Human Language Technologies – The Baltic Perspective
I. Skadiņa and R. Rozis (Eds.)

doi:10.3233/978-1-61499-701-6-167

Word Embeddings for Latvian Natural Language Processing Tools

Artūrs ZNOTIŅŠ [1]
Institute of Mathematics and Computer Science, University of Latvia

Abstract. Word embeddings or distributed representations of words in a low dimensional vector space have been shown to capture both syntactic and semantic word relationships. Recently, multiple methods have been proposed to learn good word vector representations from very large text corpora effectively. Such word representations have been used to improve performance in a variety of natural language processing tasks. This work compares multiple methods to learn word embeddings for Latvian language and applies them to part of speech tagging, named entity recognition and dependency parsing tasks achieving state-of-the-art results for Latvian without resorting to any hand crafted and language specific features or resources such as gazetteers.

Keywords. Information extraction, word embeddings, *word2vec*

1. Introduction

It has been shown that usage of word embeddings improves performance on multiple natural language processing tasks by transferring word co-occurrence information from large text corpora that could not be learned from relatively small annotated corpora for specific supervised tasks.

This work presents initial results with inferring word embeddings on Latvian, a language with a much richer morphology compared to English that increases vocabulary size and embeds additional morphological linguistic regularities that may not be desirable for some tasks. Therefore, we experiment with lemmatized word embeddings and try to combine them with unlemmatized ones to achieve better performance on multiple downstream tasks.

The various trained word embeddings are compared in three Latvian NLP tasks: part of speech tagging, named entity recognition and dependency parsing. Obtained results are compared to results of existing tools that are not based on neural network architectures. Recurrent neural network architectures allow to combine word distributional information from pre-trained word embeddings and word character based representations to capture orthographic sensitivity allowing to achieve good performance without resorting to any hand crafted and task/language specific features or resources such as gazetteers.

[1] Corresponding Author: Artūrs Znotiņš, University of Latvia, Institute of Mathematics and Computer Science, Raiņa bulvāris 29, Riga, Latvia, LV-1459; E-mail: arturs.znotins@lumii.lv

2. Data Set

Word embedding models are trained on Latvian newswire text corpus (see Table 1). Training data is prepared using the following steps:

1. remove HTML tags and tables;
2. remove all punctuation and special characters;
3. replace all digits with zeroes;
4. tokenize text into sentences and words.

A lemmatized variant of the corpus is prepared using available morphological analyzer and tagger [1] to train lemmatized word embeddings.

Table 1. Latvian news corpus statistics after preprocessing.

Number of sentences	66.8M
Number of words	1.7B
Vocabulary size (at least 10 occurrences)	968,000
Vocabulary size after lemmatization (>10)	547,000

3. Linguistic Regularities

Having trained word embeddings we can search for words which are represented by similar vectors and plot two-dimensional PCA or t-SNE [2] projections to investigate more complex word relationships.

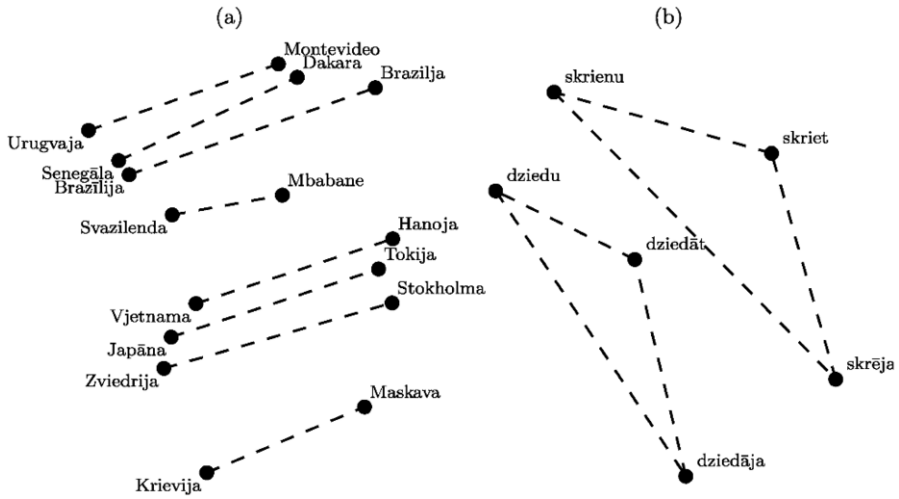

Figure 1. Two-dimensional PCA projections of Latvian word embeddings. Relationships between words are represented by a dashed line. (a) Relationship between vectors of countries and their capital cities. (b) Relationship between vectors of Latvian verb form inflections: infinitive, singular first person present-tense and third person past-tense forms of "to sing" and "to run".

Word embeddings capture relationships between word pairs that can be used to solve analogy queries, e.g. "Latvia is to Lithuania as Riga is to X" with desirable answer "Vilnius" (see Figure 1.a). Due to rich morphology in Latvian word embeddings include relationships between different inflections (see Figure 1.b).

Latvian word embeddings contain multiple types of word similarities:

- semantic;
- syntactic;
- inflections;
- spelling errors.

The most common problems seen in similarity query results are related to rare words and often together used words.

4. Experimental Setup

4.1. Word embedding models

Various state-of-the-art word embedding models are compared on multiple downstream tasks to find the the best model for each of these tasks.

CBOW and Skip-Gram (SG). The *word2vec* tool is fast and widely-used to produce two different word embedding models. SG model uses word's Huffman code as input to a log-linear classifier with a continuous projection layer, and it predicts surrounding words within given context window. CBOW model predicts a target word given its context words [3].

CWindow (CWIN) and Structured Skip-gram (SSG). [4] propose two simple modifications of CBOW and SG that account for word order information, achieving better performance in syntactically oriented downstream tasks.

Character based word embeddings (CWE). [5] propose to learn a single vector per character type and fixed set of parameters to combine them into word vectors yielding state-of-the-art results in language modeling and part-of-speech tagging. Such model should be beneficial for morphologically rich languages (e.g., Latvian).

4.2. Training corpus and parameters

To estimate how the corpus size effects the performance of word embeddings, Latvian newswire text corpus is subsampled to train word embeddings on corpora of different sizes.

Word embedding models are trained using different vector dimensions: 20, 50, 100, 200. For other hyperparameters default values are used as preliminary experiments showed that they already gave optimal results.

4.3. Tasks

Trained word embedding models are evaluated on three Latvian NLP tasks that have available manually annotated data (see Table 2) and already existing tools:

- a statistical morphological tagger which achieves 97.9% accuracy for part of speech recognition and 93.6% for the full morphological feature tag set that includes case, gender, number, person and more fine grained information [1];

- a syntactic parser [6] based on *MaltParser* and the hybrid dependency-based annotation model used in the Latvian Treebank, achieving 74.63% *UAS* (unlabeled attachment score);
- CRF-based named entity recognizer (NER) for person names, locations, organizations, achieving 84.6% F1-score [7].

Table 2. Available manually annotated Latvian datasets and their statistics: number of words/sentences.

Task	train	dev	test
Morphologically annotated corpus	88,600 (5,560)	11,500 (750)	8,000 (620)
Treebank	49,000 (3,900)	-	4,000 (220)
Universal dependencies	12,600 (670)	3,500 (190)	4,000 (220)
Named entity annotated corpus	44,000 (2,400)	-	-

For NER and POS tasks, neural architecture consisting of bidirectional long short-term memory (LSTM) with a sequential conditional random fields layer above it (LSTM-CRF) is used to combine orthographic representations of words learned from annotated corpora and pertained word embeddings [8, 9]. IOB (Inside, Outside, Beginning) tagging scheme is used to model named entities that span several tokens. NER is evaluated on three different types of named entities (locations, persons and organizations) using F1-score, best models are evaluated using 5-fold cross validation.

For dependency parsing task a continuous transition-based dependency parser based on LSTM is used to learn representations of parser state from learned orthographic representations of words and word embeddings [10, 11]. Word embedding models are compared on corpus of universal dependencies using UAS metric.

5. Results

Generally best results for all three tasks are achieved using SSG and CWIN word embeddings that are sensitive to word order (see Table 3). CWE models achieved significantly lower results than other models.

Lemma embeddings achieved lower results that could be caused by lemmatization errors of morphological analyzer and loss of information because of mixing all word inflections into one vector representation (this information could be useful for syntactically oriented tasks). Lemma embeddings can still be useful when combined with unlemmatized ones using vector averaging or concatenation. By using simple vector concatenation of SSG word embeddings and SG lemma embeddings the best performance is achieved on all three tasks. SG lemma embeddings seems to capture more semantic similarity that is useful in NER task when combining word embeddings.

From the results in Figure 2.a, we can conclude that using a larger training corpus yields better word embeddings. Especially in NER task, we need a large text corpus to train word embeddings to successfully capture semantic similarity between rare proper nouns.

Table 3. Results for different word embedding models and best achieved results by concatenating lemma embeddings with regular embeddings. Metrics used: accuracy (part of speech tagging), F1-score (named entity recognition) and unlabeled attachment score (parsing).

Model	POS (ACC, %)	NER (F1, %)	Parsing (UAS, %)
ssg_200	**98.3**	89.1	**74.9**
ssg_100	**98.3**	89.0	**74.9**
sg_100	97.5	86.0	72.6
cbow_100	97.5	85.9	72.6
cwin_100	**98.3**	88.9	74.8
cwe_100	96.9	74.6	68.2
sg_100_lem	97.3	84.5	71.2
avg(ssg_100, sg_100_lem)	**98.2**	86.5	72.8
concat(ssg_100, sg_100_lem)	**98.3**	**90.2**	**75.1**

Generally, 100-dimensional word embeddings achieve acceptable results, but larger dimensionality does help to improve the performance (in English a dimensionality of 50 typically is sufficient for NLP tasks [12]).

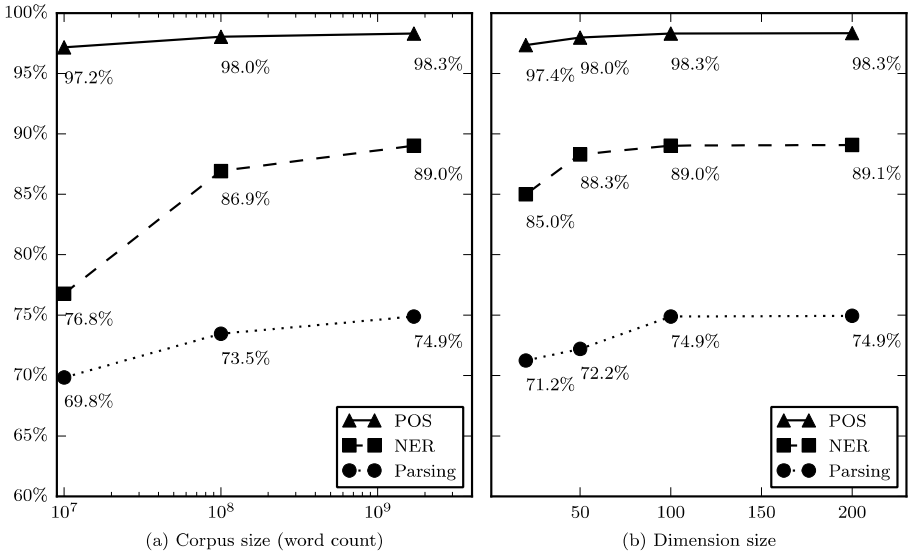

Figure 2. (a) Results using 100-dimensional SSG word embeddings trained on different size corpora (subsampled from Latvian newswire text corpus). (b) Dimensionality impact on results using SSG word embeddings.

In POS task LSTM-CRF tagger (98.3%) outperforms existing CMM tagger (97.7%). In morphological tagging with a simplified tagset (includes part of speech, case, number, gender, person and verb form mood; ~228 unique tags) LSTM-CRF achieves 94.6% accuracy. Increasing tagset granularity slows down tagger considerably because of CRF layer.

In NER task LSTM-CRF tagger significantly increases F1-score compared to currently used CRF tagger: from 83.4% to 90.5% (evaluated with 5-fold cross-validation on three named entity types). Learning curve (see Figure 3) shows that LSTM-CRF model with word embeddings achieves much better results compared to the existing CRF based NER for Latvian, especially if just a relatively small part of the annotated training corpus is used.

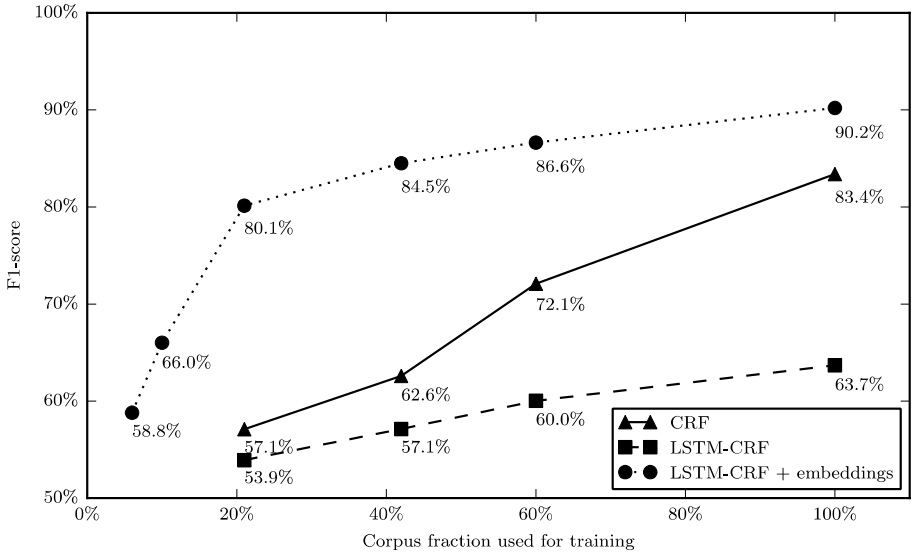

Figure 3. Learning curve using 5-fold cross-validation for NER task.

In dependency parsing LSTM based model (76.8%) outperforms the existing parser (75%) evaluated on the test set. In universal dependency parsing LSTM model achieves 75.1% precision using combined embedding model which is comparable to the best results reported for languages with relatively small treebanks (e.g., Turkish). Using part of speech tag as feature slightly increases performance to 75.4%. [11] argues that better performance could be achieved by just using learned orthographic word representations, but this was not observed for Latvian, possibly due to smaller training dataset.

6. Conclusion

In this paper multiple types of word embedding models are compared on three Latvian NLP tasks, achieving state-of-the-art results on all three of them. Which is a significant result considering used neural architectures do not rely on hand engineered features and gazetteers that increase difficulty of system maintenance and their adoption to other domains.

Overcoming problems related to rich morphology of Latvian is still a challenge. Lemmatization does not help in tasks included in this work, but it can improve results when combining regular word embeddings with lemma embeddings. Character based embeddings can help to generalize vectors of unseen words and to reduce model size, but achieved results are worse compared to word based embeddings.

References

[1] P. Paikens, L. Rituma and L. Pretkalniņa, Morphological analysis with limited resources: Latvian example, *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013),* 2013.
[2] L. van der Maaten and G. Hinton, Visualizing High-Dimensional Data Using t-SNE, *Journal of Machine Learning Research*, 2008.
[3] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, *NIPS*, 2013.
[4] W. Ling, C. Dyer, A. Black and I. Transcoso, Two/too simple adaptations of word2vec for syntax problems, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015.
[5] W. Ling, T. Luís, L. Marujo, R. Astudillo, S. Amir, C. Dyer and I. Trancoso, Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation, *CoRR, abs/1508.02096*, 2015.
[6] L. Pretkalniņa and L Rituma, Statistical syntactic parsing for Latvian, *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, 2013.
[7] A. Znotins and P. Paikens, Coreference Resolution for Latvian, *Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014)*, 2014.
[8] G. Lample, M. Ballesteros, K. Kawakami, S. Subramanian and C. Dyer, Neural Architectures for Named Entity Recognition. *Proceedings of NAACL 2016*, 2016.
[9] Z. Huang, W. Xu and K. Yu, Bidirectional LSTM-CRF Models for Sequence Tagging. *CoRR, abs/1508.01991,* 2015.
[10] C. Dyer, M. Ballesteros, W. Ling, A. Matthews and N. Smith, Transition-based Dependeny Parsing with Stack Long Short-Term Memory, *Proceedings of ACL*, 2015.
[11] M. Ballesteros, C. Dyer, and N. Smith, Improved Transition-Based Parsing by Modeling Characters instead of Words with LSTMs, *Proceedings of EMNLP 2015*, 2015.
[12] S. Lai, K. Liu, L. Xu and J. Zhao, How to Generate a Good Word Embedding? *CoRR, abs/1507.05523*, 2015.