Polyglot: Distributed Word Representations for Multilingual NLP

Rami Al-Rfou **Bryan Perozzi** **Steven Skiena**
Computer Science Dept. Stony Brook University Stony Brook, NY 11794
{ralrfou, bperozzi, skiena}@cs.stonybrook.edu

Abstract

Distributed word representations (word embeddings) have recently contributed to competitive performance in language modeling and several NLP tasks. In this work, we train word embeddings for more than 100 languages using their corresponding Wikipedias. We quantitatively demonstrate the utility of our word embeddings by using them as the sole features for training a part of speech tagger for a subset of these languages. We find their performance to be competitive with near state-of-art methods in English, Danish and Swedish. Moreover, we investigate the semantic features captured by these embeddings through the proximity of word groupings. We will release these embeddings publicly to help researchers in the development and enhancement of multilingual applications.

1 Introduction

Building multilingual processing systems is a challenging task. Every NLP task involves different stages of preprocessing and calculating intermediate representations that will serve as features for later stages. These stages vary in complexity and requirements for each individual language. Despite recent momentum towards developing multilingual tools (Nivre et al., 2007; Hajič et al., 2009; Pradhan et al., 2012), most of NLP research still focuses on rich resource languages. Common NLP systems and tools rely heavily on English specific features and they are infrequently tested on multiple datasets. This makes them hard to port to new languages and tasks (Blitzer et al., 2006).

A serious bottleneck in the current approach for developing multilingual systems is the requirement of familiarity with each language under consideration. These systems are typically carefully tuned with hand-manufactured features designed by experts in a particular language. This approach can yield good performance, but tends to create complicated systems which have limited portability to new languages, in addition to being hard to enhance and maintain.

Recent advancements in unsupervised feature learning present an intriguing alternative. Instead of relying on expert knowledge, these approaches employ automatically generated task-independent features (or word embeddings) given large amounts of plain text. Recent developments have led to state-of-art performance in several NLP tasks such as language modeling (Bengio et al., 2006; Mikolov et al., 2010), and syntactic tasks such as sequence tagging (Collobert et al., 2011). These embeddings are generated as a result of training "deep" architectures, and it has been shown that such representations are well suited for domain adaptation tasks (Glorot et al., 2011; Chen et al., 2012).

We believe two problems have held back the research community's adoption of these methods. The first is that learning representations of words involves huge computational costs. The process usually involves processing billions of words over weeks. The second is that so far, these systems have been built and tested mainly on English.

In this work we seek to remove these barriers to entry by generating word embeddings for over a hundred languages using state-of-the-art techniques. Specifically, our contributions include:

- **Word embeddings** - We will release word embeddings for the hundred and seventeen languages that have more than 10,000 articles on Wikipedia. Each language's vocabulary will contain up to 100,000 words. The embeddings will be publicly available at

(www.cs.stonybrook.edu/~dsl), for the research community to study their characteristics and build systems for new languages. We believe our embeddings represent a valuable resource because they contain a minimal amount of normalization. For example, we do not lower case words for European languages as other studies have done for English. This preserves features of the underlying language.

- **Quantitative analysis** - We investigate the embedding's performance on a part-of-speech (PoS) tagging task, and conduct qualitative investigation of the syntactic and semantic features they capture. Our experiments represent a valuable chance to evaluate distributed word representations for NLP as the experiments are conducted in a consistent manner and a large number of languages are covered. As the embeddings capture interesting linguistic features, we believe the multilingual resource we are providing gives researchers a chance to create multilingual comparative experiments.

- **Efficient implementation** - Training these models was made possible by our contributions to Theano (machine learning library (Bergstra et al., 2010)). These optimizations empower researchers to produce word embeddings under different settings or for different corpora than Wikipedia.

The rest of this paper is as follows. In Section 2, we give an overview of semi-supervised learning and learning representations related work. We then describe, in Section 3, the network used to generate the word embeddings and its characteristics. Section 4 discusses the details of the corpus collection and preparation steps we performed. Next, in Section 5, we discuss our experimental setup and the training progress over time. In Section 6 we discuss the semantic features captured by the embeddings by showing examples of the word groupings in multiple languages. Finally, in Section 7 we demonstrate the quality of our learned features by training a PoS tagger on several languages and then conclude.

2 Related Work

There is a large body of work regarding semi-supervised techniques which integrate unsupervised feature learning with discriminative learning methods to improve the performance of NLP applications. Word clustering has been used to learn classes of words that have similar semantic features to improve language modeling (Brown et al., 1992) and knowledge transfer across languages (Täckström et al., 2012). Dependency parsing and other NLP tasks have been shown to benefit from such a large unannotated corpus (Koo et al., 2008), and a variety of unsupervised feature learning methods have been shown to unilaterally improve the performance of supervised learning tasks (Turian et al., 2010). (Klementiev et al., 2012) induce distributed representations for a pair of languages jointly, where a learner can be trained on annotations present in one language and applied to test data in another.

Learning distributed word representations is a way to learn effective and meaningful information about words and their usages. They are usually generated as a side effect of training parametric language models as probabilistic neural networks. Training these models is slow and takes a significant amount of computational resources (Bengio et al., 2006; Dean et al., 2012). Several suggestions have been proposed to speed up the training procedure, either by changing the model architecture to exploit an algorithmic speedup (Mnih and Hinton, 2009; Morin and Bengio, 2005) or by estimating the error by sampling (Bengio and Senecal, 2008).

(Collobert and Weston, 2008) shows that word embeddings can almost substitute NLP common features on several tasks. The system they built, SENNA, offers part of speech tagging, chunking, named entity recognition, semantic role labeling and dependency parsing (Collobert, 2011). The system is built on top of word embeddings and performs competitively compared to state of art systems. In addition to pure performance, the system has a faster execution speed than comparable NLP pipelines (Al-Rfou' and Skiena, 2012).

To speed up the embedding generation process, SENNA embeddings are generated through a procedure that is different from language modeling. The representations are acquired through a model that distinguishes between phrases and corrupted versions of them. In doing this, the model avoids the need to normalize the scores across the vocabulary to infer probabilities. (Chen et al., 2013) shows that the embeddings generated by SENNA

Apple	apple	Bush	bush	corpora	dangerous
Dell	tomato	Kennedy	jungle	notations	costly
Paramount	bean	Roosevelt	lobster	digraphs	chaotic
Mac	onion	Nixon	sponge	usages	bizarre
Flex	potato	Fisher	mud	derivations	destructive

Table 1: Words nearest neighbors as they appear in the English embeddings.

perform well in a variety of term-based evaluation tasks. Given the training speed and prior performance on NLP tasks in English, we generate our multilingual embeddings using a similar network architecture to the one SENNA used.

However, our work differs from SENNA in the following ways. First, we do not limit our models to English, we train embeddings for a hundred and seventeen languages. Next, we preserve linguistic features by avoiding excessive normalization to the text. For example, our English model places "*Apple*" closer to IT companies and "*apple*" to fruits. More examples of linguistic features preserved by our model are shown in Table 1. This gives us the chance to evaluate the embeddings performance over PoS tagging without the need for manufactured features. Finally, we release the embeddings and the resources necessary to generate them to the community to eliminate any barriers.

Despite the progress made in creating distributed representations, combining them to produce meaning is still a challenging task. Several approaches have been proposed to address feature compositionality for semantic problems such as paraphrase detection (Socher et al., 2011), and sentiment analysis (Socher et al., 2012) using word embeddings.

3 Distributed Word Representation

Distributed word representations (word embeddings) map the index of a word in a dictionary to a feature vector in high-dimension space. Every dimension contributes to multiple concepts, and every concept is expressed by a combination of subset of dimensions. Such mapping is learned by back-propagating the error of a task through the model to update random initialized embeddings. The task is usually chosen such that examples can be automatically generated from unlabeled data (i.e so it is unsupervised). In case of language modeling, the task is to predict the last word of a phrase that consists of n words.

In our work, we start from the example construction method outlined in (Bengio et al., 2009). They train a model by requiring it to distinguish between the original phrase and a corrupted version of the phrase. If it does not score the original one higher than the corrupted one (by a margin), the model will be penalized. More precisely, for a given sequence of words $S = [w_{i-n} \dots w_i \dots w_{i+n}]$ observed in the corpus T, we will construct another corrupted sequence S' by replacing the word in the middle w_i with a word w_j chosen randomly from the vocabulary. The neural network represents a function $score$ that scores each phrase, the model is penalized through the hinge loss function $J(T)$ as shown in 1.

$$J(T) = \frac{1}{|T|} \sum_{i \in T} |1 - score(S') + score(S)|_+ \quad (1)$$

Figure 1 shows a neural network that takes a sequence of words with size $2n + 1$ to compute a score. First, each word is mapped through a vocabulary dictionary with the size $|V|$ to an index that is used to index a shared matrix C with the size $|V| * M$ where M is the size of the vector representing the word. Once the vectors are retrieved, they are concatenated into one vector called projection layer P with size $(2n + 1) * M$. The projection layer plays the role of an input to a hidden layer with size $|H|$, the activations A of which are calculated according to equation 3, where W_1, b_1 are the weights and bias of the hidden layer.

$$A = tanh(W_1 P + b_1) \quad (2)$$

To calculate the phrase score, a linear combination of the hidden layer activations A is computed using W_2 and b_2.

$$score(P) = W_2 A + b_2 \quad (3)$$

Therefore, the five parameters that have to be learned are W_1, W_2, b_1, b_2, C with a total number of parameters $(2n + 1) * M * H + H + H + 1 + |V| * M \approx M * (nH + |V|)$.

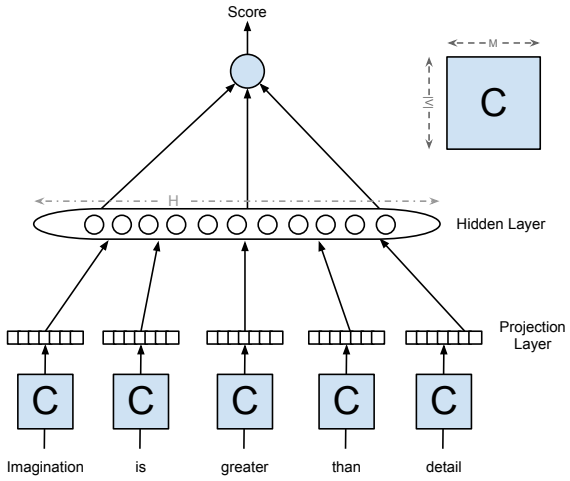

Figure 1: Neural network architecture. Words are retrieved from embeddings matrix C and concatenated at the projection layer as an input to computer the hidden layer activation. The score is the linear combination of the activation values of the hidden layer. The scores of two phrases are ranked according to hinge loss to distinguish the corrupted phrase from the original one.

4 Corpus Preparation

We have chosen to generate our word embeddings from Wikipedia. In addition to size, there are other desirable properties that we wish for the source of our language model to have:

- **Size and variety of languages** - As of this writing (April, 2013), 42 languages had more than 100,000 article pages, and 117 languages had more than 10,000 article pages.
- **Well studied** - Wikipedia is a prolific resource in the literature, and has been used for a variety of problems. Particularly, Wikipedia is well suited for multilingual applications (Navigli and Ponzetto, 2010).
- **Quality** - Wikipedians strive to write articles that are readable, accurate, and consist of good grammar.
- **Openly accessible** - Wikipedia is a resource available for free use by researchers
- **Growing** - As technology becomes more accessible, the size and scope of the multilingual Wikipedia effort continues to expand.

To process Wikipedia markup, we first extract the text using a modified version of the Bliki engine[1]. Next we must tokenize the text. We rely on an OpenNLP probabilistic tokenizer whenever possible, and default to the Unicode text segmentation[2] algorithm offered by Lucene when we have no such OpenNLP model. After tokenization, we normalize the tokens to reduce their sparsity. We have two main normalization rules. The first replaces digits with the symbol #, so "1999" becomes ####. In the second, we remove hyphens and brackets that appear in the middle of a token. As an additional rule for English, we map non-Latin characters to their unicode block groups.

In order to capture the syntactic and semantic features of words, we must observe each word several times in each of its valid contexts. This requirement, when combined with the Zipfian distribution of words in natural language, implies that learning a meaningful representation of a language requires a huge amount of unstructured text. In practice we deal with this limitation by restricting ourselves to considering the most frequently occurring tokens in each language.

Table 2 shows the size of each language corpus in terms of tokens, number of word types and coverage of text achieved by building a vocabulary out of the most frequent 100,000 tokens, $|V|$. Out of vocabulary (OOV) words are replaced with a special token $\langle$UNK$\rangle$.

While Wikipedia has 284 language specific encyclopedias, only five of them have more than a million articles. The size drops dramatically, such that the 42$^{\text{nd}}$ largest Wikipedia, Hindi, has slightly above 100,000 articles and the 100$^{\text{th}}$, Tatar, has slightly over 16,000 articles[3].

Significant Wikipedias in size have a word coverage over 92% except for German, Russian, Arabic and Czech which shows the effect of heavy usage of morphological forms in these languages on the word usage distribution.

The highest word coverage we achieve is unsurprisingly for Chinese. This is expected given the limited size vocabulary of the language - the number of entries in the Contemporary Chinese Dictionary are estimated to be 65 thousand words (Shuxiang, 2004).

[1] Java Wikipedia API (Bliki engine) - `http://code.google.com/p/gwtwiki/`

[2] `http://www.unicode.org/reports/tr29/`

[3] `http://meta.wikimedia.org/w/index.php?title=List_of_Wikipedias&oldid=5248228`

Language	Tokens $*10^6$	Words $*10^3$	Coverage
English	1,888	12,125	96.30%
German	687	9,474	91.78%
French	473	4,675	95.78%
Spanish	399	3,978	96.07%
Russian	328	5,959	90.43%
Italian	322	3,642	95.52%
Portuguese	197	2,870	95.68%
Dutch	197	3,712	93.81%
Chinese	196	423	99.67%
Swedish	101	2,707	92.36%
Czech	80	2,081	91.84%
Arabic	52	1,834	91.78%
Danish	44	1,414	93.68%
Bulgarian	39	1,114	94.35%
Slovene	30	920	94.42%
Hindi	23	702	96.25%

Table 2: Statistics of a subset of the languages processed. The second column reports the number of tokens found in the corpus in millions while the third column reports the word types found in thousands. The coverage indicates the percentage of the corpus that will be matching words in a vocabulary consists of the most frequent 100 thousand words.

5 Training

For our experiments, we build a model as the one described in Section 3 using Theano (Bergstra et al., 2010). We choose the following parameters, context window size $2n + 1 = 5$, vocabulary $|V| = 100,000$, word embedding size $M = 64$, and hidden layer size $H = 32$. The intuition, here, is to maximize the relative size of the embeddings compared to the rest of the network. This might force the model to store the necessary information in the embeddings matrix instead of the hidden layer. Another benefit is that we will avoid overfitting on the smaller Wikipedias. Increasing the window size or the embedding size slows down the training speed, making it harder to converge within a reasonable time.

The examples are generated by sweeping a window over sentences. For each sentence in the corpus, all unknown words are replaced with a special token ⟨UNK⟩ and sentences are padded with ⟨S⟩, ⟨/S⟩ tokens. In case the window exceeds the edges of a sentence, the missing slots are filled with our padding token, ⟨PAD⟩.

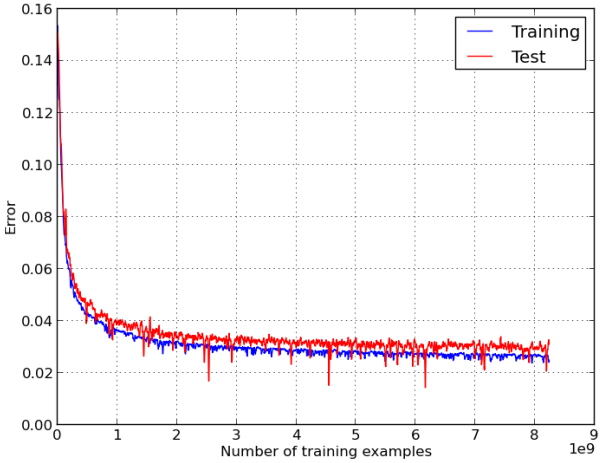

Figure 2: Training and test errors of the French model after 23 days of training. We did not notice any overfitting while training the model. The error curves are smoother the larger the language corpus is.

To train the model, we consider the data in minibatches of size 16. Every 16 examples, we estimate the gradient using stochastic gradient descent (Bottou, 1991), and update the parameters which contributed to the error using backpropagation (Rumelhart et al., 2002). Calculating an exact gradient is prohibitive given that the dataset size is in millions of examples. We calculate the development error by sampling randomly 10000 minibatches from the development dataset.

For each language, we set the batch size to 16 examples, and the learning rate to be 0.1. Following, (Collobert et al., 2011)'s advice, we divide each layer by the *fan in* of that layer, and we consider the embeddings layer to have a fan in of 1. We divide the corpus to three sets, training, development and testing with the following percentages 90, 5, 5 respectively.

One disadvantage of the approach used by (Collobert et al., 2011) is that there is no clear stopping criteria for the model training process. We have noticed that after a few weeks of training, the model's performance reaches the point where there is no significant decrease in the average loss over the development set, and when this occurs we manually stop the training. An interesting property of this model is that we did not notice any sign of overfitting for large Wikipedias. This could be explained by the infinite amount of examples we can generate by randomly choosing the re-

	Word	Translation
French	**rouge**	**red**
	juane	yellow
	rose	pink
	blanc	white
	orange	orange
	bleu	blue
Arabic	ارکش	**thanks**
	ارکشو	and thanks
	يتايحت	greetings
	ًارکش	thanks + diacritic
	ًارکشو	and thanks + diacritic
	ابحرم	hello
Russian	Путин	**Putin**
	Янукович	Yanukovych
	Троцкий	Trotsky
	Гитлер	Hitler
	Сталин	Stalin
	Медведев	Medvedev

	Word	Translation
Spanish	**dentista**	**dentist**
	peluquero	barber
	ginecólog	gynecologist
	camionero	truck driver
	oftalmólogo	ophthalmologist
	telegrafista	telegraphist
Arabic	نادلو	**two boys**
	نابذا	two sons
	نيدلو	two boys
	نلاضط	two children
	نينبا	two sons
	نانتبا	two daughters
Chinese	**Transliteration**	
	dongzhi	**Winter Solstice**
	chunfen	Vernal Equinox
	xiazhi	Summer solstice
	qiufen	Autumnal Equinox
	ziye	Midnight
	chuxi	New Year's Eve

	Word	Word
English	**Mumbai**	**Bombay**
	Chennai	Madras
	Bangalore	Shanghai
	Kolkata	Calultta
	Cairo	Bangkok
	Hyderabad	Hyderabad
German	**Eisenbahnbetrieb**	**rail operations**
	Fahrbetrieb	driving
	Reisezugverkehr	passenger trains
	Fährverkehr	ferries
	Handelsverkehr	Trade
	Schülerverkehr	students Transport
Italian	**papa**	**Pope**
	Papa	Pope
	pontefice	pontiff
	basileus	basileus
	canridnale	cardinal
	frate	friar

Table 3: Examples of the nearest five neighbors of every word in several languages. Translation is retrieved from `http://translate.google.com`.

placement word in the corrupted phrase. Figure 2 shows a typical learning curve of the training. As the number of examples have been seen so far increased both the training error and the development error go down.

6 Qualitative Analysis

In order to understand how the embeddings space is organized, we examine the subtle information captured by the embeddings through investigating the proximity of word groups. This information has the potential to help researchers develop applications that use such semantic and syntactic information. The embeddings not only capture syntactic features, as we will demonstrate in Section 4, but also demonstrate the ability to capture interesting semantic information. Table 3 shows different words in several languages. For each word on top of each list, we rank the vocabulary according to their Euclidean distance from that word and show the closest five neighboring words.

- **French & Spanish** - Expected groupings of colors and professions is clearly observed.
- **English** - The example shows how the embedding space is aware of the name change that happened to a group of Indian cities. "*Mumbai*" used to be called "*Bombay*", "*Chennai*" used to be called "*Madras* and "*Kolkata*" used to be called "*Calcutta*". On the other hand, "*Hyderabad*" stayed at a similar distance from both names as they point to the same conceptual meaning.
- **Arabic** - The first example shows the word "*Thanks*". Despite not removing the diacritics from the text, the model learned that the two surface forms of the word mean similar things and, therefore, grouped them together. In Arabic, conjunction words do not get separated from the following word. Usually, "*and thanks*" serves as a letter signature as "*sincerely*" is used in English. The model learned that both words {"*and thanks*", "*thanks*" } are similar, regardless their different forms. The second example illustrates a specific syntactic morphological feature of Arabic, where enumeration of couples has its own form.
- **German** - The example demonstrates that the compositional semantics of multi-unit words are still preserved.
- **Russian** - The model learned to group Russian/Soviet leaders and other figures related to the Soviet history together.
- **Chinese** - The list contains three solar terms that are part of the traditional East Asian lunisolar calendars. The remaining two terms correspond to traditional holidays that occur at the same dates of these solar terms.
- **Italian** - The model learned that the lower and upper cases of the word has similar meaning.

7 Sequence Tagging

Here we analyze the quality of the models we have generated. To test the quantitative performance of the embeddings, we use them as the sole features for a well studied NLP task, part of speech tagging.

To demonstrate the capability of the learned dis-

Language	Source	Test			TnT
		Unknown	Known	All	
German	Tiger† (Brants et al., 2002)	89.17%	**98.60%**	97.85%	98.10%
Bulgarian	BTB† (Simov et al., 2002)	75.74%	98.33%	96.33%	97.50%
Czech	PDT 2.5 (Bejček et al., 2012)	71.98%	**99.15%**	97.13%	99.10%
Danish	DDT† (Kromann, 2003)	73.03%	98.07%	**96.45%**	96.40%
Dutch	Alpino† (Van der Beek et al., 2002)	73.47%	95.85%	93.86%	95.00%
English	PennTreebank (Marcus et al., 1993)	75.97%	97.74%	**97.18%**	96.80%
Portuguese	Sint(c)tica† (Afonso et al., 2002)	75.36%	97.71%	95.95%	96.80%
Slovene	SDT† (Džeroski et al., 2006)	68.82%	95.17%	93.46%	94.60%
Swedish	Talbanken05† (Nivre et al., 2006)	83.54%	95.77%	**94.68%**	94.70%

Table 4: Results of our model against several PoS datasets. The performance is measured using accuracy over the test datasets. Third column represents the total accuracy of the tagger the former two columns reports the accuracy over known words and OOV words (unknown). The results are compared to the TnT tagger results reported by (Petrov et al., 2012).
†CoNLL 2006 dataset

tributed representations in extracting useful word features, we train a PoS tagger over the subset of languages that we were able to acquire free annotated resources for. We choose our tagger for this task to be a neural network because it has a fast convergence rate based on our initial experiments.

The part of speech tagger has similar architecture to the one used for training the embeddings. However we have changed some of the network parameters, specifically, we use a hidden layer of size 300 and learning rate of 0.3. The network is trained by minimizing the negative of the log likelihood of the labeled data. To tag a specific word w_i we consider a window with size $2n$ where n in our experiment is equal to 2. Equation 4 shows how we construct a feature vector F by concatenating ($\oplus$) the embeddings of the words occurred in the window, where C is the matrix that contains the embeddings of the language vocabulary.

$$F = \bigoplus_{j=i-2}^{i+2} C[w_j] \tag{4}$$

The feature vector will be fed to the network and the error will back propagated back to the embeddings.

The results of this experiment are presented in Table 4. We train and test our models on the universal tagset proposed by (Petrov et al., 2012). This universal tagset maps each original tag in a treebank to one out of twelve general PoS tags. This simplifies the comparison of classifiers performance across languages. We compare our results to a similar experiment conducted in their work, where they trained a TnT tagger (Brants, 2000) on several treebanks. The TnT tagger is based on Markov models and depends on trigram counts observed in the labeled data. It was chosen for its fast speed and (near to) state-of-the-art accuracy, without language specific tuning.

The performance of embeddings is competitive in general. Surprisingly, it is doing better than the TnT tagger in English and Danish. Moreover, our performance is so close in the case of Swedish. This task is hard for our tagger for two reasons. The first is that we do not add OOV words seen during training of the tagger to our vocabulary. The second is that all OOV words are substituted with one representation, ⟨UNK⟩ and there is no character level information used to inform the tagger about the characteristic of the OOV words.

On the other hand, the performance on the known words is strong and consistent showing the value of the features learned about these words from the unsupervised stage. Although the word coverage of German and Czech are low in the original Wikipedia corpora (See Table 2), the features learned are achieving great accuracy on the known words. They both achieve above 98.5% accuracy. It is noticeable that the Slovene model performs the worst, under both known and unknown words categories. It achieves only 93.46% accuracy on the test dataset. Given that the Slovene embeddings were trained on the least amount of data among all other embeddings we test here, we expect the quality to go lower for the other smaller Wikipedias not tested here.

In Table 5, we present how well the vocabulary of each language's embeddings covered the part of speech datasets. The datasets come from a different domain than Wikipedia, and this is reflected in the results.

In Table 6, we present the results of training the same neural network part of speech tagger without using our embeddings as initializations. We found that the embeddings benefited all the languages we considered, and observed the greatest benefit in languages which had a small number of training examples. We believe that these results illustrate the performance

Language	% Token Coverage	% Word Coverage
Bulgarian	94.58	77.70
Czech	95.37	65.61
Danish	95.41	80.03
German	94.04	60.68
English	98.06	79.73
Dutch	96.25	77.76
Portuguese	94.09	72.66
Slovene	95.33	83.67
Swedish	95.87	73.92

Table 5: Coverage statistics of the embedding's vocabulary on the part of speech datasets after normalization. Token coverage is the raw percentage of words which were known, while the Word coverage ignores repeated words.

Language	# Training Examples	Accuracy Drop
Bulgarian	200,049	-2.01%
Czech	1,239,687	-0.86%
Danish	96,581	-1.77%
German	735,826	-0.89%
English	950,561	-0.25%
Dutch	208,418	-1.37%
Portuguese	212,749	-0.91%
Slovene	27,284	-2.68%
Swedish	199,509	-0.82%

Table 6: Accuracy of randomly initialized tagger compared to our results. Using the embeddings was generally helpful, especially in languages where we did not have many training examples. The scores presented are the best we found for each language (languages with more resources could afford to train longer before overfitting).

8 Conclusion

Distributed word representations represent a valuable resource for any language, but particularly for resource-scarce languages. We have demonstrated how word embeddings can be used as off-the-shelf solution to reach near to state-of-art performance over a fundamental NLP task, and we believe that our embeddings will help researchers to develop tools in languages with which they have no expertise.

Moreover, we showed several examples of interesting semantic relations expressed in the embeddings space that we believe will lead to interesting applications and improve tasks as semantic compositionality.

While we have only considered the properties of word embeddings as features in this work, it has been shown that using word embeddings in conjunction with traditional NLP features can significantly improve results on NLP tasks (Turian et al., 2010; Collobert et al., 2011). With this in mind, we believe that the entire research community can benefit from our release of word embeddings for over 100 languages.

We hope that these resources will advance the study of possible pair-wise mappings between embeddings of several languages and their relations. Our future work in this area includes improving the models by increasing the size of the context window and their domain adaptivity through incorporating other sources of data. We will be investigating better strategies for modeling OOV words. We see improvements to OOV word handling as essential to ensure robust performance of the embeddings on real-world tasks.

Acknowledgments

This research was partially supported by NSF Grants DBI-1060572 and IIS-1017181, with additional support from TexelTek.

References

Susana Afonso, Eckhard Bick, Renato Haber, and Diana Santos. 2002. Floresta sintá (c) tica": a treebank for portuguese. In *Proc. of the Third Intern. Conf. on Language Resources and Evaluation (LREC)*, pages 1698–1703.

Rami Al-Rfou' and Steven Skiena. 2012. Speedread: A fast named entity recognition pipeline. In *Pro-*

ceedings of the 24th International Conference on Computational Linguistics (Coling 2012), pages 53–61, Mumbai, India, December. Coling 2012 Organizing Committee.

Eduard Bejček, Jarmila Panevová, Jan Popelka, Pavel Straňák, Magda Ševčíková, Jan Štěpánek, and Zdeněk Žabokrtský. 2012. Prague Dependency Treebank 2.5 – a revisited version of PDT 2.0. In *Proceedings of COLING 2012*, pages 231–246, Mumbai, India, December. The COLING 2012 Organizing Committee.

Yoshua Bengio and J-S Senecal. 2008. Adaptive importance sampling to accelerate training of a neural probabilistic language model. *Neural Networks, IEEE Transactions on*, 19(4):713–722.

Y. Bengio, H. Schwenk, J.S. Senécal, F. Morin, and J.L. Gauvain. 2006. Neural probabilistic language models. *Innovations in Machine Learning*, pages 137–186.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. 2009. Curriculum learning. In *International Conference on Machine Learning, ICML*.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June. Oral Presentation.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia.

Léon Bottou. 1991. Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nîmes 91*, Nimes, France. EC2.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The tiger treebank. In *IN PROCEEDINGS OF THE WORKSHOP ON TREEBANKS AND LINGUISTIC THEORIES*, pages 24–41.

Thorsten Brants. 2000. Tnt: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied natural language processing*, pages 224–231. Association for Computational Linguistics.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. 2012. Marginalized denoising autoencoders for domain adaptation. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pages 767–774. ACM, New York, NY, USA, July.

Yanqing Chen, Bryan Perozzi, Rami Al-Rfou', and Steven Skiena. 2013. The expressive power of word embeddings. *CoRR*, abs/1301.3226.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning, ICML*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November.

Ronan Collobert. 2011. Deep learning for efficient discriminative parsing. In *AISTATS*.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc Le, Mark Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Ng. 2012. Large scale distributed deep networks. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1232–1240.

Sašo Džeroski, Tomaž Erjavec, Nina Ledinek, Petr Pajas, Zdenek Žabokrtsky, and Andreja Žele. 2006. Towards a slovene dependency treebank. In *Proc. of the Fifth Intern. Conf. on Language Resources and Evaluation (LREC)*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the Twenty-eight International Conference on Machine Learning (ICML'11)*, volume 27, pages 97–110, June.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009), June 4-5*, Boulder, Colorado, USA.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474, Mumbai, India, December. The COLING 2012 Organizing Committee.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *In Proc. ACL/HLT*.

Matthias Trautner Kromann. 2003. The danish dependency treebank and the dtag treebank tool. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT)*, page 217.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

T. Mikolov, M. Karafiát, L. Burget, J. Cernocky, and S. Khudanpur. 2010. Recurrent neural network based language model. *Proceedings of Interspeech*.

Andriy Mnih and Geoffrey E Hinton. 2009. A scalable hierarchical distributed language model. *Advances in neural information processing systems*, 21:1081–1088.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Proceedings of the international workshop on artificial intelligence and statistics*, pages 246–252.

Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05: A swedish treebank with phrase structure and dependency annotation. In *Proceedings of the fifth International Conference on Language Resources and Evaluation (LREC)*, pages 1392–1395.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague, Czech Republic, June. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*, Jeju, Korea.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 2002. Learning representations by back-propagating errors. *Cognitive modeling*, 1:213.

Lu Shuxiang. 2004. *The Contemporary Chinese Dictionary (Xiandai Hanyu Cidian)*. Commercial Press.

Kiril Simov, Petya Osenova, Milena Slavcheva, Sia Kolkovska, Elisaveta Balabanova, Dimitar Doikoff, Krassimira Ivanova, Er Simov, and Milen Kouylekov. 2002. Building a linguistically interpreted corpus of bulgarian: the bultreebank. In *In: Proceedings of LREC 2002, Canary Islands*.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems 24*.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487. Association for Computational Linguistics.

J. Turian, L. Ratinov, and Y. 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*, pages 384–394. Association for Computational Linguistics.

Leonoor Van der Beek, Gosse Bouma, Rob Malouf, and Gertjan Van Noord. 2002. The alpino dependency treebank. *Language and Computers*, 45(1):8–22.