HyperText: Endowing FastText with Hyperbolic Geometry

Yudong Zhu Di Zhou Jinghui Xiao Xin Jiang Xiao Chen Qun Liu
Huawei Noah's Ark Lab
{zhuyudong3,zhoudi7,xiaojinghui4,Jiang.Xin,
chen.xiao2,qun.liu}@huawei.com

Abstract

Natural language data exhibit tree-like hierarchical structures such as the hypernym-hyponym relations in WordNet. FastText, as the state-of-the-art text classifier based on shallow neural network in Euclidean space, may not model such hierarchies precisely with limited representation capacity. Considering that hyperbolic space is naturally suitable for modeling tree-like hierarchical data, we propose a new model named HyperText for efficient text classification by endowing FastText with hyperbolic geometry. Empirically, we show that HyperText outperforms FastText on a range of text classification tasks with much reduced parameters.

1 Introduction

FastText (Joulin et al., 2016) is a simple and efficient neural network for text classification, which achieves comparable performance to many deep models like char-CNN (Zhang et al., 2015) and VDCNN (Conneau et al., 2016), with a much lower computational cost in training and inference. However, natural language data exhibit tree-like hierarchies in several respects (Dhingra et al., 2018) such as the taxonomy of WordNet. In Euclidean space the representation capacity of a model is strictly bounded by the number of parameters. Thus, conventional shallow neural networks (e.g., FastText) may not represent tree-like hierarchies efficiently given limited model complexity.

Fortunately, hyperbolic space is naturally suitable for modeling the tree-like hierarchical data. Theoretically, hyperbolic space can be viewed as a continuous analogue of trees, and it can easily embed trees with arbitrarily low distortion (Krioukov et al., 2010). Experimentally, Nickel and Kiela (2017) first used the Poincaré ball model to embed hierarchical data into hyperbolic space and achieved promising results on learning word embeddings in WordNet.

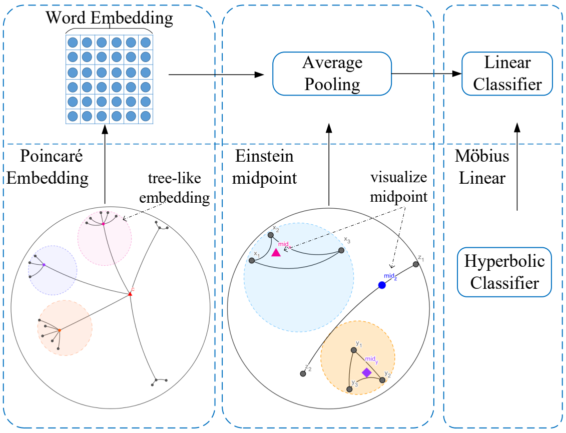

Figure 1: The architecture comparison of FastText (upper) and HyperText (lower).

Inspired by their work, we propose HyperText for text classification by endowing FastText with hyperbolic geometry. We base our method on the Poincaré ball model of hyperbolic space. Specifically, we exploit the Poincaré ball embedding of words or ngrams to capture the latent hierarchies in natural language sentences. Further, we use the Einstein midpoint (Gulcehre et al., 2018) as the pooling method to emphasize semantically specific words which usually contain more information but occur less frequently than general ones (Dhingra et al., 2018). Finally, we employ Möbius linear transformation (Ganea et al., 2018) to play the part of the hyperbolic classifier. We evaluate the performance of our approach on text classification task using ten standard datasets. We observe HyperText outperforms FastText on eight of them. Besides, HyperText is much more parameter-efficient. Across different tasks, only $17\% \sim 50\%$ parameters of FastText are needed for HyperText to achieve comparable performance. Meanwhile, the computational cost of our model increases moderately (2.6x in inference time) over FastText.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1166–1171
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

2 Method

2.1 Overview

Figure 1 illustrates the connection and distinction between FastText and HyperText. The differences of the model architecture are three-fold: First, the input token in HyperText is embedded using hyperbolic geometry, specifically the Poincaré ball model, instead of Euclidean geometry. Second, Einstein midpoint is adopted in the pooling layer so as to emphasize semantically specific words. Last, the Möbius linear transformation is chosen as the prediction layer. Besides, the Riemannian optimization is employed in training HyperText.

2.2 Poincaré Embedding Layer

There are several optional models of hyperbolic space such as the Poincaré ball model, the Hyperboloid model and the Klein model, which offer different affordances for computation. In HyperText, we choose the Poincaré ball model to embed the input words and ngrams so as to better exploit the latent hierarchical structure in text. The Poincaré ball model of hyperbolic space corresponds to the Riemannian manifold which is defined as follow:

$$\mathbb{P}^d = (\mathcal{B}^d, g_x), \tag{1}$$

where $\mathcal{B}^d = \{\boldsymbol{x} \in \mathbb{R}^d | \, \|\boldsymbol{x}\| < 1\}$ is an open d-dimensional unit ball ($\|\cdot\|$ denotes the Euclidean norm) and g_x is the Riemannian metric tensor.

$$g_x = \lambda_x^2 g^E, \tag{2}$$

where $\lambda_x = \frac{2}{1-\|\boldsymbol{x}\|^2}$ is the conformal factor, $g^E = \mathbf{I}_d$ denotes the Euclidean metric tensor. While performing back-propagation, we use the Riemannian gradients to update the Poincaré embedding. The Riemannian gradients are computed by rescaling the Euclidean gradients:

$$\bigtriangledown_R f(\boldsymbol{x}) = \frac{1}{1-\lambda_x^2} \bigtriangledown_E f(\boldsymbol{x}). \tag{3}$$

Since ngrams retain the sequence order information, given a text sequence $S = \{w_i\}_{i=1}^m$, we embed all the words and ngrams into the Poincaré ball, denoted as $X = \{\boldsymbol{x}_i\}_{i=1}^k$, where $\boldsymbol{x}_i \in \mathcal{B}^d$.

2.3 Einstein midpoint Pooling Layer

Average pooling is a normal way to summarize features as in FastText. In Euclidean space, the average pooling is

$$\bar{\boldsymbol{u}} = \frac{\sum_{i=1}^k \boldsymbol{x_i}}{k}. \tag{4}$$

To extend the average pooling operation to the hyperbolic space, we adopt a weighted midpoint method called the Einstein midpoint (Gulcehre et al., 2018). In the d-dimensional Klein model $\mathbb{K}^d$, the Einstein midpoint takes the weighted average of embeddings, which is given by:

$$\bar{\boldsymbol{m}}_{\mathbb{K}} = \frac{\sum_{i=1}^k \gamma_i \boldsymbol{x_i}}{\sum_{i=1}^k \gamma_i}, \boldsymbol{x_i} \in \mathbb{K}^d, \tag{5}$$

where $\gamma_i = \frac{1}{\sqrt{1-\|\boldsymbol{x_i}\|^2}}$ are the Lorentz factors. However, our embedding layer is based on the Poincaré model rather than the Klein model, which means we can't directly compute the Einstein midpoints using Equation (5). Nevertheless, the various models commonly used for hyperbolic geometry are isomorphic, which means we can first project the input embedding to the Klein model, execute the Einstein midpoint pooling, and then project results back to the Poincaré model.

The transition formulas between the Poincaré and Klein models are as follow:

$$\boldsymbol{x}_{\mathbb{K}} = \frac{2\boldsymbol{x}_{\mathbb{P}}}{1+\|\boldsymbol{x}_{\mathbb{P}}\|^2}, \tag{6}$$

$$\bar{\boldsymbol{m}}_{\mathbb{P}} = \frac{\bar{\boldsymbol{m}}_{\mathbb{K}}}{1+\sqrt{1-\|\bar{\boldsymbol{m}}_{\mathbb{K}}\|^2}}, \tag{7}$$

where $\boldsymbol{x}_{\mathbb{P}}$ and $\boldsymbol{x}_{\mathbb{K}}$ respectively denote token embeddings in the Poincaré and Klein models. $\bar{\boldsymbol{m}}_{\mathbb{P}}$ and $\bar{\boldsymbol{m}}_{\mathbb{K}}$ are the Einstein midpoint pooling vectors in the Poincaré and Klein models. It should be noted that points near the boundary of the Poincaré ball get larger weights in the Einstein midpoint formula. These points (tokens) are regarded to be more representative, which can provide salient information for the text classification task (Dhingra et al., 2018).

2.4 Möbius Linear Layer

The Möbius linear transformation is an analogue of linear mapping in Euclidean neural networks. We use the Möbius linear to combine features outputted by the pooling layer and complete the classification task, which takes the form:

$$\boldsymbol{o} = \boldsymbol{M} \otimes \bar{\boldsymbol{m}}_{\mathbb{P}} \oplus \boldsymbol{b}, \tag{8}$$

where $\otimes$ and $\oplus$ denote the Möbius matrix multiplication and Möbius addition defined as follows (Ganea et al., 2018):

$$\boldsymbol{M} \otimes \boldsymbol{x} = (1/\sqrt{c}) \tanh\big(\frac{\|\boldsymbol{M}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \tanh^{-1}(\sqrt{c}\,\|\boldsymbol{x}\|)\big) \frac{\boldsymbol{M}\boldsymbol{x}}{\|\boldsymbol{M}\boldsymbol{x}\|},$$

$$\boldsymbol{x} \oplus \boldsymbol{b} = \frac{(1+2c\langle \boldsymbol{x}, b\rangle + c\|\boldsymbol{b}\|^2)\boldsymbol{x} + (1-c\|\boldsymbol{x}\|^2)\boldsymbol{b}}{1+2c\langle \boldsymbol{x}, \boldsymbol{b}\rangle + c^2\|\boldsymbol{x}\|^2\|\boldsymbol{b}\|^2}.$$

where $\boldsymbol{M} \in \mathbb{R}^{d \times n}$ denotes the weight matrix, and

Model	AG	Sogou	DBP	Yelp P.	Yelp F.	Yah. A.	Amz. F.	Amz. P.	TNEWS	IFYTEK
FastText	92.5	96.8	98.6	95.7	63.9	72.3	60.2	94.6	54.6	54.0
VDCNN	91.3	96.8	98.7	95.7	**64.7**	73.4	**63.0**	**95.7**	54.8	**55.4**
DistilBERT(1-layer)*	92.9	-	**99.0**	91.6	58.6	74.9	59.5	-	-	-
FastBERT(speed=0.8)	92.5	-	**99.0**	94.3	60.7	**75.0**	61.7	-	-	-
HyperText	**93.2**	**97.3**	98.5	**96.1**	64.6	74.3	60.1	94.6	**55.9**	55.2

Table 1: Accuracy(%) of different models. *The results of DistilBERT are cited from Liu et al. (2020)

Dataset	#Classes	#Train	#Test
AG	4	120,000	7,600
Sogou	5	450,000	60,000
DBP	14	560,000	70,000
Yelp P.	2	560,000	38,000
Yelp F.	5	650,000	50,000
Yah. A.	10	1,400,000	60,000
Amz. F.	5	3,000,000	650,000
Amz. P.	2	3,600,000	400,000
TNEWS	15	53,360	10,000
IFLYTEK	119	12,133	2,599

Table 2: Dataset statistics

n denotes the number of class; $\boldsymbol{b} \in \mathbb{R}^n$ is the bias vector and c is a hyper-parameter that denotes the curvature of hyperbolic spaces. In order to obtain the categorical probability $\hat{\boldsymbol{y}}$, a softmax layer is used after the Möbius linear layer.

$$\hat{\boldsymbol{y}} = \mathrm{softmax}(\boldsymbol{o}) \tag{9}$$

2.5 Model Optimization

This paper uses the cross-entropy loss function for the multi-class classification task:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{y} \cdot \log(\hat{\boldsymbol{y}}), \tag{10}$$

where N is the number of training examples, and $\boldsymbol{y}$ is the one-hot representation of ground-truth labels. For training, we use the Riemannian optimizer (Bécigneul and Ganea, 2018) which is more accurate for the hyperbolic models. We refer the reader to the original paper for more details.

3 Experiments

3.1 Experimental setup

Datasets To make a comprehensive comparison with FastText, we choose the same eight datasets as in Joulin et al. (2016) in our experiments. Also, we add two Chinese text classification datasets from Chinese CLUE (Xu et al., 2020), which are presumably more challenging. We summarize the statistics of datasets used in our experiments in Table 2.

Hyperparameters Follow Joulin et al. (2016), we set the embedding dimension as 10 for first eight datasets in Table 1. On TNEWS and IFLYTEK datasets, we use 200-dimension and 300-dimension embeddings respectively. The learning rate is selected on a validation set from $\{0.001, 0.05, 0.01, 0.015\}$. In addition, we use PKUSEG tool (Luo et al., 2019) for Chinese word segmentation.

3.2 Experimental Results

Comparison with FastText and deep models The results of our experiments are displayed in Table 1. Our proposed HyperText model outperforms FastText on eight out of ten datasets, and the accuracy of HyperText is 0.7% higher than FastText on average. In addition, from the results, we observe that HyperText works significantly better than FastText on the datasets with more label categories, such as Yah.A., TNEWS and IFLYTEK. This arguably confirms our hypothesis that HyperText can better model the hierarchical relationships of the underlying data and extract more discriminative features for classification. Moreover, HyperText outperforms DistilBERT(Sanh et al., 2019) and FastBERT(Liu et al., 2020) which are two distilled versions of BERT. And HyperText achieves comparable performance to the very deep convolutional network (VDCNN) (Conneau et al., 2016) which consists of 29 convolutional layers. From the results, we can see that HyperText has better or comparable classification accuracy than these deep models while requiring several orders of magnitude less computation.

Embedding Dimension Since the input embeddings account for more than 90% model parameters, we investigate the impact of dimension of input embedding on the classification accuracy. The experimental results are presented in Figure 2. As we can see, on most tasks HyperText performs consistently better than FastText in various dimension settings. In particular, on IFLYTEK and TNEWS datasets, HyperText with 50-dimension respectively achieves better performance to FastText with 300-dimension and 200-dimension. On other eight less challenging datasets, the experiments are

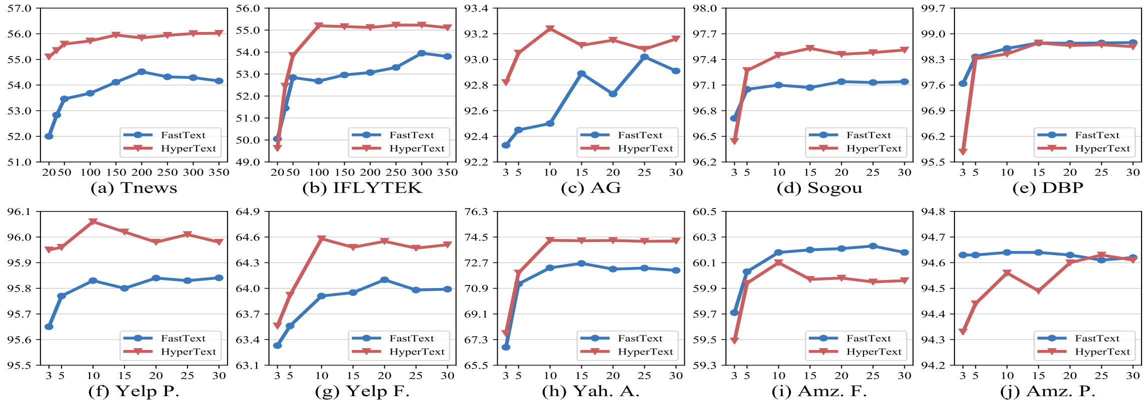

Figure 2: Accuracy vs Embedding dimension. The x-axis represents the embedding dimension, while the y-axis represents the accuracy.

conducted in the low-dimensional settings and HyperText often requires less dimensions to achieve the optimal performance in general. It verifies that thanks to the ability to capture the internal structure of the text, the hyperbolic model is more parameter efficient than its Euclidean competitor.

Computation in Inference FastText is well-known for its fast inference speed. We compare the FLOPs versus accuracy under different dimensions in Figure 3. Due to the additional non-linear operations, HyperText generally requires more ($4.5 \sim 6.7$x) computations compared to FastText with the same dimension. But since HyperText is more parameter efficient, when constrained on the same level of FLOPs, HyperText mostly performs better than FastText on the classification accuracy. Besides, the FLOPs level of VDCNN is 10^5 higher than HyperText and FastText.

Ablation study We conduct the ablation study to figure out the contribution of different layers. The results on several datasets are present in Table 3. Note that whenever we replace a hyperbolic layer with its counterpart in Euclidean geometry, the model performs worse. The results show that all the hyperbolic layers (Poincaré Embedding Layer, Einstein midpoint Pooling Layer and Möbius Linear Layer) are necessary to achieve the best performance.

Model	Yelp P.	AG	Yah.A.	TNEWS
HyperText	96.1	93.2	74.3	55.9
-PE&EM	95.9	92.8	73.9	55.6
-ML	95.6	92.3	73.2	54.6

Table 3: Ablation study of each components in HyperText (PE for Poincaré Embedding, EM for Einstein Midpoint, and ML for Möbius Linear layer).

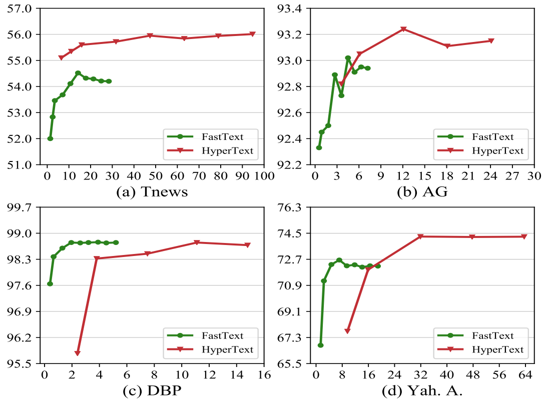

Figure 3: FLOPs($\times 10^3$) vs Accuracy(%) under different dimensions. The x-axis represents the FLOPs, while the y-axis represents the accuracy. Different points represent different embedding dimensions

4 Related Work

Hyperbolic space can be regarded as a continuous version of tree, which makes it a natural choice to represent the hierarchical data (Nickel and Kiela, 2017, 2018; Sa et al., 2018). Hyperbolic geometry has been applied to learning knowledge graph representations. HyperKG (Kolyvakis et al., 2019) extends TransE to the hyperbolic space, which obtains great improvement over TransE on WordNet dataset. Balaževic et al. (2019) proposes MURP model which minimizes the hyperbolic distances between head and tail entities in the multi-relational graphs. Instead of using the hyperbolic distance, Chami et al. (2019, 2020) uses the hyperbolic rotations and reflections to better model the rich kinds of relations in knowledge graphs. Specifically, the authors use the hyperbolic rotations to capture anti-symmetric relations and hyperbolic reflections to capture symmetric relations, and combine these operations together by the attention mechanism. It achieves significant improvement at low dimension.

Hyperbolic geometry is also applied in natural language data so as to exploit the latent hierarchies in the word sequences (Tifrea et al., 2019).

Recently, many hyperbolic geometry based deep neural networks (Gulcehre et al., 2018; Ganea et al., 2018) achieve promising results, especially when the mount of parameters is limited. There are some applications based on hyperbolic geometry, such as question answering system (Tay et al., 2018), recommendation system (Chamberlain et al., 2019) and image embedding (Khrulkov et al., 2020).

5 Conclusion

We have shown that hyperbolic geometry can endow the shallow neural networks with the ability to capture the latent hierarchies in natural language. The empirical results indicate that HyperText consistently outperforms FastText on a variety of text classification tasks. On the other hand, HyperText requires much less parameters to retain performance on par with FastText, which means neural networks in hyperbolic space could have a stronger representation capacity.

References

Ivana Balaževic, Carl Allen, and Timothy Hospedales. 2019. Multi-relational poincaré graph embeddings. In *Advances in Neural Information Processing Systems*.

Gary Bécigneul and Octavian-Eugen Ganea. 2018. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*.

Benjamin Paul Chamberlain, Stephen R. Hardwick, David R. Wardrope, Fabon Dzogang, Fabio Daolio, and Saúl Vargas. 2019. Scalable hyperbolic recommender systems. *CoRR*, abs/1902.08648.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*.

Ines Chami, Adva Wolf, Frederic Sala, and Christopher Ré. 2019. Low-dimensional knowledge graph embeddings via hyperbolic rotations. In *Graph Representation Learning NeurIPS 2019 Workshop*.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*.

Bhuwan Dhingra, Christopher J. Shallue, Mohammad Norouzi, Andrew M. Dai, and George E. Dahl. 2018. Embedding text in hyperbolic spaces. In *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing, TextGraphs@NAACL-HLT*.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, pages 5345–5355.

Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. 2018. Hyperbolic attention networks. In *International Conference on Learning Representations*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. 2020. Hyperbolic image embeddings. *arXiv preprint arXiv:1904.02239*.

Prodromos Kolyvakis, Alexandros Kalousis, and Dimitris Kiritsis. 2019. Hyperkg: Hyperbolic knowledge graph embeddings for knowledge base completion. *arXiv preprint arXiv:1908.04895*.

Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguná. 2010. Hyperbolic geometry of complex networks. *Physical Review E, 82(3):036106*.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Haotang Deng, and Qi Ju. 2020. Fastbert: a selfdistilling bert with adaptive inference time. *CoRR,abs/2004.02178*.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *CoRR*, abs/1906.11455.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, pages 6338–6347.

Maximillian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *Proc. ICML*, page 3776–3785.

Christopher De Sa, Albert Gu, Christopher Ré, and Frederic Sala. 2018. Representation tradeoffs for hyperbolic embeddings. In *Proceedings of the 35th International Conference on Machine Learning,PMLR*, volume 80, pages 4460–4469.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108.*

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Hyperbolic representation learning for fast and efficient neural question answering. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining(WSDM)*, pages 583–591.

Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. 2019. Poincaré glove: Hyperbolic word embeddings. In *International Conference on Learning Representation*.

Liang Xu, Xuanwei Zhang, Lu Li, Hai Hu, Chenjie Cao, Weitang Liu, Junyi Li, Yudong Li, Kai Sun, Yechen Xu, Yiming Cui, Cong Yu, Qianqian Dong, Yin Tian, Dian Yu, Bo Shi, Rongzhao Wang Jun Zeng, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, and Zhenzhong Lan. 2020. Clue: A chinese language understanding evaluation benchmark. *arXiv preprint arXiv:2004.05986*.

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.