HFT-CNN: Learning Hierarchical Category Structure for Multi-label Short Text Categorization

Kazuya Shimura[1], **Jiyi Li**[2] **and Fumiyo Fukumoto**[2]
Graduate School of Engineering[1]
Interdisciplinary Graduate School[2]
University of Yamanashi
4-3-11, Takeda, Kofu, 400-8511 Japan
{g17tk008,jyli,fukumoto}@yamanashi.ac.jp

Abstract

We focus on the multi-label categorization task for short texts and explore the use of a hierarchical structure (HS) of categories. In contrast to the existing work using non-hierarchical flat model, the method leverages the hierarchical relations between the categories to tackle the data sparsity problem. The lower the HS level, the worse the categorization performance. Because lower categories are fine-grained and the amount of training data per category is much smaller than that in an upper level. We propose an approach which can effectively utilize the data in the upper levels to contribute categorization in the lower levels by applying a Convolutional Neural Network (CNN) with a fine-tuning technique. The results using two benchmark datasets show that the proposed method, Hierarchical Fine-Tuning based CNN (HFT-CNN) is competitive with the state-of-the-art CNN based methods.

1 Introduction

Short text categorization is widely studied since the recent explosive growth of online social networking applications (Song et al., 2014). In contrast with documents, short texts are less topic-focused in texts. Major attempts to tackle the problem is to expand short texts with knowledge extracted from the textual corpus, machine-readable dictionaries, and thesauri (Phan et al., 2008; Wang et al., 2008; Chen et al., 2011; Wu et al., 2012). However, because of domain-independent nature of dictionaries and thesauri, it is often the case that the data distribution of the external knowledge is different from the test data collected from some specific domain, which deteriorates the overall performance of categorization. A methodology which maximizes the impact of pre-defined domains/categories is needed to improve categorization performance.

More recently, many authors have attempted to apply deep learning techniques including CNN (Wang et al., 2015; Zhang and Wallace, 2015; Zhang et al., 2017; Wang et al., 2017), the attention based CNN (Yang et al., 2016), bag-of-words based CNN (Johnson and Zhang, 2015a), and the combination of CNN and recurrent neural network (Lee and Dernoncourt, 2016; Zhang et al., 2016) to text categorization. Most of them demonstrated that neural network models are powerful for learning features from texts, while they focused on single-label or a few labels problem. Several efforts have been made to multi-labels (Johnson and Zhang, 2015b; Liu et al., 2017). Liu *et al*. explored a family of new CNN models which are tailored for extreme multi-label classification (Liu et al., 2017). They used a dynamic max pooling scheme, a binary cross-entropy loss, and a hidden bottleneck layer to improve the overall performance. The results by using six benchmark datasets where the label-set sizes are up to 670K showed that their method attained at the best or second best in comparison with seven state-of-the-art methods including FastText (Joulin et al., 2017) and bag-of-words based CNN (Johnson and Zhang, 2015a). However, all of these attempts aimed at utilizing a large volume of data.

We address the problem of multi-label short text categorization and explore the use of a HS of categories. The lower level of categories are fine-grained compared to the upper level of categories. Moreover, it is often the case that the amount of training data in a lower level is much smaller than that in an upper level which deteriorates the overall performance of categorization. We propose an approach which can effectively utilize the data in the upper levels to contribute categorization in lower levels by applying fine-tuning to the CNN which can learn a HS of categories and incorporate

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 811–816
Brussels, Belgium, October 31 - November 4, 2018.

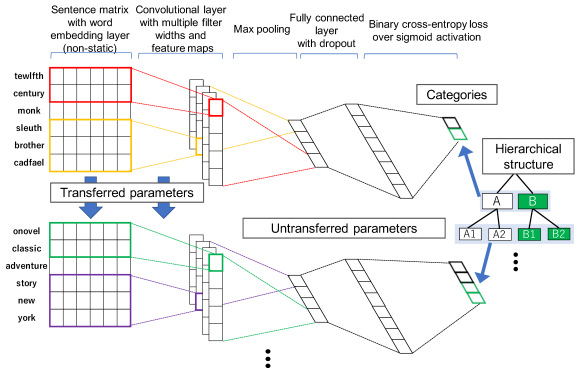


Figure 1: HFT-CNN model

granularity of categories into categorization. We transferred the parameters of CNN trained from upper to lower levels according to the HS, and finely tuned parameters. The main contributions of our work can be summarized: (1) We propose a method that maximizes the impact of pre-defined categories to alleviate data sparsity in multi-label short texts. (2) We empirically examined a fine-tuning with CNN that fits to learn a HS of categories defined by lexicographers, and (3) The results show that our method is competitive to the state-of-the-art CNN based methods by using two benchmark datasets, especially it is effective for categorization of short texts consisting of a few words with a large number of labels.

2 Hierarchical Fine-Tuning based CNN

2.1 CNN architecture

Similar to other CNN (Johnson and Zhang, 2015a; Liu et al., 2017), our HFT-CNN model shown in Figure 1 is based on (Kim, 2014). Let $\mathbf{x}_i \in \mathbb{R}^k$ be the k-dimensional word vector with the i-th word in a sentence obtained by applying skip-gram model provided in fastText[1]. A sentence with length n is represented as $\mathbf{x}_{1:n} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{nk}$. A convolution filter $\mathbf{w} \in \mathbb{R}^{hk}$ is applied to a window size of h words to produce a new feature, $c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b)$ where $b \in \mathbb{R}$ indicates a bias term and f refers to a non-linear activation function. We applied this convolution filter to each possible window size in the sentence and obtained a feature map, $m \in \mathbb{R}^{n-h+1}$. As shown in Figure 1, we then apply a max pooling operation over the feature map and obtain the maximum value $\hat{m}$ as a feature of this filter. We obtained multiple filters by varying window sizes and multiple features. These features form a pooling layer and are passed to a fully connected layer. In the fully connected layer, we applied dropout (Hinton et al., 2012). The dropout randomly sets values in the layer to 0. Finally, we obtained the probability distribution over categories. The network is trained with the objective that minimizes the binary cross-entropy (BCE) of the predicted distributions and the actual distributions by performing stochastic gradient descent.

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

2.2 Hierarchical structure learning

Our key idea is to use a fine-tuning technique in CNN to tackle the data sparsity problem, especially a lower level of a HS. Following a HS, we transferred the parameters of CNN trained in the upper levels to the lower levels which are worse trained because of the lack of data, and then finely tuned parameters of CNN for lower levels (Figure 1). This approach can effectively utilize the data in the upper levels to contribute categorization in the lower levels.

Fine-tuning is motivated by the observation that the earlier features of 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 a HS of categories as we first learn to distinguish among generic categories at the upper level of a hierarchy, then learns lower level distinctions by using only within the appropriate top level of the HS. We note that fine-tuning the last few layers are usually sufficient for transfer learning as the last few layers become more specific features. However, the HS consisting of deep level needs to fine-tune the early layers as well because the distance between the upper and lower level of categories is significant. For this reason, we transferred two layers shown in Figure 1, i.e., a layer obtained by word embedding and the convolutional layer. We used them as an initial parameter to learn the second level of a hierarchy. We repeated this procedure from the top level to the bottom level of a hierarchy. We note that a HS consists of many levels. We fine-tune between adjacent layers only because they are more correlated with each other compared to distant layers.

2.3 Multi-label categorization

Each test instance is classified into categories with probabilities/scores by applying HFT-CNN. We

Dataset	#L	Tr	Te	C
RCV1	4	23,149	781,265	103
Amazon670K	9	490,449	153,025	670,091

Table 1: Data Statistics: #L shows the depth of a hierarchy. Tr and Te refer to the # of training and test data, respectively. C indicates the total # of categories.

then utilize a constraint of a HS to obtain final results which differs from the existing work on non-hierarchical flat model (Johnson and Zhang, 2015a; Liu et al., 2017). This is done by using two scoring functions: One is a Boolean Scoring Function (BSF). Another is a Multiplicative Scoring Function (MSF). Both functions set a threshold value and categories whose scores exceed the threshold value are considered for selection. The difference is that BSF has a constraint that a category can only be selected if its ancestor categories are selected. MSF does not have such a constraint, i.e., we extracted all the categories whose scores exceeded the threshold value and sorted them in descending order as the system's assignments.

3 Experiments

3.1 Data and HFT-CNN model setting

We selected two benchmark datasets having a HS from the extreme classification repository[2]: RCV1 (Lewis et al., 2004) and Amazon670K (Leskovec and Krevl, 2015). All the documents in RCV1 and item descriptions in Amazon670K are tagged by using Tree Tagger (Schmid, 1995). We used nouns, verbs, and adjectives. We then applied fastText. Each dataset has an official training and test sets. We used each fold in the experiments. We choose titles from the training and test set on RCV1. The maximum number of words in the title was 13 words. Each text of Amazon670K consists of a product name and its item description. We extracted the first 13 words from each item description and used them in the experiments. Table 1 presents the statistics on the datasets. We divided the training data into two folds; we used 5% to tuning the parameters, and the remains to train the models. Our model setting is shown in Table 2[3]. In the experiments, we run three times for each model and obtained the averaged performance.

[2]manikvarma.org/downloads/XC/XMLRepository.html

[3]Our source code including Chainer's version of XML-CNN is available at: HTTP://github.com/ShimShim46/HFT-CNN.

3.2 Evaluation Metrics

We used the standard F1 measure. Furthermore, we evaluated our method by two rank-based evaluation metrics: the precision at top k, P@k and the Normalized Discounted Cumulated Gains, NDCG@k which are commonly used for comparing extreme multi-label classification methods (Liu et al., 2017). We calculated P@k and NDCG@k for each test data and then obtained an average over all the test data.

3.3 Basic results

We compared HFT-CNN with a method which has hierarchical-based categorization but without fine-tuning (WoFT-CNN) and Flat model to examine the effect of the fine-tuning. WoFT-CNN shows that we independently trained parameters of CNN for each level and trained parameters are not transferred. Flat means that we simply applied our CNN model. The results are shown in Table 3. The HFT-CNN is better than WoFT-CNN and Flat model except for Micro-F1 obtained by WoFT-CNN(M) in Amazon670K. We also found that the overall results obtained by MSF were better to those obtained by BSF.

3.4 Comparison with state-of-the-art method

We chose XML-CNN as a comparative method because their method attained at the best or second best compared to the seven existing methods in six benchmark datasets (Liu et al., 2017). Original XML-CNN is implemented by using Theano[4], while we implemented HFT-CNN by Chainer[5]. In order to avoid the influence of differences in libraries, we implemented XML-CNN by Chainer and compared it with HFT-CNN. We used the author-provided implementation in Chainer's version of XML-CNN. We recall that we set convolutional filters with the window sizes to (2,3,4) and the stride size to 1 because of short text. To make a fair comparison, we also evaluated XML-CNN with the same window sizes and stride size as HFT-CNN.

Liu et al. evaluated their method by using P@k and NDCG@k. We used their metrics as well as F1 measure. We did not set a threshold value on BSF and MSF when we evaluated by using these metrics, but instead, we used a ranked list of cate-

[4]https://drive.google.com/file/d/1Wwy1MNkrJRXZM3WNZNywa94c2-iEh_6U/view

[5]https://chainer.org

Description	Values	Description	Values
Input word vectors	fastText	Filter region size	(2,3,4)
Stride size	1	Feature maps (m)	128
Filters	128×3	Activation function	ReLu
Pooling	1-max pooling	Dropout	Randomly selected
Dropout rate1	0.25	Dropout rate2	0.5
Hidden layers	1,024	Batch sizes	100
Learning rate	Predicted by Adam	Epoch	40 with early stopping
Loss function	BCE loss over sigmoid activation	Threshold value for BSF and MSF	0.5

Table 2: HFT-CNN model settings: Dropout rate1 shows dropout immediately after embedding layer, and Dropout rate2 refers to dropout in a fully connected layer.

Metric	RCV1				
F1	HFT(B)	WoFT(B)	HFT(M)	WoFT(M)	Flat
Micro	79.87	79.69	**80.29**	80.06	79.51
Macro	50.31	49.59	**51.40**	50.64	47.71
Metrics	Amazon670K				
	HFT(B)	WoFT(B)	HFT(M)	WoFT(M)	Flat
Micro	49.74	50.12	$*$50.94	**50.94**	49.10
Macro	6.78	6.37	**9.87**	8.68	5.73

Table 3: Basic results: (B) and (M) refer to a BSF and MSF, respectively. Bold font shows the best result within each line. The method marked with "$*$" indicates the score is not statistically significant compared to the best one. We used a t-test, p-value < 0.05.

Metric	RCV1			
	HFT(B)	HFT(M)	XML(1)	XML(2)
P@1	92.60	**93.29**	92.93	92.55
P@3	$*$77.56	**77.70**	77.18	76.80
P@5	$*$53.96	**54.23**	53.85	53.60
G@1	92.60	**93.29**	92.93	92.55
G@3	$*$88.47	**88.79**	88.16	87.75
G@5	89.37	**89.81**	89.19	88.80
Metric	Amazon670K			
	HFT(B)	HFT(M)	XML(1)	XML(2)
P@1	**86.54**	85.39	84.62	84.12
P@3	**66.25**	65.34	64.83	64.48
P@5	**51.09**	$*$50.84	49.93	49.75
G@1	**86.54**	85.39	84.62	84.12
G@3	**76.26**	75.05	74.50	74.13
G@5	**72.84**	71.46	70.99	70.74

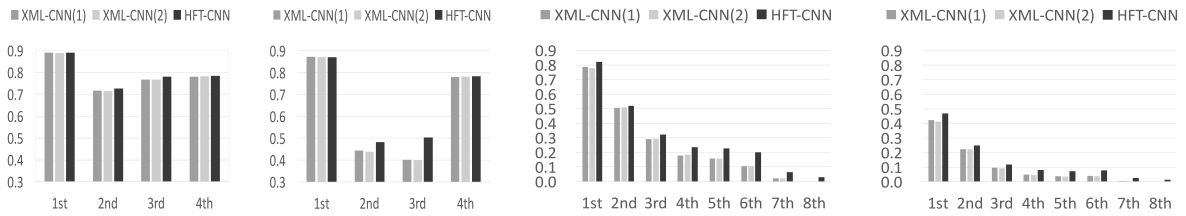
Table 4: Comparative results: "1" and "2" of XML show the stride size=1 and 2 by XML-CNN, respectively. "G" stands for NDCG.

gories assigned to the test instance. The results are shown in Table 4. HFT-CNN with BSF/MSF has the best scores with statistical significance compared to both of the XML-CNNs. On RCV1, HFT-CNN(B) in P@1 and NDCG@1 were worse than XML-CNN(1), while HFT-CNN(M) with the same metrics were statistically significant compared to XML-CNN(1). This is not surprising because hierarchical fine-tuning does not contribute to the accuracy at the top level as the trained parameters on the top level have not changed in the level.

We also examined the affection on each system performance by the depth of a hierarchical structure. Figure 2 shows Micro-F1 at each hierarchical level. The deeper the hierarchical level, the worse the system's performance. However, HFT-CNN is still better than XML-CNNs. The improvement by MSF was $1.00 \sim 1.34\%$ by Micro-F1 and $3.77 \sim 10.07\%$ by Macro-F1 on RCV1. On Amazon670K, the improvement was $1.10 \sim 9.26\%$ by Micro-F1 and $1.10 \sim 3.60\%$ by Macro-F1. This shows that hierarchical fine-tuning fits to learn the hierarchical category structure.

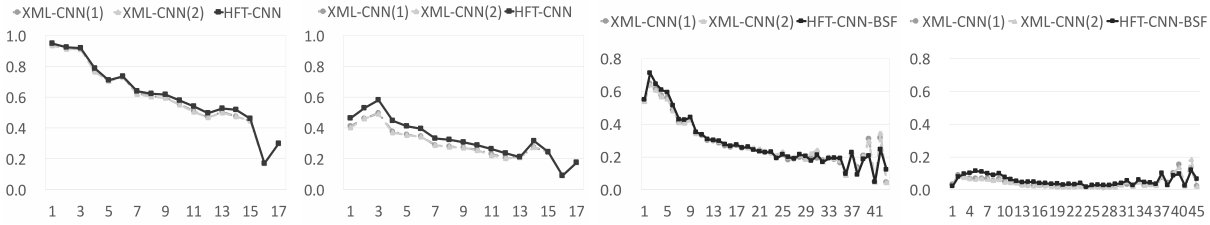
We recall that we focused on the multi-label problem. Figures 3 illustrates Micro-F1 and Macro-F1 against the number of categories per short text. We can see from RCV1 in Figure 3 that Micro-F1 obtained by HFT-CNN and XML-CNNs were not statistically significant difference in the number of categories, while Macro-F1 by HFT-CNN except for the number of 13 categories was constantly better to XML-CNNs. On Amazon670K data, when the number of categories assigned to the short text is less than 38, HFT-CNN was better than XML-CNNs or HFT-CNN was not statistically significant compared to XML-CNNs by both F1-scores. However, when it exceeds 39, HFT-CNN was worse than XML-CNNs. One possible reason is the use of BSF: a category can only be selected if its ancestor categories are selected. Therefore, once the test data could not be classified into categories correctly, their child categories also cannot be correctly assigned to the test data. In contrast, as shown in Figure 5, HFT-CNN by MSF was better than XML-CNNs in both Micro and Macro F1 even in the deep level of a hierarchy. From the observations, a robust scoring function is needed for further improvement.

It is important to note that how the ratio of training data affects the overall performance as we focused on the data sparsity problem. Figure



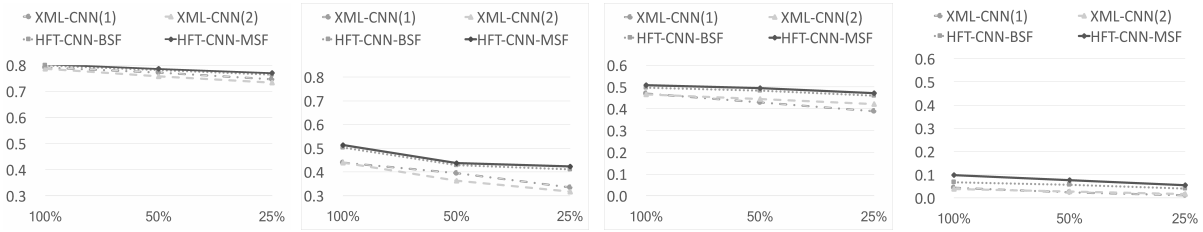
(a) Micro-F1 (RCV1) (b) Macro-F1(RCV1) (c) Micro-F1 (Amazon670K) (d) Macro-F1 (Amazon670K)

Figure 2: Performance in each hierarchical level: HFT-CNN used BSF



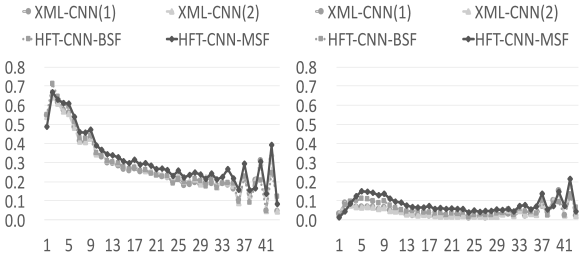
(a) Micro-F1 (RCV1) (b) Macro-F1 (RCV1) (c) Micro-F1 (Amazon670K) (d) Macro-F1 (Amazon670K)

Figure 3: Performance against the # of categories per short text: HFT-CNN used BSF



(a) Micro-F1 (RCV1) (b) Macro-F1 (RCV1) (c) Micro-F1 (Amazon670K) (d) Macro-F1 (Amazon670K)

Figure 4: Performance against a ratio of the training data



(a) Micro-F1 (Amazon670K) (b) Macro-F1 (Amazon670K)

Figure 5: Performance against the # of categories per short text: Comparison with HFT-CNN by MSF and other methods

4 shows Micro and Macro-F1 against a ratio of the training data. Overall, the curves show that more training helps the performance, while the curves obtained by HFT-CNN drop slowly compared to other methods in both datasets and evaluation metrics. From the observations mentioned in the above, we can conclude that fine-tuning works well, especially in the cases that the number of the training data per category is small.

4 Conclusion

We have presented an approach to multi-label categorization for short text. The comparative results with XML-CNN showed that HFT-CNN is competitive, especially for the cases that there exists only a small amount of training data. Future work will include: (i) incorporating lexical semantics such as named entities and domain-specific senses for further improvement, (ii) extending the method to utilize label dependency constraints (Bi and Kwok, 2011), and (iii) improving the accuracy of the top ranking categories to deal with P@1 and NDCG@1 metrics.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. This work is supported in part by Support Center for Advanced Telecommunications Technology Research, Foundation and the Grant-in-aid for the Japan Society for the Promotion of Science, No.17K00299.

References

W. Bi and J. T. Kwok. 2011. Multi-Label Classification on Tree- and DAG-Structured Hierarchies. In *Proc. of the 28th Internation Conference on Mathine Learning*, pages 17–24.

M. Chen, 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*, pages 1776–1781.

G. E. Hinton, N. Srivastava, A. Krizhevsky, H. Sutskever, and R. R. Salakhutdinov. 2012. Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. In *arXiv preprint arXiv:1207.0580*.

R. Johnson and T. Zhang. 2015a. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. In *Proc of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112.

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

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proc. of the 15th Conference of the European Chapter of the Association for Conputational Linguistics*, pages 427–431.

Y. Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

J. Y. Lee and F. Dernoncourt. 2016. Sequential Short-Text classification with Recurrent and Convolutional Neural Networks. In *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 515–520.

J. Leskovec and A. Krevl. 2015. SNAP Datasets: Stanford Large Network Dataset Collection.

D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397.

J. Liu, W-C. Chang, Y. Wu, and Y. Yang. 2017. Deep Learning for Extreme Multi-Label Text Classification. In *Proc. of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124.

X. H. Phan, 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*, pages 91–100.

H. Schmid. 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proc. of the EACL SIGDAT Workshop*, pages 47–50.

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

F. Wang, Z. Wang, Z. Li, and J-R. Wen. 2008. Concept-Based Short Text Classification and Ranking. In *Proc. of the 23rd ACM International Conference on Information and Knowledge Management*, pages 1069–1078.

J. Wang, Z. Wang, D. Zhang, and J. Yan. 2017. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In *Proc. of the 26th International Joint Conference on Artificial Intelligence*, pages 2915–2921.

P. Wang, 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*, pages 352–357.

W. Wu, 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*, pages 481–492.

Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 1480–1489.

R. Zhang, H. Lee, and D. Radev. 2016. Dependency Sensitive Convolutional Neural Networks for Modeling Sentences and Documents. In *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 1512–1521.

Y. Zhang, M. Lease, and B. C. Wallace. 2017. Exploiting Domain Knowledge via Grouped Weight Sharing with Application to Text Categorization. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 155–160.

Y. Zhang and B. C. Wallace. 2015. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. In *Proc. of the 8th International Joint Conference on Natural Language Processing*, pages 253–263.