Evaluating the Combination of Word Embeddings with Mixture of Experts and Cascading gcForest In Identifying Sentiment Polarity

by

Mounika Marreddy, Subba Reddy Oota, Radha Agarwal, Radhika Mamidi

in

25TH ACM SIGKDD CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING
(*SIGKDD-2019*)

Anchorage, Alaska, USA

Report No: IIIT/TR/2019/-1

Centre for Language Technologies Research Centre
International Institute of Information Technology
Hyderabad - 500 032, INDIA
August 2019

Evaluating the Combination of Word Embeddings with Mixture of Experts and Cascading gcForest In Identifying Sentiment Polarity

Mounika Marreddy
mounika.marreddy@research.iiit.ac.in
IIIT-Hyderabad
Hyderabad, India
mounika.marreddy@research.iiit.ac.in

Subba Reddy Oota
IIIT-Hyderabad
Hyderabad, India
oota.subba@students.iiit.ac.in

Radha Agarwal
IIIT-Hyderabad
Hyderabad, India
radha.agarwal@students.iiit.ac.in

Radhika Mamidi
IIIT-Hyderabad
Hyderabad, India
radhika.mamidi@iiit.ac.in

ABSTRACT

Neural word embeddings have been able to deliver impressive results in many Natural Language Processing tasks. The quality of the word embedding determines the performance of a supervised model. However, choosing the right set of word embeddings for a given dataset is a major challenging task for enhancing the results. In this paper, we have evaluated neural word embeddings with (i) a mixture of classification experts (MoCE) model for sentiment classification task, (ii) to compare and improve the classification accuracy by different combination of word embedding as first level of features and pass it to cascade model inspired by GcForest for extracting diverse features. We argue that each expert learns a certain positive and negative examples corresponding to its category of features and resulting features on a given task (polarity identification) can achieve competitive performance with state of the art methods in terms of accuracy, precision, and recall using gcForest.

KEYWORDS

mixtue of experts, gcForest, word embeddings, sentiment analysis
. .

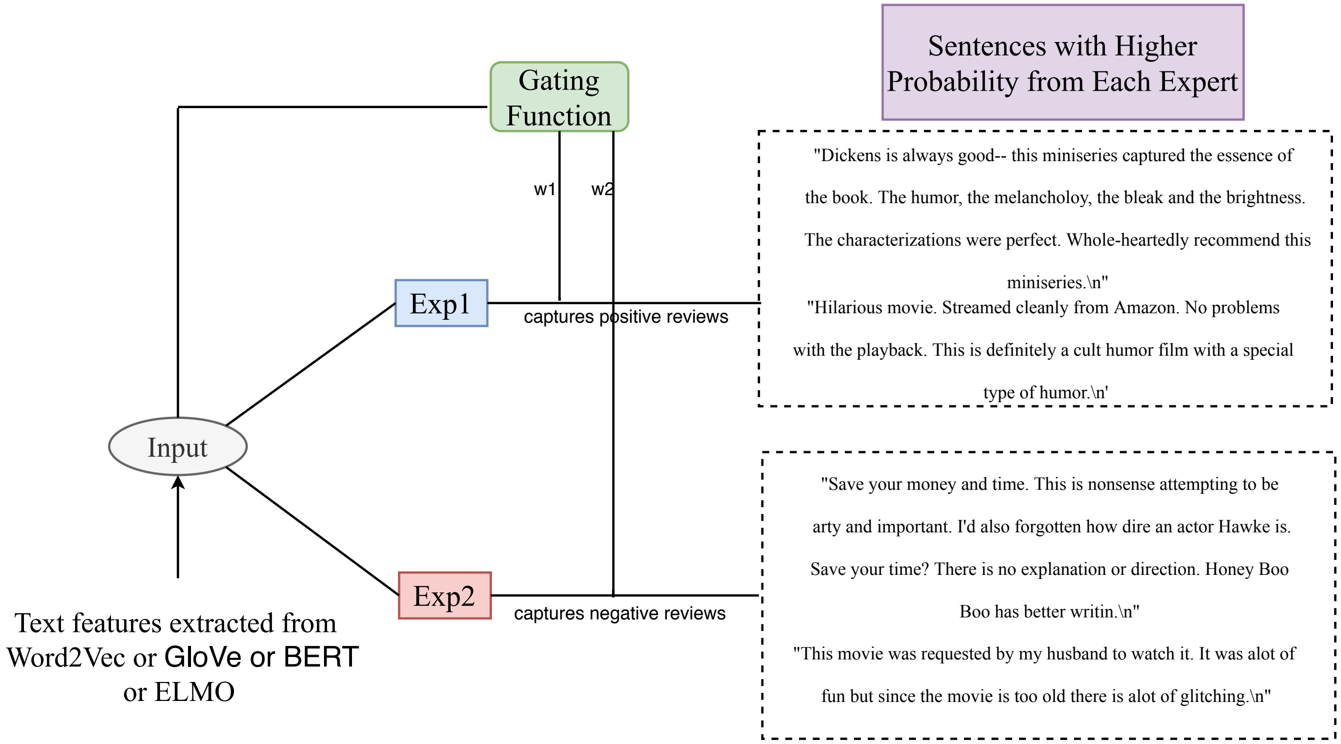
1 INTRODUCTION

Sentiment Analysis is one of the most successful and well-studied fields in Natural Language Processing [3, 9, 10]. Traditional approaches mainly focus on designing a set of features such as bag-of-words, sentiment lexicon to train a classifier for sentiment classification [18]. However, feature engineering is labor intensive and almost reaches its performance bottleneck. Moreover, as the increasing information on web like writing reviews on review sites and social media, opinions influence human behavior and help organization or individual in decision making task. With the huge success of deep learning techniques, some researchers designed an effective neural networks to generate low dimensional contextual representations and yields promising results on the sentiment analysis [7, 14, 21].

Since the work of [2], NLP community is focusing on improving the feature representation of sentence/document with continuous development in neural word embedding. Word2Vec embedding was the first powerful technique to achieve semantic similarity between words but fail to capture the meaning of a word based on context [17]. As an improvement to Word2Vec, [19] introduced GloVe embeddings, primarily focus on global co-occurrence count for generating word embeddings. Using Word2Vec & GloVe, it was easy to train with application in Question Answering task, Sentiment Analysis, Automatic Summarization [13] and also gained popularity in Word Analogy, Word similarity and Named Entity Recognition tasks [5]. However, the main challenge with GloVe and Word2Vec is unable to differentiate the word used in different context. [16] introduced a deep LSTM encoder from an attentional sequence-to-sequence model trained for machine translation (MT) to contextualize word vectors(MT-LSTM/CoVe). The main limitation with CoVe vectors was it uses zero vectors for unknown words (out of vocabulary words).

ELMO [20] and BERT [6] embeddings are two recent popular techniques outperforms many of the NLP tasks and got huge success in neural embedding techniques that represent the context in features due to the attention-based mechanism. ELMO embedding is a character based embedding, it allows the model to capture out of vocabulary words and deep contextualized word representation can capture syntax and semantic features of words and outperforms the problems like sentiment analysis [1] and named entity recognition [15]. In advancement to contextual embedding, BERT embedding is a breakthrough in neural embedding technique and built upon transformers including the self-attention mechanism. It can represent features with the relationship between all words in a sentence. BERT outperforms state of the art feature representation for a task like question answering with SQuAD [22], language modeling/sentiment classification.

In recent years, the use of neural word embeddings provide better vector representations of semantic information, there has been relatively little work on direct evaluations of these models.


Sentences with Higher Probability from Each Expert
Gating Function
w1
w2
Exp1
captures positive reviews
Input
Exp2
captures negative reviews
Text features extracted from Word2Vec or GloVe or BERT or ELMO

"Dickens is always good-- this miniseries captured the essence of the book. The humor, the melancholoy, the bleak and the brightness. The characterizations were perfect. Whole-heartedly recommend this miniseries.\n"

"Hilarious movie. Streamed cleanly from Amazon. No problems with the playback. This is definitely a cult humor film with a special type of humor.\n'

"Save your money and time. This is nonsense attempting to be arty and important. I'd also forgotten how dire an actor Hawke is. Save your time? There is no explanation or direction. Honey Boo Boo has better writin.\n"

"This movie was requested by my husband to watch it. It was alot of fun but since the movie is too old there is alot of glitching.\n"

Figure 1: Proposed Mixture of Classification Experts (MoCE) model. Here Expert1 captures positive reviews and Expert2 captures negative reviews.

There has been previous work to evaluate various word embedding techniques [8] on a specific task like word similarity or analogy, Named entity recognition [23] and evaluate it based on the obtained performance metric.

In this paper we have evaluated neural word embedding techniques with (1) a mixture of classification experts (MoCE) model for the sentiment classification task, (ii) compare and improve the classification accuracies using gcForest. The underlying mechanism of MoCE model is that it has great potential to discriminate positive and negative examples for sentiment classification task on Amazon product reviews data. In the next sections, we discuss the proposed MoCE approach, cascading gcForest and our enhancements.

2 MODEL ARCHITECTURE

We use a mixture of experts based model, whose architecture is inspired from [11]. The mixture of experts architecture is composed of gating network and several expert networks, each of which solves a function approximation problem over a local region of the input space. The detailed overview of our model is shown in Figure 1 where the input is a text vector extracted from recently successful neural embeddings such as Word2Vec, GloVe, ELMO, Amazon Small Embeddings, & BERT. These input features pass through both the gating network and two of the experts. The gating network uses a probabilistic model to choose the best expert for a given input text vector.

2.1 MoCE Architecture

Given an input feature vector $\mathbf{x}$ from the one of the neural word embedding method, we model its posterior probabilities as a mixture of posteriors produced by each expert model trained on $\mathbf{x}$.

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= \sum_{j=1}^{K} P(S_j|\mathbf{x},\theta_0)p(\mathbf{y}|\mathbf{x},S_{\theta_j}) \\ &= \sum_{j=1}^{K} g_{S_j}(\mathbf{x},\theta_0)p(\mathbf{y}|\mathbf{x},S_{\theta_j}) \end{aligned} \tag{1}$$

Here, $P(S_j|\mathbf{x},\theta_0) = g_{S_j}(\mathbf{x},\theta_0)$ is the probability of choosing S_j^{th} expert for given input $\mathbf{x}$. Note that $\sum_{j=1}^{K} g_{S_j}(\mathbf{x},\theta_0) = 1$ and $g_{S_j}(\mathbf{x},\theta_0) \geq 0,\ \forall j \in [K]$. $g_{S_j}(\mathbf{x},\theta_0)$ is also called gating function and is parameterized by θ_0.

In this paper, we choose $p(\mathbf{y}|\mathbf{x},S_{\theta_j})$ as a Gaussian probability density for each of the experts, denoted by:

$$p(\mathbf{y}|\mathbf{x},S_{\theta_j}) = \frac{1}{(|\sigma_j|2\pi)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-W_j\mathbf{x})^2\right) \tag{2}$$

where $S_{\theta_j} \in \mathbb{R}^{m\times n}$ is the weight matrix associated with the S_j^{th} expert. Thus, $S_{\theta_j} = \{W_j\}$. We use softmax function for the gating

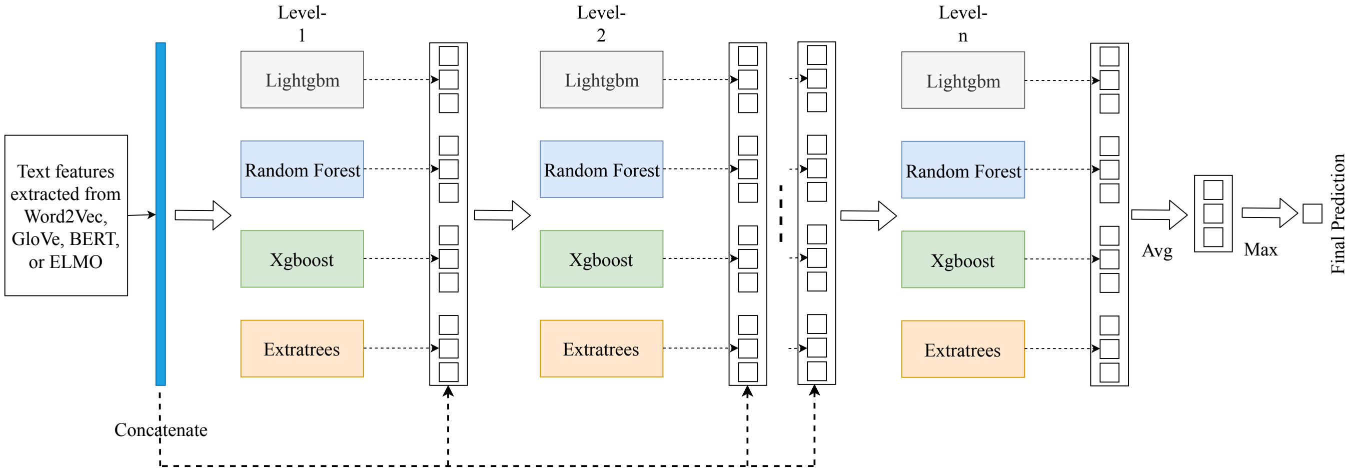


Figure 2: Cascading gcForest Architecture

variable $g_{S_j}(\mathbf{x}, \theta_0)$.

$$g_{S_j}(\mathbf{x}, \theta_0) = \frac{\exp\left(\mathbf{v}_j^T \mathbf{x}\right)}{\sum_{i=1}^{K} \exp\left(\mathbf{v}_i^T \mathbf{x}\right)} \tag{3}$$

where $\mathbf{v}_j \in \mathbb{R}^n$, $\forall j \in [K]$. Thus, $\theta_0 = \{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$. Let Θ be the set of all the parameters involved for the K-experts. Thus, $\Theta = \{\theta_0, (W_1), \ldots, (W_K)\}$. Here, we train the MoCE model and update the weights iteratively using expectation-maximization (EM) algorithm.

2.2 Multigrained gcForest Architecture

Table 1: Model-Parameters

Model	Parameters
XGB	n_foldss: 5 n_estimators: 100 max_depth: 5 learning_rate: 0.1
LGBM	n_foldss: 5 n_estimators: 100 max_depth: 5 learning_rate: 0.1
RF	n_foldss: 5 n_estimators: 100
ET	n_foldss: 5 n_estimators: 100

In order to improve the classification performance of each dataset, we passed the input feature vector to a multigrain gcForest model for better feature representation. The gcForest model we motivate from [24], where the cascade structure, as illustrated in Figure 2, where each cascading level receives input from the preceding level and the processed result passed to the next level.

The raw input feature vector is given to gcForest with different dimension associated with pretrained embeddings. Each cascading level contains different ensemble based forest models i.e an ensemble of ensembles yields the diversity in feature construction. Here, each forest produces a class distribution for each instance and finally estimate the average of all class distributions across the ensemble based forests gives an output vector. The output vector is concatenated with the original feature vector and passed to the next cascading level. In order to avoid the risk of overfitting, each forest uses K-fold cross-validation to produce the class vector. Moreover, the complexity of a model can be controlled by checking the training error and validation error to terminate the process when the training is adequate.

3 EXPERIMENTAL SETUP & RESULTS

3.1 Dataset Description

For our experiments, we are using the Dranziera protocol dataset provided in ESWC Semantic Challenge-2019 [1]. The dataset contains a total of 20 Amazon products and each product is having 50000 reviews are presented. We divided the dataset into 80 percent training and 20 percent testing. We applied five-fold cross-validation technique to observe how the model performs with overall data.

3.2 Feature Extraction

In this paper, we mainly focused on five successful pretrained word embeddings such as: Word2Vec (embeddings are of 300 dimensions) [17], GloVe (embeddings are of 300 dimensions) [19], Amazon Small Word Embeddings (embeddings are of 128 dimensions), BERT (embeddings are 768 dimensions each) [6], and ELMO (embeddings are 1024 dimensions each) [20].

3.3 Results & Discussion

Here, we conducted the experiments in two steps. In the first step, we evaluated the five word embeddings using MoCE model and the second step describes better feature representation using cascading

[1] http://www.maurodragoni.com/research/opinionmining/dranziera/embeddings-evaluation.php

Table 2: Comparison of word embedding results of various domains with our MoCE Model

	Word2vec		Glove		BERT		ELMO	
Domain	Expert1	Expert2	Expert1	Expert2	Expert1	Expert2	Expert1	Expert2
Amazon_Instant_Video	4097	4349	4058	4331	2718	2731	3571	3601
Automotive	4055	4257	4265	4103	2740	2761	3620	3641
Baby	3658	4372	4899	289	3395	3280	3897	3621
Beauty	142	5045	4320	4124	2780	2733	3412	3572
Books	4146	4183	4231	4183	2882	2853	3765	3412
Clothing_Accessories	4522	3986	4273	4448	3339	3700	3923	3654
Electronics	5012	228	4286	4078	2834	2794	3662	3761
Health	4008	4197	4096	4233	2975	2772	3570	3698
Home_Kitchen	4061	4391	4412	4162	2980	2992	3476	3671
Movies_TV	4290	4042	153	4868	2744	2897	3644	3812
Music	4010	4302	4286	4030	3241	3103	3921	3956
Office_Products	4988	119	4361	4047	3299	3263	4001	4117
Patio	196	4995	4966	225	2800	2799	3491	3379
Pet_Supplies	4140	4022	4147	4034	2734	2810	3564	3679
Shoes	4606	4235	4616	4329	3073	3369	3890	3795
Software	4142	4234	4387	3577	2768	2772	3561	3673
Sports_Outdoors	3906	4395	3971	4380	2938	2988	3452	3677
Tools_Home_Improvement	4298	3917	4292	3965	2756	2759	3545	3890
Toys_Games	4400	4252	4379	4295	3301	3292	3755	3678
Video_Games	4069	4196	234	4961	2781	2836	3572	3684

gcForest outperforms the state of the art results on amazon product review datasets.

3.3.1 *Evaluation of Embeddings using MoCE.*

Using the approach discussed in Section 2, initially we trained a separate mixture of classification experts model for each product using all the embeddings. Experiments are conducted on the dataset by passing input as text vector extracted from recent successful neural word embeddings and output as corresponding target classes positive or negative. We split the dataset into 40000 reviews in training and 10000 reviews into testing. The MoCE model performance was evaluated by training and testing the different subsets of the 50000 reviews in a 5-fold cross-validation scheme. The proposed model was trained until the model reached the convergence with a lower bound of $1e^{-5}$ or a maximum of 100 iterations.

Here, we used five different successful word embeddings to extract the input feature vector and we compare each method using the MoCE model for all the 20 datasets. Table 2 presents the performance results of each embedding scheme where the two experts discriminate both positive and negative examples. From the table 2, we can observe that both GloVe and Word2vec embeddings having better discrimination where one of the experts captures majority positive sentiment examples as other expert capture more negative sentiment examples. Here, we use test dataset of total 10000 examples out of which 5000 samples are positive and 5000 samples are negative. For example, from the Table 2 consider the Shoes domain dataset, for the GloVe Embedding: expert1 capture 4616 positive sentiment samples and expert2 capture 4329 negative sentiment samples shows better discrimination and similarly with the Word2Vec and ELMO. However, in the case of BERT embedding: expert1 capture only 3073 positive examples and expert2 captures 3369 negative examples.

3.3.2 *Polarity Identification using gcForest.*

Using the MocE results described in Table 2, we can observe the better feature representation of each pretrained word embedding model based on the experts which discriminate the positive or negative samples. In order to validate and improve the classification performance, we also built the cascading gcForest classification model described in section 2.2. We use four ensemble forest models such as LightGBM [12], XGboost [4], Random Forest, and Extra Trees classifier in each cascading layer. The configuration of the gcForest model is shown in Table 1. Here, we use a 5-fold cross-validation method to avoid the overfitting problem. With this method, the model outperforms the state of the art results for different combination features such as GloVE, Word2Vec, ELMO & BERT as shown in Table 3. We also improve the classification performance of each domain dataset by using the above mentioned five embeddings. Since, gcForest doesnot require more hyper-parameters and deeper layers to train to achieve good performance and very fast to train.

Figure 3 illustrates each domain results for all the pretrained embeddings. From the figure 3, we can observe that Word2Vec, GloVe, and ELMO methods perform better when compared to BERT embeddings and similar comparison we observed in table 2. One of the main reason why BERT & ELMO are not performed better than Word2Vec & GloVe is that to fine-tune language models (LMs) likes BERT/ELMO for a specific dataset training for few epochs getting better results instead of simply using pretrained embeddings. Amazon Small Word embeddings results are better mainly because these embeddings are domain specific.

4 CONCLUSION

Neural word embeddings have been able to deliver impressive results in many Natural Language Processing tasks. However, choosing the right set of word embeddings for a given dataset is a major challenging task for enhancing the results. In this paper, we have evaluated five neural word embedding methods such as Word2Vec, GloVe, ELMO, BERT, & Amazon Small Word embeddings with (i) a mixture of classification experts (MoCE) model for sentiment classification task, (ii) to compare and improve the classification accuracy by different combination of word embedding as first level

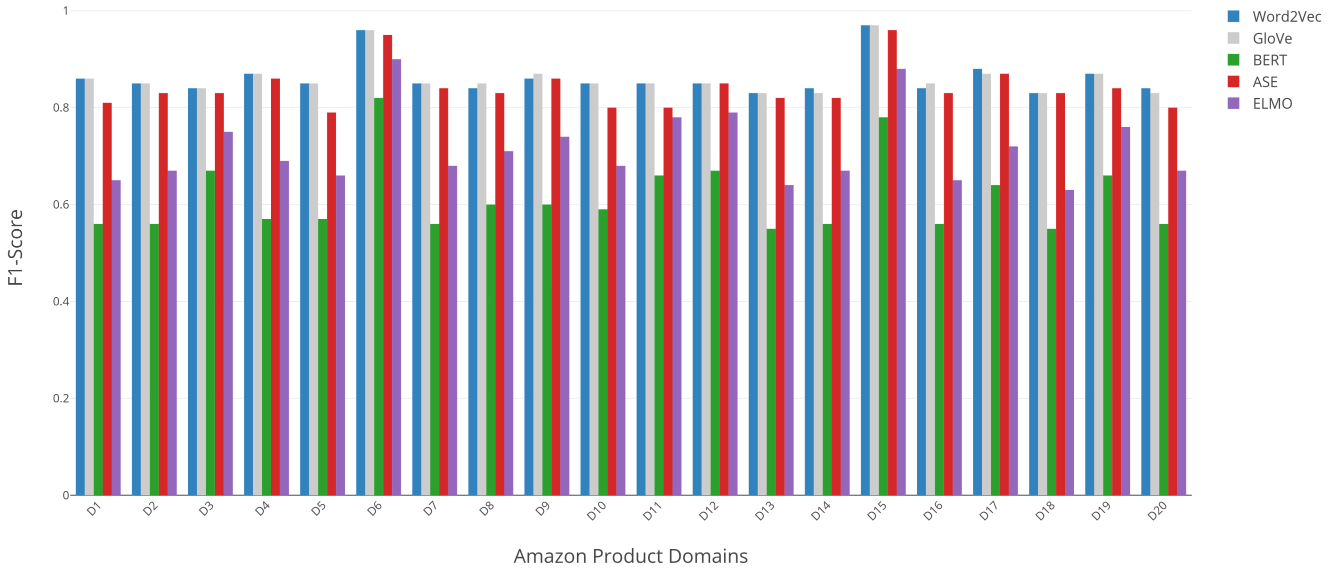


Figure 3: Figure presents the F1-score of amazon 20 products using gcForest on five word embeddings Word2Vec, GloVe, BERT, Amazon Small Embeddings, and ELMO.

Table 3: Detailed results of domains of Dranziera dataset by the Baselines and by gcForest

	Tested System							
Dom	**SVM**	**ME**	**DBP**	**DDP**	**CNN**	**GWE**	**NS**	**gcF**
(1)	0.70	0.70	0.72	0.71	0.80	0.80	0.80	**0.87**
(2)	0.72	0.71	0.72	0.70	0.73	0.79	0.85	**0.87**
(3)	0.69	0.72	0.71	0.69	0.84	0.79	0.85	**0.86**
(4)	0.69	0.72	0.74	0.73	0.82	0.81	0.85	**0.88**
(5)	0.69	0.69	0.69	0.69	0.78	0.75	0.79	**0.86**
(6)	0.69	0.72	0.80	0.78	0.77	0.81	0.86	**0.97**
(7)	0.68	0.69	0.73	0.70	0.79	0.77	0.86	**0.87**
(8)	0.67	0.66	0.69	0.69	0.78	0.79	0.86	**0.86**
(9)	0.72	0.69	0.71	0.69	0.75	0.82	0.87	**0.88**
(10)	0.73	0.72	0.70	0.71	0.75	0.79	0.80	**0.86**
(11)	0.69	0.65	0.71	0.72	0.76	0.77	0.80	**0.86**
(12)	0.73	0.73	0.72	0.70	0.79	0.80	0.87	**0.87**
(13)	0.69	0.71	0.70	0.69	0.86	0.80	0.86	**0.86**
(14)	0.68	0.73	0.67	0.66	0.82	0.79	0.84	**0.85**
(15)	0.67	0.73	0.83	0.81	0.81	0.84	0.86	**0.97**
(16)	0.74	0.69	0.72	0.71	0.79	0.76	0.85	**0.86**
(17)	0.67	0.73	0.71	0.71	0.76	0.81	0.87	**0.89**
(18)	0.73	0.73	0.68	0.69	0.79	0.79	0.85	**0.85**
(19)	0.66	0.69	0.74	0.71	0.77	0.84	0.86	**0.88**
(20)	0.69	0.70	0.70	0.70	0.72	0.78	0.82	**0.84**

(1) NS (NeuroSent), (2) gcF(gcForest)

of features and pass it to cascade model inspired by GcForest for extracting diverse features.

In the future, we plan to experiment on all NLP tasks by using a hierarchical mixture of experts and conduct experiments on other standard datasets with a primary focus on all aspects of word embeddings.

REFERENCES

[1] Jorge A Balazs, Edison Marrese-Taylor, and Yutaka Matsuo. 2018. IIIDYT at IEST 2018: Implicit Emotion Classification With Deep Contextualized Word Representations. *arXiv preprint arXiv:1808.08672* (2018).

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3, Feb (2003), 1137–1155.

[3] Erik Cambria and Bebo White. 2014. Jumping NLP curves: A review of natural language processing research. *IEEE Computational intelligence magazine* 9, 2 (2014), 48–57.

[4] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.

[5] Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* 4 (2016), 357–370.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers)*, Vol. 2. 49–54.

[8] Sahar Ghannay, Benoit Favre, Yannick Esteve, and Nathalie Camelin. 2016. Word embedding evaluation and combination.. In *LREC*. 300–305.

[9] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 151–160.

[10] Yohan Jo and Alice H Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 815–824.

[11] Michael I Jordan and Lei Xu. 1995. Convergence results for the EM approach to mixtures of experts architectures. *Neural Networks* 8, 9 (1995), 1409–1431.

[12] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*. 3146–3154.

[13] Tom Kenter and Maarten De Rijke. 2015. Short text similarity with word embeddings. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. ACM, 1411–1420.

[14] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[15] Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In *Asia Information Retrieval Symposium*. Springer, 581–587.

[16] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*. 6294–6305.

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

[18] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 79–86.

[19] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543. http://www.aclweb.org/anthology/D14-1162

[20] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

[21] Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[22] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-End Open-Domain Question Answering with BERTserini. *arXiv preprint arXiv:1902.01718* (2019).

[23] Mengnan Zhao, Aaron J Masino, and Christopher C Yang. 2018. A Framework for Developing and Evaluating Word Embeddings of Drug-named Entity. In *Proceedings of the BioNLP 2018 workshop*. 156–160.

[24] Zhi-Hua Zhou and Ji Feng. 2017. Deep forest: Towards an alternative to deep neural networks. *arXiv preprint arXiv:1702.08835* (2017).