Title Categorization based on Category Granularity

Kazuya Shimura and Fumiyo Fukumoto*

Graduate School of Engineering, Graduate Faculty of Interdisciplinary Research*, University of Yamanashi
4-3-11, Takeda, Kofu, Yamanashi, 400-8510, Japan
{g17tk008,fukumoto}@yamanashi.ac.jp

Abstract

We focus on a problem of short text categorization, *i.e.* categorization of newspaper titles, and present a method that maximizes the impact of informative words due to the sparseness of titles. We used hierarchical structure of categories and a transfer learning technique based on pre-training and fine-tuning in order to incorporate granularity of categories into categorization. According to the hierarchical structure of categories, we transferred trained parameters of Convolutional Neural Network(CNN) on upper layers to the related lower layers, and finely tuned parameters of CNN. The method was tested on titles collected from the Reuters corpus, and the results showed the effectiveness of the method.

1. Introduction

There has been a great deal of interest in short text categorization with the availability of online social networks. Automatic categorization of short text such as search snippets, reviews, and Web page titles supports many applications, *e.g.*, retrieving information on the Internet, and creating digital libraries. In this paper, we focus on news titles as short texts. A basic assumption in categorization with machine learning techniques is that the distribution of words between training and test data is identical. However, when the data consists of short texts, *e.g.* titles, it is often the case that the word distribution between them are different from each other, and thus the performance of categorization is deteriorated. Moreover, similar to the categorization task with normal documents such as news articles and academic papers, titles are often assigned to several categories which make categorization task more problematic.

In this paper, we present a method for title categorization that maximizes the impact of informative words in titles. We used a transfer learning technique based on supervised pre-training and fine-tuning in Convolutional Neural Network (CNN) in order to incorporate granularity of title categories into categorization. The idea of supervised pre-training is to use a data-rich auxiliary dataset and task, to initialize the parameters of CNN. The CNN can then be used on the small dataset tagged by fine-grained categories, directly, as a feature extractor. The network can be updated by continued training on the small dataset. The process is so called fine-tuning. The supervised pre-training and fine-tuning are widely used in visual recognition tasks, and showed that they are effective when training data is scarce (Girshick et al., 2014). We applied a hierarchical structure of categories to learning process on CNN. We first learn to distinguish among categories at the top level of a hierarchy, *e.g.*, corporate/industrial and economics, by using pre-training, then learns lower level distinctions, *e.g.*, distinction between legal/judicial and inflation/prices, by using fine-tuning only within the appropriate top level of the hierarchical structure. These sub-problems can be solved more accurately, and efficiently.

2. Related Work

With the growing online social network application and e-commerce (Deng and Peng, 2006), short text categorization has been widely studied (Song et al., 2014). Unlike normal documents such as news articles and academic papers, short texts are sparseness and limited contextual information. Major attempts to tackle these problems are to expand short texts with knowledge extracted from textual corpus, Machine-readable dictionaries and thesauri. Several authors followed this approach, and construct a high-quality categorization model (Phan et al., 2008; Chen et al., 2011; Chang and Lin, 2011; Long et al., 2012; Wu et al., 2012). However, when the data distribution of the external knowledge and test data are not identical, the classification accuracy might perform worse. Moreover, manual collection and tuning the data is very expensive and time-consuming. The methodology for accurate short text categorization by making the maximum use of training data only is needed to improve categorization performance.

To our knowledge, there have been only a few previous work on short text classification utilizing short texts only. Zhang *et al.* classified short text by detecting information path (Zhang et al., 2013). Their assumption is that ordered subsets of short text, called information path consists of sequential subsets in the test dataset. Instances of former classified subsets can assist classification of later subset followed by this path. Experimental results on the search snippets and the paper titles show the effectiveness of the method. However, they reported that term selection for detecting information path is an entropy-based simple statistical method, and thus needs to involve some existing techniques, including those leveraging auxiliary resource into their framework for further accuracy gains.

Over the past few years, many authors have attempted to apply deep learning methods including CNN (Zhang et al., 2015; Wang et al., 2015; Zhang and Wallace, 2015) to text categorization task, and a sequence of results has demonstrated that CNN outperformed state-of-the-art bag-of-words based machine learning techniques such as SVMs and Boosting. Kim applied CNN to several benchmarks including movie review classification, question classification, and opinion polarity detection (Kim,

2014). He reported that CNN outperformed remarkably well, although it is a simple architecture with one layer of convolution on top of word vectors obtained from an unsupervised neural language model, word2vec (Mikolov et al., 2013). Several researchers have attempted to incorporate character-level information into convolutional neural networks (Santos and Gatti, 2014; Zhang et al., 2015). Santos *et al.* proposed a deep convolutional neural network that exploits from character- to sentence-level information to perform sentiment analysis of short texts. They used two convolutional layers to extract relevant features from words and sentences of any size. Experiments on two data sets, *i.e.*, the Stanford Sentiment Treebank and the Stanford Twitter Sentiment corpus showed the effectiveness of the method. Zhang *et al.* applied character-level convolutional networks to text classification (Zhang et al., 2015). They compared their method with a large number of traditional and deep learning models including word-based convolutional network, long-short term memory using several large-scale datasets such as AG's news corpus, Sogou news corpus. They reported that character-level convolutional network is effective compared with other deep learning models, while they focused on single-label problem.

Other researchers focused on multi-label categorization problem. One attempt is Berger's method which makes use of word embeddings in both of a convolutional neural network and a recurrent neural network model with a gated recurrent unit to label documents with a large label set (Berger, 2015). They demonstrated that a convolutional neural network and a recurrent neural network with a gated recurrent unit outperformed the traditional binary relevance method with bag-of-words features on a large scale multi-label classification problem. Johnson *et al.* explored a semi-supervised method with convolutional neural networks for categorization (Johnson and Zhang, 2015). The model learns embeddings of small text regions from unlabeled data for integration into a supervised CNN. The idea is based on two-view semi-supervised learning which is intended to be effective for the task of interest even though the training is done on unlabeled data. The model is applied to the two tasks, *i.e.*, sentiment classification and topic classification tasks. All of these approaches aimed at utilizing large scale data set to construct a high-quality classification model.

In contrast with the aforementioned works, here we propose a method for title categorization that maximizes the impact of informative words by using category granularity.

3. Framework of the system

The method consists of three procedures, (1) Word embedding learning, (2) Transfer learning with CNN, and (3) multi-label categorization. Figure 1 illustrates our system.

3.1. Word embedding learning

Models based on neural networks have become very popular as these models attained at a very good performance in practice. However, they often tend to be relatively slow both at train and test time, limiting their use on very large datasets (Joulin et al., 2016). To tackle the problem, many researchers have attempted to use word-embedding procedure such as word2vec by Mikolov *et al.* (Mikolov et al., 2013), and fastText by Joulin *et al.* (Joulin et al., 2016). The first procedure is to learn distributed word representation. We used fastText tools[1]. FastText developed by Joulin *et al.* is a method to learn vector representations for each word using a neural network language model. The model architecture consists of hidden layer and output layer to predict nearby words. Unlike word2vec, fastText uses a hierarchical softmax (Goodman, 2001) based on the Huffman coding tree (Mikolov et al., 2013) which makes the running time more efficiently. They also used a bag of n-grams as additional features to capture some partial information about the local word order to make computational cost more efficiency. As shown in (1) of Figure 1, we used all the texts and applied fastText to obtain distributed word representation.

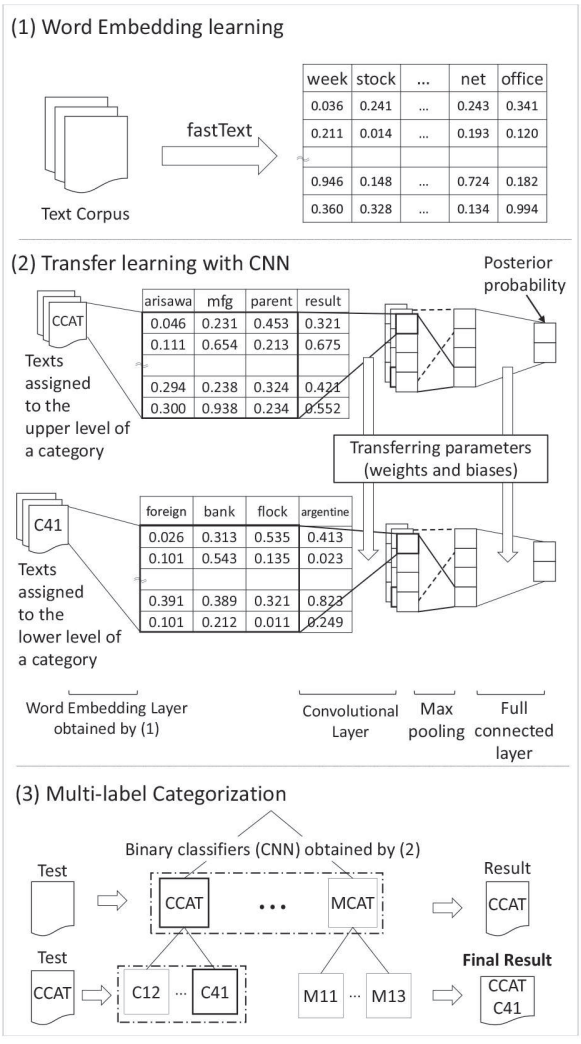

Figure 1: Overview of the system

3.2. Transfer learning with CNN

The transfer learning is a learning technique that retains and applies the knowledge learned in one or more domains (henceforth we call it a source domain) to efficiently de-

[1] https://github.com/facebookresearch/fastText

velop an effective hypothesis for a new domain (we call it a target domain). An essential requirement for successful knowledge transfer is that the source domain and the target domain should be closely related (Tan et al., 2017). We used titles consisting of source domain which is a coarse granularity compared to the target domain, and learned a model by using CNN. CNN is a feedforward network equipped with convolution layers interleaved with pooling layers.

As shown in (2) of Figure 1, we use word embeddings obtained by procedure (1) to initialize a lookup table, and lower levels extract more complex feature. Let $T = \{w_1, w_2, \cdots, w_N\}$ be a title. It's projected matrix $M \in R^{d*N}$ is obtained by a table of the word embedding layer, where d is the dimension of word embedding. The next layer, convolutional layer is to extract patterns of discriminative word sequences occurred in the input titles across the training data. The output from the convolutional layer are passed to the pooling layer in order to reduce the representation. We used max pooling which returns the maximum value. Finally, pooling layers are passed to a fully connected softmax layer. It calculates the probability distribution over the labels as a classifier. For each category within the top level hierarchy (CCAT in Figure 1), we applied this procedure. For the results, we repeatedly applied fine-tuning strategy, and learned models for lower levels (C41 in Figure 1).

Fine-tuning begins with transferring the parameters(weights and biases) from a pre-trained network to the network we wish to train (Tajbakhsh et al., 2016). After the parameters of the last few layers are initialized, the new network can be fine-tuned in a layer-wise manner. Fine-tuning is motivated by the observation that the earlier features of a CNN contain more generic features that should be effective for many tasks, but later layers of the CNN becomes progressively more specific to the details of the classes contained in the original dataset. The motivation is identical to the hierarchical structure of categories, *i.e.* we first learn to distinguish among categories at the top level of a hierarchy, then learn lower level distinctions by using only within the appropriate top level of the hierarchical structure. Fine-tuning the last few layers is usually sufficient for transfer learning. However, if the distance between the upper and lower level of categories is significant, one may need to fine-tune the early layers as well. In this work, we transferred the convolutional layer, the last two layers within the fully connected layers.

3.3. Multi-label categorization

The final procedure is multi-label categorization shown in Figure 1. We compute the probabilities of the test title, being in each of the top-level categories, *e.g.*, CCAT and MCAT, and each of the second-level categories, *e.g.*, C12 and M13, respectively. We note that for the non-hierarchical case, models were learned to distinguish each category from all other categories. For the hierarchical case, models were learned to distinguish each category from only those categories within the same top-level probabilities from the first and second level for the hierarchical approach. We compute a Boolean function, *i.e.* we first set a category at the top level, and only match second-level categories that pass this test.

4. Experiments

4.1. Data

We had experiments to evaluate our method. We used Reuters'96 corpus. The Reuters'96 corpus from 20th Aug. 1996 to 19th Aug. 1997 consists of 806,791 documents organized into coarse-grained categories, *i.e.*, 126 categories with a four-level hierarchy. We selected 30 categories and used them in the experiments. The selection is made according to the number of documents belonging to the categories. We created three sets of categories, *i.e.*, a set of categories assigned to more than 15,000 documents, a set of categories assigned to less than 15,000 and more than 3,000 documents, and a set of categories with less than 3,000 documents. We call these, *large*, *medium*, and *small*, respectively. The data used in the experiments is shown in Table 1.

The total number of titles with Reuters corpus used in the experiment is 557,290, and average number of categories per a title is 1.496. We divided data into two: the training data consisting of 80% of the data, 445,832 titles, and the test data consisted of 20% of the data, 111,458 titles. All the titles are tagged by using Tree Tagger (Schmid, 1995). We used nouns, verbs, and adjectives in the experiments. As shown in Table 1, the hierarchical level of the categories used in the experiments is three.

We examined two types of fine-tuning in the experiments. The first fine-tuning is initialize the parameters in each level. Namely, the model is learned in the top-level, then initialize the parameters obtained in the top-level and learned in the second level. The parameters obtained by the second-level are initialized, and models are learned in the third level of a hierarchy. We call it Gradual fine-tuning. The second type of fine-tuning is so called Global fine-tuning, *i.e.*, models for the second and third levels are learned by initializing parameters which are obtained in the top level.

We compared our method with the results obtained by Support Vector Machines(SVMs) and fastText (Joulin et al., 2016). We used Bag-of-words for SVMs and distributed word representation for CNN and fastText. We used all the Reuters corpus with titles and contents to learn word representation. We set word- embedding dimension to 300. The batch size of CNN was set to 100, and the number of epoch was 40. The filter size of convolutional layer is 300 $\times$3, and the number of filters is 128. These parameters are empirically determined. For all of three methods including our method, we evaluated results with non-hierarchical flat and hierarchical cases.

In the experiments, we used five cross validation to evaluate the method. For evaluating the effectiveness of category assignments, we used the standard recall, precision, and $F1$ measures. Recall denotes the ratio of correct assignments by the system divided by the total number of correct assignments. Precision is the ratio of correct assignments by the system divided by the total number of the system's assignments. The $F1$ measure which combine recall(r) and precision(p) with an equal weight is $F1(r,p) = \frac{2rp}{r+p}$.

Table 1: The Statistics of the data

Cat	# of doc	Level	Size	Cat	# of doc	Level	Size	Cat	# of doc	Level	Size
CCAT	315,946	top	large	C12	9,552	second	medium	C172	10,192	third	medium
MCAT	157,245	top	large	E13	5,747	second	medium	E131	4,943	third	medium
ECAT	77,557	top	large	C22	5,241	second	medium	C312	2,163	third	small
M11	41,138	second	large	C14	3,279	second	medium	E511	2,074	third	small
M13	37,639	second	large	C34	3,172	second	medium	E513	1,957	third	small
C17	31,215	second	large	E31	2,093	second	small	E311	1,514	third	small
C31	26,191	second	large	E61	218	second	small	C311	1,241	third	small
M12	16,811	second	large	M131	20,557	third	large	C313	959	third	small
E51	14,854	second	medium	M132	16,913	third	large	E132	768	third	small
C41	10,246	second	medium	C171	13,486	third	medium	E313	85	third	small

4.2. Results

The results are shown in Tables 2 and 3. Micro-$F1$ indicates the score computed globally over all the $n \times m$ binary decisions where n is the number of total test documents, and m is the number of categories in consideration. Macro-$F1$ refers to the score computed for the binary decisions on each individual category first, and then be averaged over categories (Yang and Lin, 1999). "Flat" refers to the results of each method obtained by not applying category hierarchy. "Not fine-tuning" shows the results without fine-tuning.

Table 2: Categorization accuracy

Approach	Micro-$F1$	Macro-$F1$
SVMs(Flat)	0.625	0.085
SVMs(Hierarchy)	0.907	0.745
fastText(Flat)	0.691	0.297
fastText(Hierarchy)	0.909	0.747
CNN(Flat)	0.754	0.238
CNN(Not fine-tuning)	**0.930**	**0.795**
CNN(Gradual fine-tuning)	**0.930**	0.794
CNN(Global fine-tuning)	**0.930**	**0.795**

Table 3: Accuracy by category level(Average macro-F1)

Approach	Top	Second	Third
SVMs(Flat)	0.771	0.016	0.000
SVMs(Hierarchy)	0.933	0.744	0.703
fastText(Flat)	0.796	0.336	0.141
fastText(Hierarchy)	0.938	0.747	0.701
CNN(Flat)	0.852	0.310	0.020
CNN(Not fine-tuning)	**0.954**	**0.799**	**0.753**
CNN(Gradual fine-tuning)	**0.954**	0.798	**0.753**
CNN(Global fine-tuning)	**0.954**	**0.799**	**0.753**

We can see from Table 2 that the overall performance of CNN was better to those of SVMs and fast Text. The best Micro-F1 was obtained by CNN with a hierarchy, and that of Macro-$F1$ were CNN (Not fine-tuning) and CNN (Global fine-tuning) with hierarchical case. The results with hierarchy were better to those without hierarchy in all of the methods, as both of Micro-$F1$ and Macro-$F1$ with hierarchy outperformed than flat non-hierarchical case. As shown in Table 3, there is a drop in performance in going from the top to lower level of categories in all of the method. However, the performance obtained by CNN is still better than other methods. This demonstrates that CNN is effective for categorization.

We recall that we used fine-tuning technique as it is effective for small training dataset, since it is motivated by the observation that the earlier features of a CNN contain more generic features that should be effective for categorization, but later layers of the CNN is more specific to the details of the classes contained in the original dataset. Thus, it is important to compare the results with and without fine-tuning in the experiments. Table 3 shows that the results obtained by Not fine-tuning, Gradual fine-tuning and Global fine-tuning are not statistically significant with each other, especially Not fine-tuning and Global fine-tuning were the same accuracy, while they improved overall performance compared with other baseline methods. We examined how the number of training data affects the overall performance of these two methods. We decreased the number of training data ranging from 100% from 10% in steps 10%, and 10%, 5%, 2% and 1%. There were no significant difference between Not-fine-tuning and Global fine-tuning methods when the ratio ranged from 100% to 10%. Figure 2 illustrates the average macro-$F1$ against the number of training data obtained by each method ranged from 10% to 1%. Each value of Macro-$F1$ indicates the results obtained by five cross validation. The x-axis refers to the ratio of training data and y-axis shows the average Macro-$F1$. As shown in Figure 2, when the ratio is less than 10% the average macro-$F1$ obtained by Global fine-tuning was slightly better to those obtained by Not fine-tuning. From the results, We can conclude that when the training data consists of a small number of titles, fine-tuning works well.

5. Conclusions

We have developed an approach for titles categorization by using hierarchical structure of categories and a transfer learning technique based on pre-training and fine-tuning to incorporate granularity of categories into categorization. The results showed that CNN with hierarchical structure attained at 0.930 Micro-$F1$, and especially, it works well for lower level of a hierarchy. Future work will include: (i) extending the method to make use of hierarchical structure of words, *e.g.*, WordNet structure, (ii) tuning several parameters such as the number of layers and the number of epochs to obtain further accuracy, and (iii) evaluating the method by using other data such as the search

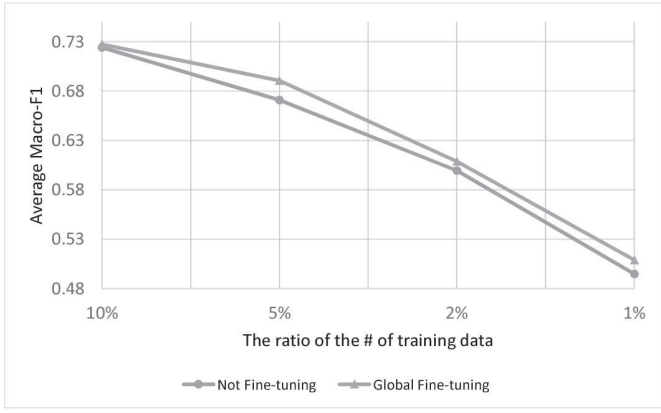

Figure 2: Average macro-F1 against the # of training data

snippets (Phan et al., 2008) and titles of scientific papers [2].

Acknowledgements

The authors would like to thank anonymous reviewers for their helpful comments. This work was supported by the Telecommunications Advancement Foundation, and Support Center for Advanced Telecommunications Technology Research, Foundation.

6. References

Berger, M. J., 2015. Large Scale Multi-label Text Classification with Semantic Word Vector.

Chang, C. C. and C. J. Lin, 2011. Libsvm: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(27):1–27.

Chen, M., X. Jin, and D. Shen, 2011. Short Text Classification Improved by Learning Multi-granularity Topics. In *Proc. of the 22nd International Joint Conference on Artificial Intelligence*.

Deng, W. W. and H. Peng, 2006. Research on a Naive Bayesian based Short Message Filtering System. In *Proc. of the 5th International Conference on Machine Learning and Cybernetics*.

Girshick, D., J. Donahue, T. Darrell, and J. Malik, 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*.

Goodman, J., 2001. Classes for Fast Maximum Entropy Training. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Johnson, R. and T. Zhang, 2015. Semi-supervised Convolutional Neural Networks for Text Categorization vis Region Embedding. *Proc. of the Advances in Neural Information Processing Systems*, 28:919–927.

Joulin, A., E. Grave, P. Bojanowski, and T. Mikolov, 2016. Bag of Tricks for Efficient Text Classification. In *ArXiv preprint arXiv 1607.01759*.

Kim, Y., 2014. Convolutional Neural Networks for Sentence Classification. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*.

[2] www.jst.go.jp/EN/index.html

Long, G., L. Chen, X. Zhu, and C. Zhang, 2012. TCSST: Transfer Classification of Short & Sparse Text Using External Data. In *Proc. of the 21st ACM International Conference on Information and Knowledge Management*.

Mikolov, T., K. Chen, G. Corrado, and J. Dean, 2013. Efficient Estimation of Word Representations in Vector Space. In *Proc. of the International Conference on Learning Representations Workshop*.

Phan, X. H., L. M. Nguyen, and S. Horiguchi, 2008. Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-Scale Data Collections. In *Proc. of the 17th International World Wide Web Conference*.

Santos, C. N. and M. Gatti, 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In *Proc. of the 25th International Conference on Computational Linguistics*.

Schmid, H., 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proc. of the EACL SIGDAT Workshop*.

Song, G., Y. Ye, X. Du, X. Huang, and S. Bie, 2014. Short Text Classification: A Survey. *Multimedia*, 9(5):635–643.

Tajbakhsh, N., J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, N. B. Gotway, and H. Liang, 2016. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312.

Tan, B., Y. Zhang, S. J. Pan, and Q. Yang, 2017. Distant domain transfer learning. In *Proc. of the 31st AAAI Conference on Artificial Intelligence*.

Wang, P., J. Xu, B. Xu, C-L. Liu, H. Zhang, F. Wang, and H. Hao, 2015. Semantic Clustering and Convolutional Neural Network for Short Text Categorization. In *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Wu, W., H. Li, H. Wang, and K. Q. Zhu, 2012. A Probabilistic Taxonomy for Text Understanding. In *Proc. of the 2012 ACM SIGMOD International Conference on Management of Data*.

Yang, Y. and X. Lin, 1999. A Re-examination of Text Categorization. In *Proc. of the 22nd International Conference on Research and Development in Information Retrieval*.

Zhang, S., X. Jin, D. Shen, B. Cao, X. Ding, and X. Zhang, 2013. Short Text Classification by Detecting Information Path. In *Proc. of the 22rd ACM International Conference on Information and Knowledge Management*.

Zhang, X., J. Zhao, and Y. LeCun, 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing systems*.

Zhang, Y. and B. C. Wallace, 2015. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *Computing Research Repository*.