pregnant models of composition, ranging from a bigram additive model (Hermann and Blunsom, 2014) to complex syntactically inspired constitutionality. Given a sentence representation, whether it is trained with `PV-DBOW` or otherwise, any model of composition could in principle be used to obtain word embeddings.

8 Conclusion

We discovered a fundamental difference between the usage of local and global correlation to establish word embeddings. Without aligning words in parallel data, there is no apparent way to introduce local correlation in cross-lingual induction of word embeddings. However, using global (sentence level) correlation, we can create embeddings that are useful for IR-related tasks. We introduced a model to train `PV-DBOW` sentence representations on parallel data in any number of languages. But we also show that we can obtain word embeddings from sentence representations that are trained on another language, as long as there is parallel data available.

References

[Cettolo et al.2012] Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.

[Hermann and Blunsom2013] Karl Moritz Hermann and Phil Blunsom. 2013. Multilingual distributed representations without word alignment.

[Hermann and Blunsom2014] Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual Models for Compositional Distributional Semantics. In *Proceedings of ACL*.

[Hermann2014] Karl Moritz Hermann. 2014. *Distributed Representations for Compositional Semantics*. Ph.D. thesis.

[Klementiev et al.2012] Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words.

[Le and Mikolov2014] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents.

[Mikolov et al.2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al.2013b] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013b. Exploiting similarities among languages for machine translation.