A Hierarchical Neural Attention-based Text Classifier

Koustuv Sinha [1,2], Yue Dong [1,2], Jackie C.K. Cheung [1,2] and Derek Ruths [1]
[1] School of Computer Science, McGill University, Canada
[2] Montreal Institute of Learning Algorithms, Canada
{koustuv.sinha, yue.dong2, jcheung, derek.ruths }
@{mail.mcgill.ca, mail.mcgill.ca, cs.mcgill.ca, mcgill.ca}

Abstract

Deep neural networks have been displaying superior performance over traditional supervised classifiers in text classification. They learn to extract useful features automatically when sufficient amount of data is presented. However, along with the growth in the number of documents comes the increase in the number of categories, which often results in poor performance of the multiclass classifiers. In this work, we use external knowledge in the form of topic category taxonomies to aide the classification by introducing a deep hierarchical neural attention-based classifier. Our model performs better than or comparable to state-of-the-art hierarchical models at significantly lower computational cost while maintaining high interpretability.

1 Introduction

A large number of documents are being generated all over the world everyday, and as a result automatic text classification has become an essential tool for searching, retrieving, and managing the text (Allahyari et al., 2017). There has been an increasing trend in developing data-driven neural text classifiers (Collobert et al., 2011; Lai et al., 2015; Zhang et al., 2015; Yogatama et al., 2017; Conneau et al., 2017), due to their ability to handle large-scale corpora and their robustness in automatic feature extraction.

However, text classification has become increasingly challenging as the number of categories grows with continually expanding corpus. To alleviate this problem, one form of the external knowledge – class taxonomy – has been introduced to aid the classification in a hierarchical fashion (Koller and Sahami, 1997). In general, hierarchical classifiers can be categorized into two broad approaches: *local* (*top-down* and *bottom-up*) and *global* (or *big-bang*) (Silla and Freitas, 2011). The local approaches create a unique classifier for each parent node in the taxonomy (Liu et al., 2001; Quinn and Laier, 2006; Vens et al., 2008; Kowsari et al., 2017), while global approaches create a single classifier for the entire taxonomy (Silla Jr and Freitas, 2009).

Kowsari et al. (2017) recently proposed a hierarchical neural-based model called HDLTex, which displayed superior performance over traditional non-neural-based models with a top-down structure. However, HDLTex suffers the inherited disadvantage of the *top-down* approach: the number of sub-models grows exponentially with respect to the number of sub-trees. This is especially problematic in HDLTex, as it uses deep networks with a large number of parameters for the sub-models, and the combined model itself grows exponentially with the depth of taxonomy.

In contrast, we propose a unified *global* deep neural-based classifier that overcomes the problem of exploding models. The backbone of our approach is one encoder-decoder structure that sequentially predicts the class label of the next level, conditioned on a dynamic document representation obtained based on a variant of an attention mechanism (Bahdanau et al., 2015). The contribution of our paper is as follows:

1. We propose an end-to-end *global* neural attention-based model for hierarchical classification, which performs better than the state-of-the-art hierarchical classifier at lower computation cost.

2. We empirically show that the use of hierarchical taxonomy provides a robust classifier, by comparing with state-of-the-art flat classifiers.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 817–823
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

2 Literature Review

Traditional text classification methods focus on selecting a good set of features (for example,TF-IDF (Salton and Buckley, 1987)) to represent the documents and employing non-linear classifiers such as SVM (Dumais et al., 1998; Joachims, 1999; Tong and Koller, 2001), decision trees (Apté et al., 1994), or Naive Bayes (McCallum et al., 1998; Kim et al., 2006) methods for text classification. More recent work has employed deep neural networks to merge feature extraction and classification into one joint process, where the model parameters can be learned through back-propagation (Xue et al., 2008; Lai et al., 2015; Zhang et al., 2015). A common theme in these convolutional neural networks (CNN)-based or recurrent neural network (RNN)-based approaches is to create a document representation from either the last hidden state of the RNN or via some pooling operations on all hidden states.

Furthermore, the attention mechanism (Bahdanau et al., 2015; Sutskever et al., 2014) has been adapted for these CNN/RNN structures for text classification (Lin et al., 2017), providing high interpretability and allowing us to inspect which parts of the text are discriminative for a particular sample.

In addition, external knowledge has been examined as a way to boost the performance of text classifiers (Collobert and Weston, 2008a; Ngiam et al., 2011; Howard and Ruder, 2018). One form of external knowledge is built on top of the hierarchical relations of the classes (Koller and Sahami, 1997), where a class taxonomy is used to improve the performance of the end-level classification[1]. Most of the hierarchical classifiers[2] perform classification by navigating through the hierarchy in *top-down* approaches (Liu et al., 2001; Quinn and Laier, 2006; Vens et al., 2008), where a local classifier is constructed at each parent node. The state-of-the-art hierarchical classifier HDLTex is proposed by Kowsari et al. (2017). It combines deep neural networks in the *top-down* fashion where a separate neural network (either CNN or RNN) is built at each parent node to classify its children.

3 Model

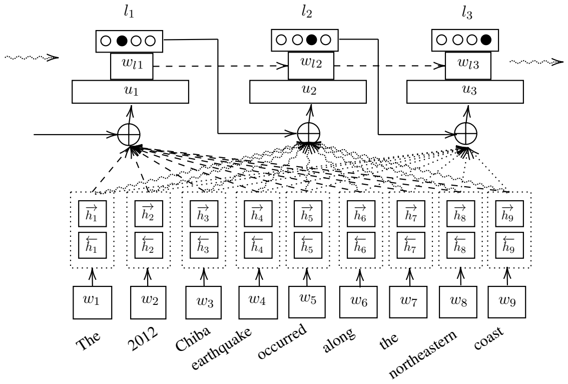

l_1
l_2
l_3
w_{l1}
w_{l2}
w_{l3}
u_1
u_2
u_3
$\overrightarrow{h_1}$ $\overrightarrow{h_2}$ $\overrightarrow{h_3}$ $\overrightarrow{h_4}$ $\overrightarrow{h_5}$ $\overrightarrow{h_6}$ $\overrightarrow{h_7}$ $\overrightarrow{h_8}$ $\overrightarrow{h_9}$
$\overleftarrow{h_1}$ $\overleftarrow{h_2}$ $\overleftarrow{h_3}$ $\overleftarrow{h_4}$ $\overleftarrow{h_5}$ $\overleftarrow{h_6}$ $\overleftarrow{h_7}$ $\overleftarrow{h_8}$ $\overleftarrow{h_9}$
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9
The
2012
Chiba earthquake
occurred
along
the
northeastern
coast

Figure 1: Proposed model architecture

Our proposed model (Figure 1) consists of three parts: 1) a bidirectional LSTM encoder (Hochreiter and Schmidhuber, 1997) that transforms each word into vector representations based on their context. 2) an attention module that helps to generate dynamic document representations across different level of classification, 3) multi-layer perceptron (MLP) classifiers at each level that makes the prediction of classes at that level based on the dynamically generated document representation and the level masking.

Our hierarchical classification model can be viewed as a sequence-to-sequence model, where a sequence of word embeddings is used to generate a sequence of hierarchical class labels. In addition, we employ a modified attention module from the traditional attention mechanism used in sequential generation tasks (Bahdanau et al., 2015; Sutskever et al., 2014). Instead of computing attention weights conditioned on the hidden state of the decoder at time step i, we condition on the parent category embedding c_{k-1}. This is intuitive in our setting as the document representation should depend on the parent class predicted by the model.

Formally, suppose we are given a document with n tokens $D = (w_1, w_2, ..., w_n)$ and its category labels of m levels $C = (c_1, \ldots, c_m), c_k \in \{c_1^{l_k}, \ldots, c_{s_k}^{l_k}\}$ where l_k indicates the k-th level of the class taxonomy and s_k represents the number of classes in level k [3]. A bidirectional LSTM is

[1] Classifiers that do not take into account the hierarchy and are only concerned with predicting the leaf nodes are termed *flat* classifiers in this work.

[2] We use the term "hierarchical classifiers" to refer the models that follow the external taxonomy of class labels, which is substantially different from hierarchical attention networks (Yang et al., 2016). In Yang et al. (2016), hierarchical attention networks refer to the hierarchical nature of their attention mechanism; the model attends to the sentences first and then attends to the words.

[3] We suppose w_i and c_i are word embeddings and class embedding respectively.

first used to extract features of the document:

$$\begin{aligned}\overrightarrow{h_t} &= \overrightarrow{LSTM}(w_t, \overrightarrow{h_{t-1}}),\\ \overleftarrow{h_t} &= \overleftarrow{LSTM}(w_t, \overleftarrow{h_{t+1}}).\end{aligned} \tag{1}$$

The encoder's hidden states $H = (h_1, \ldots, h_n)$ are constructed by the concatenation of $(\overrightarrow{h_t})$ and $(\overleftarrow{h_t})$ as $h_i = [\overrightarrow{h_t}, \overleftarrow{h_t}]$.

When classifying the class label at level k, we first form contextual word features $\bar{H}_k$ by concatenating the previously predicted category embedding c_{k-1} (parent) with each of the encoder's outputs $H = (h_1, \ldots, h_n)$:

$$\bar{H}_k = H \oplus c_{k-1}. \tag{2}$$

Then, we transform these n vectors in $\bar{H}_k$ into n attention scores (scalars) through a series of linear and non-linear transformations:

$$a_k = \text{softmax}(w_{s2}\text{tanh}(W_{s_1}\bar{H}_k^T)). \tag{3}$$

As one single attention distribution might only focus on a specific component of the semantics in the document, we follow Lin et al. (2017)'s work to perform m hops of attention and form the multi-head attention matrix A_k $(m \times n)$. To encourage diversity over the multiple hops of the attention distributions, we employ the Frobenius norm penalty (Lin et al., 2017) $P = \left\|A_k A_k^\top - I\right\|_F^2$ to force the attention hops to focus on different aspects of the semantics.

The document representation for level k is obtained by multiplying the multi-head attention matrix and the contextual word features:

$$D_k = W_{s_3} A_k \bar{H}_k. \tag{4}$$

Finally, a two layered multi-layer perceptron (MLP) is employed to classify the category at level k:

$$\begin{aligned} d_k &= \text{RELU}(W_D[D_k, d_k - 1]),\\ y_k &= \text{softmax}(W_k d_k)\end{aligned} \tag{5}$$

Normally, the softmax in Equation 5 is computed over all class labels across the entire taxonomy levels. This is not desirable when the taxonomy is deep and the number of classes is large. We solve this by employing a *level masking* technique where we mask out all the classes that are not in the current classification level k. The loss is then calculated as the joint cross entropy loss among all levels of the taxonomy: $l = \sum_{i=1}^{m} l_i$.

	DBpedia	WOS
Level 1 Categories	9	7
Level 2 Categories	70	134
Level 3 Categories	219	NA
Number of documents	381,025	46,985
Mean document length	106.9	200.7

Table 1: Dataset Comparison

4 Experimental Setup

Dataset Two datasets are used for our experiments: Web of Science (WOS) and DBpedia. Web of Science (WOS) is a hierarchical two-level taxonomy dataset that contains 46,985 documents collected from *Web of Science* (Reuters, 2012) by Kowsari et al. (2017). Despite its small size, WOS is used as a benchmark dataset for hierarchical classification as it provides the raw text for deep neural models to train on[4].

As deep learning models usually contain a large number of parameters that need to be learned, to prevent over-fitting (Lawrence et al., 1997; Srivastava et al., 2014) we usually need a large dataset to train upon. Thus, we curated a bigger dataset with hierarchical labels from Wikipedia meta information provider DBpedia[5]. Compared to WOS, our DBpedia dataset is larger in two aspects: the number of data instances and the number of hierarchical levels (Table 1). The DBpedia ontology was first used in Zhang et al. (2015) for flat text classification. We instead use the DBpedia ontology to construct a dataset with a three-level taxonomy of classes. In order to ensure enough documents per-class, we only extract leaf-classes with more than 200 documents. We also randomly subsample 3,000 documents per category to balance the number of leaf-level categories. This results in 381,025 documents in total, which we split into 90% for training (from which 10% were kept aside for validation) and 10% on testing, on which we report our classification metrics[6].

Baselines State-of-the-art *flat* classifiers such as FastText (Joulin et al., 2017), Bi-directional

[4] The LSHTC dataset (Partalas et al., 2015) has been widely used as a benchmark for hierarchical text classification. However, the raw texts are not available which makes it difficult to extract features for modern neural approaches. Instead, only the tf-idf vectors are provided as inputs with no option to retrieve the original text (even after consulting with the original authors we were unable to procure it).

[5] http://wiki.dbpedia.org/

[6] Our code and data will be released at https://github.com/koustuvsinha/hier-class

	DBpedia				WOS		
Flat Baselines				Overall			Overall
FastText				86.2			61.3
BiLSTM + MLP + Maxpool				94.20			**77.69**
BiLSTM + MLP + Meanpool				**94.68**			73.08
Structured Self Attention (m=1)				94.04			77.40
Hierarchical Models	l_1	l_2	l_3	Overall	l_1	l_2	Overall
HDLTex (5B params)	99.26	97.18	95.5	92.10	90.45	84.66	76.58
Our model (34M params)	99.21	96.03	95.32	**93.72**	89.32	82.42	**77.46**

Table 2: Test accuracy on the WOS and DBpedia datasets. The flat baseline models are trained without the hierarchical taxonomy of classes and therefore only have results on the leaf-node classification.

LSTM with max/mean pooling (Collobert and Weston, 2008b; Lee and Dernoncourt, 2016) and the Structured Self-attentive classifier (Lin et al., 2017) are used for the comparison. We noticed that using the default hyperparameters of the Structured Self-attentive classifier with high attention hops ($m >= 8$) performed poorly compared to use just one attention hop ($m = 1$). Therefore, we reported the results of using one attention hop ($m = 1$) as our baselines for fair comparison. We also compare our classifier to the state-of-the-art hierarchical classifier HDLTex (Kowsari et al., 2017).

Hyperparameters We use 300-dimensional word embeddings which are randomly initialized and fine-tuned during training. Two-layer Bidirectional LSTM with 300 hidden units in each layer are employed. In the multi-head attention mechanism, we use 4 heads (hops) with 0.1 Frobenius norm penalty because it gives the best validation performance. The final fully-connected MLP layer W_D has 1200 hidden units. In addition, we add 0.4 dropout on BiLSTM layers and MLP layers to prevent over-fitting.

For optimization, we use the standard Adam optimizer (Kingma and Ba, 2014) with the learning rate of 0.001, weight decay of 10^{-4} and 10^{-6} for WOS and DBpedia, respectively. The gradients are clipped to 0.5 in order to prevent exploding gradients. All the results are obtained after 25 epochs of training. After every 10 epochs, we reduce the learning rate by half if the validation accuracy is not improving. We employ early-stopping to select the best model. In addition, a weighted loss function is utilized to balance the performance on under-represented classes.

Hierarchical Evaluation For evaluating hierarchical models, we present the *teacher-forcing* result on each level, such as l_1, l_2 and l_3. This indicates the per-level classification performance when we provide the true parent class to the classifier while predicting the next class. However, this is not desirable as during inference we should not have access to the correct parent class. Hence we also present the *Overall* score in Table 2, where the classifier uses its own prediction as the parent class.

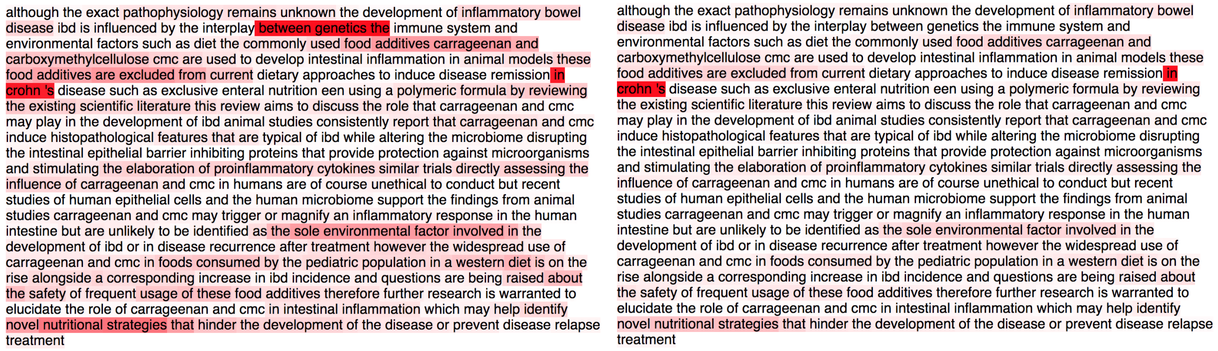
5 Results

Our model is significantly better than the existing state-of-the-art hierarchical baseline (Table 2). Although, we also see that both hierarchical classifiers (ours and HDLTex) perform comparably with or slightly worse than the state-of-the-art flat classifiers in terms of accuracy. However, the robustness analysis we performed in Table 3 indicates that hierarchical models are more robust in their errors since most of the errors generated by hierarchical classifiers remain within the correct tree of the parent class, while flat classifiers do worse. For example, on WOS, 88.57% of all classified data by our hierarchical model is within the correct subtree compared to 85.56% for the flat classifier.

Classifier	Correct parent	Predicted parent
Flat classifier - BiLSTM Max Pooling	90.74	85.56
Hierarchical approach - Our model	93.03	88.57

Table 3: Robustness analysis of taxonomy on the WOS dataset. We compare the success rate of our model and the BiLSTM flat classifier. The success rate is defined as the number of times the predicted class is within the same subtree as the correct parent. We calculate this in two scenarios: 1. when the true parent class is manually provided, or *teacher-forced* (Correct parent), and 2. when the true parent class is predicted by our model (Predicted parent)

Interestingly, the class taxonomy seems to be more beneficial in boosting the performance of hierarchical classifiers on WOS than DBpedia. The hierarchical classifiers perform better on the



although the exact pathophysiology remains unknown the development of inflammatory bowel disease ibd is influenced by the interplay between genetics the immune system and environmental factors such as diet the commonly used food additives carrageenan and carboxymethylcellulose cmc are used to develop intestinal inflammation in animal models these food additives are excluded from current dietary approaches to induce disease remission in crohn 's disease such as exclusive enteral nutrition een using a polymeric formula by reviewing the existing scientific literature this review aims to discuss the role that carrageenan and cmc may play in the development of ibd animal studies consistently report that carrageenan and cmc induce histopathological features that are typical of ibd while altering the microbiome disrupting the intestinal epithelial barrier inhibiting proteins that provide protection against microorganisms and stimulating the elaboration of proinflammatory cytokines similar trials directly assessing the influence of carrageenan and cmc in humans are of course unethical to conduct but recent studies of human epithelial cells and the human microbiome support the findings from animal studies carrageenan and cmc may trigger or magnify an inflammatory response in the human intestine but are unlikely to be identified as the sole environmental factor involved in the development of ibd or in disease recurrence after treatment however the widespread use of carrageenan and cmc in foods consumed by the pediatric population in a western diet is on the rise alongside a corresponding increase in ibd incidence and questions are being raised about the safety of frequent usage of these food additives therefore further research is warranted to elucidate the role of carrageenan and cmc in intestinal inflammation which may help identify novel nutritional strategies that hinder the development of the disease or prevent disease relapse treatment

(a) Level 1 - correct class : Medical

although the exact pathophysiology remains unknown the development of inflammatory bowel disease ibd is influenced by the interplay between genetics the immune system and environmental factors such as diet the commonly used food additives carrageenan and carboxymethylcellulose cmc are used to develop intestinal inflammation in animal models these food additives are excluded from current dietary approaches to induce disease remission in crohn 's disease such as exclusive enteral nutrition een using a polymeric formula by reviewing the existing scientific literature this review aims to discuss the role that carrageenan and cmc may play in the development of ibd animal studies consistently report that carrageenan and cmc induce histopathological features that are typical of ibd while altering the microbiome disrupting the intestinal epithelial barrier inhibiting proteins that provide protection against microorganisms and stimulating the elaboration of proinflammatory cytokines similar trials directly assessing the influence of carrageenan and cmc in humans are of course unethical to conduct but recent studies of human epithelial cells and the human microbiome support the findings from animal studies carrageenan and cmc may trigger or magnify an inflammatory response in the human intestine but are unlikely to be identified as the sole environmental factor involved in the development of ibd or in disease recurrence after treatment however the widespread use of carrageenan and cmc in foods consumed by the pediatric population in a western diet is on the rise alongside a corresponding increase in ibd incidence and questions are being raised about the safety of frequent usage of these food additives therefore further research is warranted to elucidate the role of carrageenan and cmc in intestinal inflammation which may help identify novel nutritional strategies that hinder the development of the disease or prevent disease relapse treatment

(b) Level 2 - correct class : Crohn's disease

Figure 2: WOS dataset attention rereading per level. Highlighted words indicate the attented words. Stronger color denote higher focus of attention. We note that the attention spread becomes much more focused in Level 2 compared to its parent Level 1.

leaf-node level classification of WOS than that on DBpedia. We observe this behaviour due to the dataset of DBpedia being shorter in average length making it easier to classify for *flat* classifiers, hence hierarchical classifiers overfit on the training data.

In addition to the performance improvement on both datasets over HDLTex, our model takes significantly less time and resources to train, especially when the dataset is large in terms of the intermediate non-leaf nodes in the output taxonomy. As HDLTex needs to build one sub-classifier for each parent nodes, the number of sub-classifiers grows quickly. For example, there are 80 parent nodes in the taxonomy of the DBpedia dataset and HDLTex needs to build 80 RNNs, where each sub-classifier contains around 67 million parameters. As a consequence, we can barely fit the whole model of HDLTex on our CPU [7] because it requires 60 GB RAM to build these 80 deep neural networks.

6 Discussion

Analysis of Attention The intuition behind building dynamic document representations, using multiple attentions across different hierarchical levels, is to have a re-reading effect over the taxonomy. When we first encounter an article as humans, we tend to read it carefully, but on subsequent reads we can easily identify the key aspects of the article. We find in our exploratory experiments the attention vectors behave exactly the same. For the first level, the attention values are more spread out to help our classifier pick various important aspects of the article, but on the subsequent levels, the attention is more focused towards specific keywords for that subclass, as the example shown in Figure 2 [8]. We perform additional qualitative analysis of attention spread which is provided in Appendix.

7 Conclusion

In this work, we propose a light-weight neural-based hierarchical classifier that performs better than or comparable to the state-of-the-art hierarchical model at lower computation cost. Our model employs an adapted version of attention to represent documents dynamically through the hierarchy, which provides additional interpretability of the dynamic document representations. In addition, we demonstrate that the robustness of flat text classification can be improved by using external knowledge such as a hierarchical taxonomy. As a future direction, we will advance our model to automatically construct the hierarchical taxonomy in order to improve text classification with a large number of classes.

Acknowledgements

We thank Compute Canada and Calcul Quebec for providing the GPU compute resources. We would also express our appreciation to our sponsors, Pierre Arbour Foundation and FRQNT for supporting Koustuv Sinha and NSERC for supporting Yue Dong for this research.

[7] It is not possible to fit the entire model in one GPU as our best GPU has the RAM capacity of 12GB, one needs to have multiple GPU's and parallel execution for this task.

[8] We use the same visualization script as of Lin et al. (2017).

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. 2017. A brief survey of text mining: Classification, clustering and extraction techniques. In *Proceedings of KDD Bigdas, Halifax, Canada.*

Chidanand Apté, Fred Damerau, and Sholom M Weiss. 1994. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems (TOIS)*, 12(3):233–251.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR).*

Ronan Collobert and Jason Weston. 2008a. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Ronan Collobert and Jason Weston. 2008b. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

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

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1107–1116.

Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. 1998. Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 148–155. ACM.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung Hyon Myaeng. 2006. Some effective techniques for naive bayes text classification. *IEEE transactions on knowledge and data engineering*, 18(11):1457–1466.

Diederik P Kingma and Jimmy Lei Ba. 2014. Adam: Amethod for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Representations.*

Daphne Koller and Mehran Sahami. 1997. Hierarchically classifying documents using very few words. Technical report, Stanford InfoLab.

Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. 2017. HDLTex: Hierarchical deep learning for text classification. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 364–371.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273.

Steve Lawrence, C Lee Giles, and Ah Chung Tsoi. 1997. Lessons in neural network training: Overfitting may be harder than expected. In *AAAI/IAAI*, pages 540–545. Citeseer.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 515–520.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations (ICLR 2017).*

Shaohui Liu, Mingkai Dong, Haijun Zhang, Rong Li, and Zhongzhi Shi. 2001. An approach of multi-hierarchy text classification. In *Info-tech and Info-net, 2001. Proceedings. ICII 2001-Beijing. 2001 International Conferences on*, volume 3, pages 95–100. IEEE.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*. Citeseer.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696.

Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artieres, George Paliouras, Eric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Galinari. 2015. Lshtc: A benchmark for large-scale text classification. *arXiv preprint arXiv:1503.08581*.

Michael J Quinn and Mary L Laier. 2006. Method and apparatus for fast lookup of related classification entities in a tree-ordered classification hierarchy. US Patent 7,032,072.

Thomson Reuters. 2012. Web of science.

Gerard Salton and Chris Buckley. 1987. Term weighting approaches in automatic text retrieval. Technical report, Cornell University.

Carlos N Silla and Alex A Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72.

Carlos N Silla Jr and Alex A Freitas. 2009. A global-model naive bayes approach to the hierarchical prediction of protein functions. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 992–997. IEEE.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel. 2008. Decision trees for hierarchical multi-label classification. *Machine Learning*, 73(2):185.

Gui-Rong Xue, Dikan Xing, Qiang Yang, and Yong Yu. 2008. Deep classification in large-scale text hierarchies. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 619–626. ACM.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Dani Yogatama, Chris Dyer, Wang Ling, and Phil Blunsom. 2017. Generative and discriminative text classification with recurrent neural networks. *arXiv preprint arXiv:1703.01898*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.