LAMB: A Good Shepherd of Morphologically Rich Languages

Sebastian Ebert and **Thomas Müller** and **Hinrich Schütze**
Center for Information and Language Processing
LMU Munich, Germany
ebert@cis.lmu.de

Abstract

This paper introduces STEM and LAMB, embeddings trained for stems and lemmata instead of for surface forms. For morphologically rich languages, they perform significantly better than standard embeddings on word similarity and polarity evaluations. On a new WordNet-based evaluation, STEM and LAMB are up to 50% better than standard embeddings. We show that both embeddings have high quality even for small dimensionality and training corpora.

1 Introduction

Despite their power and prevalence, embeddings, i.e., (low-dimensional) word representations in vector space, have serious practical problems. First, large text corpora are necessary to train high-quality embeddings. Such corpora are not available for underresourced languages. Second, morphologically rich languages (MRLs) are a challenge for standard embedding models because many inflectional forms are rare or absent even in a large corpus. For example, a Spanish verb has more than 50 forms, many of which are rarely used. This leads to missing or low quality embeddings for such inflectional forms, even for otherwise frequent verbs, i.e., sparsity is a problem. Therefore, we propose to compute normalized embeddings instead of embeddings for surface/inflectional forms (referred to as *forms* throughout the rest of the paper): STem EMbeddings (STEM) for word stems and LemmA eMBeddings (LAMB) for lemmata.

Stemming is a heuristic approach to reducing form-related sparsity issues. Based on simple rules, forms are converted into their stem.[1] However, often the forms of one word are converted into several different stems. For example, present indicative forms of the German verb "brechen" (to break) are mapped to four different stems ("brech", "brich", "bricht", "brecht"). A more principled solution is lemmatization. Lemmatization unites many individual forms, many of which are rare, in one equivalence class, represented by a single lemma. Stems and equivalence classes are more frequent than each individual form. As we will show, this successfully addresses the sparsity issue.

Both methods can learn high-quality semantic representations for rare forms and thus are most beneficial for MRLs as we show below. Moreover, less training data is required to train lemma embeddings of the same quality as form embeddings. Alternatively, we can train lemma embeddings that have the same quality but fewer dimensions than form embeddings, resulting in more efficient applications.

If an application such as parsing requires inflectional information, then stem and lemma embeddings may not be a good choice since they do not contain such information. However, NLP applications such as similarity benchmarks (e.g., MEN (Bruni et al., 2014)) and (as we show below) polarity classification are semantic and are largely

[1] In this paper, we use the term "stem" not in its linguistic meaning, but to refer to the character string that is produced when a stemming algorithm like SNOWBALL is applied to a word form. The stem is usually a prefix of the word form, but some orthographic normalization (e.g., "possibly" → "possible" or "possibli") is often also performed.

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 742–752,
Austin, Texas, November 1-5, 2016. ©2016 Association for Computational Linguistics

independent of inflectional morphology.

Our contributions are the following. (i) We introduce the normalized embeddings STEM and LAMB and show their usefulness on different tasks for five languages. This paper is the first study that comprehensively compares stem/lemma-based with form-based embeddings for MRLs. (ii) We show the advantage of normalization on word similarity benchmarks. Normalized embeddings yield better performance for MRL languages on most datasets (6 out of 7 datasets for German and 2 out of 2 datasets for Spanish). (iii) We propose a new intrinsic relatedness evaluation based on WordNet graphs and publish datasets for five languages. On this new evaluation, LAMB outperforms form-based baselines by a big margin. (iv) STEM and LAMB outperform baselines on polarity classification for Czech and English. (v) We show that LAMB embeddings are efficient in that they are high-quality for small training corpora and small dimensionalities.

2 Related Work

There have been a large number of studies on English, a morphologically simple language, that show that the effect of normalization, in particular stemming, is different for different applications. For instance, Karlgren and Sahlgren (2001) analyze the impact of morphological analysis on creating word representations for synonymy detection. They compare several stemming methods. Bullinaria and Levy (2012) use stemming and lemmatization before training word representations. The improvement of morphological normalization in both studies is moderate in the best case. Melamud et al. (2014) compute lemma embeddings to predict related words given a query word. They do not compare form and lemma representations.

A finding about English morphology does not provide insight into what happens with the morphology of an MRL. In this paper we use English to provide a data point for morphologically poor languages. Although we show that normalization for embeddings increases performance significantly on some applications – a novel finding to the best of our knowledge – morphologically simple languages (for which normalization is expected to be less important) are not the main focus of the paper. Instead, MRLs are the main focus. For these, we show large improvements on several tasks.

Recently, Köper et al. (2015) compared form and lemma embeddings on English and German focusing on morpho-syntactic and semantic relation tasks. Generally, they found that lemmatization has limited impact. We extensively study MRLs and find a strong improvement on MRLs when using normalization, on intrinsic as well as extrinsic evaluations.

Synonymy detection is a well studied problem in the NLP community (Turney, 2001; Turney et al., 2003; Baroni and Bisi, 2004; Ruiz-Casado et al., 2005; Grigonytė et al., 2010). Rei and Briscoe (2014) classify hyponomy relationships through embedding similarity. Our premise is that semantic similarity comprises all of these relations and more. Our ranking-based word relation evaluation addresses this issue. Similar to Melamud et al. (2014), our motivation is that, in contrast to standard word similarity benchmarks, large resources can be automatically generated for any language with a WordNet. This is also exploited by Tsvetkov et al. (2015). Their intrinsic evaluation method requires an annotated corpus, e.g., annotated with WordNet supersenses. Our approach requires only the WordNet.

An alternative strategy of dealing with data sparsity is presented by Soricut and Och (2015). They compute morphological features in an unsupervised fashion in order to construct a form embedding by the combination of the word's morphemes. We address scenarios (such as polarity classification) in which morphological information is less important, thus morpheme embeddings are not needed.

3 Stem/Lemma Creation

The main hypothesis of this work is that normalization addresses sparsity issues, especially for MRLs, because although a particular word form might not have been seen in the text, its stem or lemma is more likely to be known. For all stemming experiments we use SNOWBALL,[2] a widely used stemmer. It normalizes a form based on deterministic rules, such as *replace the suffix 'tional' by 'tion'* for English.

For lemmatization we use the pipeline version of the freely available, high-quality lemmatizer LEM-

[2] snowball.tartarus.org

MING (Müller et al., 2015). Since it is a language-independent token-based lemmatizer it is especially suited for our multi-lingual experiments. Moreover, it reaches state-of-the-art performance for the five languages that we study. We train the pipeline using the Penn Treebank (Marcus et al., 1993) for English, SPMRL 2013 shared task data (Seddah et al., 2013) for German and Hungarian, and CoNLL 2009 (Hajič et al., 2009) datasets for Spanish and Czech. We additionally use a unigram list extracted from Wikipedia datasets and the ASPELL dictionary of each language.[3]

4 Experiments

4.1 Word Similarity

Our first experiment evaluates how well STEM/LAMB embeddings predict human word similarity judgments. Given a pair of words (m, n) with a human-generated similarity value and a set of embeddings E we compute their similarity as cosine similarity. For form embeddings E^F, we directly use the embeddings of the word pairs' forms (E^F_m and E^F_n) and compute their similarity. For STEM we use $E^S_{stem(w)}$, where $stem(w)$ is the stem of w. For LAMB we use $E^L_{lemma(w)}$, where $lemma(w)$ is the lemma of w; we randomly select one of w's lemmata if there are several. We conduct experiments on English (en), German (de) and Spanish (es). Table 1 gives dataset statistics.

For good performance, high-quality embeddings trained on large corpora are required. Hence, the training corpora for German and Spanish are web corpora taken from COW14 (Schäfer, 2015). Preprocessing includes removal of XML, conversion of HTML characters, lowercasing, stemming using SNOWBALL and lemmatization using LEMMING. We use the entire Spanish corpus (3.7 billion tokens), but cut the German corpus to approximately 8 billion tokens to be comparable to Köper et al. (2015). We train CBOW models (Mikolov et al., 2013) for forms, stems and lemmata using WORD2VEC[4] with the following settings: 400 dimensions, symmetric context of size 2 (no dynamic window), 1 training iteration, negative sampling with 15 samples, a learning rate of 0.025, minimum count of words of 50, and a sampling parameter of 10^{-5}. CBOW is chosen, because it trains much faster than skip-gram, which is beneficial on these large corpora.

Since the morphology of English is rather simple we do not expect STEM and LAMB to reach or even surpass highly optimized systems on any word similarity dataset (e.g., Bruni et al. (2014)). Therefore, for practical reasons we use a smaller training corpus, namely the preprocessed and tokenized Wikipedia dataset of Müller and Schütze (2015).[5] Embeddings are trained with the same settings (using 5 iterations instead of only 1, due to the smaller size of the corpus: 1.8 billion tokens).

Table 1 shows results. We also report the Spearman correlation on the vocabulary intersection, i.e., only those word pairs that are covered by the vocabularies of all models.

Results. Although English has a simple morphology, LAMB improves over form performance on MEN and SL. A tie is achieved on RW. These are the three largest English datasets, giving a more reliable result. Both models perform comparably on WS. Here, STEM is ahead by 1 point. Forms are better on the small datasets MC and RG, where a single word pair can have a large influence on the result. Additionally, these are datasets with high frequency forms, where form embeddings can be well trained. Because of the simple morphology of English, STEM/LAMB do not outperform forms or only by a small margin and thus they cannot compete with highly optimized state-of-the-art systems.[6]

On German, both STEM and LAMB perform better on all datasets except WS. We set the new state-of-the-art of 0.79 on Gur350 (compared to 0.77 (Szarvas et al., 2011)) and 0.39 on ZG (compared to 0.25 (Botha and Blunsom, 2014)); 0.83 on Gur65 (compared to 0.79 (Köper et al., 2015)) is the best performance of a system that does not need additional knowledge bases (cf. Navigli and Ponzetto (2012), Szarvas et al. (2011)).

LAMB's results on Spanish are equally good. 0.82 on MC and 0.58 on WS are again the best per-

[3] ftp://ftp.gnu.org/gnu/aspell/dict

[4] code.google.com/p/word2vec/

[5] cistern.cis.lmu.de/marmot/naacl2015

[6] Baroni et al. (2014)'s numbers are higher on some of the datasets for the *best* of 48 different parameter configurations. In contrast, we do not tune parameters.

formances of a system not requiring an additional knowledge base (cf. Navigli and Ponzetto (2012)). The best performance before was 0.64 for MC and 0.50 for WS (both Hassan and Mihalcea (2009)). STEM cannot improve over form embeddings, showing the difficulty of Spanish morphology.

4.2 Word Relations

Word similarity benchmarks are not available for many languages and are expensive to create. To remedy this situation, we create word similarity benchmarks that leverage WordNets, which are available for a great number of languages.

Generally, a representation is deemed good if words related by a lexical relation in WordNet – synonymy, hyponymy etc. – have high cosine similarity with this representation. Since the gold standard necessary for measuring this property of a representation can be automatically derived from a WordNet, we can create very large similarity benchmarks with up to 50k lemmata for the five languages we investigate: Czech, English, German, Hungarian and Spanish.

We view each WordNet as a graph whose edges are the lexical relations encoded by the WordNet, e.g., synonymy, antonymy and hyponymy. We then define $\mathcal{L}$ as the set of lemmata in a WordNet and the distance $d(l, l')$ between two lemmata l and l' as the length of the *shortest path* connecting them in the graph. The k-neighborhood $N^k(l)$ of l is the set of lemmata l' that have distance k or less, excluding l: $N^k(l) := \{l' | d(l, l') \leq k, l \neq l'\}$. The rank of l for an embedding set E is defined as:

$$\mathrm{rank}_E^k(l) := \operatorname*{argmin}_i \, l_i \in N^k(l) \tag{1}$$

where l_i is the lemma at position i in the list of all lemmata in the WordNet, ordered according to cosine similarity to l in descending order. We restrict $i \in [1, 10]$ and set $k = 2$ for all experiments in this paper. We omit the indexes k and E when they are clear from context.

To measure the quality of a set of embeddings we compute the mean reciprocal rank (MRR) on the rank results of all lemmata:

$$\mathrm{MRR}_E = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \frac{1}{\mathrm{rank}_E(l)} \tag{2}$$

We create large similarity datasets for five languages: Czech (cz), English (en), German (de), Hungarian (hu) and Spanish (es) by extracting all lemmata from the WordNet version of the respective language. For English and Spanish we use the preprocessed WordNets from the Open Multilingual WordNet (Bond and Paik, 2012). We use the Czech and Hungarian WordNets (PALA and SMRZ, 2004; Miháltz et al., 2008) and GermaNet (Hamp and Feldweg, 1997) for German. We keep all lemmata that have a known form in the form embeddings and that exist in the lemma embeddings. Moreover, we filter out all synsets that contain only one lemma and discard all multiword phrases. The split into development and test sets is done in a way that the distribution of synset sizes (i.e., the number of lemmata per synset) is nearly equal in both sets.

The number of lemmata in our evaluation sets can be found in Table 2. For more insight, we report results on all parts-of-speech (POS), as well as separately for nouns (n), verbs (v) and adjectives (a).[7] The data is provided as supplementary material.[8]

We propose the following models for the embedding evaluation. For form embeddings we compare three different strategies, a realistic one, an optimistic one and a lemma approximation strategy. In the realistic strategy (*form real*), given a query lemma we randomly sample a form, for which we then compute the k-neighborhood. If the neighbors contain multiple forms of the same equivalence class, we exclude the repetitions and use the next neighbors instead. For instance, if *house* is already a neighbor, then *houses* will be skipped. The optimistic strategy (*form opt*) works similarly, but uses the embedding of the most frequent surface form of a lemma. This is the most likely form to perform best in the form model. This strategy presupposes the availability of information about lemma and surface form counts. As a baseline lemma approximation strategy, we sum up all surface form embeddings that belong to one equivalence class (*form sum*). For STEM we repeat the same experiments as described for forms, leading to *stem real*, *stem opt* and *stem sum*.

For embedding training, Wikipedia comes as a

[7] The all-POS setting includes all POS, not just n, v, a.

[8] All supplementary material is available at www.cis.uni-muenchen.de/ebert/

				full vocabulary				vocabulary intersection			
lang	dataset	reference	pairs	form	STEM	LAMB	cov.	form	STEM	LAMB	cov.
de	Gur30	Gurevych (2005)	29	0.76	**0.83**	0.80	29, 29, 29	0.76	**0.83**	0.80	29
	Gur350	Gurevych (2005)	350	0.74	**0.79**	**0.79**	336, 340, 339	0.74	**0.79**	**0.79**	336
	Gur65	Gurevych (2005)	65	0.80	**0.83**	0.82	65, 65, 65	0.80	**0.83**	0.82	65
	MSL	Leviant and Reichart (2015)	999	0.44	0.44	**0.47**	994, 995, 995	0.44	0.44	**0.47**	994
	MWS	Leviant and Reichart (2015)	350	0.60	0.61	**0.62**	348, 350, 350	0.60	**0.61**	**0.61**	348
	WS	Köper et al. (2015)	280	**0.72**	**0.72**	0.71	279, 280, 280	**0.72**	0.71	0.71	279
	ZG	Zesch and Gurevych (2006)	222	0.36	0.38	**0.39**	200, 207, 208	0.36	0.40	**0.41**	200
en	MC	Miller and Charles (1991)	30	**0.82**	0.77	0.80	30, 30, 30	**0.82**	0.77	0.80	30
	MEN	Bruni et al. (2014)	1000	0.72	0.73	**0.74**	1000, 1000, 1000	0.72	0.73	**0.74**	1000
	RG	Rubenstein et al. (1965)	65	**0.82**	0.79	0.79	65, 65, 65	**0.82**	0.79	0.79	65
	RW	Luong et al. (2013)	2034	**0.47**	**0.47**	**0.47**	1613, 1947, 1819	0.47	0.47	**0.48**	1613
	SL	Hill et al. (2014)	999	0.42	0.38	**0.43**	998, 999, 999	0.42	0.38	**0.43**	998
	WS	Finkelstein et al. (2002)	353	0.63	**0.64**	0.63	353, 353, 353	0.63	**0.64**	0.63	353
es	MC	Hassan and Mihalcea (2009)	30	0.70	0.69	**0.82**	30, 30, 30	0.70	0.69	**0.82**	30
	WS	Hassan and Mihalcea (2009)	352	0.54	0.54	**0.58**	350, 352, 352	0.54	0.54	**0.58**	350

Table 1: Word similarity results. The left part shows dataset information. The right part shows Spearman correlation (ρ) for the models with their *full vocabulary* and for the intersection of vocabularies. Coverage is shown for all models in order of appearance. Bold is best per vocabulary and row.

lang	set	all	a	n	v
cz	dev	9694	852	6436	2315
	test	9763	869	6381	2433
de	dev	51682	6347	40674	5018
	test	51827	6491	40623	5085
en	dev	44448	9713	30825	5661
	test	44545	9665	30736	5793
es	dev	12384	1711	8634	1989
	test	12476	1727	8773	1971
hu	dev	19387	1953	15268	2057
	test	19486	1928	15436	2011

Table 2: Number of lemmata in WordNet datasets

natural choice as corpus, because it is available for many languages. Therefore, we use the preprocessed and tokenized Wikipedia datasets of Müller and Schütze (2015). We train 50-dimensional skip-gram embeddings (Mikolov et al., 2013) with WORD2VEC on the original, the stemmed and the lemmatized corpus, respectively. Embeddings are trained for all tokens, because we need high coverage; the context size is set to 5, all remaining parameters are left at their default value.[9]

Results. The MRR results in the left half of Table 3 (“unfiltered”) show that for all languages and for all POS, *form real* has the worst performance among the form models. This comes at no surprise since this model does barely know anything about word forms and lemmata. The *form opt* model improves these results based on the additional information it has access to (the mapping from lemma to its most frequent form). *form sum* performs similar to *form opt*. For Czech, Hungarian and Spanish it is slightly better (or equally good), whereas for English and German there is no clear trend. There is a large difference between these two models on German nouns, with *form sum* performing considerably worse. We attribute this to the fact that many German noun forms are rare compounds and therefore lead to badly trained form embeddings, which summed up do not lead to high quality embeddings either.

Among the stemming models, *stem real* also is the worst performing model. We can further see that for all languages and almost all POS, *stem sum* performs worse than *stem opt*. That indicates that stemming leads to many low-frequency stems or many words sharing the same stem. This is especially apparent in Spanish verbs. There, the stemming models are clearly inferior to form models.

Overall, LAMB performs best for all languages and POS types. Most improvements of LAMB are

[9] We train smaller embeddings than before, because we have more models to train and training corpora are smaller.

significant. The improvement over the best form-model reaches up to 6 points (e.g., Czech nouns). In contrast to *form sum*, LAMB improves over *form opt* on German nouns. This indicates that the sparsity issue is successfully addressed by LAMB.

In general, morphological normalization in terms of stemming or lemmatization improves the result on all languages, leading to an especially substantial improvement on MRLs. For the morphologically very rich languages Czech and Hungarian, the relative improvement of STEM or LAMB to form-based models is especially high, e.g., Hungarian all: 50%. Moreover, we find that MRLs yield lower absolute performance. This confirms the findings of Köper et al. (2015). Surprisingly, LAMB yields better performance on English despite its simple morphology.

The low absolute results – especially for Hungarian – show that we address a challenging task and that our new evaluation methodology is a good evaluation for new types of word representations.

For further insight, we restrict the nearest neighbor search space to those lemmata that have the same POS as the query lemma. The general findings in the right half of Table 3 ("filtered") are similar to the unrestricted experiment: Normalization leads to superior results. The *form real* and *stem real* models yield the lowest performance. *Form opt* improves the performance and *form sum* is better on average than *form opt*. *Stem sum* can rarely improve on *stem opt*. The best stemming model most often is better than the best form model. LAMB can benefit more from the POS type restriction than the form models. The distance to the best form model generally increases, especially on German adjectives and Spanish verbs. In all cases except on English adjectives, LAMB yields the best performance. Again, in almost all cases LAMB's improvement over the form-models is significant.

4.3 Polarity Classification

Our first two evaluations were intrinsic. We now show the benefit of normalization on an extrinsic task. The task is classification of Czech movie reviews (CSFD, Habernal et al. (2013)) into positive, negative or neutral (Table 4). We reimplement lingCNN (Ebert et al., 2015), a Convolutional Neural Network that uses linguistic information to improve polarity classification. This model reaches close to state-of-the-art performance on data of the SemEval 2015 Task 10B (message level polarity). LingCNN takes several features as input: (i) embedding features, (ii) linguistic features at word level and (iii) linguistic features at review level.

We reuse the 50-dimensional Wikipedia embeddings from Section 4.2 and compare three experimental conditions: using forms, STEM and LAMB.

Linguistic word level features are: (i) SubLex 1.0 sentiment lexicon (Veselovská and Bojar, 2013) (two binary indicators that word is marked positive/negative); (ii) SentiStrength[10] (three binary indicators that word is an emoticon marked as positive/negative/neutral); (iii) prefix "ne" (binary negation indicator in Czech).[11]

All word level features are concatenated to form a single word representation of the review's input words. The concatenation of these representations is the input to a convolution layer, which has several filters spanning the whole representation height and several representations (i.e., several words) in width. The output of the convolution layer is input to a k-max pooling layer (Kalchbrenner et al., 2014). The max values are concatenated with the following linguistic review level features: (i) the count of elongated words, such as "cooool"; (ii) three count features for the number of positive/negative/neutral emoticons using the SentiStrength list; (iii) a count feature for punctuation sequences, such as "!!!"; (iv) and a feature that counts the number of negated words. (v) A final feature type comprises one count feature each for the number of sentiment words in a review, the sum of sentiment values of these words as provided by the sentiment lexicon, the maximum sentiment value and the sentiment value of the last word (Mohammad et al., 2013). The concatenation of max values and review level features is then forwarded into a fully-connected three-class (positive, negative, neutral) softmax layer. We train lingCNN with AdaGrad (Duchi et al., 2011) and early stopping, batch size = 100, 200 filters per width of 3-6; k-max pooling with $k = 5$; learning rate 0.01; and ℓ_2 regularization ($\lambda = 5 \cdot 10^{-5}$).

We also perform this experiment for English on

[10] `sentistrength.wlv.ac.uk/`

[11] We disregard words with the prefix "nej", because they indicate superlatives. Exceptions are common negated words with this prefix, such as "nejsi" (engl. "you are not").

		unfiltered							filtered						
		form			STEM				form			STEM			
lang	POS	real	opt	sum	real	opt	sum	LAMB	real	opt	sum	real	opt	sum	LAMB
cz	a	0.03	0.04	0.05	0.02	0.05	0.05	**0.06**	0.03‡	0.05†	0.07	0.04†	0.08	0.08	**0.09**
	n	0.15‡	0.21‡	0.24‡	0.18‡	0.27‡	0.26‡	**0.30**	0.17‡	0.23‡	0.26‡	0.20‡	0.29‡	0.28‡	**0.32**
	v	0.07‡	0.13‡	0.16†	0.08‡	0.14‡	0.16‡	**0.18**	0.09‡	0.15‡	0.17‡	0.09‡	0.17†	0.18	**0.20**
	all	0.12‡	0.18‡	0.20‡	0.14‡	0.22‡	0.21‡	**0.25**	-	-	-	-	-	-	-
de	a	0.14‡	0.22‡	0.25†	0.17‡	0.26	0.21‡	**0.27**	0.17‡	0.25‡	0.27‡	0.23‡	**0.33**	**0.33**	**0.33**
	n	0.23‡	0.35‡	0.30‡	0.28‡	0.35†	0.33‡	**0.36**	0.24‡	0.36‡	0.31‡	0.28‡	0.36	0.35‡	**0.37**
	v	0.11‡	0.19‡	0.18‡	0.11‡	0.22	0.18‡	**0.23**	0.13‡	0.20‡	0.21‡	0.13‡	0.24‡	0.23‡	**0.26**
	all	0.21‡	0.32‡	0.28‡	0.24‡	0.33†	0.30‡	**0.34**	-	-	-	-	-	-	-
en	a	0.22‡	0.25‡	0.24‡	0.16‡	0.26‡	0.25‡	**0.28**	0.25‡	0.28‡	0.28‡	0.18‡	0.29‡	**0.32**	0.31
	n	0.24‡	0.27‡	0.28‡	0.22‡	**0.30**	0.28‡	**0.30**	0.25‡	0.28‡	0.29‡	0.23‡	0.31†	0.31‡	**0.32**
	v	0.29‡	0.35‡	**0.37**	0.17‡	0.35	0.24‡	**0.37**	0.33‡	0.39‡	0.42‡	0.21‡	0.42†	0.39‡	**0.44**
	all	0.23‡	0.26‡	0.27‡	0.20‡	0.28‡	0.25‡	**0.29**	-	-	-	-	-	-	-
es	a	0.20‡	0.23‡	0.23‡	0.08‡	0.21‡	0.18‡	**0.27**	0.21‡	0.25‡	0.26‡	0.10‡	0.26‡	0.26‡	**0.30**
	n	0.21‡	0.25‡	0.25‡	0.16‡	0.25‡	0.23‡	**0.29**	0.22‡	0.26‡	0.27‡	0.17‡	0.27‡	0.26‡	**0.30**
	v	0.19‡	0.35†	0.36	0.11‡	0.29‡	0.19‡	**0.38**	0.22‡	0.36‡	0.36‡	0.16‡	0.36‡	0.33‡	**0.42**
	all	0.20‡	0.26‡	0.26‡	0.14‡	0.24‡	0.21‡	**0.30**	-	-	-	-	-	-	-
hu	a	0.02‡	0.06‡	0.06‡	0.05‡	0.08	0.08	**0.09**	0.04‡	0.08‡	0.08‡	0.06‡	**0.12**	0.11	**0.12**
	n	0.01‡	0.04‡	0.05‡	0.03‡	**0.07**	0.06‡	**0.07**	0.01‡	0.04‡	0.05‡	0.04‡	**0.07**†	0.06‡	**0.07**
	v	0.04‡	0.11‡	0.13‡	0.07‡	0.14‡	0.15	**0.17**	0.05‡	0.13‡	0.14‡	0.07‡	0.15‡	0.16†	**0.19**
	all	0.02‡	0.05‡	0.06‡	0.04‡	0.08‡	0.07‡	**0.09**	-	-	-	-	-	-	-

Table 3: Word relation results. MRR per language and POS type for all models. *unfiltered* is the unfiltered nearest neighbor search space; *filtered* is the nearest neighbor search space that contains only one POS. ‡ (resp. †): significantly worse than LAMB (sign test, $p < .01$, resp. $p < .05$). Best unfiltered/filtered result per row is in bold.

the SemEval 2015 Task 10B dataset (cf. Table 4). We reimplement Ebert et al. (2015)'s lexicon features. They exploit the fact that there are many more sentiment lexicons available in English. Other word level features are the same as above. Sentiment count features at review level are computed separately for the entire tweet, for all hashtag words and for each POS type (Ebert et al., 2015).

Considering the much smaller dataset size and shorter sentences of the SemEval data we chose the following hyperparameters: 100k most frequent word types, 100 filters per filter width of 2-5; and k-max pooling with $k = 1$.

Results. Table 5 lists the 10-fold cross-validation results (accuracy and macro F_1) on the CSFD dataset. LAMB/STEM results are consistently better than form results.

In our analysis, we found the following example for the benefit of normalization: "popis a název zajmavý a film je taková filmařská prasárna" (engl. "description and title are interesting, but it is bad film-making"). This example is correctly classified as negative by the LAMB model because it has an embedding for "prasárna" (bad, smut) whereas the form model does not.

The out-of-vocabulary counts for form and LAMB on the first fold of the CSFD experiment are 26.3k and 25.5k, respectively. The similarity of these two numbers suggests that the quality of word embeddings (form vs. LAMB) are responsible for the performance gain.

On the SemEval data, LAMB improves the results over form and stem (cf. Table 5).[12] Hence, LAMB can still pick up additional information despite the simple morphology of English. This is probably due to better embeddings for rare words. The SemEval 2015 winner (Hagen et al., 2015) is a highly domain-dependent and specialized system that we do not outperform.

In the introduction, we discussed that normalization removes inflectional information that is necessary for NLP tasks like parsing. For polarity classification, comparatives and superlatives can be important. Further analysis is necessary to deter-

[12] To be comparable with published results we report the macro F_1 of positive and negative classes.

dataset	total	pos	neg	neu
CSFD	91379	30896	29716	30767
SemEval train	9845	3636	1535	4674
SemEval dev	3813	1572	601	1640
SemEval test	2390	1038	365	987

Table 4: Polarity classification datasets

lang	features	acc	F_1
cz	Brychcin et al. (2013)	-	**81.53**
	form	80.86	80.75
	STEM	**81.51**	81.39
	LAMB	81.21	81.09
en	Hagen et al. (2015)	-	**64.84**
	form	66.78	62.21
	STEM	66.95	62.06
	LAMB	**67.49**	63.01

Table 5: Polarity classification results. Bold is best per language and column.

mine whether their normalization hurts in our experiments. However, note that we evaluate on polarity only, not on valence.

5 Analysis

Normalized embeddings deal better with sparsity than form embeddings. In this section, we demonstrate two additional benefits of LAMB based on its robustness against sparsity.

Embedding Size. We now show that LAMB can train embeddings with fewer dimensions on the same amount of data and still reach the same performance as larger form embeddings. We repeat the word relation experiments of Section 4.2 (all POS) and train all models with embedding sizes 10, 20, 30 and 40 for Spanish. We choose Spanish because it has richer morphology than English and more training data than Czech and Hungarian.

Figure 1 depicts the MRR results of all models with respect to embedding size. The relative ranking of form models is *real* < *opt* < *sum*. That comes from the additional information the more complex models have access to. All stemming models reach lower performance than their form counterparts (similar to results in Table 3). That suggests that stemming is not a proper alternative to correctly dealing with Spanish morphology. LAMB reaches higher performance than *form real* with already 20 dimensions. The 30 dimensional LAMB model is better than all other models. Thus, we can create lower-dimensional lemma embeddings that are as good as higher-dimensional form embeddings; this has the benefits of reducing the number of parameters in models using these embeddings and of reducing training times and memory consumption.

Corpus Size. Our second hypothesis is that less training data is necessary to train good embeddings. We create 10 training corpora consisting of the first k percent, $k \in \{10, 20, \ldots, 100\}$, of the randomized Spanish Wikipedia corpus. With these 10 subcorpora we repeat the word relation experiments of Section 4.2 (all POS). As query lemmata, we use the lemmata from before that exist in all subcorpora.

Figure 2 shows that the relative ranking among the models is the same as before. This time however, *form sum* yields slightly better performance than *form opt*, especially when little training data is available. The stemming models again are inferior to their form counterparts. Only *stem opt* is able to reach performance similar to *form opt*. LAMB always reaches higher performance than *form real*, even when only 10% of the training corpus is used. With 30% of the training corpus, LAMB surpasses the performance of the other models.[13] Again, by requiring less than 30% of the training data, embedding training becomes much more efficient. Furthermore, in low-resource languages that lack the availability of a large homogeneous corpus, LAMB can still be trained successfully.

6 Conclusion

We have presented STEM and LAMB, embeddings based on stems and lemmata. In three experiments we have shown the superiority compared to commonly used form embeddings. Especially (but not only) on MRLs, where data sparsity is a problem, both normalized embeddings perform better than form embeddings by a large margin. In a new challenging WordNet-based experiment we have shown four methods of adding morphological information

[13] Recall that *form opt* is similar to an approach that is used in most systems that have embeddings, which just use the available surface forms.

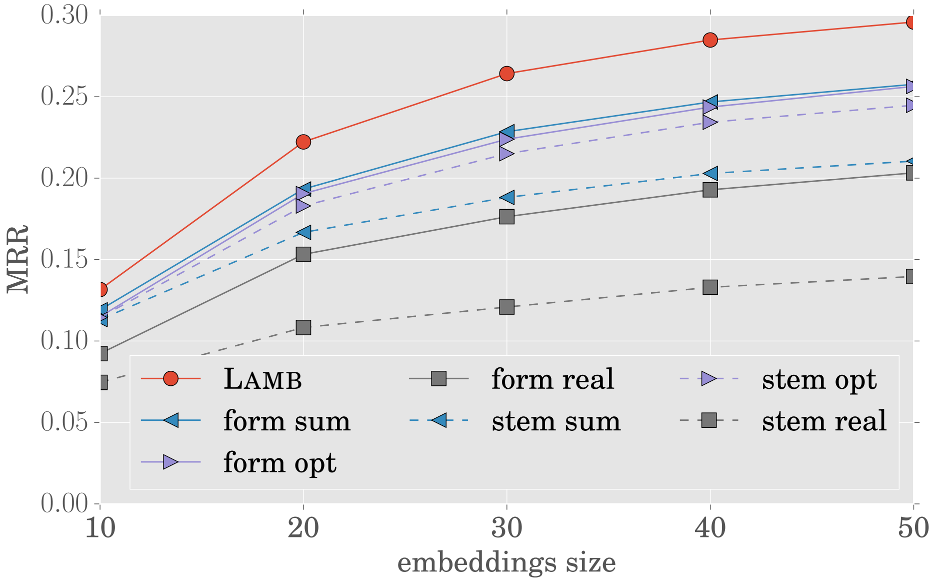

Figure 1: Embedding size analysis

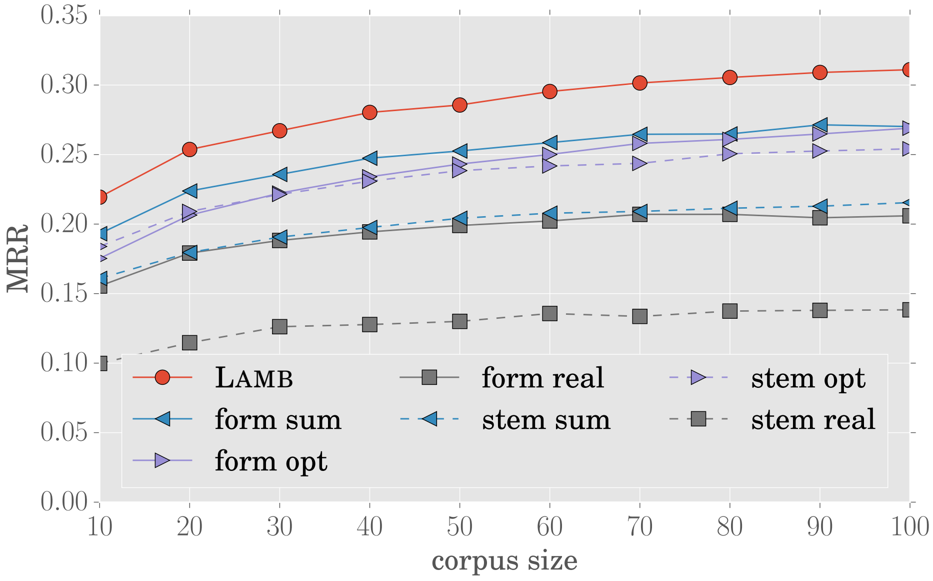

Figure 2: Corpus size analysis

(*opt*, *sum*, STEM, LAMB). Here, LAMB is the best of the proposed ways of using morphological information, consistently reaching higher performance, often by a large margin. STEM methods are not consistently better, indicating that the more principled way of normalization as done by LAMB is to be preferred. The datasets are available as supplementary material at `www.cis.uni-muenchen.de/ebert/`.

Our analysis shows that LAMB needs fewer embedding dimensions and less embedding training data to reach the same performance as form embeddings, making LAMB appealing for underresourced languages.

As morphological analyzers are becoming more widely available, our method – which is easy to implement, only requiring running the analyzer – should become applicable to more and more languages.

Acknowledgments This work was supported by DFG (grant SCHU 2246/10).

References

Marco Baroni and Sabrina Bisi. 2004. Using cooccurrence statistics and the web to discover synonyms in a technical language. In *Proceedings of LREC*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of ACL*.

Francis Bond and Kyonghee Paik. 2012. A Survey of Wordnets and their Licenses. In *Proceedings of the 6th Global WordNet Conference*.

Jan A. Botha and Phil Blunsom. 2014. Compositional Morphology for Word Representations and Language Modelling. In *Proceedings of ICML*.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research*, 49.

John A. Bullinaria and Joseph P. Levy. 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior Research Methods*, 44(3):890–907.

John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12.

Sebastian Ebert, Ngoc Thang Vu, and Hinrich Schütze. 2015. A Linguistically Informed Convolutional Neural Network. In *Proceedings of WASSA*.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. *ACM Trans. Inf. Syst.*, 20(1).

Gintarė Grigonytė, Joao Cordeiro, Gaël Dias, Rumen Moraliyski, and Pavel Brazdil. 2010. Paraphrase alignment for synonym evidence discovery. In *COLING*.

Iryna Gurevych. 2005. Using the Structure of a Conceptual Network in Computing Semantic Relatedness. In *Proceedings of IJCNLP*.

Ivan Habernal, Tomáš Ptáček, and Josef Steinberger. 2013. Sentiment Analysis in Czech Social Media Using Supervised Machine Learning. In *Proceedings of WASSA*.

Matthias Hagen, Martin Potthast, Michel Büchner, and Benno Stein. 2015. Webis: An Ensemble for Twitter Sentiment Detection. In *Proceedings of SemEval*.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL*.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet - a Lexical-Semantic Net for German. In *In Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Samer Hassan and Rada Mihalcea. 2009. Cross-lingual Semantic Relatedness Using Encyclopedic Knowledge. In *Proceedings of EMNLP*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *CoRR*, abs/1408.3456.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of ACL*.

Jussi Karlgren and Magnus Sahlgren. 2001. From Words to Understanding. In *Foundations of Real World Intelligence*. CSLI Publications.

Maximilian Köper, Christian Scheible, and Sabine Schulte im Walde. 2015. Multilingual Reliability and "Semantic" Structure of Continuous Word Spaces. In *Proceedings of IWCS*.

Ira Leviant and Roi Reichart. 2015. Judgment Language Matters: Multilingual Vector Space Models for Judgment Language Aware Lexical Semantics. *CoRR*, abs/1508.00106.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better Word Representations with Recursive Neural Networks for Morphology. In *Proceedings of CoNLL*.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational linguistics*.

Oren Melamud, Ido Dagan, Jacob Goldberger, Idan Szpektor, and Deniz Yuret. 2014. Probabilistic Modeling of Joint-context in Distributional Similarity. In *Proceedings of CoNLL*.

Márton Miháltz, Csaba Hatvani, Judit Kuti, György Szarvas, János Csirik, Gábor Prószéky, and Tamás Váradi. 2008. Methods and Results of the Hungarian WordNet Project. In *Proceedings of the 4th Global WordNet Conference*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of ICLR: Workshop*.

George A. Miller and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1).

Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets. In *Proceedings of SemEval*.

Thomas Müller and Hinrich Schütze. 2015. Robust morphological tagging with word representations. In *Proceedings of NAACL*.

Thomas Müller, Ryan Cotterell, Alexander Fraser, and Hinrich Schütze. 2015. Joint lemmatization and morphological tagging with Lemming. In *Proceedings of EMNLP*.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelRelate! A Joint Multilingual Approach to Computing Semantic Relatedness July 22-26, 2012, Toronto, Ontario, Canada. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Karel PALA and Pavel SMRZ. 2004. Building Czech Wordnet. *Romanian Journal of Information Science and Technology*, 7(1-2).

Marek Rei and Ted Briscoe. 2014. Looking for Hyponyms in Vector Space Language Learning, CoNLL 2014, Baltimore, Maryland, USA, June 26-27, 2014. In *Proceedings of CoNLL*.

Maria Ruiz-Casado, Enrique Alfonseca, and Pablo Castells. 2005. Using context-window overlapping in synonym discovery and ontology extension. In *Proceedings of RANLP*.

Roland Schäfer. 2015. Processing and querying large web corpora with the COW14 architecture. In *Proceedings of CMLC*.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, et al. 2013. Overview of the SPMRL 2013 shared task: Cross-Framework evaluation of parsing morphologically rich languages. In *Proceddings of SPMRL*.

Radu Soricut and Franz Josef Och. 2015. Unsupervised Morphology Induction Using Word Embeddings. In *Proceedings of NAACL-HLT*.

György Szarvas, Torsten Zesch, and Iryna Gurevych. 2011. Combining Heterogeneous Knowledge Resources for Improved Distributional Semantic Models. In *Proceedings of CICLing*.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of Word Vector Representations by Subspace Alignment. In *Proceedings of EMNLP*.

Peter D. Turney, Michael L. Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. *ACM Transactions on Information Systems*.

Peter D. Turney. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of ECML*.

Kateřina Veselovská and Ondřej Bojar. 2013. Czech SubLex 1.0.

Torsten Zesch and Iryna Gurevych. 2006. Automatically Creating Datasets for Measures of Semantic Relatedness. In *Proceedings of the Workshop on Linguistic Distances*.