Word Embeddings for Multi-label Document Classification

Ladislav Lenc[†‡]
[‡] NTIS – New Technologies for the Information Society, University of West Bohemia, Plzeň, Czech Republic
llenc@kiv.zcu.cz

Pavel Král[†‡]
[†] Department of Computer Science and Engineering, University of West Bohemia, Plzeň, Czech Republic
pkral@kiv.zcu.cz

Abstract

In this paper, we analyze and evaluate word embeddings for representation of longer texts in the multi-label document classification scenario. The embeddings are used in three convolutional neural network topologies. The experiments are realized on the Czech ČTK and English Reuters-21578 standard corpora. We compare the results of word2vec static and trainable embeddings with randomly initialized word vectors. We conclude that initialization does not play an important role for classification. However, learning of word vectors is crucial to obtain good results.

1 Introduction

Text classification (or categorization) is one of the core tasks in natural language processing (NLP) field. The applications of text classification are numerous as for instance sentiment analysis, automatic categorization of e-mails or spam filtering. The main goal of document classification is to assign one or more labels to a given document. The categorization then helps the users to find appropriate documents. Nowadays, computers are heavily utilized for this task and can save a great amount of human labor.

In this paper, we concentrate on multi-label document classification which means that one document can belong to more classes simultaneously. More formally, given a set of documents D and a set of all possible labels C we create a model that assigns a set $C_d \subset C$ to the document $d \in D$.

Multi-label classification is often solved using an ensemble of binary classifiers (Tsoumakas and Katakis, 2006). However, nowadays, neural nets outperform majority of artificial intelligence approaches including computer vision and natural language processing. Therefore, in this work we use different approaches based on convolutional neural nets (CNNs) which were already presented in Kim (2014) and Lenc and Král (2017).

Usually, the pre-trained word vectors obtained by some semantic model (e.g. word2vec (w2v) (Mikolov et al., 2013a) or glove (Pennington et al., 2014)) are used for initialization of the embedding layer of the particular neural net. These vectors can then be progressively adapted during neural network training. It was shown in many experiments that it is possible to obtain better results using these vectors compared to the randomly initialized vectors. Moreover, it has been proven that even "static" vectors (initialized by pre-trained embeddings and fixed during the network training) usually bring better performance than randomly initialized and trained ones.

However, the experiments were often realized on rather shorter texts and in single-label classification task. In this paper, we would like to analyze and evaluate the use of word embeddings for representation of longer texts in multi-label classification scenario. The embeddings are used in three different convolutional neural network topologies.

The experiments are realized on the Czech ČTK and English Reuters-21578 standard corpora. The Czech language has been chosen as a representative of highly inflectional Slavic language with a free word order. English is used to compare the results of our method with state of the art.

We compare the results with word2vec static and trainable embeddings with randomly initialized word vectors. We conclude that the initialization does not play an important role for classification of these documents. We further analyze and compare the word2vec embeddings with the learned ones from the semantic point of view and discuss the results.

Proceedings of Recent Advances in Natural Language Processing, pages 431–437,
Varna, Bulgaria, Sep 4–6 2017.
https://doi.org/10.26615/978-954-452-049-6_057

The rest of the paper is organized as follows. The following section contains a short review of the usage of neural networks for document classification including word embeddings. Section 3 describes topologies of the convolutional networks. Section 4 deals with experiments realized on the ČTK and Reuters corpora and then analyzes and discusses the obtained results. In the last section, we conclude the experimental results and propose some future research directions.

2 Related Work

Nowadays, neural nets belong to the state-of-the art approaches on many natural language processing tasks as for instance POS tagging, chunking, named entity recognition, semantic role labeling or document classification (Collobert et al., 2011).

First, we mention traditional feed-forward neural nets as shown for instance in (Manevitz and Yousef, 2007). The authors obtain F-measure about 78% on the standard Reuters dataset with a simple multi-layer perceptron with three layers. The standard backpropagation algorithm for multi-label learning of an MLP was improved in (Zhang and Zhou, 2006). The authors use a novel error function which gives better results on functional genomics text categorization.

Nam et al. (2014) propose a novel learning strategy of feed-forward nets for multi-label text classification task. The authors use cross-entropy algorithm for training with rectified linear units activation (Srivastava et al., 2014). The documents are represented by tf-idf and multi-label classification is realized by a simple thresholding of the output layer. The networks are evaluated on several multi-label datasets and obtain results comparable with the state of the art.

Both, standard convolutional networks and recurrent convolutional neural nets are also successfully used for text categorization. The authors (Lai et al., 2015) demonstrated that recurrent CNNs outperform CNNs on four corpora in single-label document classification task.

Another CNN based method with word embeddings as inputs (Kurata et al., 2016) leverages the co-occurrence of labels in the multi-label classification. Some neurons in the output layer capture the patterns of label co-occurrences, which improves the classification accuracy. This method is evaluated on the natural language query classification in a document retrieval system.

An alternative multi-label classification approach is proposed by Yang and Gopal (2012). The conventional representations of texts and categories are transformed into meta-level features. These features are then utilized in a learning-to-rank algorithm. Experiments on six benchmark datasets show good abilities of this approach in comparison with other methods.

Three different types of word embeddings with CNNs are compared in Kim (2014) on 7 NLP tasks including sentiment analysis and question classification. The author proposes a novel CNN topology and shows that word2vec initialization and a subsequent learning plays a crucial role for all sentence-level single-label classification tasks.

3 Network Topologies

In this section we describe the three CNN network topologies used in our experiments. The network inputs are the sequences of word indices into a vocabulary V of the size $|V|$. In order to ensure the fixed length, all documents are padded or shortened to a specified length M. These are then represented as real-valued word vectors of dimension E in the embedding layer. The embedding layer is either initialized randomly or by pre-trained word vectors from word2vec[1]. In the case of word2vec initialization, the layer is either further learned during the training process or is kept static. The input and the embedding layer is similar in all three following network topologies. The concrete values of the main hyper-parameters of the following networks are specified in Section 4.

3.1 Convolutional Network 1

This architecture was proposed in Lenc and Král (2017) for multi-label document classification.

The embedding layer is followed by a convolutional layer, where we use N_C convolution kernels of the size $k \times 1$. It uses rectified linear unit (ReLU) activation function. The following layer performs max pooling over the length $M - k + 1$ resulting in N_C $1 \times E$ vectors. The output of this layer is then flattened and connected with a fully connected layer with d_1 neurons. The output layer uses sigmoid activation function and its size corresponds to the number of categories $|C|$. The final result is obtained by thresholding of the output layer.

[1]It is possible to use other semantic model, however based on our previous experiments, we keep word2vec.

This architecture will hereafter be referenced as CNN1.

3.2 Convolutional Network 2

The second architecture is a modified version of a successful net proposed by Kim (2014).

Contrary to the first topology, this network uses two-dimensional convolutional kernels of various widths. The sizes of the kernels are $k \times E$ which means that it takes the whole length of the embedding. The original version is used for single-label classification and therefore it uses softmax activation function in the output layer. In our case, sigmoid is more appropriate due to the multi-label classification task. Similarly as in the previous topology, we also added one fully connected layer before the output. The output layer of the size $|C|$ is also thresholded to determine the set of assigned categories.

This network will be further called CNN2. The architecture is described in detail in Kim (2014). The threshold values for both CNN1 and CNN2 are set on the development corpus.

3.3 Two-level CNN

The first level of this network is the CNN1 topology described above. However, the output is not thresholded as in the former case. This network uses a multi-layer perceptron with one hidden layer to predict the number of labels.

It takes the output of the CNN S and learns a function $l = f(S)$ that maps the vector S to the number of relevant labels l. The output layer has softmax activation. After determining the number of labels the l categories with the highest activations are assigned to the document.

Figure 1 shows the architecture of this network where the CNN and 2nd-level FNN are merged. This topology will be hereafter referenced as 2L-CNN.

4 Experiments

This section describes first the tools and corpora used for evaluation of the approaches. Then we deal with the preprocessing stage and set-up of hyper-parameters of our networks. We describe further our experiments on Czech and English standard corpora and analyze the results.

4.1 Tools and Corpora

For implementation of all neural nets we used Keras tool-kit (Chollet, 2015) which is based on

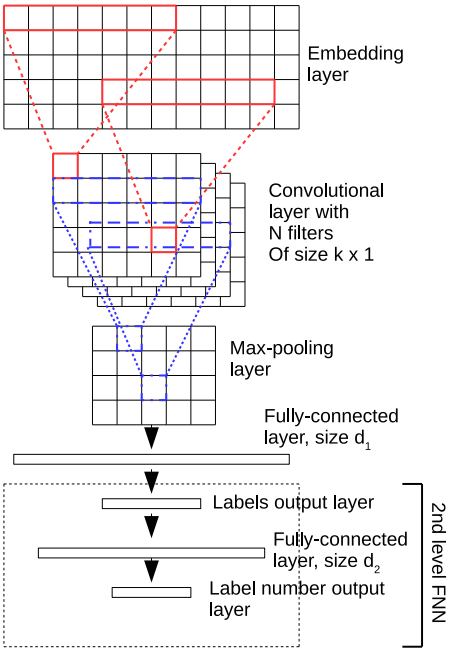

Figure 1: Two-level CNN architecture (2L-CNN).

the Theano deep learning library (Bergstra et al., 2010). It has been chosen mainly because of good performance and our previous experience with this tool. For evaluation of the multi-label document classification results, we use the standard recall, precision and F-measure (*F1*) metrics (Powers, 2011). The values are micro-averaged.

Word2vec vectors for Czech experiments are trained on Czech Wikipedia (Svoboda and Brychcín, 2016). For the English experiments we utilize the standard vectors trained on part of Google News dataset (Mikolov et al., 2013b).

4.1.1 Czech Text Document Corpus v 1.0

This corpus is composed of 11,955 news articles provided by the Czech News Agency (ČTK). The documents are annotated from a set of 60 categories as for instance agriculture, weather, politics or sport out of which we used 37 most frequent ones. The average number of categories per document is 2.55 and the average length of the documents is 277 words. 500 randomly chosen documents are reserved for development set while the remaining part is used for training and testing of our models. Figure 2 shows the distribution of the document lengths (in word tokens). This corpus is freely available for research purposes at `http://home.zcu.cz/~pkral/sw/`. We use the five-fold cross validation procedure for all experiments on this corpus.

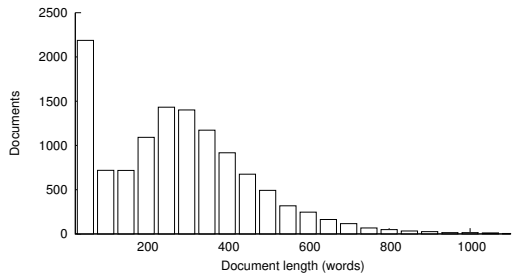

Figure 2: Document lengths in ČTK dataset.

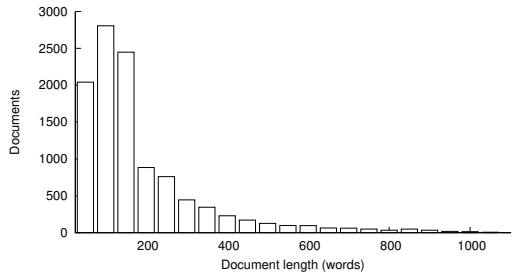

Figure 3: Document lengths in Reuters dataset.

4.1.2 Reuters-21578 English Corpus

The Reuters-21578[2] corpus is a collection of 21,578 documents. However, these documents include examples with no topics and with errors. Therefore, we use the commonly utilized version where the training part is composed of 7769 documents, while 3019 documents are reserved for testing. The number of possible categories is 90 and average label/document number is 1.23. Average document length is 159. This dataset is used in order to compare the performance of our networks with the state of the art. Distribution of document lengths is shown in Figure 3.

4.2 Preprocessing

The same preprocessing was performed for both Czech and English corpora. First we replaced all numbers in the texts by one common token "NUMERIC". The following characters were removed: [.,-_?!#:%()+"] The texts were then lowercased and a simple tokenization according to a space was done. To ensure a fixed length we either shortened the documents to a size of $M = 400$ or padded it by a token "PADDING" to the same length. We used two vocabulary sizes, namely 1000 and 20,000 most frequent words. Words not present in the vocabulary are replaced by an "OOV" (out of vocabulary) token.

[2]http://www.daviddlewis.com/resources/testcollections/reuters21578/

4.3 Hyper-parameters Set-up

The embedding vector length is set to 300 in all cases to allow the utilization of the pre-trained vectors and for a straightforward comparison with the learned ones.

The first network (CNN1) uses 40 kernels of length 16 according to (Lenc and Král, 2017). ReLU activation is used in convolutional layer. The following layer contains 256 neurons and the output layer has 37 or 90 neurons according to the used corpus.

The second network (CNN2) utilizes three kernel widths $k \in 3, 4, 5$ as proposed in (Kim, 2014) (100 filters are used for each width). The convolutional layer is followed by a fully-connected layer with 256 neurons and the output layer is the same as in the previous case.

The two-level network (2L-CNN) merges the CNN1 network and an MLP with 100 neurons in the hidden layer and 8 neurons in the output layer indicating the number of assigned labels. The output layer uses softmax activation function.

All networks are trained for 20 epochs using adaptive moment estimation optimization algorithm (Kingma and Ba, 2014). Mean square error loss function is used for the MLP determining the number of labels while binary cross-entropy is used for all CNNs.

4.4 Results on the Czech Corpus

We first present the results of the CNNs on the Czech Corpus (see Table 1). CNN1 and CNN2 are used with thresholding, the thresholds are set experimentally on the development set. Hyper-parameters of the 2L-CNN were also set on this set. The upper part shows the results for vocabulary size 1000 while the lower one uses 20,000 words.

This table shows that the size of the vocabulary plays an important role for document classification. The second interesting observation is that the results of the usage of the embeddings for all networks are consistent. The lowest scores are in all cases obtained with w2v static embeddings while the best classification results are generally achieved with randomly initialized embeddings.

We can thus conclude that initialization of the embeddings with w2v pre-trained vectors does not have any positive impact in this experiment. The best results are obtained using CNN1 and 2L-CNN both with randomly initialized embeddings.

Method	Prec.	Recall	F1[%]
Vocabulary size 1000			
CNN1, w2v static	57.58	70.92	63.56
CNN1, random	71.56	70.84	71.20
CNN1, w2v trainable	72.97	70.02	71.47
CNN2, w2v static	63.86	75.19	69.07
CNN2, random	70.33	71.37	70.84
CNN2, w2v trainable	69.55	71.66	70.59
2L-CNN, w2v static	67.12	62.21	64.57
2L-CNN, random	75.24	68.20	71.55
2L-CNN, w2v trainable	75.76	66.97	71.10
Vocabulary size 20,000			
CNN1, w2v static	64.89	79.88	71.61
CNN1, random	84.84	83.55	84.19
CNN1, w2v trainable	84.62	82.86	83.73
CNN2, w2v static	77.58	80.26	78.90
CNN2, random	80.36	79.59	79.97
CNN2, w2v trainable	79.83	80.96	80.39
2L-CNN, w2v static	76.36	70.20	73.15
2L-CNN, random	87.67	80.95	84.17
2L-CNN, w2v trainable	87.60	79.05	83.10

Table 1: Results of the CNNs on the Czech corpus.

Another interesting observation is that the role of training of the embeddings in the CNN2 is very small compared to the other two nets. The reason for this behavior can be different size and number of convolutional kernels.

In the second experiment, we show in Table 2 the impact of the optimal thresholds for classification using CNN1 and CNN2 nets. This experiment is done in order to determine the threshold values for English corpus where the development set is missing. Moreover, we also would like to analyze the impact of this optimal value on classification. The thresholds are thus set on the whole corpus. As in the previous case, the upper section of the table uses the vocabulary size 1000 and the lower one 20,000.

We must note that these data cannot be presented as "fair" results for the Czech dataset. However, our goal is to compare the different types of embeddings and this experiment well illustrates the ceiling which can be reached using a particular setting.

This experiment also shows that these results are comparable to the Table 1, therefore we can conclude that the threshold values set on development corpus are appropriate and that the methods are robust to the sub-optimally set thresholds. The second observation is that the behavior of all networks with different kinds of embeddings is similar as in the previous case.

Method	Prec.	Recall	F1[%]
Vocabulary size 1000			
CNN1, w2v static	69.00	62.93	65.82
CNN1, random	75.98	68.10	71.82
CNN1, w2v trainable	78.12	66.46	71.82
CNN2, w2v static	74.13	68.43	71.16
CNN2, random	76.38	67.27	71.54
CNN2, w2v trainable	75.93	67.10	71.24
Vocabulary size 20,000			
CNN1, w2v static	76.06	72.01	73.98
CNN1, random	86.50	82.24	84.32
CNN1, w2v trainable	86.60	81.20	83.81
CNN2, w2v static	82.50	76.45	79.36
CNN2, random	83.16	77.47	80.21
CNN2, w2v trainable	83.75	77.97	80.75

Table 2: Results of CNN1 and CNN2 on the Czech corpus with optimal threshold values.

4.5 Results on the English Reuters Dataset

This experiment is realized in order to show the impact of the different embeddings on the standard English corpus. We use the 300-dimensions English word2vec embeddings trained on Google News[3].

This table shows that the role of the embeddings on English language is similar to the previous one, however it slightly differs in the case of the CNN2.

This network gives comparable results for all embedding types. Moreover the best score is obtained with w2v static embeddings, however this difference is not statistically significant.

The behavior of the embeddings in the other two CNNs is similar as in the previous Czech experiments, where word2vec initialization does not play any positive role for classification. A reason for such difference in Czech and English could be caused by the quality of the word2vec embeddings. The best performing network is the 2L-CNN with randomly initialized embeddings. The resulting F-measure is comparable to the value of 87.89% presented in (Nam et al., 2014).

4.6 Embedding Analysis

In this experiment we analyze the semantic similarity of the embedding vectors learned during the network training and compare them with the standard word2vec vectors. We employ the cosine distance to identify 5 most similar words.

We have chosen word "Británie" (Britain) and show the most similar words both for Czech and English embeddings. The results of the Czech experiment are reported in Table 4 while the results

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

Method	Prec.	Recall	F1[%]
CNN1 w2v static	84.37	73.40	78.50
CNN1 random	87.26	86.14	86.69
CNN1 w2v trainable	90.63	82.18	86.20
CNN2 w2v static	89.42	78.34	83.51
CNN2 random	89.82	76.79	82.79
CNN2 w2v trainable	89.10	78.42	83.42
2L-CNN, w2v static	82.89	74.41	78.42
2L-CNN, random	90.39	84.96	87.59
2L-CNN, w2v trainable	91.03	82.39	86.50

Table 3: Results of the CNNs on the Reuters dataset, vocabulary size is set to 20,000.

Word	Cos.	Word	Cos.
w2v (static)		**CNN1, random**	
Německo (Germany)	0.65	třetí (third)	0.20
usa	0.63	výroba (production)	0.19
velká (great)	0.60	hlavně (mainly)	0.19
spojené (united)	0.59	Rusko (Russia)	0.18
Rusko (Russia)	0.58	procent (percent)	0.18
CNN2, random		**CNN2, w2v trainable**	
premiér (p. minister)	0.51	německo (Germany)	0.51
vlády (governments)	0.49	vláda (government)	0.43
vláda (government)	0.48	Londýn (London)	0.40
ústavu (institute)	0.47	tun (tons)	0.39
vládní (governmental)	0.46	prezident (president)	0.38

Table 4: 5 closest words to "Británie (Britain)" in Czech.

on English embeddings shows Table 5. The upper part compares the word2vec (static) embeddings with the vectors learned by CNN1. The lower part then compares the randomly initialized embeddings learned by CNN2 with vectors that were initialized by word2vec and progressively adapted during the training of CNN2.

Table 4 shows that all similarity values except CNN1 values are comparable. The CNN1 vectors differ significantly from the word2vec ones regarding the similarity values. We can observe similar results for both variants learned by CNN2.

From the point of view of the semantic similarity of words, this experiment shows, that the lists of words differ and there is just only few common ones. On the other hand, the majority of words are for all cases really semantically close and related to the word Britain.

Table 5 shows the same experiment carried out with the English word2vec. There is again higher difference between word2vec and CNN1 vectors. However, the variants of CNN2 vectors differ more significantly than in the case of Czech embeddings. This experiment shows, that the lists of words differ as in the previous case and that there is only few common ones.

Word	Cos.	Word	Cos.
w2v (static)		**CNN1, random**	
mining	0.41	based	0.24
cents	0.39	tird	0.21
opec	0.38	statistics	0.18
gold	0.34	amount	0.16
quarterly	0.33	october	0.16
CNN2, random		**CNN2, w2v trainable**	
line	0.19	intervention	0.33
accord	0.19	october	0.33
estimates	0.18	adding	0.31
share	0.18	investment	0.30
same	0.16	ministers	0.29

Table 5: Closest words to "Britain" in English.

5 Conclusions and Future Work

This paper analyzed and evaluated word embeddings in convolutional neural networks for representation of longer texts in multi-label classification task. Three different CNNs topologies were used. The experiments were realized on Czech ČTK and English Reuters-21578 corpora.

We compared the results of word2vec static and trainable embeddings with randomly initialized word vectors. We concluded that initialization does not play an important role for multi-label document classification in both languages. However, learning of word vectors is crucial to obtain good classification score. This behavior should be justified by a sufficient amount of the relatively long documents. This fact improves the convergence of our models during training and also decreases the impact of the particular words for the whole classification. We further analyzed both word2vec static and learned embeddings from the semantic point of view and discussed the results. We can conclude that although the semantically closest words differ significantly, they are all close from the semantic point of view.

In future work, we would like to study the impact of document length to the classification results with randomly initialized embeddings and pre-trained word vectors. We could also have used different network topologies to further improve the performance of document classification. A subsequent analysis of the behavior of the embeddings in these nets will be also realized.

Acknowledgments

This work was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports.

References

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)*. Austin, TX, volume 4, page 3.

Franois Chollet. 2015. keras. https://github.com/fchollet/keras.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research* 12:2493–2537.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Gakuto Kurata, Bing Xiang, and Bowen Zhou. 2016. Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence. In *Proceedings of NAACL-HLT*. pages 521–526.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification .

Ladislav Lenc and Pavel Král. 2017. Deep neural networks for Czech multi-label document classification. *CoRR* abs/1701.03849. http://arxiv.org/abs/1701.03849.

L. Manevitz and M. Yousef. 2007. One-class document classification via neural networks. *Neurocomputing* 70(7-9):1466–1481. https://doi.org/10.1016/j.neucom.2006.05.013.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale multi-label text classification - revisiting neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pages 437–452.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

DMW Powers. 2011. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies* 2(1):37–63.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

Lukáš Svoboda and Tomávs Brychcín. 2016. New word analogy corpus for exploring embeddings of czech words. *CoRR* abs/1608.00789. http://arxiv.org/abs/1608.00789.

Grigorios Tsoumakas and Ioannis Katakis. 2006. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining* 3(3).

Yiming Yang and Siddharth Gopal. 2012. Multilabel classification with meta-level features in a learning-to-rank framework. *Machine Learning* 88(1-2):47–68.

Min-Ling Zhang and Zhi-Hua Zhou. 2006. Multi-label neural networks with applications to functional genomics and text categorization. *Knowledge and Data Engineering, IEEE Transactions on* 18(10):1338–1351.