CTI Project: Deep Networks as a Semantic Platform for Modeling User Behavior Data Milestone 1

1plusX AG and ETH Zürich

May 16, 2017

1 Introduction

In this CTI project we intend to model user behavior on the web. In the first milestone, we take a closer look at embeddings of content and users. We start out with textual content, as it is present on almost all websites.

We investigate both monolingual and bilingual embeddings of words and documents, and propose extensions of current methods to train these embeddings. We then evaluate our own approaches on well-established tasks.

Relatively little work has been performed on training user embeddings, most probably due to the limited availability of publicly accessible datasets. We therefore employ algorithms from document modeling for this task, and attempt to design a meaningful evaluation ourselves.

2 Related Work

In the following we give a short summary of the state of the art in the research areas that are relevant to our project.

2.1 Word embeddings

While word embeddings existed before, the excitement for the area was caused by a method called word2vec [17]. It comes in two variants and either learns the embedding of a word by predicting its surrounding words (skip-gram), or predicts a target word from its context words (CBOW). Shortly after that, GloVe [18] proposed to train word embeddings directly from the global co-occurrence statistics of a corpus. While there was continued interest in the field, newer methods failed to surpass these two consistently and across tasks, so they still remain the two most widely used today. An interesting extension of word2vec is called fastText [8], which additionally uses n-gram features in their training algorithm.

2.2 Document embeddings

Document modeling was invigorated by the introduction of topic models, most notably LDA [2], which sees a document as a mixture over latent topics that define word distributions. The Replicated Softmax [7] then introduced a (simple) neural network architecture, with a layer of visible input units coupled with a layer of binary hidden units. While these models looked at the entire input as a bag-of-words, DocNADE [11] proposed to make use of the information given by the word sequence. It's extension DeepDocNADE [12] investigates the use of deeper architectures, but has to revert back to bag-of-words input as a consequence. An extension of word2vec by the name of Paragraph Vectors (and inofficially known as doc2vec) [13] takes the skip-gram and CBOW idea to the document level. Finally, the Neural Variational Document Model (NVDM) [16] applies the variational autoencoder (VAE), which has proven successful in other domains, to documents.

2.3 Bilingual word embeddings

A recent study [20] has surveyed the currently best-performing methods for creating bilingual word embeddings. They can be divided into two groups. The first requires an alignment between the words in the training data, whereas the second operates on aligned sentences, but does not require word alignment. The bivec method [15] uses the former more fine-grained alignment, and generates these alignments with the help of the Berkeley Aligner [14]. While this alignment is important for the algorithm, it is by no means easy to generate. Consequently, more training data is available for the second approach, and more methods have been designed for this input data. Both BilBOWA [4] and BiCVM [6] combine the monolingual loss from word2vec with a bilingual objective to align the word embeddings of two languages. A standard autoencoder is used by BAE-cr [3] to encode a sentence and reconstruct it in its own, as well as the other language.

2.4 User embeddings

Little work has been performed in the area of user embeddings. The dominating approach for analyzing user interaction data has for a long time been collaborative filtering. A well-known example task is the prediction of users' movie watching preferences, as in the Netflix prize challenge [1]. The dominating approach in that challenge were recommender systems that compute these predictions by a matrix factorization of the given user-rating matrix. Implicitly, users' affinities to concepts are also learned in this approach. As a more recent explicit example, the skip-gram idea has been directly applied to products and users [5].

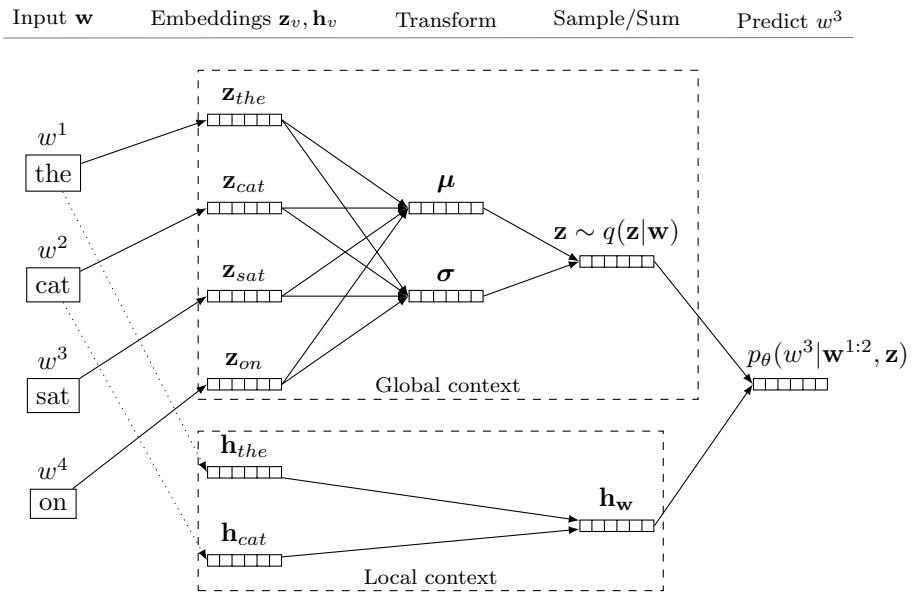


Figure 1: Illustration of next word prediction in SAVAE. The example document consists of 4 words ("the cat sat on"), and w^3 ("sat") shall be predicted next.

3 Document Models

When comparing the state-of-the-art document models, we took to the idea of doc2vec, which uses an explicit representation of the entire document coupled with a local context of a few words in a window that slides over the document. While it generates document embeddings efficiently for the training data, these embeddings have to be retrained on unseen data (e.g. at production time), however. This makes doc2vec impractical to use in a system with changing document collection (such as the web). In a joint Master's thesis between 1plusX and ETH, we thus developed a generative document model with a proper probabilistic model of documents, that could generate document embeddings at test time in a simple feedforward pass.

3.1 SAVAE

We developed the Sequence-Aware Variational Autoencoder (SAVAE) for document modeling. An illustration of the next word prediction process is given in Figure 1 and explained in the following. We first extract global word embeddings $\mathbf{z}_v$ for all words and then another set of local word embeddings $\mathbf{h}_v$ for the k previous words. The global word embeddings are then summed up and transformed through N layers of an MLP, producing $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, the parameters of our variational (normal) distribution $q(\mathbf{z}|\mathbf{w})$. Then we sample from this dis-

Model	20 Newsgroups	Reuters
Replicated Softmax	650.4	-
DocNade	223.4	487.4
DeepDocNADE	369.5	535.1
NVDM	379.6	552.3
SAVAE	**143.8**	**377.51**

Table 1: Perplexity on test set.

tribution to obtain a document representation $\mathbf{z}$ which is concatenated with the local embedding $\mathbf{h_w}$ and finally used to predict the next word w^3.

3.2 Experiments

We evaluated the various document models on several established tasks. In the following, we show some interesting results that we picked. The relative performance of the algorithms is stable between different tasks and datasets.

Perplexity. With perplexity, the probabilistic model of a method can be evaluated. Intuitively, perplexity quantifies how surprised a model is, given a set of previously unseen test documents. The results are given in Table 1 for two datasets, 20 Newsgroups and Reuters RCV1-v2. SAVAE outperforms the other methods that either ignore the word sequence (Replicated Softmax, DeepDocNADE and NVDM), or simply have less model capacity (DocNADE). A comparison with doc2vec cannot be made, as it does not have a probabilistic model to calculate the perplexity.

Document retrieval. In the document retrieval task, the model is presented with an unseen test document and has to output the closest documents in the training set. The goal is to first return documents in the training set with the same label as the test document. The 20 Newsgroups corpus consists of messages from Usenet discussion groups. They include metadata such as headers, footers and quotes that contain information about the discussion participants that helps in assigning documents to the correct label. We therefore strip this metadata and plot the precision-recall curve in Figure 2.

Nearest neighbors. As a convenient by-product of training document representations, our document models also learn word embeddings. For a qualitative inspection of the results, we plot a selection in Table 2. The closest word embeddings look very similar for the first two terms *weapons* and *medical*, but for *define* SAVAE discovers synonyms, while the two models operating on bag-of-words simply retrieve words that often co-occur in the same document. We believe that this is the effect of the local context that SAVAE takes into account.

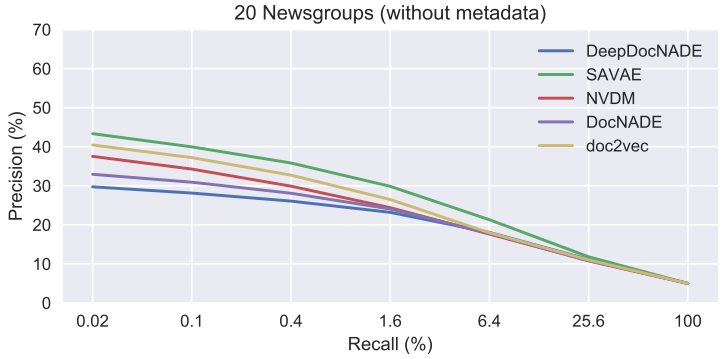


Figure 2: Document retrieval evaluation on 20 Newsgroups dataset. Metadata (headers, footers, quotes) has been removed.

SAVAE			**DeepDocNADE**			**NVDM**		
weapons	**medical**	**define**	**weapons**	**medical**	**define**	**weapons**	**medical**	**define**
weapon	health	draw	weapon	treatment	defined	guns	medicine	defined
firearms	disease	realize	guns	medicine	int	weapon	disease	null
arms	medicine	assume	firearms	health	function	batf	health	int
guns	patients	count	military	patients	value	firearms	patients	morality
crime	treatment	notice	amendment	disease	apply	militia	treatment	constitution

Table 2: Learned word embeddings for SAVAE, DeepDocNADE and NVDM.

4 Bilingual Embeddings

In BAE-cr [3], the autoencoder framework is used to train bilingual embeddings with a neat adaptation: The encoded latent representation of a sentence is used to reconstruct the sentence in the same language, as well as the one in the other language. This forces the encoder networks in the two languages to produce similar latent representations for parallel sentences. Since the generative document models can generate a latent representation of their input as well as generate a word distribution from a latent code, they fit the autoencoder framework and can be employed for this idea. Since we view sentences as documents in the bilingual context, the length of the documents is vastly reduced for the document models, compared to their usual application.

4.1 Bilingual NVDM

For the bilingual NVDM (BiNVDM), we adapt the monolingual NVDM to a bilingual model by setting up one NVDM for both languages l_1 and l_2. The objective of the monolingual NVDM is given by the variational lower bound on

the log-likelihood

$$\log p_\theta(\mathbf{w}|l) \geq \mathbf{E}_{q(\mathbf{z}|\mathbf{w})}\left[\sum_{t=1}^{l} \log p_\theta(w^t|\mathbf{z})\right] - D_{\mathrm{KL}}[q(\mathbf{z}|\mathbf{w}) \parallel p(\mathbf{z})], \qquad (1)$$

where $\mathbf{w}$ is our document, l is the length of that document, p_θ is our generative distribution and $p(\mathbf{z})$ is the prior on the latent variable $\mathbf{z}$. The first term on the right-hand side of Equation 1 is called the reconstruction or decoder loss, while the second term is the encoder loss. We now extend this objective by adding a cross-lingual term to the reconstruction loss, i.e. for reconstructing a sentence $\mathbf{w}$ in the first language from the latent representation $\mathbf{z}$ of the parallel sentence in the second language. The individual encoder losses of the NVDMs stay the same, but we add a Euclidean distance term to the overall encoder loss, to force the latent representations in the two languages to align.

As a small remark, we note that our model is not restricted to the bilingual case, and can be easily extended to more than two languages.

4.2 Experiments

Previous work has focused on creating bilingual word embeddings, and has created representations for larger pieces of text (sentences, documents) mostly by averaging the constituent words. This is reflected in the previously used evaluation tasks, which we nevertheless use for comparison here. We look at two monolingual tasks that perform intrinsic evaluation of the embeddings, and a bilingual classification task. For the monolingual tasks, we also compare with state-of-the-art monolingual word embedding methods. All of the methods were trained on the Europarl corpus [10]. The corpus contains the minutes of the European parliament and translations into all the languages spoken in the European Union. We apply standard preprocessing (tokenization and lowercasing) with the help of the tools distributed together with the corpus.

Training details. The word analogy and QVEC experiments have been performed with the default parameters of word2vec, which train 200-dimensional embeddings. The CLDC experiments have been run with the parameters reported in the paper and a dimensionality of 40 (except for BiCVM, which uses 128 dimensional embeddings in the paper).

Word analogy. The English word analogy task was popularized in the word2vec paper [17] and asks questions of the form "*man* is to *king* as *woman* is to?" The task has 10675 syntactic (noun, verb and adjective forms) and 8869 semantic questions (countries with capitals, currencies, cities and family relations). The numbers as well as some relations differ only slightly for a German and an Italian version, with one exception: the German semantic task only has 540 relations. The German task additionally has 255 questions with opposites. We did not find a French version of the task. The results for the syntactic questions are

Language	Method	Accuracy
en	word2vec (skip-gram)	30.4%
en	fastText (skip-gram)	54.5%
en	GloVe	19.4%
en (de-en)	bivec	26.4%
en (de-en)	BiCVM	0.2%
en (de-en)	BAE-cr	1.0%
en (de-en)	BilBOWA	15.9%
en (de-en)	BiNVDM	23.8%
de	word2vec (skip-gram)	25.9%
de (de-en)	bivec	27.0%
de (de-en)	BiCVM	0.4%
de (de-en)	BAE-cr	0.9%
de (de-en)	BilBOWA	19.1%
de (de-en)	BiNVDM	19.9%

Table 3: Results of the syntactic word analogy task for models trained on the Europarl corpus.

shown in Table 3. We observe that word2vec performs better than GloVe, presumably because of the use of the exact sequence over the co-occurrence counts used in GloVe. fastText performs even better, which can be explained by the incorporation of n-grams into their model. All of the three monolingual methods compare similarly on the semantic questions (cf. Table 7). As for the bilingual algorithms, bivec performs worse than word2vec for English, but better for German. The algorithms trained on sentence-parallel data perform worse than bivec, which is to be expected, as the individual word relationships (especially for rarer words) are not as easy to learn from the aggregation that the sentence essentially is. BiNVDM performs the best of these methods, followed by BilBOWA. We could not get BiCVM and BAE-cr to work after a number of trials, and their numbers are abysmal correspondingly. The complete results are given in appendix A.

QVEC. The QVEC score is supposed to correlate strongly with downstream NLP tasks, according to its authors [19]. It tests how well the word vectors correlate with manually crafted linguistic features. The results can be found in Table 4. We see that both bivec and BilBOWA perform on almost the same level as monolingual embeddings. BiNVDM performs very badly in this evaluation, which we did not find an explanation for. A complete set of results is reported in appendix B.

CLDC. The cross-lingual document classification (CLDC) task is a standardized comparison developed by Klementiev et. al [9]. It uses the Reuters RCV2

Language	Method	QVEC	QVEC_CCA
en	word2vec	21.519	0.395331
en	fastText	22.3843	0.406191
en	GloVe	22.624	0.35837
en (de-en)	bivec	19.2968	0.389775
en (de-en)	BilBOWA	20.2689	0.335055
en (de-en)	BiNVDM	6.40112	0.239493

Table 4: QVEC score on models trained with Europarl.

Method	en → de	de → en
bivec	0.8601 (0.876)	0.7598 (0.778)
BiCVM	0.3147 (0.864)	0.3072 (0.747)
BAE	0.7483 (0.918)	0.6467 (0.742)
BilBOWA	0.8319 (0.865)	0.7114 (0.750)
BiNVDM	0.8335	0.7098

Table 5: Classification accuracy on CLDC from reproduction experiments. Numbers in brackets are as reported in the papers.

dataset as the multilingual corpus. It trains a classifier (MLP) in one language and then evaluates it on a different language. Classification is done only over the four top-level categories of the Reuters topics: CCAT (Corporate/Industrial), ECAT (Economics), GCAT (Government/Social), and MCAT (Markets). The results can be found in Table 5 both for training on English texts to then classify German articles and vice versa. While bivec and BilBOWA are within reasonable distance of the numbers reported in their respective papers, we were not able to reproduce the results of BiCVM and BAE. Overall, bivec performs the best, but is the most expensive to train (requires word-aligned training data). BilBOWA and BiNVDM perform about equally on this task.

Nearest neighbors. We also perform a qualitative inspection of the produced word bilingual word embeddings by giving the ten nearest neighbors of a word in the other language, measured by cosine similarity of their embeddings. The results are in Table 6. The first three columns show the nearest German neighbors to three English words, and the last column shows the opposite direction. The words were selected according to the expectation that they would be central in the Europarl corpus, the minutes of the European parliament.

en → de			de → en
president	**european**	**commission**	**präsident**
präsident	europäischen	kommission	president
herr	europäische	kommissionsvorschlag	mr
geehrter	union	mitteilung	es
ratspräsident	europäischer	kommissionsvorschlags	commissioner
kommissar	europäisches	vorlegt	office
herrn	europäischem	kommissionspräsidenten	madam
verehrter	aufbauwerk	kommissionsvorschläge	prodi
präsidentin	aufbauwerks	prüft	bolkestein
kommissionspräsident	unionsbürger	teilt	ladies
geehrte	europäisch	vorschlägt	gentlemen

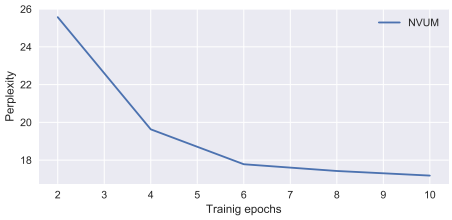
Table 6: Nearest neighbors in the other language.

5 User Embeddings

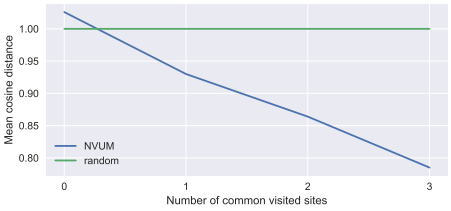
In document modeling, a document is seen as a set or sequence of words that describe its semantics. In LDA, for example, the assumption is that a small number of topics is present in any given document, and the word distributions of these topics govern the words that can appear in the document. Analogously, we can view a user as a set of website visits. The topics correspond to the user's interests, and these interests determine which sites a user visits. Since this analogy seems quite fitting, we employ methods from document modeling to create user embeddings in the following.

5.1 NVUM

With the above analogy, a straightforward adaptation of NVDM results in the Neural Variational User Model (NVUM), where documents are users and words are website visits. During training, we observed that the input distribution of website visits looks quite different to that of words in documents. Several of these factors can be attributed to the limited possibilities of collecting the data, as we only get to see a subset of a user's visited websites. This leads to a much larger vocabulary (factor 100, even after removing seldomly visited sites), much sparser data (down to two distinct sites visited per user), and conversely much higher counts for a single visited site. This leads to sharply peaked distributions that are harder to learn, at least given the same dimensionality of the latent representation. We combat these specialities by normalizing a user's visit distribution, so absolute counts don't impact the learning process. An interesting question is whether preprocessing from natural language processing (NLP), such as stopword removal, stemming and downsampling of frequent words, can be applied to website visits as well.



(a) Test perplexity.



(b) Proximity of test users.

5.2 Experiments

Since we could not find established comparison tasks, we rely on the proven measure of perplexity from NLP, and design a clustering metric we call proximity.

Perplexity. As in the perplexity evaluation for documents, we withhold some users from the training process, and then predict their distributions at test time. While the absolute numbers are hard to interpret, we believe that this metric is suited for comparison of different document models (cf. Section 6). We plot the perplexity as a function of the training epochs in Figure 3a.

Proximity. We designed a clustering metric called proximity, that evaluates the mean cosine distance of a test user's embedding vector to those of training users. We first cluster the training users with respect to the test user by the number of common site visits. Subsequently, we compute the mean cosine distance between the user's embedding and each of the users in such a cluster. Qualitatively, we expect the representations for users with less visits in common to have a larger distance than those with more common visits. In Figure 3b, we can see that this is indeed the case. In absolute terms, we expect a random assignment of user representations to produce orthogonal embeddings, with a cosine distance of 1. The more user representations are aligned, the closer the cosine distance gets to 0. We observe that the cosine distance decreases linearly for NVUM in Figure 3b.

6 Stock Take and Future Directions

After milestone 1, we have a much better understanding of how data from different domains can be co-embedded, and how methods to do so can be implemented and run on larger datasets. As already mentioned in the previous sections, there are a couple of areas where further investigation could provide more evidence on how universally applicable these methods are.

Comparison with other document models. The document models from Section 3 are directly applicable to user embeddings. For bilingual embeddings,

the situation is a bit trickier. DeepDocNADE constructs several prediction problems in its training process by splitting a document into two, and then predicting the words in the second part given those in the first. This requires word-aligned input data to correctly perform this split. The algorithm can then predict the missing words in language l_2, given a partial sentence from l_1. The word alignments could be generated by the Berkeley Aligner, for example. It is still unclear to us, how noisy the alignments of this method are (although it seems to be the latest method developed). The same story is true for SAVAE, since it does a next word prediction in its training process, which again requires word-aligned bilingual training data.

Dictionary induction evaluation. An interesting task to validate the alignment quality of the word embeddings in two languages could be that of dictionary induction. The goal in this task is to create a dictionary by selecting the word in language l_2 with the highest cosine similarity to a given word in l_1, similar to the anecdotal analysis in Section 4.2. There are several datasets with ground truth available for a quantitative evaluation.

Bilingual document embedding evaluation. So far, we are solely evaluation the bilingual word embeddings. The CLDC task could be adapted to take the document representation as input (instead of an average of word embeddings of the constituent words of the document). However, we are not too happy with this task anyway, as a classification into four classes, of which two are very similar (Economics and Markets), seems to have limited validity. As to the best of our knowledge there exist no other tasks for an evaluation of multilingual document embeddings, a different approach could be to design our own task.

Website content. Until now we are looking at a user's visited websites merely as abstract symbols of the vocabulary, without making use of any clues in the URL or, even more importantly, in the website's content. There are two different models that could be explored. The first sees the user as one big document, the concatenation of the content of all the user's visited websites. The second sees a user more as a corpus, a collection of documents (the visited websites). To our knowledge, there has not been any work on this topic. Standard clustering algorithms would not apply, for example, since in the web setting a document can appear in a potentially very large number of collections.

In our next milestone, we want to interpret the dimensions of the latent user representations. The interpretability of embeddings is currently an ongoing research topic in the machine learning community, so we expect this to be an interesting, yet challenging task.

7 Conclusion

We have analyzed and compared several state-of-the-art methods for document models, bilingual word embeddings and user embeddings. We have designed our own extensions of these algorithms for document modeling (SAVAE), bilingual embeddings of documents and words (BiNVDM), and user embeddings (NVUM) that promise to be well-suited for our use cases. The models have shown decent performance on several evaluation tasks, but more investigation is needed to provide definitive evidence for their usefulness. We propose to continue with the project along the directions outlined in Section 6.

References

[1] James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA, 2007.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[3] Sarath Chandar, Stanislas Lauly, Hugo Larochelle, Mitesh Khapra, Balaraman Ravindran, Vikas C Raykar, and Amrita Saha. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, pages 1853–1861, 2014.

[4] Stephan Gouws, Yoshua Bengio, and Greg Corrado. BilBOWA: Fast Bilingual Distributed Representations without Word Alignments. *Proceedings of The 32nd International Conference on Machine Learning*, pages 748–756, 2015.

[5] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1809–1818. ACM, 2015.

[6] Karl Moritz Hermann and Phil Blunsom. Multilingual models for compositional distributional semantics. In *Proceedings of ACL*, jun 2014.

[7] Geoffrey E Hinton and Ruslan R Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, pages 1607–1614, 2009.

[8] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

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

[10] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

[11] Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, pages 2708–2716, 2012.

[12] Stanislas Lauly, Yin Zheng, Alexandre Allauzen, and Hugo Larochelle. Document neural autoregressive distribution estimation. *arXiv preprint arXiv:1603.05962*, 2016.

[13] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196, 2014.

[14] Percy Liang, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 104–111. Association for Computational Linguistics, 2006.

[15] Thang Luong, Hieu Pham, and Christopher D Manning. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, 2015.

[16] Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In *Proc. ICML*, 2016.

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

[18] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.

[19] Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. Evaluation of word vector representations by subspace alignment. 2015.

[20] Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. Cross-lingual models of word embeddings: An empirical comparison. In *Proc. of ACL*, 2016.

A Word Analogy

The full results on the word analogy task are given in Table 7. In case that one of the four words of an analogy task is not present in the vocabulary of the embedding, that question is skipped. This is why we see so little questions

being evaluated in the semantic task for English and Italian, and basically none for German.

B QVEC

The full results for the QVEC task are given in Table 8

Language	Method	Corpus	Results
en	word2vec (skip-gram)	Europarl	syntactic: 2419/7950 = 30.4% semantic: 85/210 = 40.5%
en	fastText (skip-gram)	Europarl	syntactic: 4336/7950 = 54.5% semantic: 83/210 = 39.5%
en	GloVe	Europarl	syntactic: 1541/7950 = 19.4% semantic: 84/210 = 40.0%
en	GloVe (pretrained)	6B word corpus (Wiki + Giganet)	syntactic: 5582/9076 = 61.5% semantic: 433/506 = 85.6%
en (de-en)	bivec	Europarl	syntactic: 2086/7888 = 26.4% semantic: 71/182 = 39.0%
en (de-en)	BiCVM	Europarl	syntactic: 12/7888 = 0.2% semantic: 0/182 = 0.0%
en (de-en)	BAE-cr	Europarl	syntactic: 81/7888 = 1.0% semantic: 2/182 = 1.1%
en (de-en)	BilBOWA	Europarl	syntactic: 1251/7888 = 15.9% semantic: 25/182 = 13.7%
en (de-en)	BiNVDM	Europarl	syntactic: 1879/7888 = 23.8% semantic: 42/182 = 23.1 %
de	word2vec	Wikipedia (April 2016)	syntactic: 3433/9308 = 36.9% semantic: 346/540 = 64.1% opposites: 72/255 = 28.2%
de	word2vec	Europarl	syntactic: 1725/6666 = 25.9% semantic: 0/1 = 0.0% opposites: 18/123= 14.6%
de (de-en)	bivec	Europarl	syntactic: 1795/6650 = 27.0% semantic: 1/1 = 100.0% opposites: 28/123 = 22.8%
de (de-en)	BiCVM	Europarl	syntactic: 27/6650 = 0.4% semantic: 0/1 = 0.0% opposites: 0/123 = 0.0%
de (de-en)	BAE-cr	Europarl	syntactic: 62/6650 = 0.9% semantic: 0/1 = 0.0% opposites: 2/123 = 1.6%
de (de-en)	BilBOWA	Europarl	syntactic: 1268/6650 = 19.1% semantic: 0/1 = 0.0% opposites: 2/123 = 1.6%
de (de-en)	BiNVDM	Europarl	syntactic: 1321/6650 = 19.9% semantic: 0/1 = 0.0% opposites: 8/123 = 6.5%
it	word2vec	Europarl	syntactic: 1198/5385 = 22.2% semantic: 41/210 = 19.5%

Table 7: Full results of the word analogy task.

Language	Method	Corpus	QVEC	QVEC_CCA
en	word2vec	Europarl	21.519	0.395331
en	fastText	Europarl	22.3843	0.406191
en	GloVe	Europarl	22.624	0.35837
en	GloVe	6B words	24.7799	0.427152
en (de-en)	bivec	Europarl	19.2968	0.389775
en (de-en)	BiCVM	Europarl	8.60193	0.243564
en (de-en)	BAE	Europarl	3.36974	0.148676
en (de-en)	BilBOWA	Europarl	20.2689	0.335055
en (de-en)	BiNVDM	Europarl	6.40112	0.239493
it	word2vec	Europarl	20.1172	0.349258

Table 8: Full results of the QVEC evaluation.