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 on sentiment analysis task in two steps: (i) proposed 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 in the first step, each expert learns a certain positive or negative examples corresponding to its category and in the second step 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

ACM Reference Format:
Mounika Marreddy, Subba Reddy Oota, Radha Agarwal, and Radhika Mamidi. 2019. Evaluating the Combination of Word Embeddings with Mixture of Experts and Cascading gcForest in Identifying Sentiment Polarity. In *Proceedings of KDD 2019 (WISDOM'19): 8th KDD Workshop on Issues of Sentiment Discovery and Opinion Mining, August 4, 2019.* ACM, Anchorage, Alaska, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

WISDOM'19, August 4, 2019, Anchorage, Alaska

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Sentiment Analysis is one of the most successful and well-studied fields in Natural Language Processing [1–3]. Traditional approaches mainly focus on designing a set of features such as bag-of-words, sentiment lexicon to train a classifier for sentiment classification [4]. 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 [5–7].

Since the work of [8], NLP community is focusing on improving the feature representation of sentence/document with continuous development in a 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 [9]. As an improvement to Word2Vec, [10] 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 [11], sentiment analysis [12], automatic summarization [13] and also gained popularity in word analogy, word similarity and named entity recognition tasks [14]. However, the main challenge with GloVe and Word2Vec is unable to differentiate the word used in a different context. [15] introduced a deep LSTM (Long short-term memory) 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 (Embeddings from Language Models) [16] and BERT (Bidirectional Encoder Representations from Transformers) [17] 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

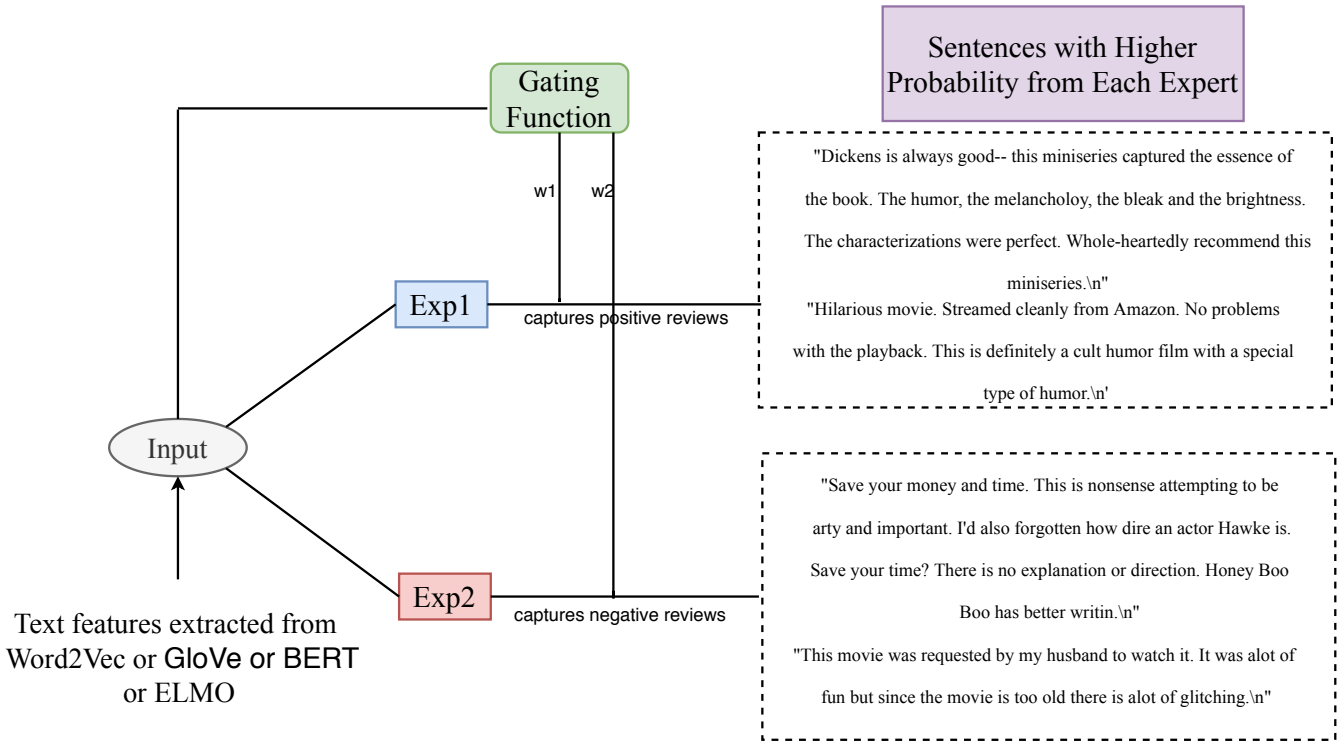

Figure 1: Proposed Mixture of Classification Experts (MoCE) model. Here, Expert1 captures positive reviews and Expert2 captures negative reviews.

features of words and outperforms the problems like sentiment analysis [18] and named entity recognition [19]. 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 [20], 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. There has been previous work to evaluate various word embedding techniques [21] on a specific task like word similarity or analogy, Named entity recognition [22] and evaluate it based on the obtained performance metric.

In this paper, we have evaluated four successful pretrained neural word embeddings: Word2Vec, GloVe, ELMo and BERT on sentiment analysis task in two steps (1) proposed a mixture of classification experts (MoCE) model for the sentiment classification task, (ii) to compare and improve the classification accuracies by combining the popular word embedding as first level of features and pass it to cascade model inspired by 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 first step, a mixture of classification expert uses a combination of the simpler learner to improve predictions. Each learner divides the dataset into several different regions based on the relationship between input and output. In our case, it will divide the region of the different polarities region with the help of probabilistic gating network. 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 second step, we validated and improve the classification accuracy by combining the four embedding vectors and passed it to cascaded gcForest for better feature representation. The gcForest model with combined word embeddings is able to perform better results with the sentiment analysis task.

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 [23]. 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, & BERT. These input features pass through both the gating network and two of the

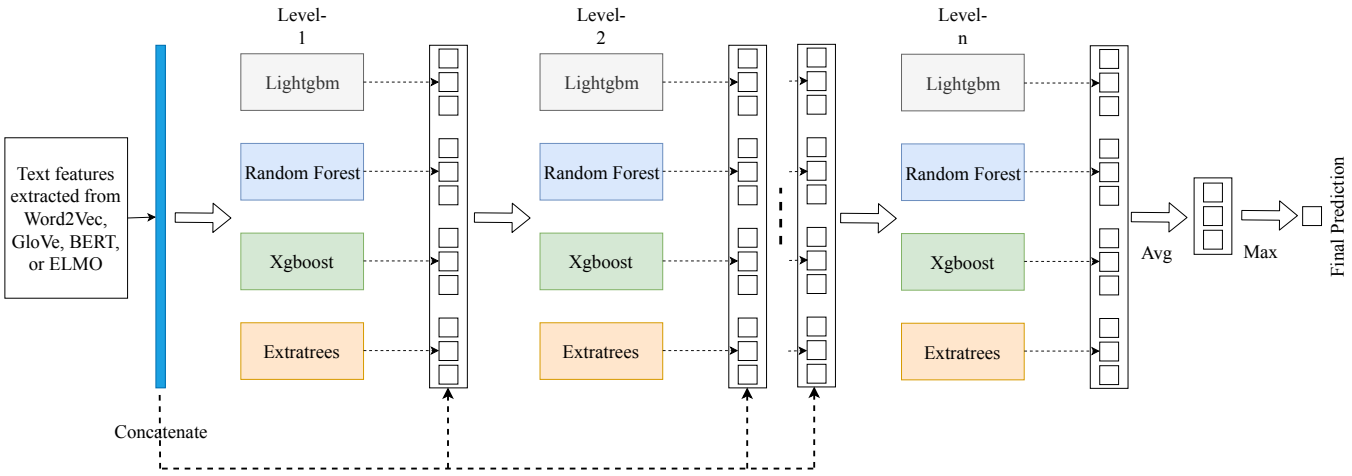

Figure 2: Cascading gcForest Architecture

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 $\mathbf{x}$.

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

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.

Since the class labels $\{\mathbf{y}_1, \mathbf{y}_2,...,\mathbf{y}_n\}$ are independent and identically distributed sample of outcome variables from a population modelled by a K-component finite mixture model. Here, the outcome variable is discrete (either positive or negative sentiment). Due to this reason, 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:

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

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

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

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 of Cascading gcForest

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

Table 2: Comparison of word embedding results of 20 domains of Dranziera dataset with our MoCE Model. The values in the table indicates the percentage of positive reviews captured by Expert1 and percentage of negative reviews captured by Expert2.

Domain	Word2vec		GloVe		BERT		ELMo	
	Expert1	Expert2	Expert1	Expert2	Expert1	Expert2	Expert1	Expert2
Amazon_Instant_Video	0.81	0.86	0.81	0.86	0.54	0.55	0.71	0.72
Automotive	0.81	0.85	0.85	0.82	0.54	0.55	0.72	0.72
Baby	0.73	0.87	0.97	0.05	0.61	0.67	0.77	0.72
Beauty	0.02	0.98	0.86	0.82	0.55	0.54	0.68	0.71
Books	0.82	0.83	0.84	0.83	0.57	0.57	0.75	0.68
Clothing_Accessories	0.90	0.79	0.85	0.88	0.66	0.74	0.78	0.73
Electronics	0.98	0.04	0.85	0.81	0.56	0.55	0.73	0.75
Health	0.80	0.83	0.81	0.84	0.59	0.55	0.71	0.73
Home_Kitchen	0.81	0.87	0.88	0.83	0.59	0.59	0.69	0.73
Movies_TV	0.85	0.80	0.03	0.97	0.54	0.57	0.72	0.76
Music	0.80	0.86	0.85	0.80	0.64	0.62	0.78	0.79
Office_Products	0.99	0.02	0.87	0.80	0.65	0.64	0.80	0.82
Patio	0.03	0.99	0.99	0.04	0.31	0.55	0.69	0.67
Pet_Supplies	0.82	0.80	0.82	0.80	0.54	0.56	0.71	0.73
Shoes	0.92	0.84	0.92	0.86	0.60	0.65	0.77	0.75
Software	0.82	0.84	0.87	0.71	0.55	0.55	0.71	0.73
Sports_Outdoors	0.78	0.87	0.79	0.87	0.58	0.59	0.69	0.73
Tools_Home_Improvement	0.85	0.78	0.85	0.79	0.55	0.54	0.70	0.77
Toys_Games	0.88	0.85	0.87	0.85	0.45	0.43	0.75	0.73
Video_Games	0.81	0.83	0.04	0.99	0.43	0.39	0.71	0.73

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

In order to evaluate the word embeddings, we choose sentiment analysis task to perform the experiments. Here, we briefly describe the dataset Amazon Product Reviews.

3.1 Dataset Description

Amazon product domains: This corpus is a collection of 20 product reviews derived from Task-1 of ESWC Semantic Challenge-2019. The 20 different Amazon product domains names are mentioned here [1], and this corpus belongs to sentiment analysis task. The data for the Task-1 will consist of 50k reviews for each domain of which 25k reviews are positive and 25k reviews are negative. The evaluation metrics for method evaluation are precision, recall, and macro F1-score.

3.2 Feature Extraction

In this paper, we mainly focused on four successful pretrained word embeddings such as: Word2Vec (embeddings are of 300 dimensions) [9], GloVe (embeddings are of 300 dimensions) [10], BERT (embeddings are 768 dimensions each) [17], and ELMo (embeddings are 1024 dimensions each) [16].

3.3 Training Strategy

Using the approach discussed in Section 2, we trained a separate mixture of classification experts model (MoCE) for the dataset Amazon Product Reviews with the associated task sentiment analysis using all the embeddings. The input to the MoCE model is a text vector and output is the corresponding classes based on a specific task. Here, we select the number of experts based on the number of output classes. The gating function selects one of the experts with higher probability score for the corresponding input. The selected expert predicts the target label using that particular expert weights. Both expert parameters and gating parameters are updated using the iterative expectation-maximization (EM) algorithm. The training model is validated by K-fold approach in which the model is repeatedly trained on K-1 folds and the remaining one fold is used for validation. The proposed model is trained until the model reaches the convergence with a lower bound of $1e^{-5}$ or a maximum of 100 iterations.

[1] http://www.maurodragoni.com/research/opinionmining/events/challenge-2019/

4 RESULTS & DISCUSSION

Here, we conducted the experiments in two steps. In the first step, we evaluated the four word embeddings using MoCE model and the second step describes better feature representation using cascading gcForest outperforms the state-of-the-art results on amazon product review datasets.

4.1 Evaluation of Embeddings using MoCE

Experiments are conducted on the 20 Amazon product domains 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.

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

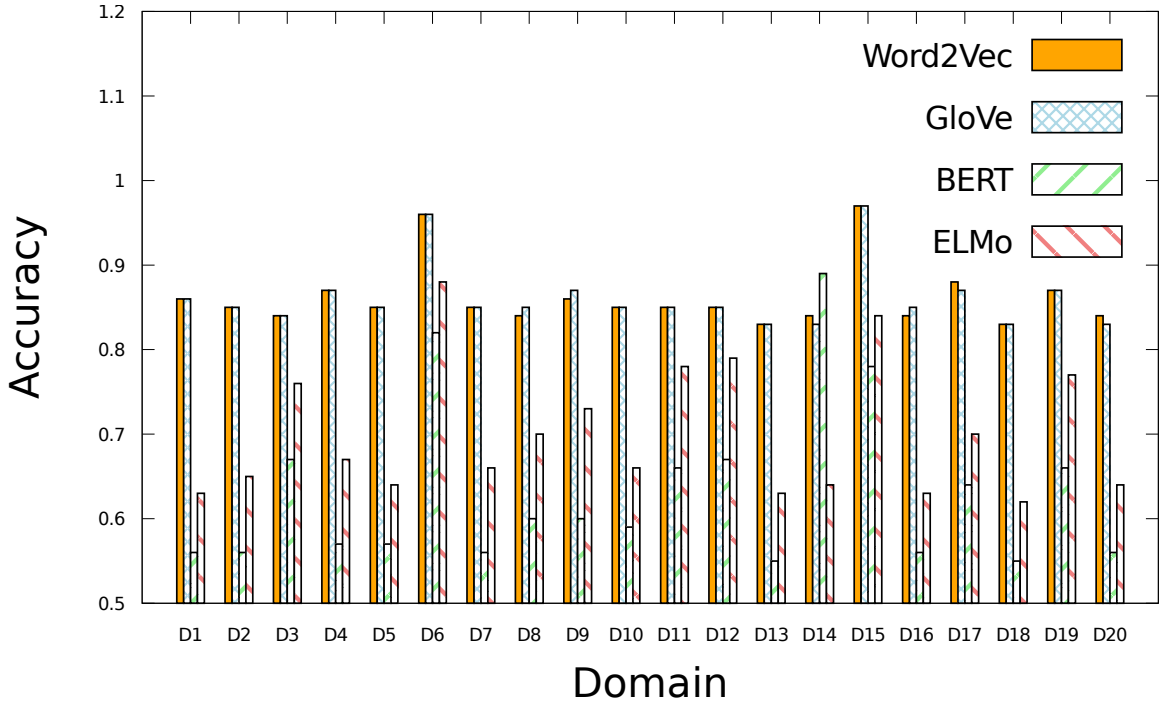

Figure 3: Figure presents the accuracy of amazon 20 products using gcForest on four word embeddings Word2Vec, GloVe, BERT, and ELMo.

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 captures 92% positive sentiment samples and expert2 captures 86% negative sentiment samples, shows better discrimination and similarly with the Word2Vec and ELMo. Word embeddings like Word2Vec and GloVe embedding feature as input, MoCE model isolate the positive and negative examples by two experts. In contrary, for the domains Baby, Electronics, Office_Products and Patio (here expert1 only captures all the positive and negative samples), this is mainly because of expressing the opinion in reviews are almost similar in both classes. However, in the case of BERT and ELMo embedding: both experts isolate the samples for all the domains to capturing of context-sensitive information.

4.2 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 [25], XGboost [26], 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 mentioned in [27] 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 four 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 with an evaluation metric accuracy. From the figure 3, we can observe that Word2Vec, GloVe, and ELMO methods perform better when compared to BERT embeddings in terms of accuracy and similar comparison we observed in table 2 using an evaluation metric F1-score. One of the main reason why BERT & ELMo do not perform 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. In Table 3, we describes the comparison between previous state-of-the-art methods and using gcForest. The combination of word embedding results comparison we observed in Table 3 and it outperforms the state-of-the-art results.

5 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 four neural word embedding methods such as Word2Vec,

Table 3: Detailed results of domains (Dom) of Amazon product reviews dataset by the Baselines, existing method results and by passing combination of word embeddings to gcForest

	Tested System (Macro F1- Score)							
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**

SVM (Support Vector Machines), ME (Maximum Entropy)
DBP (Domain Belonging Polarity), NS (NeuroSent)
DDP (Domain Detection Polarity), gcF(gcForest)
CNN (Convolutional Neural Networks)
GWE(Google Word Embeddings)

GloVe, ELMo, & BERT on sentiment analysis task in two steps (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. 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] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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*.

[8] 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.

[9] 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.

[10] 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

[11] Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*. 2397–2406.

[12] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.

[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] 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.

[15] 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.

[16] 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).

[17] 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).

[18] 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).

[19] 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.

[20] 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).

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

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

[27] Mauro Dragoni and Giulio Petrucci. 2017. A neural word embeddings approach for multi-domain sentiment analysis. *IEEE Transactions on Affective Computing* 8, 4 (2017), 457–470.