Predicting Croatian Phrase Sentiment Using a Deep Matrix-Vector Model

Siniša Biđin, Jan Šnajder, Goran Glavaš

University of Zagreb, Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab
Unska 3, 10000 Zagreb, Croatia
sinisa@bidin.cc, {jan.snajder, goran.glavas}@fer.hr

Abstract

Many sentiment analysis tasks rely on the existence of a sentiment lexicon. Such lexicons, however, typically contain single words annotated with prior sentiment. Problems arise when trying to model the sentiment of multiword phrases such as *"very good"* or *"not bad"*. In this paper, we use a recently proposed deep neural network model to classify the sentiment of phrases in Croatian. The experimental results suggest that reasonable classification of phrase-level sentiment for Croatian is achievable with such a model, reaching a performance comparable to that of an analogous model for English.

Napovedovanje sentimenta besednih zvez v hrvaščini z uporabo globinskega modela matrik vektorjev

Napovedovanje sentimenta besednih zvez v hrvaščini z uporabo globinskega modela matrik vektorjev Mnogo analiz sentimenta se zanaša na obstoj leksikona z informacijami o sentimentu. Vendar takšni leksikoni tipično vsebujejo samo posamezne besede, označene z vnaprejšnjim sentimentom. Problemi se pojavijo, ko bi želeli modelirati sentiment večbesednih enot, kot so »zelo dobro« ali »ni slabo«. V prispevku uporabimo pred kratkim predlagano globinsko nevronsko mrežo, s katero klasificiramo sentiment besednih zvez v hrvaščini. Eksperimentalni rezultati nakazujejo, da je s takim modelom mogoče doseči razmeroma dobro klasifikacijo besednih zvez glede na njihov sentiment, saj je delovanje modela primerljivo z analognim modelom za angleški jezik.

1. Introduction

The sentiment of a word, a phrase, or a document refers to its subjective attitude, polarity, or expression of feeling. The phrase *"nicely done"* has a positive, whereas *"horribly wrong"* has a negative sentiment. Sentiment analysis explores the ways of identifying or extracting sentiment from text. Applying methods of sentiment analysis on larger amounts of text, nowadays widely available on the web, allows us to do things such as attempt to judge the popularity of a product or predict the outcome of an election.

In this paper, we focus on classifying the sentiment of Croatian phrases consisting of two words. Given sentiment-labeled phrases such as *"very bad"*, *"not bad"*, and *"very good"*, we aim to train a model to correctly learn that *"bad"* bears a negative sentiment, and *"good"* a positive one. Also, the model should learn that *"very"* is an intensifier: it amplifies the sentiment of a word it is paired with. Likewise, *"not"* should be recognized as a negator, a word that inverts the sentiment of the word or a phrase it appears next to.

To learn the sentiment of Croatian bigrams, we employ a deep neural network model proposed by Socher et al. (2012). This model has shown to have good results when applied to the English language, which is something we aim to replicate for Croatian. We train and evaluate the deep neural model on two datasets of phrases, achieving performance comparable to the results obtained for English phrases.

2. Related work

This work is most closely related to two prominent areas of natural language processing: sentiment analysis and compositionality in vector spaces. Compositionality in vector spaces refers to the problem of learning a useful representation of a composition of multiple vector representations.

Focusing on compositionality, the model we use (Socher et al., 2012) is a generalization of earlier models. One model proposes vector composition through additive and multiplicative functions (Mitchell and Lapata, 2010), while another captures compositionality of words by linear combinations of nouns represented as vectors and adjectives as matrices (Baroni and Zamparelli, 2010). Finally, a general approach for sentiment analysis of phrases was laid out by Yessenalina and Cardie (2011), interesting also in that it introduces a model that uses matrices to represent words and matrix multiplication to compose them.

Another related work focusing also on sentiment analysis is the one by Socher et al. (2011), where predictions of sentence-level sentiment distributions are made using a recursive model that attempts to model sentiment via compositional semantics. Later models improve on this and achieve state-of-the-art results for the tasks of sentence-level sentiment classification (Socher et al., 2012; Socher et al., 2013), the first of which is the very model we are using here.

3. Training the matrix-vector model

To classify the phrase sentiment, we use the MV-RNN model proposed by Socher et al. (2012). This model can be applied by recursive operators to any n-gram, but we simplify it to the point where it only handles bigrams. The MV recursive neural network model derives its name from the matrix-vector representation of words. In essence, this means that each word w of a lexicon is modeled using two separate pieces of data: an n-dimensional vector $\mathbf{x}$ representing some semantic property of the word (such as sentiment) and an n-by-n matrix $\mathbf{X}$ representing the way the word influences the same semantic property of other words with which it constitutes a phrase.

$$w = (\mathbf{x}, \mathbf{X}), \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{X} \in \mathbb{R}^{n \times n}$$

Given an initial set of word MV-representations and some initial shared weights $\mathbf{W}$, all initialized to some (e.g.,

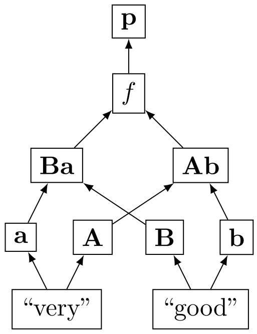


Figure 1: Two words, *"very"* and *"good"*, having MV-representations $(\mathbf{a}, \mathbf{A})$ and $(\mathbf{b}, \mathbf{B})$ respectively, affect each others' meaning (via $\mathbf{Ba}$ and $\mathbf{Ab}$) and combine using f to form a basis for phrase sentiment classification $\mathbf{p}$.

random) continuous values, in addition to a non-linear function g (e.g., a sigmoid), we can use a combining function f to determine the vector representation $\mathbf{p}$ of an entire phrase. This is depicted in Fig. 1. The function represents possible effects the two words have on each others sentiment by multiplying each one's matrix with the others vector.

$$\mathbf{p} = f(\mathbf{Ba}, \mathbf{Ab}) = g\left(\mathbf{W}\begin{bmatrix}\mathbf{Ba}\\ \mathbf{Ab}\end{bmatrix}\right), \quad \mathbf{W} \in \mathbb{R}^{n\times 2n}$$

We can then use the vector $\mathbf{p}$ to determine the sentiment of the phrase it represents. Instead of focusing on only two classes of sentiment (negative and positive), the model can predict a sentiment distribution over K classes. Applying the softmax function to $\mathbf{p}$ in combination with some weights $\mathbf{W}_{\text{class}}$, element-wise, gives us an estimate $\mathbf{d}$ of membership probability for each of the K sentiment classes:

$$\mathbf{d} = \operatorname{softmax}\left(\mathbf{W}_{\text{class}}\,\mathbf{p}\right), \quad \mathbf{W}_{\text{class}} \in \mathbb{R}^{K\times n}, \quad \mathbf{d} \in \mathbb{R}^{K}$$

$$\operatorname{softmax}_i(\mathbf{z}) = \frac{e^{\mathbf{z}_i}}{\sum_{i\neq j}^{n} e^{\mathbf{z}_j}}$$

To determine the amount of error between the reference and predicted sentiment probability distributions, $\mathbf{y} \in \boldsymbol{Y}$ and $\mathbf{d}$, respectively, we compute the binary cross entropy errors for each of the K classes. The loss function J is simply the mean error across all training instances:

$$E(\mathbf{y}, \mathbf{d}) = -\frac{1}{K}\sum_{i=1}^{K}\left(\mathbf{y_i}\ln\left(\mathbf{d_i}\right) + \left(1-\mathbf{y_i}\right)\ln\left(1-\mathbf{d_i}\right)\right)$$

$$J = \frac{1}{N}\sum_{i=1}^{N} E\left(\boldsymbol{Y}^{(i)}, \mathbf{d}^{(i)}\right)$$

While the initial vector components $\mathbf{x}$ of all the word MV-representations could be initialized to random values, we can also pretrain them, which has been shown to be beneficial for many tasks (Erhan et al., 2010). Following these insights, we initialize the vectors to word embeddings produced by *word2vec*,[1] an implementation of the skip-gram model by Mikolov et al. (2013), trained on the fHrWaC[2] corpus (Šnajder et al., 2013; Ljubešić and Erjavec, 2011).

Similarly, we set all the initial word matrix components $\mathbf{X}$ to the identity matrix, adding a small amount of noise. Since $\mathbf{X} \approx \mathbf{I}$, it ceases to have an effect on the sentiment of a word when multiplied with that word's vector, as in the definition of function f. This ensures that words by default do not function as operators; they neither intensify, attenuate, nor flip the sentiment of the words they are paired with.

The model's total number of parameters equals $2n^2 + Kn + L(n + n^2)$, corresponding to sizes of $\mathbf{W}$, $\mathbf{W}_{\text{class}}$, and the MV-representations of all L words in the lexicon. We optimize these parameters by minimizing J with stochastic gradient descent, using a starting learning rate of $\alpha = 0.1$ and diminishing it linearly towards zero. Due to the large space complexity ($O(Ln^2)$), there are practical restrictions on the value of n. However, it has been shown that setting n to larger values (larger than 11) does not improve the performance (Socher et al., 2012).

4. Evaluation

We evaluate the model on two different datasets of phrases:[3] (1) a synthetic dataset where phrases have been assembled and their sentiment distributions labeled manually and (2) a dataset of manually translated common phrases extracted from movie reviews in English.

Since movies are commonly rated on a scale of 1 to 10, and indeed our source for the second dataset uses that very same rating scheme, we will be classifying phrases into K=10 sentiment classes that each correspond to a particular rating ranging from 1 (the worst) to 10 (the best). Additionally, we will use the same model trained for K=10 classes and apply it to classification of sentiment into K=3 classes.

4.1. Datasets

The datasets consist of unique two-word phrases paired with their sentiment distributions over a certain number K of classes. It should be noted that a reference sentiment distribution is never assigned to an individual word but exclusively to phrases. Each phrase occurs only once in a dataset, but an individual word may occur multiple times, as a part of different phrases (e.g., *"good"*).

Synthetic dataset. The first set consists of 1500 different phrases composed of Croatian words, assembled by pairing each of the 25 different adverbs with each of the 60 different adjectives. The set is divided into 1200 training phrases and 300 test phrases. Each of the phrases is manually labeled by a probability distribution over the K=10 sentiment classes, determined subjectively by a single author considering the phrase outside of context. None of the phrases have been labeled with ambiguous sentiment, meaning their sentiment probability distributions contain only one single maximum.

Movie reviews dataset. The second dataset is based on a publicly available dataset of bigrams extracted from movie reviews written in English.[4] Each of the phrases is associated with its frequency of occurrence within reviews with each of 10 different possible ratings. Note that here we

[1] https://code.google.com/p/word2vec/

[2] http://takelab.fer.hr/data/fhrwac/

[3] Datasets are available from http://takelab.fer.hr/data/crophrasesent

[4] http://compprag.christopherpotts.net/iqap-experiments.html

assume a correlation between a review's rating and the sentiment of phrases expressed within it, and so use the frequencies of occurrence to construct for each unique phrase a probability distribution over K=10 sentiment classes. Such a simplistic assumption might not hold in all cases (e.g., a positive phrase might, for whatever reason, appear often in negatively scored reviews and vice versa). Each phrase that occurred in total at least 300 times was manually translated into Croatian by a single annotator using his subjective judgment. The translated phrases are then compiled into a dataset consisting of 1026 different phrases containing 208 unique words. The dataset is divided into a training set consisting of 821 and a test set consisting of 205 instances.

4.2. Results and discussion

We evaluate the MV-RNN model for several different sizes of the word vector (n = 8, 10, 13, and 15). We present the results using two different measures: (1) the F1-score and (2) the mean Kullback-Leibler divergence (KL-divergence). The KL-divergence measures the (dis)similarity between the reference and predicted probability distributions $\mathbf{y}$ and $\mathbf{d}$, respectively:

$$\mathrm{KL}(\mathbf{y},\mathbf{d}) = \sum_i \mathbf{y}_i \ln \frac{\mathbf{y}_i}{\mathbf{d}_i}$$

We compute two F1-scores: (1) for K=10 classes and (2) for K=3 classes (the *positive*, *negative*, and *neutral* class). The F1-score for the K=3 case is derived from the results of the K=10 case, by splitting the sentiment probability distribution into three ranges ($1 \leq$ negative ≤ 3; $4 \leq$ neutral ≤ 7; $8 \leq$ positive ≤ 10), for which we sum the probabilities assigned to individual scores. Such binning allows us to evaluate the model in a commonly used *negative/neutral/positive* sentiment classification setting.

For the K=3 classification setting, we compare the MV-RNN against two baselines: a simple sentiment lexicon-based model (SentiLex) and a support vector machine (SVM) model. The SentiLex model assigns a positive (+1), negative (−1), or neutral (0) score to each word in a phrase, and then simply sums up these polarities. The SVM model is trained on a concatenation of two word vectors as features, either two one-hot vectors ($\mathrm{SVM}_{\text{1-hot}}$) or two 100-dimensional pretrained vectors ($\mathrm{SVM}_{\text{Pre}}$).

The evaluation results for the synthetic and movie review dataset are given in Tables 1 and 2, respectively. The MV-RNN models perform very well on the synthetic dataset, clearly outperforming the baselines. However, good performance on this dataset should come as no surprise, because the dataset is (1) very *clean* – there is no sentiment ambiguity (e.g., one phrase having high probabilities for both positive and negative scores) and (2) each word occurs in the dataset paired with every other and is found within different phrases many times. Individual words in real datasets will occur much less frequently. Reference and predicted probability distributions for four example phrases from the synthetic dataset are depicted in Fig. 2.

On the more realistic move reviews dataset, with significantly more sentiment ambiguity and a smaller number of occurrences of single words, the model performs worse than on the synthetic set. The performance is, nonetheless, well above the baselines for K=3, and comparable to the performance achieved by the same model for English (Socher et al., 2012). Example reference and predicted probability distributions are depicted in Fig. 3.

		F1-score		
	n	K=3	K=10	KL
SentiLex	–	43.0	–	–
$\mathrm{SVM}_{\text{1-hot}}$	85	83.9	–	–
$\mathrm{SVM}_{\text{Pre}}$	100	91.8	–	–
$\mathrm{MV\text{-}RNN}_{\text{Rand}}$	8	93.0	63.1	0.025
$\mathrm{MV\text{-}RNN}_{\text{Rand}}$	10	90.1	71.6	0.025
$\mathrm{MV\text{-}RNN}_{\text{Rand}}$	13	92.7	70.0	**0.021**
$\mathrm{MV\text{-}RNN}_{\text{Rand}}$	15	**93.1**	69.6	**0.021**
$\mathrm{MV\text{-}RNN}_{\text{Pre}}$	8	91.2	68.7	0.026
$\mathrm{MV\text{-}RNN}_{\text{Pre}}$	10	92.8	**76.4**	0.023
$\mathrm{MV\text{-}RNN}_{\text{Pre}}$	13	91.2	74.6	0.024
$\mathrm{MV\text{-}RNN}_{\text{Pre}}$	15	92.4	74.8	0.023

Table 1: Results for the **synthetic** dataset, using random (Rand) and pretrained (Pre) initial vectors.

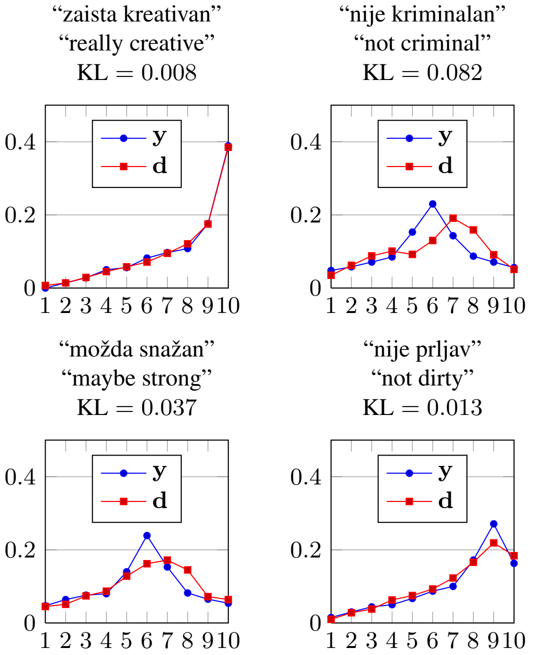
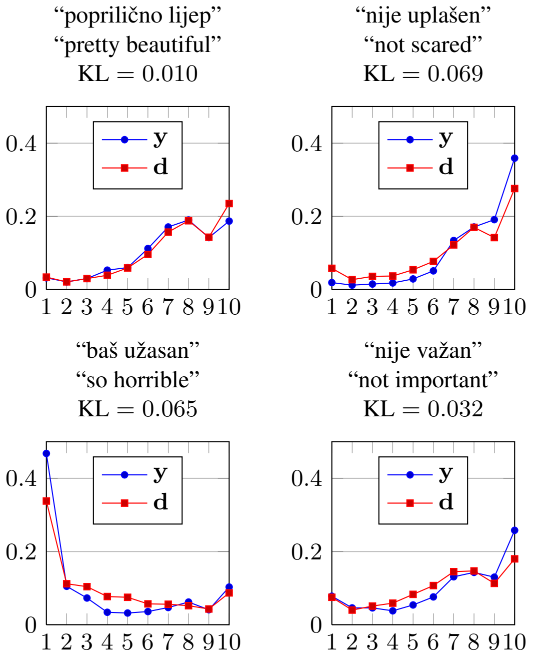


Figure 2: Results for selected phrases from the **synthetic** test set. The x-axis shows the K=10 sentiment classes, while the y-axis shows the sentiment probability distribution (the probability of the phrase belonging to a specific sentiment class). Reference sentiment probability distributions are shown in blue and classifier predictions in red.

It is apparent from the results that the model can correctly capture the way words can intensify, attenuate, or flip entirely the sentiment inherent in words they are paired with. A lower performance on the movie reviews dataset may perhaps be traced down to the assumption upon which the reference distributions were created: phrases are negative if they more frequently occur in generally negative reviews and positive if they more frequently occur in positive reviews. However, an unambiguously negative phrase still may occur

		F1-score		
	n	K=3	K=10	KL
SentiLex	–	45.2	–	–
SVM$_{\text{1-hot}}$	134	63.8	–	–
SVM$_{\text{Pre}}$	100	61.2	–	–
MV-RNN$_{\text{Rand}}$	8	68.9	34.6	0.055
MV-RNN$_{\text{Rand}}$	10	67.2	36.5	0.055
MV-RNN$_{\text{Rand}}$	13	**69.2**	34.9	0.056
MV-RNN$_{\text{Rand}}$	15	67.8	40.8	**0.054**
MV-RNN$_{\text{Pre}}$	8	63.7	33.3	0.065
MV-RNN$_{\text{Pre}}$	10	67.6	38.3	0.066
MV-RNN$_{\text{Pre}}$	13	64.3	**43.1**	0.067
MV-RNN$_{\text{Pre}}$	15	67.7	37.1	0.066

Table 2: Results for the **movie reviews** dataset, using random (Rand) and pretrained (Pre) initial vectors.



"prilično lijep"
"pretty beautiful"
KL = 0.010

"nije uplašen"
"not scared"
KL = 0.069

"baš užasan"
"so horrible"
KL = 0.065

"nije važan"
"not important"
KL = 0.032

Figure 3: Results similar to those from Fig. 2, but for chosen phrases from the **movie reviews** test set.

in an otherwise very positive review with a high rating, and vice-versa. Similarly, a phrase may be ambiguous in that it can be used in both positive and negative contexts. These ambiguities are likely to affect the model's performance.

Surprisingly, pretraining the word vectors does not improve the performance. Moreover, in some cases having word vectors pretrained actually degrades performance. This is likely due to the fact that pretraining serves to learn the semantic meaning of the words, which may often conflict with their sentiment. For example, two antonyms will, after pretraining, have similar word vector representations, but their sentiment is directly opposite (e.g., *"better"* vs. *"worse"*).

5. Conclusion

While lexicons of prior sentiment are useful in many sentiment analysis tasks, multiword phrases often have a sentiment different from the prior sentiment of their constituent words. In this paper we used a deep neural network model proposed by Socher et al. (2012) to learn the sentiment of two-word Croatian phrases. We evaluated the model on two different datasets: one synthetic and the other realistic. Experimental results suggest that deep learning models are well-suited for the task of modeling the sentiment of Croatian phrases, confirming previous results for English.

We have not exploited the key capability of the MV-RNN model: the recursive application to arbitrary length n-grams, which has been shown to be very effective for modeling the sentiment of complete sentences (Socher et al., 2013). We intend to pursue this line of work and experiment with predicting the sentiment of complete sentences in Croatian.

6. References

M. Baroni and R. Zamparelli. 2010. Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space. In *Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193. ACL.

D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. 2010. Why does unsupervised pretraining help deep learning? *Journal of Machine Learning Research*, 11:625–660.

N. Ljubešić and T. Erjavec. 2011. hrWaC and slWac: compiling web corpora for Croatian and Slovene. In *In Proc. of Text, Speech and Dialogue 2011*, Lecture Notes in Computer Science, pages 395–402. Springer.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

J. Mitchell and M. Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

J. Šnajder, S. Padó, and Ž. Agić. 2013. Building and evaluating a distributional memory for Croatian. In *51st Annual Meeting of the Association for Computational Linguistics*, pages 784–789. ACL.

R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. ACL.

R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. ACL.

R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. ACL.

A. Yessenalina and C. Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 172–182. ACL.