Concatenated Power Mean Word Embeddings as Universal Cross-Lingual Sentence Representations

Andreas Rücklé[†], **Steffen Eger**[†], **Maxime Peyrard**[†‡], **Iryna Gurevych**[†‡]

[†]Ubiquitous Knowledge Processing Lab (UKP)
[‡]Research Training Group AIPHES
Department of Computer Science, Technische Universität Darmstadt
[†] www.ukp.tu-darmstadt.de
[‡] www.aiphes.tu-darmstadt.de

Abstract

Average word embeddings are a common baseline for more sophisticated sentence embedding techniques. However, they typically fall short of the performances of more complex models such as InferSent. Here, we generalize the concept of average word embeddings to *power mean word embeddings*. We show that the concatenation of different types of power mean word embeddings considerably closes the gap to state-of-the-art methods *monolingually* and substantially outperforms these more complex techniques *cross-lingually*. In addition, our proposed method outperforms different recently proposed baselines such as SIF and Sent2Vec by a solid margin, thus constituting a much harder-to-beat monolingual baseline. Our data and code are publicly available.[1]

1 Introduction

Sentence embeddings are dense vectors that summarize different properties of a sentence (e.g. its meaning), thereby extending the very popular concept of word embeddings (Mikolov et al., 2013a; Pennington et al., 2014) to the sentence level.

Universal sentence embeddings have recently gained considerable attention due to their wide range of possible applications in downstream tasks. In contrast to task-specific representations, such as the ones trained specifically for tasks like textual entailment or sentiment, such sentence embeddings are trained in a task-agnostic manner on large datasets. As a consequence, they often perform better when little labeled data is available (Subramanian et al., 2018).

To a certain degree, the history of sentence embeddings parallels that of word embeddings, but on a faster scale: early word embeddings models were complex and often took months to train (Bengio et al., 2003; Collobert and Weston, 2008; Turian et al., 2010) before Mikolov et al. (2013a) presented a much simpler method that could train substantially faster and therefore on much more data, leading to significantly better results. Likewise, sentence embeddings originated from the rather resource-intensive 'Skip-thought' encoder-decoder model of Kiros et al. (2015), before successively less demanding models (Hill et al., 2016; Kenter et al., 2016; Arora et al., 2017) were proposed that are much faster at train and/or test time.

The most popular state-of-the-art approach is the so-called InferSent model (Conneau et al., 2017), which learns sentence embeddings with a rather simple architecture in single day (on a GPU), but on very high quality data, namely, Natural Language Inference data (Bowman et al., 2015). Following previous work (e.g. Kiros et al. 2015), InferSent has also set the standards in measuring the usefulness of sentence embeddings by requiring the embeddings to be "universal" in the sense that they must yield stable and high-performing results on a wide variety of so-called "transfer tasks".

We follow both of these trends and posit that sentence embeddings should be simple, on the one hand, and universal, on the other. Importantly, we extend universality to the cross-lingual case: universal sentence embeddings should perform well across multiple tasks and across natural languages.

The arguably simplest sentence embedding technique is to average individual word embeddings. This so-called mean word embedding is the starting point of our extensions.

First, we observe that average word embeddings have partly been treated unfairly in previous work such as Conneau et al. (2017) because the newly proposed methods yield sentence embeddings of rather large size (e.g., $d = 4096$) while they have been compared to much smaller average word em-

拿高维的句子编码和低维的词平均比较

[1] https://github.com/UKPLab/arxiv2018-xling-sentence-embeddings

beddings (e.g., $d = 300$). Increasing the size of individual—and thus average—word embeddings is likely to improve the quality of average word embeddings, but with an inherent limitation: there is (practically) only a finite number of words in natural languages, so that the additional dimensions will not be used to store additional information, beyond a certain threshold. To remedy, (i) we instead propose to concatenate diverse word embeddings that store *different* kinds of information, such as syntactic, semantic or sentiment information; concatenation is a simple but effective technique in different setups (Zhang et al., 2016).

Secondly, and more importantly, (ii) we posit that 'mean' has been defined too narrowly by the corresponding NLP community. Standard mean word embeddings stack the word vectors of a sentence in a matrix and compute per-dimension *arithmetic* means on this matrix. We perceive this mean as a summary of all the entries in a dimension. In this work, we instead focus on *power means* (Hardy et al., 1952) which naturally generalize the arithmetic mean.

Finally, (iii) we combine concatenation of word embeddings with different power means and show that our sentence embeddings satisfy our requirement of universality: they substantially outperform different other strong baselines across a number of tasks monolingually, and substantially outperform other approaches cross-lingually.

2 Related Work

Monolingual word embeddings are typically learned to predict context words in fixed windows (Mikolov et al., 2013a; Pennington et al., 2014). Extensions predict contexts given by dependency trees (Levy and Goldberg, 2014) or combinations of windows and dependency context (Komninos and Manandhar, 2016), leading to more syntactically oriented word embeddings. Fasttext (Bojanowski et al., 2017) represents words as the sum of their n-gram representations trained with a skip-gram model. Attract-repel (Mrkšić et al., 2017) uses synonymy and antonymy constraints from lexical resources to fine tune word embeddings with linguistic information. Vulić et al. (2017) morph-fit word embeddings using language-specific rules so that derivational antonyms ("expensive" vs. "inexpensive") move far away in vector space.

Cross-lingual word embeddings originate from the idea that not only monolingually but also cross-lingually similar words should be close in vector space. Common practice is to learn a mapping between two monolingual word embedding spaces (Faruqui and Dyer, 2014; Artetxe et al., 2016). Other approaches predict mono- and bilingual context using word alignment information as an extension to the standard skip-gram model (Luong et al., 2015) or inject cross-lingual synonymy and antonymy constraints similar as in the monolingual setting (Mrkšić et al., 2017). As with monolingual embeddings, there exists a veritable zoo of different approaches, but they have been reported to nonetheless often perform similarly in applications (Upadhyay et al., 2016).

In this work, we train one of the simplest approaches: BIVCD (Vulić and Moens, 2015). This creates bilingual word embeddings from aligned bilingual documents by concatenating parallel document pairs and shuffling the words in them before running a standard word embedding technique.

Monolingual sentence embeddings usually built on top of existing word embeddings, and different approaches focus on computing sentence embeddings by composition of word embeddings. Wieting et al. (2015) learned paraphrastic sentence embeddings by fine-tuning skip-gram word vectors while using additive composition to obtain representations for short phrases. SIF (Arora et al., 2017) computes sentence embeddings by taking weighted averages of word embeddings and then modifying them via SVD. Sent2vec (Pagliardini et al., 2017) learns n-gram embeddings and averages them. Siamese-CBOW (Kenter et al., 2016) trains word embeddings that, when averaged, should yield good representations of sentences. However, even non-optimized average word embeddings can encode valuable information about the sentence, such as its length and word content (Adi et al., 2017).

Other approaches consider sentences as additional tokens whose embeddings are learned jointly with words (Le and Mikolov, 2014), use auto-encoders (Hill et al., 2016), or mimick the skip-gram model (Mikolov et al., 2013a) by predicting surrounding sentences (Kiros et al., 2015).

Recently, InferSent (Conneau et al., 2017) achieved state-of-the-art results across a wide range of different transfer tasks. Their model uses bidirectional LSTMs and was trained on the SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2017) corpora. This is novel in that previous work,

如果词向量维度增加，能提升句子编码性能，但是词向量维度应该和词汇表大小一致

句法信息怎么得到？

确实，平均就是将一个矩阵变成一个向量，应该有其它更deep的方法

which likewise used LSTMs to learn sentence embeddings but trained on other tasks (i.e. identifying paraphrase pairs), usually did not achieve significant improvements compared to simple word averaging models (Wieting et al., 2016).

非SNLI任务上用LSTM表示不一定有效

Cross-lingual sentence embeddings have received comparatively less attention. Hermann and Blunsom (2014) learn cross-lingual word embeddings and infer document-level representations with simple composition of unigrams or bigrams, finding that added word embeddings perform on par with the more complex bigram model. Several authors proposed to extend ParagraphVec (Le and Mikolov, 2014) to the cross-lingual case: Pham et al. (2015) add a bilingual constraint to learn cross-lingual representations using aligned sentences; Mogadala and Rettinger (2016) add a general cross-lingual regularization term to ParagraphVec; Zhou et al. (2016) train task-specific representations for sentiment analysis based on ParagraphVec by minimizing the distance between paragraph embeddings of translations. Finally, Chandar et al. (2013) train a cross-lingual auto-encoder to learn representations that allow reconstructing sentences and documents in different languages, and Schwenk and Douze (2017) use representations learned by an NMT model for translation retrieval.

To our best knowledge, all of these cross-lingual works evaluate on few individual datasets, and none focuses on *universal* cross-lingual sentence embeddings that perform well across a wide range of different tasks.

3 Concatenated Power Mean Embeddings

Power means Our core idea is generalizing average word embeddings, which summarize a sequence of embeddings $\mathbf{w}_1, ..., \mathbf{w}_n \in \mathbb{R}^d$ by component-wise arithmetic averages:

$$\forall i = 1, \ldots, d: \quad \frac{w_{1i} + \cdots + w_{ni}}{n}$$

这个时间序列是什么

This operation summarizes the 'time-series' $(w_{1i}, \ldots, w_{ni})$ of variable length n by their arithmetic mean. Of course, then, we might also compute other statistics on these time-series such as standard deviation, skewness (and further moments), Fourier transformations, etc., in order to capture different information from the sequence.

For simplicity and to focus on only one type of extension, we consider in this work so-called *power means* (Hardy et al., 1952), defined as:

$$\left(\frac{x_1^p + \cdots + x_n^p}{n}\right)^{1/p}; \quad p \in \mathbb{R} \cup \{\pm\infty\}$$

for a sequence of numbers $(x_1, \ldots, x_n)$. This generalized form retrieves many well-known means such as the arithmetic mean ($p = 1$), the geometric mean ($p = 0$), and the harmonic mean ($p = -1$). In the extreme cases, when $p = \pm\infty$, the power mean specializes to the minimum ($p = -\infty$) and maximum ($p = +\infty$) of the sequence.

Concatenation For vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$, concisely written as a matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_n] \in \mathbb{R}^{n \times d}$, we let $H_p(\mathbf{W})$ stand for the vector in $\mathbb{R}^d$ whose d components are the power means of the sequences $(w_{1i}, \ldots, w_{ni})$, for all $i = 1, \ldots, d$.

Given a sentence $s = w_1 \cdots w_n$ we first look up the embeddings $\mathbf{W}^{(i)} = [\mathbf{w}_1^{(i)}, \ldots, \mathbf{w}_n^{(i)}] \in \mathbb{R}^{n \times d_i}$ of its words from some embedding space $\mathbb{E}^i$. To get summary statistics of the sentence, we then compute K power means of s and concatenate them:

w2v · glove属于不同的向量空间

$$\mathbf{s}^{(i)} = H_{p_1}(\mathbf{W}^{(i)}) \oplus \cdots \oplus H_{p_K}(\mathbf{W}^{(i)})$$

where $\oplus$ stands for concatenation and $p_1, \ldots, p_K$ are K different power mean values. Our resulting sentence representation, denoted as $\mathbf{s}^{(i)} = \mathbf{s}^{(i)}(p_1, \ldots, p_k)$, lies in $\mathbb{R}^{d_i \cdot K}$.

To get further representational power from different word embeddings, we concatenate different power mean sentence representations $\mathbf{s}^{(i)}(p_1, \ldots, p_k)$ obtained from different embedding spaces $\mathbb{E}^i$:

$$\bigoplus_i \mathbf{s}^{(i)} \tag{1}$$

The dimensionality of this representation is $K \sum_i d_i$. When all embedding spaces have the same dimensionality d, this becomes $K \cdot L \cdot d$, where L is the number of spaces considered.

4 Monolingual Experiments

4.1 Experimental Setup

Tasks We replicate the setup of Conneau et al. (2017) and evaluate on the six transfer tasks listed in their table 1. Since their selection of tasks is slightly biased towards sentiment analysis, we add three further tasks: AM, an argumentation mining task based on Stab and Gurevych (2017) where sentences are classified into the categories major

claim, claim, premise, and non-argumentative; AC, a further argumentation mining task with very few data points based on Peldszus and Stede (2015) in which the goal is to classify sentences as to whether they contain a claim or not; and CLS, a task based on Prettenhofer and Stein (2010) to identify *individual sentences* as being part of a positive or negative book review.[2]

We summarize the different tasks in Table 1.

Word embeddings We use four diverse, potentially complementary types of word embeddings as basis for our sentence representation techniques: GloVe embeddings (GV) (Pennington et al., 2014) trained on Common Crawl; Word2Vec (Mikolov et al., 2013b) trained on GoogleNews (GN); Attract-Repel (AR) (Mrkšić et al., 2017) and MorphSpecialized (MS) (Vulić et al., 2017).

We use pre-trained word embeddings except for Attract-Repel where we use the retrofitting code from the authors to tune the embeddings of Komninos and Manandhar (2016).

Evaluated approaches For each type of word embedding, we evaluate the standard average ($p = 1$) as sentence embedding as well as different power mean concatenations. We also evaluate concatenations of embeddings $\mathbf{s}^{(i)}(1, \pm\infty)$, where i ranges over the word embeddings mentioned above.[3] We motivate this choice of power means later in our analysis.

We compare against the following four approaches: SIF (Arora et al., 2017), applied to GloVe vectors; average Siamese-CBOW embeddings (Kenter et al., 2016) based on the Toronto Book Corpus; Sent2Vec (Pagliardini et al., 2017), and InferSent.

While SIF ($d = 300$), average Siamese-CBOW ($d = 300$), and Sent2Vec ($d = 700$) embeddings are relatively low-dimensional, InferSent embeddings are high-dimensional ($d = 4096$). In all our experiments the maximum dimensionality of our concatenated power mean sentence embeddings does not exceed $d = 4 \cdot 3 \cdot 300 = 3600$.

Evaluation procedure We train a logistic regression classifier on top of sentence embeddings *for our added tasks* with random subsample validation (50 runs) to mitigate the effects of different random initializations. We use SGD with Adam and tune the learning rate on the validation set. In contrast, for a direct comparison against previously published results, we use SentEval (Conneau et al., 2017) to evaluate MR, CR, SUBJ, MPQA, TREC, and SST. For most tasks, this approach likewise uses logistic regression with cross-validation.

We report macro F1 performance for AM, AC, and CLS to account for imbalanced classes, and accuracy for all tasks evaluated using SentEval.

4.2 Results

Table 2 compares models across all 9 transfer tasks. The results show that we can substantially improve sentence embeddings when concatenating multiple word embedding types. All four embedding types concatenated achieve 2pp improvement over the best individual embeddings (GV). Incorporating further power means also substantially improve performances. GV improves by 0.6pp on average, GN by 1.9pp, MS by 2.1pp and AR by 3.7pp when concatenating $p = \pm\infty$ to the standard value $p = 1$ (dimensionality increases to 900). The combination of concatenation of embedding types and power means gives an average improvement of 3pp over the individually best embedding type.

However, there is one caveat with concatenated power mean embeddings: both the concatenated embeddings as well as the different power means live in their own "coordinate system", i.e., may have different ranges. Thus, we subtract the column-wise mean of the embedding matrix as well as divide by the standard deviation, which is the z-norm operation as proposed in (LeCun et al., 1998). This indeed improves the results by 1.0pp. We thereby reduce the gap to InferSent from 4.6pp to 0.6pp (or 85%), while having a lower dimensionality (3600 vs 4096). For InferSent, this normalization decreased scores by 0.1pp on average.

We consistently outperform the lower-dimensional SIF, Siamese-CBOW, and Sent2Vec embeddings. We conjecture that these methods underperform as universal sentence embeddings[4] because they each discard important information. For instance, SIF assigns low weight to common words such as discourse markers, while Siamese-CBOW similarly tends to assign low vector norm to function words (Kenter et al., 2016). However, depending on the task, function words may have

不同的方法关注的信息不同

[2] The original CLS was built for *document* classification.

[3] Monolingually, we limit our experiments to the three named power means to not exceed the dimensionality of InferSent.

[4] SIF and others were originally only evaluated in textual similarity tasks.

Task	Type	Size	X-Ling	C	Example (X-Ling)
AM	Argumentation	7k	HT†	4	Viele der technologischen Fortschritte helfen der Umwelt sehr. *(claim)*
AC	Argumentation	450	HT†	2	Too many promises have not been kept. *(none)*
CLS	Product-reviews	6k	HT	2	En tout cas on ne s'ennuie pas à la lecture de cet ouvrage! *(pos)*
MR	Sentiment	11k	MT	2	Dunkel und verstörend, aber auch überraschend witzig. *(pos)*
CR	Product-reviews	4k	MT	2	This camera has a major design flaw. *(neg)*
SUBJ	Subjectivity	10k	MT	2	On leur raconte l'histoire de la chambre des secrets. *(obj)*
MPQA	Opinion-polarity	11k	MT	2	sind eifrig *(pos)* | nicht zu unterstützen *(neg)*
TREC	Question-types	6k	MT	6	What's the Olympic Motto? *(desc—question asking for description)*
SST	Sentiment	70k	MT	2	Holm... incarne le personnage avec un charisme regal [...] *(pos)*

Table 1: Evaluation tasks with examples from our transfer languages. The first three tasks include human-generated cross-lingual data (HT), the last 6 tasks contain machine translated sentences (MT). C is the number of classes. † indicates that a dataset contains machine translations for French.

Model	Σ	AM	AC	CLS	MR	CR	SUBJ	MPQA	SST	TREC
Arithmetic mean										
GloVe (GV)	77.2	50.0	70.3	76.6	77.1	78.3	91.3	87.9	80.2	83.4
GoogleNews (GN)	76.1	50.6	69.4	75.2	76.3	74.6	89.7	88.2	79.9	81.0
Morph Specialized (MS)	73.5	47.1	64.6	74.1	73.0	73.1	86.9	88.8	78.3	76.0
Attract-Repel (AR)	74.1	50.3	63.8	75.3	73.7	72.4	88.0	89.1	78.3	76.0
GV ⊕ GN ⊕ MS ⊕ AR	**79.1**	**53.9**	**71.1**	**77.2**	**78.2**	**79.8**	91.8	**89.1**	**82.8**	**87.6**
power mean [p-values]										
GV $[-\infty, 1, \infty]$	77.9	54.4	69.5	76.4	76.9	78.6	92.1	87.4	80.3	85.6
GN $[-\infty, 1, \infty]$	77.9	55.6	71.4	75.8	76.4	78.0	90.4	88.4	80.0	85.2
MS $[-\infty, 1, \infty]$	75.8	52.1	66.6	73.9	73.1	75.8	89.7	87.1	79.1	84.8
AR $[-\infty, 1, \infty]$	77.6	55.6	68.2	75.1	74.7	77.5	89.5	88.2	80.3	89.6
GV ⊕ GN ⊕ MS ⊕ AR $[-\infty, 1, \infty]$	80.1	58.4	71.5	77.0	78.4	80.4	**93.1**	88.9	83.0	90.6
→ with z-norm†	**81.1**	**60.5**	**75.5**	**77.3**	**78.9**	**80.8**	93.0	**89.5**	**83.6**	**91.0**
Baselines										
GloVe + SIF	76.1	45.6	72.2	75.4	77.3	78.6	90.5	87.0	80.7	78.0
Siamese-CBOW	60.7	42.6	45.1	66.4	61.8	63.8	75.8	71.7	61.9	56.8
Sent2Vec	78.0	52.4	**72.7**	75.9	76.3	80.3	91.1	86.6	77.7	**88.8**
InferSent	**81.7**	**60.9**	72.4	**78.0**	**81.2**	**86.7**	**92.6**	**90.6**	**85.0**	88.2

Table 2: Monolingual results. Brackets show the different power means that were applied to all individual embeddings. † we normalized the embeddings of our full model with the z-norm as proposed by LeCun et al. (1998).

crucial signaling value. For instance, in AM, words like "thus" often indicate argumentativeness.

While the representations learned by Siamese-CBOW and SIF are indeed lower-dimensional than both our own representations as well as those of InferSent, we find it remarkable that they both perform below the (likewise low-dimensional) GV baseline on average. Sent2Vec (700d) outperforms GV, but performs below the concatenation of GV and GN (600d). This challenges their statuses as hard-to-beat baselines when evaluated on many different transfer tasks.

We further note that our concatenated power mean word embeddings outperform much more resource-intensive approaches such as Skip-thought in 4 out of 6 common tasks reported in Conneau et al. (2017) and the neural MT (en-fr) system reported there in 5 of 5 common tasks.

Dimensionality vs. Average Score Our initial motivation stated that a fair evaluation of sentence embeddings should compare embeddings of similar sizes. Figure 1 investigates the relationship of dimensionality and performance based on our conducted experiments. We see that, indeed, larger embedding sizes lead to higher average performance scores; more precisely, there appears to be a sub-linear (logarithmic-like) growth in average performance as we increase embedding size through concatenation of diverse word embeddings. This holds for both the standard concatenation of average ($p = 1$) embeddings and the p-mean concatenation with $p = 1, \pm\infty$. Further, we observe that the concatenation of diverse average ($p = 1$) word embeddings typically outperforms the p-mean summary of the same dimensionality ($p = 1, \pm\infty$). For example, concatenating arithmetic averages of GV,

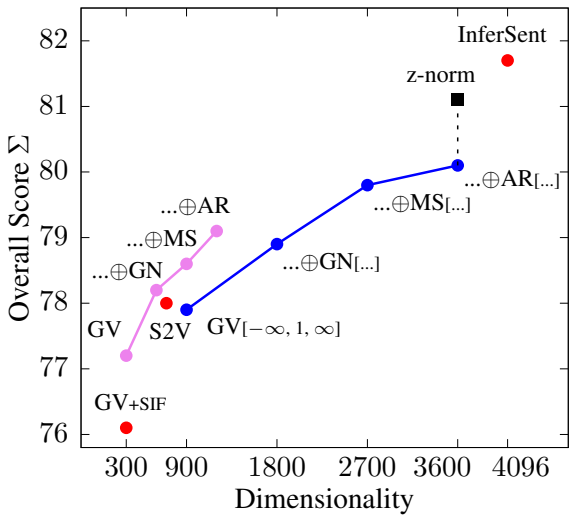

Figure 1: The average monolingual performance for the different sentence embeddings in relation to their dimensionality. We visually group related embeddings (i.e., average and power mean embeddings). S2V is Sent2Vec.

GN, and MS embeddings ($d = 900$) outperforms the p-mean embeddings ($p = 1, \pm\infty$) of GV. Similarly, GV⊕GN⊕MS⊕AR ($d = 1200$) outperforms the even higher-dimensional GV⊕GN$_{[1,\pm\infty]}$ score ($d = 1800$). This suggests that there is a trade-off for the considered p-mean concatenations: while they typically improve performance, the increase is accompanied by an increase in embedding size, which makes alternatives (e.g., concatenation of arithmetic average embeddings) competitive. However, when further concatenation of more embedding types is not possible (because no more are available) or when re-training of a given embedding type with higher dimensionality is unfeasible (e.g., because the original resources are not available or because training times are prohibitive), concatenation of power mean embeddings offers a strong performance increase that is based on a better summary of the present information. Finally, we remark the very positive effect of z-normalization, which is particularly beneficial in our situation of the concatenation of heterogeneous information; as stated, InferSent did not witness a corresponding performance increase. Our overall final result is very close to that of InferSent, while being lower dimensional and considerably cheaper at test time. At the same time, we do not rely on high quality inference data at train time, which is unavailable for most languages other than English.

5 Cross-Lingual Experiments

5.1 Experimental Setup

Tasks We obtained German (de) and French (fr) translations of all sentences in our 9 transfer tasks.

Sentences in AC are already parallel (en, de), having been (semi-)professionally translated by humans from the original English. For AM, we use student translations from the original English into German (Eger et al., 2018). CLS (en, de, fr) is also available bilingually. For the remaining datasets we created machine translated versions using Google Translate for the directions en-de and en-fr.

Word embeddings Since our monolingual embeddings are not all available cross-lingually, we use alternatives:

- We train en-de and en-fr `BIVCD` (BV) embeddings on aligned sentences from the Europarl (Koehn, 2005) and the UN corpus (Ziemski et al., 2016), respectively, using word2vec;
- `Attract-Repel` (AR) (Mrkšić et al., 2017) provide pre-trained cross-lingual word embeddings for en-de and en-fr;
- Monolingual `Fasttext` (FT) word embeddings (Bojanowski et al., 2017) of multiple languages trained on Wikipedia, which we map into shared vector space with a non-linear projection method similar to the ones proposed in Wieting et al. (2015), but with necessary modifications to account for the cross-lingual setting. (technical details are given in the appendix).

We also re-map the BV and AR embeddings using our technique. Even though BV performances were not affected by this projection, AR embeddings were greatly improved by it. All our cross-lingual word embeddings have $d = 300$.

Evaluated approaches Similar to the monolingual case, we evaluate standard averages ($p = 1$) for all embedding types, as well as different concatenations of word embedding types and power means. Since we have only three rather than four base embeddings here, we additionally report results for $p = 3$. Again, we motivate our choice of p-means below.

We also evaluate bilingual SIF embeddings, i.e., SIF applied to bilingual word embeddings, CVM-add of Hermann and Blunsom (2014) with dimensionality $d = 1000$ which we trained using sen-

tences from Europarl and the UN corpus,[5] and three novel cross-lingual variants of InferSent:

(1) `InferSent MT`: We translated all 569K sentences in the SNLI corpus (Bowman et al., 2015) to German and French using Google Translate. To train, e.g., en-de InferSent, we consider all 4 possible language combinations over each sentence pair in the SNLI corpus. Therefore, our new SNLI corpus is four times as large as the original.

(2) `InferSent TD`: We train the InferSent model on a different task where it has to differentiate between translations and unrelated sentences (translation detection), i.e., the model has two output classes but has otherwise the same architecture. To obtain translations, we use sentence translation pairs from Europarl (en-de) and the UN corpus (en-fr); unrelated sentences were randomly sampled from the respective corpora. We limited the number of training samples to the size of the SNLI corpus to keep the training time reasonable.[6]

(3) `InferSent MT+TD`: This is a combination of the two previous approaches where we merge translation detection data with cross-lingual SNLI. The two label sets are combined, resulting in 5 different classes.

We trained all InferSent adaptations using the cross-lingual AR word embeddings.

We do not consider cross-lingual adaptations of ParagraphVec and NMT approaches here because they already underperform simple word averaging models monolingually (Conneau et al., 2017).

Evaluation procedure We replicate the monolingual evaluation procedure and train the classifiers on English sentence embeddings. However, we then measure the transfer performance on German and French sentences (en→de, en→fr).

5.2 Results

For ease of consideration, we report average results over en→de and en→fr in Table 3. Per-language scores can be found in the appendix.

As in the monolingual case, we observe substantial improvements when concatenating different types of word embeddings of ∼2pp on average. However, when adding FT embeddings to the already strong concatenation BV⊕AR, the performance only slightly improves on average.

Conversely, using different power means is more effective, considerably improving performance values compared to arithmetic mean word embeddings. On average, concatenation of word embedding types plus different power means beats the best individual word embeddings by 4.4pp cross-lingually, from 69.2% for AR to 73.6%.

We not only beat all our InferSent adaptations by more than 2pp on average cross-lingually, our concatenated power mean embeddings also outperform the more complex InferSent adaptations in 8 out of 9 individual transfer tasks.

Further, we perform on par with InferSent already with dimensionality $d = 900$, either using the concatenation of our three cross-lingual word embeddings or using AR with three power means ($p = 1, \pm\infty$). In contrast, CVD-add and AR+SIF stay below InferSent, and, as in the monolingual case, even underperform relative to the best individual cross-lingual average word embedding baseline (AR), indicating that they are not suitable as universal feature representations.

6 Analysis

Machine translations To test the validity of our evaluations that are based on machine translations, we compared performances when evaluating on machine (MT) and human translations (HT) of our two parallel AM and AC datasets.

We re-evaluated the same 14 methods as in Table 3 using MT target data. We find a Spearman correlation of $\rho = 96.5\%$ and a Pearson correlation of $\tau = 98.4\%$ between HT and MT for AM. For AC we find a ρ value of 83.7% and a τ value of 89.9%. While the latter correlations are lower, we note that the AC scores are rather close in the direction en→de, so small changes (which may also be due to chance, given the dataset's small size) can lead to rank differences. Overall, this indicates that our MT experiments yield reliable rankings and that they strongly correlate to performance values measured on HT. Indeed, introspecting the machine translations, we observed that these were of very high perceived quality.

Different power means We performed additional cross-lingual experiments based on the concatenation of BV ⊕ AR ⊕ FT with additional p-means. In particular, we test (i) if some power means are more effective than others, and (ii) if us-

[5] We observed that $d = 1000$ performs slightly better than higher-dimensional CVM-add embeddings of $d = 1500$ and much better than the standard configuration with $d = 128$. This is in line with our assumption that single-type embeddings become better with higher dimension, but will not generate additional information beyond a certain threshold, cf. §1.

[6] Also, adding more data did not improve performances.

Model	Σ	AM	AC	CLS	MR	CR	SUBJ	MPQA	SST	TREC
Arithmetic mean										
BIVCD (BV)	67.3	40.5	67.6	66.3	64.4	71.7	81.1	81.6	65.7	67.0
	(3.7)	(5.6)	(3.1)	(4.0)	(1.9)	(0.6)	(3.5)	(3.1)	(3.8)	(7.7)
Attract-Repel (AR)	69.2	38.6	68.8	68.9	68.2	73.9	82.8	84.4	72.5	64.5
	(3.6)	(4.8)	(0.8)	(4.3)	(3.4)	(2.1)	(3.0)	(1.8)	(3.4)	(9.2)
FastText (FT)	68.3	38.4	63.4	70.0	69.1	73.1	85.1	81.5	69.3	65.1
	(5.6)	(8.5)	(2.9)	(4.1)	(4.1)	(2.5)	(3.6)	(4.5)	(8.6)	(11.7)
BV ⊕ AR ⊕ FT	**71.2**	40.0	67.7	**71.6**	**70.3**	**76.8**	**86.2**	84.7	73.3	70.5
	(5.9)	(11.8)	(3.3)	(3.6)	(5.1)	(1.4)	(3.8)	(3.3)	(6.8)	(13.9)
power mean [p-values]										
BV $[-\infty, 1, \infty]$	68.7	48.0	68.8	65.8	63.7	72.2	82.5	81.3	66.9	69.5
	(4.3)	(4.7)	(1.5)	(4.9)	(3.5)	(1.4)	(3.7)	(3.6)	(3.9)	(11.1)
AR $[-\infty, 1, \infty]$	71.1	44.2	67.8	68.7	68.8	75.5	84.3	84.4	73.0	73.5
	(4.5)	(8.1)	(1.2)	(5.0)	(3.8)	(2.8)	(3.1)	(2.5)	(4.9)	(8.8)
FT $[-\infty, 1, \infty]$	69.4	43.9	64.2	69.4	67.6	73.4	85.8	81.4	73.2	65.5
	(6.2)	(9.7)	(2.5)	(4.4)	(5.8)	(3.0)	(3.7)	(5.1)	(5.3)	(16.4)
BV ⊕ AR ⊕ FT $[-\infty, 1, \infty]$	73.2	50.2	**69.3**	**71.5**	70.4	**76.7**	86.7	84.5	75.2	74.3
	(5.0)	(6.8)	(1.5)	(3.8)	(5.0)	(2.4)	(4.1)	(3.8)	(5.9)	(12.0)
BV ⊕ AR ⊕ FT $[-\infty, 1, 3, \infty]$	**73.6**	**52.5**	69.1	71.1	**70.6**	**76.7**	**87.5**	**84.9**	**75.5**	**74.8**
	(5.0)	(5.7)	(1.6)	(4.3)	(5.2)	(2.7)	(4.0)	(3.3)	(5.1)	(12.8)
Baselines										
AR + SIF	68.1	38.4	67.7	**69.1**	67.7	73.8	81.6	**81.7**	70.0	63.2
	(3.5)	(3.8)	(2.6)	(3.0)	(4.2)	(1.9)	(2.9)	(3.1)	(6.2)	(3.5)
CVM-add	67.4	47.8	68.9	64.2	63.4	70.3	79.5	79.3	70.2	67.8
	(5.7)	(5.7)	(-0.1)	(5.9)	(4.5)	(5.5)	(6.8)	(4.5)	(6.8)	(12.1)
InferSent MT	71.0	49.3	69.8	67.9	69.2	**76.3**	**84.6**	76.4	**73.4**	72.3
	(7.4)	(8.5)	(2.7)	(7.3)	(5.1)	(4.5)	(3.8)	(11.7)	(6.4)	(16.5)
InferSent TD	71.0	**51.1**	**72.0**	67.9	68.9	74.7	84.3	76.8	72.7	71.0
	(6.9)	(8.3)	(1.2)	(7.3)	(5.2)	(4.2)	(4.0)	(10.5)	(6.1)	(15.0)
InferSent MT+TD	**71.3**	50.2	71.3	67.7	**69.6**	76.2	84.4	77.0	72.1	**73.2**
	(7.5)	(8.3)	(2.2)	(8.2)	(5.4)	(5.1)	(4.3)	(11.1)	(7.1)	(15.9)

Table 3: Cross-lingual results averaged over en→de and en→fr. Numbers in parentheses are the in-language results minus the given cross-language value.

ing more power means, and thus increasing the dimensionality of the embeddings, further increases performances.

We chose several intuitive values for power mean in addition to the ones already tried out, namely $p = -1$ (harmonic mean), $p = 0.5$, $p = 2$ (quadratic mean), and $p = 3$ (cubic mean). Table 4 reports the average performances over all tasks. We notice that $p = 3$ is the most effective power mean here and $p = -1$ is (by far) least effective. We discuss below why $p = -1$ hurts performances in this case. For all cases with $p > 0$, additional means tend to further improve the results, but with decreasing marginal returns. This also means that improvements are not merely due to additional dimensions but due to addition of complementary information.

power mean-values	Σ X-Ling	Σ In-Language
$p = 1, \pm\infty$	73.2	78.2
$p = 1, \pm\infty, -1$	59.9	61.6
$p = 1, \pm\infty, 0.5$	73.0	78.6
$p = 1, \pm\infty, 2$	73.4	78.5
$p = 1, \pm\infty, 3$	73.6	78.6
$p = 1, \pm\infty, 2, 3$	**73.7**	78.7
$p = 1, \pm\infty, 0.5, 2, 3$	73.6	**78.9**

Table 4: Average scores (en→de and en→fr) for additional power means (based on BV ⊕ AR ⊕ FT).

7 Discussion

Why is it useful to concatenate power means? The average of word embeddings discards a lot of information because different sentences can be represented by similar averages. The concatenation of different power means yields a more precise summary because it reduces uncertainty about the semantic variation within a sentence. For example, knowing the min and the max guarantees that embedding dimensions are all within certain ranges.

Which power means promise to be beneficial? Large $|p|$ quickly converge to min ($p = -\infty$) and

max ($p = \infty$). Hence, besides min and max, further good power mean-values are typically small numbers, e.g., $|p| < 10$. If they are integral, then odd numbers are preferable over even ones because even power means lose sign information. Further, positive power means are preferable over negative ones (see results in Table 4) because negative power means are in a fundamental sense discontinuous when input numbers are negative: they have multiple poles (power mean value tends toward $\pm\infty$) and different signs around the poles, depending on the direction from which one approaches the pole.

合适的p值

Cross-lingual performance decrease For all models and tasks, we observed decreased performances in the cross-lingual evaluation compared to the in-language evaluation. Most importantly, we observe a substantial difference between the performance decreases of our best model (5pp) and our best cross-lingual InferSent adaptation (7.5pp). Two reasons may explain these observations.

First, InferSent is a complex approach based on a bidirectional LSTM. In the same vein as Wieting et al. (2016), we hypothesize that embeddings learned from complex approaches transfer less well across domains compared to embeddings derived from simpler methods such as power mean embeddings. In our case, we transfer across languages, which is a pronounced form of domain shift.

Second, InferSent typically requires large amounts of high-quality training data. In the cross-lingual case we rely on translated sentences for training. Even though we found these translation to be of high quality, they can still introduce noise because some aspects of meaning in languages can only be approximately captured by translations. This effect could increase with more distance between languages. In particular, we observe a higher cross-language drop for the language transfer en→fr than for en→de. Furthermore, this difference is less pronounced for our power mean embeddings than it is for InferSent, potentially supporting this assumption (see the appendix for individual cross-language results).

Applications for cross-lingual embeddings A fruitful application scenario of cross-lingual sentence embeddings are cases in which we do not have access to labeled target language training data. Even though we could, in theory, machine translate sentences from the target language into English and apply a monolingual classifier, state-of-the-art MT systems like Google Translate currently only cover a small fraction of the world's ∼7000 languages.

Using cross-lingual sentence embeddings, however, we can train a classifier on English and then directly apply it to low-resource target language sentences. This so-called direct transfer approach (Zhang et al., 2016) on sentence-level can be beneficial when labeled data is scarce, because sentence-level approaches typically outperform task-specific sentence representations induced from word-level models in this case (Subramanian et al., 2018).

8 Conclusion

We proposed concatenated power mean word embeddings, a conceptually and computationally simple method for inducing sentence embeddings using two ingredients: (i) the concatenation of diverse word embeddings, which injects complementary information in the resulting representations; (ii) the use of power means to perform different types of summarizations over word embedding dimensions.

Our proposed method narrows the monolingual gap to state-of-the-art supervised methods while substantially outperforming cross-lingual adaptations of InferSent in the cross-lingual scenario.

We believe that our generalizations can be widely extended and that we have merely laid the conceptual ground-work: automatically learning power mean-values is likely to result in further improved sentence embeddings as we have observed that some power mean-values are more suitable than others, and using different power mean-values for different embedding dimensions could introduce more diversity.

自动学习p值，以及不同维度使用不同的p值

Finally, we believe that even in monolingual scenarios, future work should consider (concatenated) power mean embeddings as a challenging and truly hard-to-beat baseline across a wide array of transfer tasks.

Acknowledgments

This work has been supported by the German Research Foundation as part of the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1), by the German Federal Ministry of Education and Research (BMBF) under the promotional reference 01UG1816B (CEDIFOR), and by the German Research Foundation (DFG) as part of the Research Training Group "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES) under grant No. GRK 1994/1.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In *International Conference on Learning Representations (ICLR 2017)*. http://arxiv.org/abs/1608.04207.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough to beat baseline for sentence embeddings. In *International Conference on Learning Representations (ICLR 2017)*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*. Association for Computational Linguistics, pages 2289–2294. https://doi.org/10.18653/v1/D16-1250.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3:1137–1155. http://dl.acm.org/citation.cfm?id=944919.944966.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association of Computational Linguistics (TACL)* 5:135–146. https://doi.org/1511.09249v1.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*. Association for Computational Linguistics, pages 632–642. https://doi.org/10.18653/v1/D15-1075.

Sarath Chandar, Mitesh M. Khapra, Balaraman Ravindran, Vikas Raykar, and Amrita Saha. 2013. Multilingual Deep Learning. *Deep Learning Workshop (NIPS)* .

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*. ACM, pages 160–167. https://doi.org/10.1145/1390156.1390177.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. Association for Computational Linguistics, pages 681–691. http://www.aclweb.org/anthology/D17-1071.

Steffen Eger, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2018. Cross-lingual argumentation mining: Machine translation (and a bit of projection) is all you need! In *Proceedings of the 29th International Conference on Computational Linguistics (COLING 2018)*. pages 831–844. http://tubiblio.ulb.tu-darmstadt.de/105431/.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*. Association for Computational Linguistics, Gothenburg, Sweden, pages 462–471. http://www.aclweb.org/anthology/E14-1049.

G.H. Hardy, J.E. Littlewood, and G. Pólya. 1952. *Inequalities*. Cambridge University Press, Cambridge, England.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual Models for Compositional Distributed Semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*. Association for Computational Linguistics, pages 58–68. https://doi.org/10.3115/v1/P14-1006.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning Distributed Representations of Sentences from Unlabelled Data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*. San Diego, California, pages 1367–1377. https://doi.org/10.18653/v1/N16-1162.

Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese CBOW: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. Association for Computational Linguistics, pages 941–951. https://doi.org/10.18653/v1/P16-1089.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: a Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR 2015)*. https://arxiv.org/abs/1412.6980.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS 2015)*. MIT Press, pages 3294–3302.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *In Proceedings of the tenth Machine Translation Summit*. AAMT, pages 79–86.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*. Association for Computational Linguistics, pages 1490–1500. http://www.aclweb.org/anthology/N16-1175.

Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning (ICML 2014)*. ACM, pages 1188–1196. http://dl.acm.org/citation.cfm?id=3044805.3045025.

Yann LeCun, Leon Bottou, Genevieve B. Orr, and Klaus Robert Müller. 1998. *Efficient BackProp*, Springer Berlin Heidelberg.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*. Association for Computational Linguistics, pages 302–308. https://doi.org/10.3115/v1/P14-2050.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *NAACL Workshop on Vector Space Modeling for NLP*. The Association for Computational Linguistics, pages 151–159. http://www.aclweb.org/anthology/W15-1521.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint* http://arxiv.org/abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems 26 (NIPS 2013)* pages 3111–3119. http://arxiv.org/abs/1310.4546.

Aditya Mogadala and Achim Rettinger. 2016. Bilingual word embeddings from parallel and non-parallel corpora for cross-language text classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*. Association for Computational Linguistics, pages 692–702. http://www.aclweb.org/anthology/N16-1083.

Nikola Mrkšić, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gašić, Anna Korhonen, and Steve Young. 2017. Semantic Specialisation of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints. *Transactions of the Association of Computational Linguistics (TACL)* 5:309–324. http://www.aclweb.org/anthology/Q17-1022.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2017. Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. *arXiv preprint* http://arxiv.org/abs/1703.02507.

Andreas Peldszus and Manfred Stede. 2015. An annotated corpus of argumentative microtexts. In *Argumentation and Reasoned Action: Proceedings of the 1st European Conference on Argumentation*. Lisbon, Portugal, pages 801–815.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. Association for Computational Linguistics, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162.

Hieu Pham, Minh-Thang Luong, and Christopher D. Manning. 2015. Learning Distributed Representations for Multilingual Text Sequences. *Workshop on Vector Modeling for NLP* pages 88–94.

Peter Prettenhofer and Benno Stein. 2010. Cross-Lingual Adaptation using Structural Correspondence Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*. Association for Computational Linguistics, pages 1118–1127. http://www.aclweb.org/anthology/P10-1114.

Holger Schwenk and Matthijs Douze. 2017. Learning Joint Multilingual Sentence Representations with Neural Machine Translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP (REPL4NLP 2017)*. Association for Computational Linguistics, pages 157–167. http://www.aclweb.org/anthology/W17-2619.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics* 43(3):619–659.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations (ICLR 2018)*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*. Association for Computational Linguistics, pages 384–394. https://www.aclweb.org/anthology/P10-1040.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, and Dan Roth. 2016. Cross-lingual models of word embeddings: An empirical comparison. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. Association for Computational Linguistics, pages 1661–1670. http://www.aclweb.org/anthology/P16-1157.

Ivan Vulić and Marie-Francine Moens. 2015. Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*. Association for Computational Linguistics, pages 719–725. https://doi.org/10.3115/v1/P15-2118.

Ivan Vulić, Nikola Mrkšić, Roi Reichart, Diarmuid Ó Séaghdha, Steve Young, and Anna Korhonen. 2017. Morph-fitting: Fine-Tuning Word Vector Spaces with Simple Language-Specific Rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. Association for Computational Linguistics, pages 56–68. https://doi.org/10.18653/v1/P17-1006.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. From paraphrase database to compositional paraphrase model and back. *Transactions of the Association of Computational Linguistics (TACL)* 3:345–358. http://www.aclweb.org/anthology/Q15-1025.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *International Conference on Learning Representations (ICLR 2016)*. http://arxiv.org/abs/1511.08198.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *ArXiv preprint* http://arxiv.org/abs/1704.05426.

Ye Zhang, Stephen Roller, and Byron C. Wallace. 2016. MGNC-CNN: A simple approach to exploiting multiple word embeddings for sentence classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*. Association for Computational Linguistics, pages 1522–1527. http://www.aclweb.org/anthology/N16-1178.

Xinjie Zhou, Xianjun Wan, and Jianguo Xiao. 2016. Cross-Lingual Sentiment Classification with Bilingual Document Representation Learning. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)* pages 1403–1412.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA).

A Supplemental Material

A.1 Details on our Projection Method

Here we describe the necessary conceptual and technical details to reproduce the results of our non-linear projection method that we use to map word embeddings of two languages into a shared embedding space (cf. §5).

Formalization We learn a projection of two embedding spaces $\mathbb{E}^l$ and $\mathbb{E}^k$ with dimensionality e and f, respectively, into a shared space of dimensionality d using two non-linear transformations:

$$f_l(\mathbf{x}_l) = \tanh\left(\mathbf{W}_l\mathbf{x}_l + \mathbf{b}_l\right)$$
$$f_k(\mathbf{x}_k) = \tanh\left(\mathbf{W}_k\mathbf{x}_k + \mathbf{b}_k\right)$$

where $\mathbf{x}_l \in \mathbb{R}^e$, $\mathbf{x}_k \in \mathbb{R}^f$ are original input embeddings and $\mathbf{W}_l \in \mathbb{R}^{d\times e}$, $\mathbf{W}_k \in \mathbb{R}^{d\times f}$, $\mathbf{b}_l \in \mathbb{R}^d$, $\mathbf{b}_k \in \mathbb{R}^d$ are parameters to be learned. Here $\mathbf{x}_l$ and $\mathbf{x}_k$ are monolingual representations.

For each sentence s and its translation t we randomly sample one unrelated sentence u from our data and obtain sentence representations $\mathbf{r}_s = f_l(\mathbf{x}_s)$, $\mathbf{r}_t = f_k(\mathbf{x}_t)$, and $\mathbf{r}_u = f_k(\mathbf{x}_u)$. We then optimize the following max-margin hinge loss:

$$\mathcal{L} = \max\left(0,\ m - \mathsf{sim}(\mathbf{r}_s, \mathbf{r}_t) + \mathsf{sim}(\mathbf{r}_s, \mathbf{r}_u)\right)$$

where sim is cosine similarity and m is the margin parameter. This objective moves embeddings of translations closer to and embeddings of random cross-lingual sentences further away from each other.

Training We use minibatched SGD with the Adam optimizer (Kingma and Ba, 2015) for training. We train on >130K bilingually aligned sentence pairs from the TED corpus (Hermann and Blunsom, 2014), which consists of translated transcripts from TED talks. Each sentence s is represented by its average (monolingual) word embedding, i.e., H_1.

We set the margin parameter to $m = 0.5$ as we have observed that higher values lead to a faster convergence. We furthermore randomly set 50% of the input embedding dimensions to zero during training (dropout).

Training of one epoch usually takes less than a minute in our TensorFlow implementation (on CPU), and convergence is usually achieved after less than 100 epochs.

Model	$\sum$ X-Ling	$\sum$ In-Language
FT (monolingual)	-	80.8
FT (CCA[‡])	71.1	79.3
FT (our projection)	74.6	79.7
BV (orig)	70.9	75.8
BV (our projection)	71.0	74.6
AR (orig)	61.8	79.3
AR (our projection)	74.5	77.9

Table 5: The performance of our average word embeddings with our projection method in comparison to other approaches. [‡]We trained CCA on word-alignments extracted from TED transcripts using fast_align (i.e., CCA uses the same data source as our method).

Application Even though we learn our non-linear projection on the sentence level, we later apply it on the word level, i.e., we map individual word embeddings from each of two languages via $f_\psi(\mathbf{x}_\psi)$ where $\psi = l, k$. This is valid because average word embeddings live in the same space as individual word embeddings. The reason for doing so is that otherwise we would have to learn individual transformations for each of our power means, not only the average (= H_1), which would be too costly particularly when incorporating many different p-values. Working on the word-level, in general, also allows us to resort to word-level projection techniques using, e.g., word-alignments rather than sentence alignments.

However, in preliminary experiments, we found that our suggested approach produces considerably better cross-lingual word embeddings in our setup. Results are shown in Table 5, where we report the performance of average word embeddings for cross-lingual en→de task transfer (averaged over MR, CR, SUBJ, MPQA, SST, TREC). Compared to the word-level projection method CCA we obtain substantially better cross-lingual sentence embeddings, and even stronger improvements when re-mapping AR embeddings, even though these are already bilingual.

A.2 Individual Language-Transfer Results

We report results for the individual language transfer across en→de and en→fr in Table 6.

Model Transfer Language	Σ	Σ **de**	Σ **fr**	**AM** de	 fr	**AC** de	 fr	**CLS** de	 fr	**MR** de	 fr	**CR** de	 fr	**SUBJ** de	 fr	**MPQA** de	 fr	**SST** de	 fr	**TREC** de	 fr
Arithmetic mean																					
BIVCD (BV)	67.3	65.5	68.1	39.2	41.9	68.9	66.4	65.0	67.7	62.2	66.5	70.6	72.9	79.8	82.4	79.8	83.3	61.2	70.2	72.2	61.8
	(3.7)	(3.9)	(3.6)	(7.0)	(4.8)	(1.5)	(4.7)	(4.3)	(3.7)	(2.3)	(1.5)	(1.1)	(0.1)	(3.8)	(3.2)	(3.9)	(2.4)	(7.2)	(0.3)	(3.6)	(11.8)
Attract-Repel (AR)	69.2	69.4	68.9	39.0	38.2	**71.1**	66.5	67.4	70.4	66.6	69.7	73.7	74.0	81.8	83.8	84.0	84.8	71.3	73.6	69.6	59.4
	(3.6)	(3.4)	(3.8)	(4.7)	(4.7)	(0.4)	(1.3)	(5.8)	(2.8)	(4.7)	(2.2)	(1.9)	(2.4)	(3.2)	(2.7)	(2.1)	(1.5)	(4.4)	(2.4)	(3.8)	(14.6)
FastText (FT)	68.4	68.7	68.0	36.9	40.0	63.8	62.9	70.1	70.0	68.3	69.9	73.9	72.3	86.3	84.0	81.7	81.3	69.5	69.2	67.8	62.4
	(5.6)	(5.5)	(5.7)	(10.5)	(6.6)	(4.0)	(1.8)	(4.2)	(4.0)	(5.2)	(3.0)	(2.0)	(3.1)	(2.4)	(4.8)	(4.5)	(4.5)	(8.8)	(8.5)	(8.2)	(15.2)
BV ⊕ AR	71.1	70.9	**71.3**	**40.8**	**42.3**	70.0	**67.6**	69.4	70.9	67.2	70.5	75.4	76.5	83.3	84.8	84.1	**85.4**	**72.5**	**75.6**	**75.8**	**68.0**
	(4.2)	(4.2)	(4.1)	(9.7)	(7.6)	(2.2)	(3.9)	(4.6)	(2.9)	(4.7)	(2.1)	(1.1)	(0.4)	(3.8)	(3.1)	(3.3)	(2.5)	(5.3)	(3.0)	(3.4)	(11.6)
BV ⊕ AR ⊕ FT	**71.2**	**71.9**	70.6	39.2	40.7	71.0	64.5	**71.2**	**72.1**	**69.8**	**70.8**	**76.6**	**76.9**	**86.8**	**85.7**	**84.6**	84.9	71.8	74.7	**75.8**	65.2
	(5.9)	(5.4)	(6.4)	(12.9)	(11.0)	(1.5)	(5.0)	(3.8)	(3.4)	(6.1)	(4.0)	(1.6)	(1.2)	(3.2)	(4.4)	(3.3)	(3.3)	(8.5)	(5.1)	(7.8)	(20.0)
p-**mean** [p-values]																					
BV $[-\infty, 1, \infty]$	68.7	68.0	69.5	48.0	47.9	70.7	**66.8**	64.4	67.3	60.5	66.8	71.1	73.3	81.1	83.9	79.9	82.7	64.4	69.4	72.0	67.0
	(4.3)	(4.1)	(4.4)	(4.1)	(5.4)	(0.8)	(2.3)	(5.5)	(4.2)	(4.2)	(2.7)	(2.2)	(0.6)	(4.4)	(2.9)	(3.9)	(3.3)	(3.1)	(4.7)	(8.4)	(13.8)
AR $[-\infty, 1, \infty]$	71.1	71.3	71.0	45.7	42.7	70.5	65.1	67.1	70.3	67.4	70.2	75.3	75.7	83.7	84.9	84.0	84.8	71.2	74.9	76.6	70.4
	(4.5)	(4.4)	(4.5)	(6.7)	(9.1)	(-0.1)	(2.6)	(6.6)	(3.5)	(5.2)	(2.3)	(3.5)	(2.1)	(3.2)	(3.1)	(2.7)	(2.3)	(6.6)	(3.2)	(5.6)	(12.0)
FT $[-\infty, 1, \infty]$	69.4	70.2	68.5	42.7	45.1	67.1	61.3	69.6	69.2	68.3	67.0	73.4	73.4	86.7	84.9	81.6	81.2	**74.4**	72.0	68.2	62.8
	(6.2)	(5.4)	(7.0)	(11.1)	(8.2)	(1.3)	(3.7)	(4.2)	(4.6)	(4.9)	(6.6)	(3.0)	(3.0)	(2.7)	(4.7)	(5.2)	(5.0)	(4.0)	(6.7)	(12.0)	(20.8)
BV ⊕ AR ⊕ FT $[-\infty, 1, \infty]$	73.2	73.7	72.7	50.8	49.6	**72.0**	66.6	**70.9**	**72.0**	**70.6**	70.2	**77.3**	76.1	86.6	86.8	84.1	84.9	73.4	**77.0**	77.6	71.0
	(5.0)	(4.6)	(5.5)	(5.6)	(7.7)	(0.3)	(2.8)	(4.1)	(3.4)	(4.7)	(5.2)	(2.3)	(2.5)	(4.2)	(4.0)	(3.9)	(3.6)	(8.1)	(3.7)	(7.8)	(16.8)
BV ⊕ AR ⊕ FT $[-\infty, 1, 3, \infty]$	**73.6**	**74.0**	**73.3**	**51.4**	**53.6**	**72.0**	66.3	70.8	71.5	70.5	**70.7**	77.1	**76.2**	**87.7**	**87.3**	**84.2**	**85.6**	74.1	76.9	**78.4**	**71.2**
	(5.0)	(4.5)	(5.3)	(6.6)	(4.8)	(0.5)	(2.7)	(4.3)	(3.8)	(5.5)	(4.9)	(2.7)	(2.6)	(3.6)	(4.3)	(3.8)	(2.8)	(6.6)	(3.6)	(7.2)	(17.8)
Baselines																					
AR + SIF	68.1	67.5	68.7	40.2	36.5	70.3	65.1	**67.7**	**70.4**	66.2	69.2	73.7	73.8	80.0	83.2	**82.9**	80.5	68.0	71.9	58.4	68.0
	(3.5)	(4.3)	(2.7)	(2.3)	(5.2)	(1.8)	(3.4)	(4.5)	(1.5)	(5.5)	(3.0)	(2.3)	(1.6)	(3.8)	(2.0)	(2.0)	(4.1)	(8.8)	(3.7)	(7.4)	(-0.4)
CVM-Add	67.4	65.3	69.4	45.6	**49.9**	69.6	68.1	62.3	66.0	60.7	66.1	68.6	72.0	76.6	82.3	76.1	**82.5**	62.7	67.7	65.2	**70.4**
	(5.7)	(6.6)	(4.8)	(7.7)	(3.7)	(-0.7)	(-0.1)	(6.3)	(5.5)	(4.9)	(4.0)	(6.5)	(4.5)	(8.5)	(5.0)	(5.7)	(3.3)	(7.7)	(5.9)	(13.2)	(11.0)
InferSent MT	71.0	71.8	**70.2**	50.3	48.3	70.9	68.7	67.0	68.9	69.3	69.2	**76.7**	75.8	84.4	**84.9**	77.9	74.9	**72.5**	**74.3**	77.4	67.2
	(7.4)	(6.7)	(8.2)	(8.2)	(9.5)	(1.2)	(4.2)	(8.5)	(6.1)	(5.2)	(4.9)	(3.5)	(5.5)	(3.7)	(3.8)	(10.2)	(13.3)	(7.4)	(5.5)	(12.2)	(20.8)
InferSent TD	71.0	72.1	70.0	**52.7**	49.5	73.4	**70.6**	66.8	69.1	68.6	69.2	74.9	74.4	84.3	84.2	77.4	76.3	**72.5**	72.8	78.2	63.8
	(6.9)	(5.9)	(7.9)	(7.0)	(10.0)	(1.7)	(0.7)	(8.9)	(5.7)	(4.7)	(5.8)	(3.8)	(4.6)	(3.7)	(4.3)	(9.7)	(11.3)	(5.9)	(6.3)	(7.4)	(22.6)
InferSent MT+TD	**71.3**	**72.4**	**70.2**	52.0	48.4	**73.5**	69.2	66.6	68.8	**69.5**	**69.7**	76.4	**76.0**	**84.7**	84.2	78.3	75.7	72.1	72.2	**78.8**	67.6
	(7.5)	(6.6)	(8.4)	(7.0)	(9.6)	(1.4)	(3.0)	(9.6)	(6.8)	(5.2)	(5.6)	(4.8)	(5.5)	(4.2)	(4.4)	(9.7)	(12.4)	(7.1)	(7.0)	(10.8)	(21.0)

Table 6: Individual cross-lingual results for the language transfer en→de and en→fr. Numbers in parentheses are the in-language results minus the given cross-language value.. ⊕ denotes the concatenation of different embeddings (or p-means), brackets show the different p-means of the model.