Experiments with Neural Word Embeddings for Croatian

Leo Zuanović, Mladen Karan, Jan Šnajder

University of Zagreb, Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab
Unska 3, 10000 Zagreb, Croatia
{leo.zuanovic, mladen.karan, jan.snajder}@fer.hr

Abstract

Word representations extracted from a large corpus have been shown to be very useful in a variety of natural language processing tasks. Recently, there has been much work on using neural networks to learn good word representations from raw text. We adopt this approach and train neural word embeddings from a large Croatian web corpus. We evaluate the embeddings on three lexico-semantic tasks: synonym detection, semantic relatedness, and analogy modeling. Results on all three tasks are remarkably good and some of them markedly above the state-of-the-art results for Croatian. In particular, on the synonym detection and semantic relatedness tasks, the model achieves an accuracy of 73% and a correlation of 0.67 with human judgments, respectively.

Eksperimenti z nevronskimi vstavki besed za hrvaščino

Predstavitve besed, izluščene iz velikega korpusa, so se izkazale kot zelo koristne za raznovrstne naloge pri računalniški obravnavi naravnega jezika. V zadnjem času je bilo izvedenih veliko raziskav uporabe nevronskih mrež za učenje dobrih predstavitev besed iz neobdelanega besedila. V prispevku prevzamemo ta pristop in ga iz velikega korpusa hrvaščine naučimo nevronskih vstavkov besed. Vstavke evalviramo na treh leksikalnosemantičnih nalogah: detekciji sinonimov, semantični sorodnosti in modeliranju analogij. Rezultati na vseh treh nalogah so izredno dobri in nekateri bistveno boljši kot najboljši trenutni rezultati za hrvaščino. To še posebej velja za detekcijo sinonimov, kjer model doseže natančnost 73 %, ter za semantično sorodnost, ker model doseže korelacijo 0,67 s človeškimi odločitvami.

1. Introduction

In many natural language processing (NLP) tasks, model performance can be improved using word features induced from a large corpus, so-called *word representations*. A word representation is a mathematical object (typically a vector) associated with each word. *Distributional word representations* are derived from corpus-extracted co-occurrence matrix of words (rows) in some contexts (columns) (Turian et al., 2010). A number of design decisions have to be made when building such representations: the type and size of the context (e.g., a word window, a sentence, or document), how the counts are weighted (raw frequency, binary, tf-idf, etc.), and which dimensionality reduction technique to apply. Popular approaches include Latent Semantic Analysis (Deerwester et al., 1990), Random Indexing (Sahlgren, 2005), and Latent Dirichlet Allocation (Blei et al., 2003).

An alternative approach to word representations, which is gaining a lot of attention recently, is to learn a distributed representation in a supervised manner. Generally speaking, a *distributed representation* of a symbol is a vector of features, which characterize the meaning of the symbol while not being mutually exclusive, i.e., each of the features can be independently active or inactive, thus enabling the characterization of an exponential number of symbols (Bengio, 2008). In particular, distributed representations of words are called *word embeddings*, because the words are embedded into a dense, low-dimensional, real-valued vector space. The main idea is that functionally similar words will become close to each other after being embedded in this space.

In this paper, we experiment with word embeddings for Croatian using the recently proposed neural network-based models of Mikolov et al. (2013a). We evaluate these representations on three standard lexico-semantic tasks, namely synonym detection, semantic relatedness, and syntactic and semantic analogies. We show that the obtained word representations markedly outperform previous state-of-the-art results for Croatian.

2. Related work

Word embeddings are typically obtained as a by-product of training neural network-based language models (NNLMs). Language modeling is a classical NLP task of predicting the probability distribution over the "next" word, given some preceding words. In NNLMs, a sequence of words is first transformed into a sequence of word vectors via a projection matrix (weights between the input and the hidden layer), and then the network learns the probability distribution over these vectors. The advantage of using distributed representations is that they allow the model to generalize well to sequences that did not occur in the training set, but that are similar in terms of their features (i.e., their distributed representation), thus ameliorating the notorious data sparseness problem (Bengio, 2008).

NNLMs were first studied in the context of feed-forward networks (Bengio et al., 2003), and later in the context of recurrent neural network models (Mikolov et al., 2010; Mikolov et al.,). Computationally more efficient models were obtained by using hierarchical prediction (Morin and Bengio, 2005; Mnih and Kavukcuoglu, 2013a; Le et al., ; Mikolov et al., 2010; Mikolov et al.,).

Unlike the above-described architectures, which aim at learning good language models, the architectures described in (Mikolov et al., 2013a; Mikolov et al., 2013b) are primarily concerned with learning good word representations, and therefore are free to move away from the paradigm of predicting the target word from the previous words. Since these are the models we used in this work, we describe them in more detail in the next section.

3. CBOW and continuous skip-gram models

In the Continuous Bag of Words (CBOW) model (Mikolov et al., 2013a), the training objective is to learn distributed representations of the surrounding words (both the preceding and the succeeding ones), which, when combined, are good at predicting the intermediate target word. In the continuous skip-gram model (Mikolov et al., 2013a), on the other hand, the objective is to learn a distributed representation of the input word that is good at predicting its context in the same sentence.

These neural architectures are perhaps more easily understood as a log-linear classifier. Given a sequence of training words $w_1, w_2, \ldots, w_T$, the objective of the skip-gram model is to maximize the average log-probability:

$$\frac{1}{T}\sum_{t=1}^{T}\left[\sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t)\right]$$

where c is the size of the training context (window).[1] That is, the correct labels for the current word w_t are its surrounding words, w_{t+j}. In contrast, the CBOW model aims to maximize the probability $p(w_t|w_{t+j})$, i.e., the correct label for the surrounding words w_{t+j} is the intermediate word w_t.[2]

These models have two sets of word representations: one for the "input" words (w_{t+j} in the CBOW model and w_t in the skip-gram model) and one for the "output" (target) words (i.e., the words being predicted: w_t in CBOW and w_{t+j} in skip-gram model). These "input" representations are the ones we actually use for the semantic modeling of words. The conditional probabilities $p(w_t|w_{t+j})$ and $p(w_{t+j}|w_t)$ are defined as:

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\mathrm{T}} \cdot v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^{\mathrm{T}} \cdot v_{w_I}\right)}$$

where v_w and v'_w are the "input" and "output" vector representations of w, and W is the number of words in the vocabulary. However, this formulation is impractical because the cost of computing $\nabla \log p(w_O|w_I)$ is proportional to W, so a computationally efficient approximation of the full softmax – hierarchical softmax that uses a Huffman binary tree representation of the output layer – is used instead. Other methods used to speed-up the computation are Negative Sampling and subsampling of frequent words.

The models are trained by minimizing the negative log-likelihood using stochastic gradient descent. The gradient is computed using the well-known backpropagation rule (Rumelhart et al., 1988). Training can be performed on a large corpus in a short time (billions of words in hours). Mikolov et al. (2013a) have shown that skip-gram gives better word representations when the data is small, whereas the CBOW is faster and more suitable for larger datasets.

For details, refer to Mnih and Kavukcuoglu (2013b), who provide a good introduction to this type of models and describe a more general log-linear model.

4. Experimental setup

For training the CBOW and continuous skip-gram models, we used the publicly available `word2vec` implementation.[3] All models were trained on fhrWaC, a filtered version of Croatian web corpus described in (Ljubešić and Erjavec, 2011; Šnajder et al., 2013).[4] The corpus consists of 51M sentences and 1.2G tokens. All the words that occurred less than five times in the training data were discarded from the vocabulary, which resulted in a vocabulary of 1.4M words.

The parameters we varied are: the type of the model (CBOW or skip-gram), vector size, and the size of the context window. In what follows, we name the models to reflect their parameters (e.g., skip_100_5 is a skip-gram model with 100-dimensional vectors and a context window of at most five words). We used a hierarchical softmax in the output layer and subsampled frequent words with a threshold of 10^{-3}. The training times range from less than an hour for the CBOW model to several hours for the skip-gram model.

4.1. Task 1: Synonym detection

We evaluate the embeddings on a standard task from lexical semantics, namely synonym detection. We use the dataset created by Karan et al. (2012), with word choice questions for nouns, verbs, and adjectives (1000 questions each).[5] Each question consists of one target word with four synonym candidates, of which one is correct. The questions were extracted automatically from a machine readable dictionary of Croatian. For instance, *težak (husbandman, farmer): poljoprivrednik (agriculturalist, farmer), umjetnost (art), radijacija (radiation), bod (point)*. To make predictions, we compute pairwise cosine similarities of the target word vectors with the four candidates and predict the candidate(s) with maximum similarity.

We compare against the LSA-based synonym detection model of Karan et al. (2012), which uses 500 latent dimensions and paragraphs as contexts (LSA500P), and against a similar model that uses documents as context (LSA500D). We also compare against a Distributional Memory model of Šnajder et al. (2013), which is a state-of-the-art model on this task for Croatian.

4.2. Task 2: Semantic relatedness

For the semantic relatedness task, we use the dataset created by Janković et al. (2011), [6] containing 450 word pairs with human-annotated semantic relatedness judgments on a scale from 1 to 5. The annotations were made by 12 judges,

[1] The parameter c is actually the maximum window size. For each target word, a number R is drawn randomly from the $[1, c]$ range, and then R neighboring words to each side are taken.

[2] The surrounding words are not presented one-by-one, rather their vector representations are averaged and the resulting vector is used as the input to the classifier. We can consider this vector to be a predicted representation of the target (middle) word.

[3] `https://code.google.com/p/word2vec/`

[4] `http://takelab.fer.hr/data/fhrwac/`

[5] `http://takelab.fer.hr/data/crosyn/`

[6] `http://takelab.fer.hr/data/crosemrel450/`

out of which six with strongest agreement were selected and their scores averaged. For example, the pair *mlad (young) – star (old)* is assigned a score of 5.0, while the pair *utorak (Tuesday) – srijeda (Wednesday)* is assigned a score of 4.5.

As in the previous task, we use cosine as the similarity measure. We compare the computed similarities against the human judgments using Pearson's and Spearman's correlation coefficients. We use LSA500D as the baseline model.

4.3. Task 3: Syntactic and semantic analogies

Mimicking the experiments presented by Mikolov et al. (2013b), we also evaluate the embeddings on two analogy-based challenge sets. These consist of questions of the form "a is to b as c is to __", denoted as $a : b \rightarrow c : ?$. The task is to correctly predict the omitted fourth word, with only the exact word match deemed correct. Let $\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c$ be the corresponding word embeddings (all normalized to unit norm). Then the expected answer to $a : b \rightarrow c : ?$ is given by $\mathbf{y} = \mathbf{x}_b - \mathbf{x}_a + \mathbf{x}_c$. Of course, there might not exist a word at that exact position in the vector space, thus we search for a word w^* that is most similar to word y (excluding the input question words):

$$w^* = \arg\max_{w} \frac{\mathbf{x}_w \mathbf{y}}{\|\mathbf{x}_w\| \|\mathbf{y}\|}$$

We test the syntactic analogies on the task of finding the correct comparative form of an adjective. To build the dataset, we first selected 50 adjectives with frequent comparatives in the corpus. Next, out of those 50 adjectives, we selected 10 most common adjectives (and their comparatives), and for each we randomly selected 35 out of the 49 remaining pairs. Then, each of the 10 most common pairs is written down 35 times followed by the corresponding 35 pairs, yielding a total of 350 questions. An example item is *bogat (rich)* : *bogatiji (richer)* $\rightarrow$ *opasan (dangerous)* : ? [*opasniji (more dangerous)*]. The motivation for how the dataset was constructed is that the ten most common pairs will very well capture the "idea" of the comparative form.

To test the semantic analogy, we use the set of most common countries and their capitals obtained by translating the English version of the dataset created by Mikolov et al. (2013a).[7] The form is similar to the comparatives set, with one of the 23 pairs being repeated 22 times, each time followed by a different pair, resulting in 506 (i.e., 22 × 23) questions. For example, *Tokio (Tokyo)* : *Japan* $\rightarrow$ *Pariz (Paris)* : ? [*Francuska (France)*]. We make the analogies dataset freely available.[8]

As a baseline, we use the LSA500D model. These vectors were learned over a lemmatized corpus, hence there are no vector representations for the comparative forms.

5. Results

Our preliminary experiments have shown that network parameters (sizes of the layers) influence the results considerably, especially in the semantic analogies task. In this work we did not perform a systematic parameter optimization and we leave this for future work. Nonetheless, it should be noted that in most cases, even with the worst parameter settings, the neural network models still outperformed the simpler models by a considerable margin.

Model	N	A	V
Dm.Hr	70.0	66.3	63.2
LSA500P	67.2	68.9	61.0
LSA500D	60.0	60.8	50.7
skip_100_5	71.9	69.9	71.3
skip_200_5	73.4	71.9	74.1
skip_200_10	75.6	72.6	70.1
skip_500_5	75.5	**73.0**	**75.8**
skip_1000_10	**76.8**	72.7	72.2
cbow_100_5	61.7	69.3	69.0
cbow_100_10	62.5	67.3	64.9
cbow_200_5	66.2	70.6	72.1
cbow_200_10	64.7	67.8	68.6
cbow_500_5	66.9	70.3	72.8
cbow_1000_5	66.6	70.3	72.1
cbow_1000_10	29.8	25.9	27.6

Table 1: Results for the synonym detection task.

5.1. Task 1: Synonym detection

Table 1 shows the results for the considered models on nouns (N), adjectives (A), and verbs (V). Word embeddings outperform the baseline models across all considered parts of speech. continuous skip-gram models generally perform better than CBOW models. Overall, the biggest improvement over the baselines is achieved for verbs and the smallest for adjectives. This could be due to the fact that Croatian adjectives can have more than 40 different forms, which results in over 40 word embeddings for a single word, while for the evaluation we only consider a single vector – that of the word's lemma. It would be interesting to investigate whether better word representations for lemmas could be obtained by averaging the vectors of all the different forms of a word or by training the models over a lemmatized corpus.

Regardless of parameter setting, the neural network models outperform the state-of-the-art synonym detection model (Dm.Hr) from Šnajder et al. (2013).

5.2. Task 2: Semantic relatedness

The results for the semantic relatedness task are given in Table 2. Word embeddings markedly outperform the baseline. Skip-gram models again outperform CBOW. Spearman's coefficient is lower than Pearson's, indicating the presence of outliers. We also conducted experiments on the version of the set where all 12 judges are included. This, expectedly, decreases the results slightly (by 1–2 points).

All neural network models substantially outperform the LSA baseline. We could not compare against the Random Indexing model from Janković et al. (2011), because the authors did not use correlation coefficients for evaluation.

5.3. Task 3: Syntactic and semantic analogies

The performance of various CBOW and skip-gram models on the word analogy set is shown in Table 3. We have no baseline for comparative forms of adjectives, but selecting

[7] Available from `http://goo.gl/OR5W05`

[8] `http://takelab.fer.hr/data/croanalogy`

Model	Pearson	Spearman
LSA500D	0.438	0.225
skip_100_5	0.670	0.575
skip_200_5	0.665	0.600
skip_200_10	**0.677**	0.591
skip_500_5	0.673	0.573
skip_1000_10	0.649	**0.623**
cbow_100_5	0.533	0.438
cbow_100_10	0.501	0.432
cbow_200_5	0.570	0.468
cbow_200_10	0.537	0.453
cbow_500_5	0.576	0.504
cbow_1000_5	0.560	0.490
cbow_1000_10	0.466	0.351

Table 2: Results for the semantic relatedness task.

Model	Comparatives	Capitals
LSA500D	–	30.9
skip_100_5	36.6	13.4
skip_200_5	47.1	18.6
skip_200_10	**48.3**	28.8
skip_500_5	42.0	24.9
skip_1000_10	34.0	**35.7**
cbow_100_5	30.3	9.3
cbow_100_10	24.6	9.1
cbow_200_5	31.4	8.4
cbow_200_10	28.9	9.1
cbow_500_5	31.1	10.8
cbow_1000_5	23.4	12.3
cbow_1000_10	0	0

Table 3: Results for the word analogy task.

the right word out of 1M words in almost 50% of cases is a remarkable result. Skip-gram models again outperform CBOW. The cbow_1000_10 performs suspiciously poorly; we believe this may be due to a technical issue in the training procedure (e.g., insufficient training iterations).

6. Conclusion

Distributed word representations (aka word embeddings) have gained a lot of attention recently. We have built word embeddings for 1.4M Croatian words using CBOW and continuous skip-gram models. We evaluated the embeddings on three lexico-semantic tasks, showing a remarkable improvement in performance over the state of the art for Croatian. The skip-gram model outperformed the CBOW model.

For future work, we intend to investigate how various preprocessing steps (e.g., lemmatization) and properties of the corpus influence word representations. Another line of research is the application of word embeddings to tasks such as POS tagging and named entity recognition.

7. References

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Y. Bengio. 2008. Neural net language models. *Scholarpedia*, 3(1):3881.

D. M. Blei, A. Y. Ng, and M. I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. 1990. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.

V. Janković, J. Šnajder, and B. D. Bašić. 2011. Random indexing distributional semantic models for Croatian language. In *Text, Speech and Dialogue*, pages 411–418. Springer.

M. Karan, J. Šnajder, and B. D. Bašić. 2012. Distributional semantics approach to detecting synonyms in Croatian language. *Information Society, Proc. of the Eighth Language Technologies Conference*, pages 111–116.

H.-S. Le, I. Oparin, A. Allauzen, J. Gauvain, and F. Yvon. Structured output layer neural network language model. In *Proc. of ICASSP'11*.

N. Ljubešić and T. Erjavec. 2011. hrWaC and slWaC: Compiling web corpora for Croatian and Slovene. In *Text, Speech and Dialogue*, pages 395–402. Springer.

T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur. Extensions of recurrent neural network language model. In *Proc. of ICASSP'11*.

T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

A. Mnih and K. Kavukcuoglu. 2013a. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems 26*, pages 2265–2273.

A. Mnih and K. Kavukcuoglu. 2013b. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems*, pages 2265–2273.

F. Morin and Y. Bengio. 2005. Hierarchical probabilistic neural network language model. In *AISTATS*, volume 5, pages 246–252.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. 1988. Learning representations by back-propagating errors. *Cognitive modeling*.

M. Sahlgren. 2005. An introduction to random indexing. In *Methods and applications of semantic indexing, TKE*, volume 5.

J. Šnajder, S. Padó, and Ž. Agić. 2013. Building and evaluating a distributional memory for Croatian. In *Proc. of ACL 2013*, pages 784–789.

J. Turian, L. Ratinov, and Y. Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL 2010*, pages 384–394.