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

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 (Sarath Chandar et al., 2014). 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) that we extend for this purpose, and explain how we use it to obtain word embeddings.

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 (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.

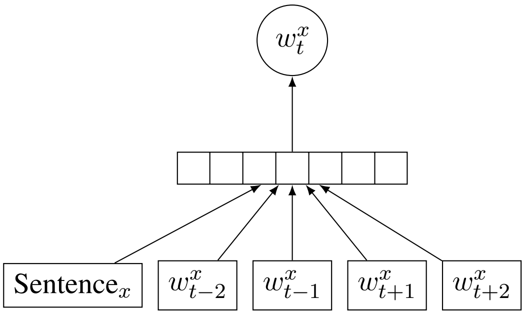
In the DBOW model, no word embeddings are trained. Rather, the sentence embedding is trained by trying to predict the indexes of all words that occur in the sentence (see figure 1b).

3.2 Embeddings for parallel sentences

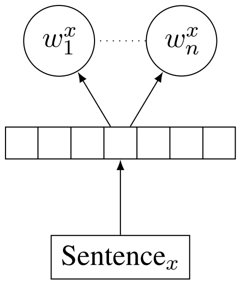
Our first approach consisted of running `PV-DM` from (Le and Mikolov, 2014), but using the same paragraph vector when training word vectors from parallel sentences. This means the sentences embedding is used (together with the surrounding words) to predict every word that occurs in that sentence for both languages. 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 parellel sentences. From the embedding of a single parallel sentence representation, the network tries to predict all words that occur in the sentence 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.

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



(a) PV-DM



(b) PV-DBOW

Figure 1: `paragraph2vec` models

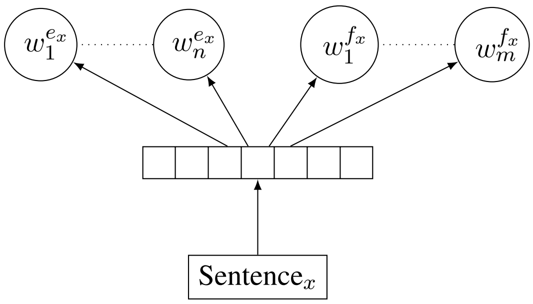


Figure 2: Bilingual PV-DBOW

4 M[ultilingual word embeddings

4.1 Words from PV-DM

Our first approach to obtaining joint multilingual word embeddings was based on using the word embeddings from 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. By predicting either words in one language or words in the other language, but never both, the model is trained to distinguish between the languages. Both languages are explicitly prevented from having similar embeddings, because that encourages the model to make prediction errors, namely predicting the word index from one language instead of the other.

The second approach was to use two classifiers and strong initialization of the paragraph vectors. The sentence embeddings that resulted from the dbow training were used and kept fixed. The word word embeddings from the previous experiment to initialize the multilingual DM setting with two separate classifiers. The idea is to further refine the word embeddings using a smaller context. However, the training occurs independently for both languages and the commonality of the semantic space relies solely on the sentence embeddings.

4.2 Words as sentence averages

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:

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

This means the word embeddings are not trained on any information about their direct context. As with (Hermann and Blunsom, 2013) and (Sarath Chandar et al., 2014), we assume a bag-of-words sentence representation and train word embeddings from their sentence co-occurrence.

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

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.

The classification labels in this set are technology, culture, science, global issues, design, business, entertainment, arts, politics, education, art, health, creativity, economics, and biology. 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 50 000 sentences of Europarl data, unless

	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.

stated otherwise. The documents 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. This means that every sentence is available in all 7 languages. All documents were tokenized and lowercased. No other preprocessing, such as stemming, was applied.

In all experiments, words that occurred fewer than five times were excluded. The resulting vocabulary sizes in this dataset are presented in table 1.

6.2 Implementation

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, multi are all 7 aligned languages, 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 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 on TED

	Classification [train]-[test]			
	EN-EN	DE-DE	EN-DE	DE-EN
Majority	.468	.468	.468	.468
I-matrix	.830	.665	.473	.643
dbow d-e	.837	.450	.405	.700

Table 3: Accuracy score on RCV evaluation task with 1000 training documents, for word representations from I-matrix training and our own models

data 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. Note that we use 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, and a version with all languages. The results are reported in the four rightmost columns table 2.

It is interesting to see that

In table 3, we compare to the multitask-learning approach by (Klementiev et al., 2012) described in section 2. A distribution of their resulting 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 the amount an nature of training data is quite different from ours, as are the vectors lengths: 40.

We explore how the resulting sentence embeddings can be used to induce word embeddings in two languages. The word embeddings are in the same space and anticipated to be aligned cross-lingually, because the sentence representations for bilingual sentences are equal.

In one setting, we define the word representation as the average of the embeddings of all sentences it occurs in. We evaluate the word embeddings that result from this.

Sentences trained on:	sentence quality	Classification [train]-[test]			
		EN-EN	DE-DE	EN-DE	DE-EN
EN	.293	.186	.134	.084	.153
DE	.305	.132	.091	.076	.132
DE-EN	.378	.194	.127	.100	.136
multi		.297	.196	.206	.226

Table 2: F1 scores on TED classification task for sentence representations and word representations.

7 Discussion and future work

A distribution of word embeddings in four language pairs (German-English, Czech-English, French-English, and Spanish-English) `http://klementiev.org/data/distrib/`

8 Conclusion

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.

[Sarath Chandar et al.2014] A P Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh M. Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. 2014. An Autoencoder Approach to Learning Bilingual Word Representations. page 10.