Contextualized Word Representations for Self-Attention Network

Mariam Essam
Computer and Systems Engineering Department
Ain Shams University
Cairo, Egypt
7662@eng.asu.edu.eg

Seif Eldawlatly
Computer and Systems Engineering Department
Ain Shams University
Cairo, Egypt
seldawlatly@eng.asu.edu.eg

Hazem Abbas
Computer and Systems Engineering Department
Ain Shams University
Cairo, Egypt
hazem.abbas@eng.asu.edu.eg

***Abstract*—Transfer learning is one approach that could be used to better train deep neural networks. It plays a key role in initializing a network in computer vision applications as opposed to implementing a network from scratch which could be time-consuming. Natural Language Processing (NLP) shares a similar concept of transferring from large-scale data. Recent studies demonstrated that pretrained language models can be used to achieve state-of-the-art results on a wide range of NLP tasks such as sentiment analysis, question answering, text summarization, and textual entailment. In this paper, we demonstrate that a free RNN/CNN self-attention model used for sentiment analysis can be improved with 2.53% by using contextualized word representation learned in a bilingual machine translation task.**

I. Introduction

Over the past few years, deep learning for Natural Language Processing (NLP) has attracted a lot of interest. Numerous models have been introduced to deep learning to deal with several NLP tasks. Sentiment analysis [1], text summarization [2], question answering [3], dialogue agents [4] and machine translation [5] are some of the problems in which neural network models have outperformed traditional approaches. However, there are still many challenges that need to be addressed. NLP is a challenging problem due to language ambiguity where the same word can have different meanings in different contexts. NLP is challenging also due to the complexity of languages used by humans where the same sentence can be expressed in different ways in the same language.

Sentiment analysis is an important tool for many purposes such as getting a feedback signal for social media and advertising campaigns, movies reviews and books reviews [6, 7]. Traditional methods treat a sentence as a bag-of-words, ignore word order, consult a curated list of "positive" and "negative" words with their frequencies to determine the sentiment of the sentence [8]. Given that order is ignored in this case, such methods miss fundamental aspects of sentiments such as negation. From a machine learning perspective, sentiment analysis is a classification or regression problem. Recurrent neural nets (RNN) [9], recursive neural nets [1] and convolution neural nets (CNN) [10, 11] models outperform the traditional model.

Attention mechanisms in Neural Networks are based on the visual attention mechanism found in humans [12]. Human visual attention can focus on a certain region of an image with high resolution while perceiving the surrounding image in low resolution, and then adjusting the focal point over time. Attention mechanisms have been used in image recognition in addition to NLP for neural machine translation (NMT) task [13]. NMT uses sequence-to-sequence model where it consists of an encoder to read the source language and a decoder to generate the destination language. Language mapping is not one-to-one, therefore, attention helps to find all the related words in the source language to the generated destination language.

Recent studies demonstrate that RNN with attention is able to achieve state-of-the-art for machine translation task [14]. Attention uses a feedforward network to calculate the scores for words dependent on a vector query to assign higher values to more related words and lower values to less related words. Attention captures the dependencies that make significant contributions to the task regardless of the distance between the elements. Equipping RNN with an attention mechanism has demonstrated a lot of success in several other tasks such as machine translation [15] and summarization [16].

Google AI research team has recently introduced a new sequence to sequence model, transformer, that solely uses attention. The model uses neither RNN nor CNN; it is composed only of a feedforward neural network [5]. The free RNN/CNN model achieves the state-of-the-art in neural machine translation. The transformer outperforms the models that use the RNN/CNN in both time efficiency and prediction, and this is due to the fact that attention requires only multiplication which is highly parallelizable. The proposed attention, "Multi-head-Attention", consists of three ways of attention encoder self-attention, masked decoder self-attention, and encoder-decoder attention.

This transformer has been designed for NMT task. It can be cast easily into the tasks that use seq2seq model. However, in a more recent work, Shen et al. introduced a new model, Directional Self-Attention Network (DiSAN) [17], that has been able to cast the transformer to sentence encoding models. It can be generally utilized in different NLP tasks, such as natural language inference, sentiment analysis, sentence classification, and semantic relatedness. DiSAN is a multi-dimensional and directional attention. Unlike the transformer, neither encoder-decoder pairs stack nor encoder-decoder structure is needed. DiSAN is a simple model with few parameters leads to less

computation and better prediction. It achieved the state-of-the-art in the sentiment analysis task using Stanford Sentiment trees (SST) dataset [1]. The model uses the pretrained word embedding Global Vectors for word representation (GloVe) [18].

Word to vector (word2vec) and GloVe are the most recently used pretrained word embedding techniques. Word2vec transforms words into vectors by predicting a given word using their neighboring words (Continuous Bag of Words) or predicting neighboring words given a word (Skip-grams) [19]. GloVe is an unsupervised learning algorithm that finds the vector representation for words by performing training on aggregated global word-word co-occurrence statistics from a corpus [18]. The resulted word vector representations show interesting relations in the word vector space e.g. (man + woman - king = queen).

Word2vec and GloVe are pretrained on large amounts of unlabeled data then used to initialize the first layer of a neural network. Word2vec and GloVe do not provide the model with any contextual information. Recent works demonstrate that using pretrained language models can be used to achieve state-of-the-art results in a wide range of NLP tasks [20].

In this paper, we demonstrate that using a deep contextualized word representations generated from a deep bi-directional language model (biLM), which is pre-trained on a large text corpus, can improve the state-of-the-art of DiSAN for sentiment analysis task.

II. Related Work

A. Sentiment Treebank

Stanford Sentiment Treebank (SST) is a corpus with fully labeled parse trees that consists of 11,855 single sentences extracted from movie reviews [1]. It was parsed with the Stanford parser to 215,154 unique phrases and labeled with a value in range [0,1] [21]. The phrases are classified under one of 5-classes based on the value: (0,0.2) very negative, (0.2,0.4) negative, (0.4,0.6) neutral, (0.6,0.8) positive and (0.8,1) very positive.

Fig. 1 is a sample of a fully labeled tree from SST data set, the leaves of this tree are the words and each parent node represents a phrase, the number in the top right represents the sentiment value. The model is trained on all phrases (all the tree node), i.e. the model will be trained on {One, the, best, of, year, ., the best, the year, of the year, the best of the year, One of the best of the year, One of the best of the year .}.

B. Transfer Learning

Transfer Learning is the technique of using the knowledge gained by a model trained for a source task to solve a target task which helps the model to generalize. Recent transfer learning techniques, such as Context Vectors (CoVe) [22], Embeddings from Language Models (ELMo) [23], Universal Language Model Fine-tuning (ULMFiT) [24] and openAI transformer [20], demonstrate that pretrained language models can be used to achieve state-of-the-art results on a wide range of NLP tasks. As ELMo has demonstrated success in several supervised models, in this paper, we test it on a supervised model.

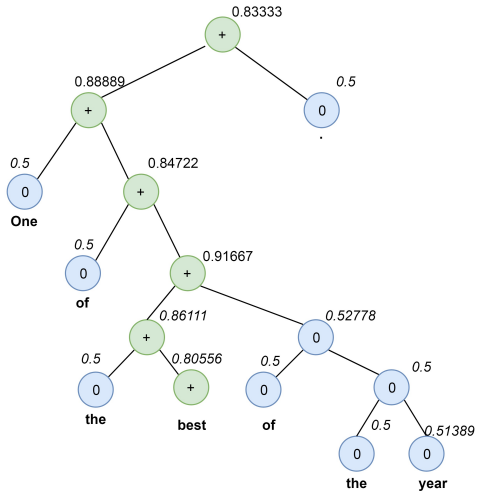

Fig. 1: SST Sample of fully labeled tree.

C. Attention

The key concept of attention is to calculate an attention weight vector to amplify the most relevant parts of the input sequence, no matter how far they were, and drown out the irrelevant parts. Additive attention (or Bahdanau attention) [15] and multiplicative attention (or Luong attention) [25] are the two major types of attention. $f(x_i, q)$ is the alignment score function that measures the dependencies between the token embedding $x_i \in [x_1, x_2, ..., x_n]$ and the vector representation of the query q.

Additive attention passes the weighted inputs $W^{(1)}x_i$ and $W^{(2)}q$ to a single layer feedforword neural network and outputs the score as shown in the following equation,

$$f(x_i, q) = w^T \sigma(W^{(1)}x_i + W^{(2)}q), \tag{1}$$

where $\sigma(.)$ is an activation function, usually $\tanh$ function, $w \in \mathbf{R}^{d_e}$, where d_e is the dimensional vector and all weight matrices $W^{(1)}$, $W^{(2)}$ and w are learned during the training process.

Multiplicative attention calculates the inner product between the token x_i and the query q. Geometrically, the dot product of two vectors is equal to multiplying the lengths of the two vectors by the cosine of the angle between them. If the angle between the two vectors is zero, then the cosine output is one and it decreases the wider the angle becomes. The dot product measures the similarities, thus if we have two vectors with the same length, the smaller the angle between the two vectors, the larger the dot product output. The score function is

$$f(x_i, q) = Dotproduct(W^{(1)}x_i, W^{(2)}q). \tag{2}$$

Additive and Multiplicative are different only in the score function, but they follow all the other steps. After calculating the alignment function, we find the $softmax$ score

$$score(x_i, q) = [f(x_i, q)]_{i=1}^{n} \tag{3}$$

$$p(z = i|x, q) = \frac{\exp(score(x_i, q))}{\sum_{i=1}^{n} \exp(score(x_i, q))}, \tag{4}$$

where n is the number of tokens, then we multiply the $softmax$ scores by each embedding x_i to get the level of expression of each of these vectors. We then sum them up producing our attention context vector,

$$c = \sum_{i=1}^{n} p(z = i|x, q)x_i, \tag{5}$$

where c $\in \mathbf{R}^{d_e}$. Additive attention outperforms the multiplicative one, but multiplicative is less time consuming due to optimized matrix multiplication using GPU.

Self-Attention, shown in Fig. 2, also called intra-attention, is a special case of the previous attention models, in which the query q is replaced by the token x_j. The token is compared or scored against each of the other words in the input vector, by this way we check the previous and the latter words and highly score the most relevant ones. Self-Attention has demonstrated success in a variety of tasks including reading comprehension [26], abstractive summarization [27] and machine translation [5].

Multi-dimensional self-attention, introduced by Shen et al., extends the additive attention at the feature level. Instead of computing single score $f(x_i, q)$, it computes a feature-wise score vector by replacing $w \in \mathbf{R}_{d_e}$ in Eq.(1) with $W \in \mathbf{R}_{d_e \times d_e}$. Token2token and source2token are two variants of multi-dimensional self-attention.

In token2token self-attention, we find the alignment score between x_i and x_j,

$$f(x_i, x_j) = W^T \sigma(W^{(1)}x_i + W^{(2)}x_j + b^{(1)}) + b. \tag{6}$$

Then we find the $softmax$ scores $p(z = i|x, x_j)$. The output of x_j is

$$s_j = \sum_{i=1}^{n} P_i^j \cdot x_i, \tag{7}$$

where the output for all elements $s = [s_1, s_2, \cdots, s_n] \in \mathbf{R}^{d_e \times n}$.

In source2token self-attention, the dependency is computed between x_i and the entire sequence x. q is totally removed, the alignment score is calculated by,

$$f(x_i) = W^T \sigma(W^{(1)}x_i + b^{(1)}) + b, \tag{8}$$

the output s is,

$$s = \sum_{i=1}^{n} P_i \cdot x_i, \tag{9}$$

where $P_i = p(z = i|x)$.

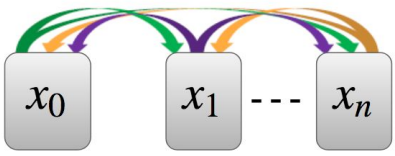

Fig. 2: Self-Attention. Attention score is calculated between each token x_j and all the other tokens x_i such that $i \neq j$.

III. Methods

In this section, we describe how we use the deep contextualized Word Representation for a self-attention model to improve the state-of-the-art of sentiment analysis task. Using ELMo for a self-attention model is done in two steps: the first step is Deep Contextualized Word Representations Generation. In this step, we describe how ELMo is used for DiSAN. In the second step, Contextualized word representation for Directional Self-Attention Network is used where we explain how after generating the contextualized word representation, we use it in the self-attention model (DiSAN).

A. Deep Contextualized Word Representations Generation

Deep contextualized word representation is an efficient pretrained word representation that models the complex characteristics of word use such as syntax and the semantics.

ELMo is known as a deep contextualized word representations technique. It is characterized by being contextual where the representation of each word is a function of all the words in the same context (sentence). Two similar words in the same sentence can have different representations as they have different meanings (context-sensitive). It is also characterized by being deep; the generation of a word representation is done by the combination of all the trained bidirectional language model (biLM) layers.

Fig. 3 shows the architecture of biLM which is composed of bidirectional Gated Recurrent Units (GRU) or Long Short-Term Memory network (LSTM) where n is number of the tokens. For a forward language model (LM), the tokens are passed to RNN starting from index 1 to index n. We compute the probability of the sequence given the previous tokens by,

$$p(x_1, x_2, ...x_n) = \prod_{k=1}^{n} p(x_k|x_1, x_2, ..., x_{k-1}). \tag{10}$$

For a backward language model (LM), the tokens are passed to RNN starting from index n to index 1. We compute the probability of the sequence given the next tokens by,

$$p(x_1, x_2, ...x_n) = \prod_{k=1}^{n} p(x_k|x_{k+1}, x_{k+2}, ..., x_n). \tag{11}$$

A biLM combines both forward LM and backward LM. The log likelihood of the forward and backward directions:

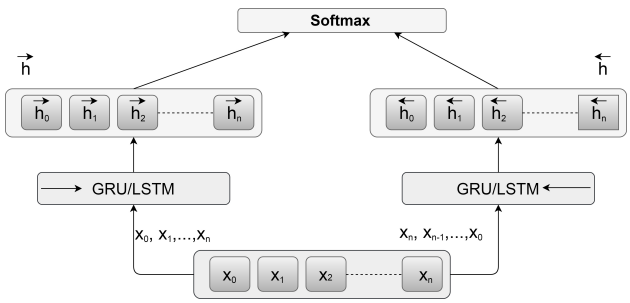

Fig. 3: Bidirectional Language Model (biLM) Architecture. It is composed of bidirectional Gated Recurrent Units (GRU) or Long Short-Term Memory network (LSTM).

$$\sum_{k=1}^{n}(\log p(x_k|x_1, x_2, ..., x_{k-1}; \Theta_t, \overrightarrow{\Theta}_{LSTM}, \Theta_s) + \log p(x_k|x_{k+1}, x_{k+2}, ..., x_n; \Theta_t, \overleftarrow{\Theta}_{LSTM}, \Theta_s)). \quad (12)$$

In the previous equation, (Θ_t) is the token representation, (Θ_s) is Softmax layer in the forward and backward direction, $\overrightarrow{\Theta}_{LSTM}$ is the LSTM parameter for forward direction and $\overleftarrow{\Theta}_{LSTM}$ is the LSTM parameter for backward direction.

ELMo is task-specific. It combines the intermediate representations layer in biLM. Fig. 4 describes its architecture, where L can be one layer or any number. In this paper, we use a pretrained model that has two biLM layers. Each token has $2L+1$ representations. For downstream tasks, ELMo collapses all the layers in one vector R, we compute a task specific of all biLM layers:

$$ELMo_k^{task} = E(R_k; \Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM}, \quad (13)$$

$$h_{k,j}^{LM} = \langle \overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} \rangle, \quad (14)$$

where s^{task} are softmax-normalized weights, γ^{task} is a scalar parameter that allows the task model to scale the entire ELMo vector and aids the optimization process, $h_{k,j}^{LM}$ concatenate hidden parameters for the forward LM and backward LM.

B. Contextualized Word Representation for Directional Self-Attention Network

After biLM is trained, we use it to generate the word vector for each token in the dataset phrases. We then use the generated embedding $ELMo^{task} = [E_1, E_2, ..., E_n]$ as an input for DiSAN. Unlike GloVe, the word representations generated using ELMo are functions of the entire input sentence.

DiSAN is composed of bidirectional (forward/backward) Directional self-attention (DiSA) blocks, shown Fig. 5. Their outputs are denoted by $\overrightarrow{u}, \overleftarrow{u}$. The outputs are concatenated vertically $[\overrightarrow{u}; \overleftarrow{u}]$ and used as input for multidimensional source2token self-attention block.

DiSA is composed of a fully connected layer whose input is E (ELMo word representations), a masked multi-dimensional

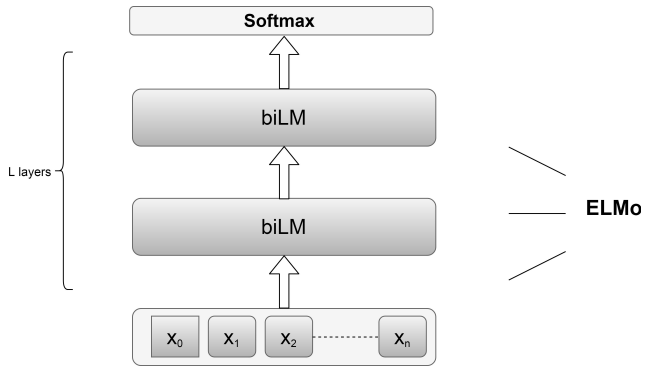

Fig. 4: Embeddings from Language Model (ELMo) Architecture.

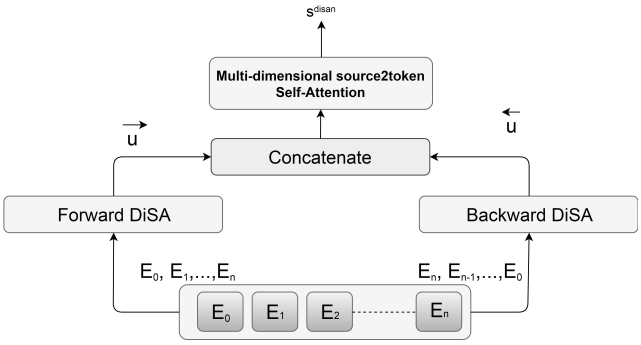

Fig. 5: DiSAN Architecture.

token2token self-attention, and a fusion gate to compute the result u by combining the output and the input of the attention block.

First, we pass the input sequence $E = [E_1, E_2, \cdots, E_n]$ to a fully connected layer and we get $h = [h_1, h_2, \cdots, h_n]$ by,

$$h = \sigma_h(W^{(h)}E + b^{(h)}), \quad (15)$$

where $W^{(h)}$ and $b^{(h)}$ are learnable parameters.

After getting h, we apply multi-dimensional token2token self-attention to it and generate context-aware vector representations s for all elements from the input sequence. For simplicity and to reduce the number of the trained parameters to enhance the performance, two modifications are done to Eq.(6), first we replace W by $const$ and we divide the activation function $\sigma(.)$ by it. In the second modification, we apply a positional mask $M \in \{0, -\infty\}^{n \times n}$ so the attention between the elements can be asymmetric and b is replaced by an all-one vector I. Hence, the alignment function becomes,

$$f(h_i, h_j) = const \cdot \tanh([W^{(1)}h_i + W^{(2)}h_j + b^{(1)}]/const) + M_{ij}I, \quad (16)$$

where the activation function $\sigma(.)$ is $\tanh(.)$.

The mask is an important feature in DiSA, it can be diagonal-disabled M_{ij}^{diag}, forward M_{ij}^{fw} or backward M_{ij}^{bw}. When $M_{ij} = -\infty$, the value of the aligned score function $f(h_i, h_j) = -\infty$, the attention is disabled as the $softmax$ result will be zero. On the contrary, when we have $M_{ij} = 0$ then $f(h_i, h_j) > 0$, then $softmax > 0$, therefore, attention is enabled.

To disable the attention of each token to itself we apply diagonal-disabled mask

$$M_{ij}^{diag} = \begin{cases} 0, & i \neq j \\ -\infty, & i = j \end{cases}. \quad (17)$$

To encode temporal order information into attention output, we use forward or backward mask,

$$M_{ij}^{fw} = \begin{cases} 0, & i < j \\ -\infty, & otherwise \end{cases}, \quad (18)$$

$$M_{ij}^{bw} = \begin{cases} 0, & i > j \\ -\infty, & otherwise \end{cases}. \quad (19)$$

After getting $f(h_i, h_j)$, we use token2token attention softmax function, Eq.(7), to find the score for all j. The output result u is the combination of the s with h, the combination is done in a dimension fusion gate,

$$F = sigmoid(W^{(f1)}s + W^{(f2)}h + b^{(f)}), \quad (20)$$

$$u = F \cdot h + (1 - F) \cdot s, \quad (21)$$

where $W^{(f1)}$, $W^{(f2)}$ and $b^{(f)}$ are trainable parameters.

Each token, using ELMo, is a vector of 1024D, compared to GloVe which is 300D. We pass the input sequence to a 900D fully connected layer. All the biases are initialized with zeros. Unlike GloVe, all words should have a distinguished word vector so there are no Out-of-Vocabulary words. The learning rate used is 0.5, the dropout rate is 0.7 and for optimization, Adadelta is used. The word vectors representation is fine-tuned during the training phase. The model is implemented with Tensorflow and run on an Ubuntu 18.0 machine with a Single graphics card unit (GPU) GTX 1080Ti and 16GB RAM.

IV. Results

We apply the contextualized self-attention model to sentiment analysis tasks and we test the impact of this addition.

The model is tested for sentiment analysis task using Stanford Sentiment Treebanking dataset (SST). We use the same dataset used for DiSAN with GloVe to evaluate the performance of the proposed approach [17]. In the beginning, we split the data to training, development and testing datasets with 8544, 1101, and 2210 samples, respectively. In the training phase, we use the phrases to train the model. During development and testing phase, we use the full sentence. The calculated accuracy percentage indicates how much the predicted result matches the expected result.

Table I shows the results achieved using the proposed approach compared to DiSAN with Glove [17]. Using contextualized word representation in the model enhances the results by 2.53%. Fig. 6 shows the accuracy per epoch of the two models. The figure demonstrates that the addition of the contextualized word representation enhanced the overall performance. Our model reaches the best accuracy at epoch 6 and can keep same performance for another two epochs then the accuracy drops down as the model is overtrained after epoch 8.

TABLE I: TEST ACCURACY FOR SENTIMENT ANALYSIS.

Model	**Test accuracy(%)**
DiSAN+ GloVe	51.72
DiSAN+ ELMo	**54.25**

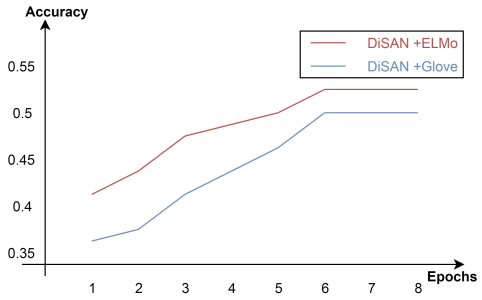

Fig. 6: Test Accuracy per epoch.

TABLE II: DISAN WITH GLOVE CONFUSION MATRIX.

		Predicted Class				
		Very Negative	**Negative**	**Neutral**	**Positive**	**Very Positive**
Actual Class	**Very Negative**	95	157	8	18	1
	Negative	63	447	17	99	7
	Neutral	7	171	34	170	7
	Positive	1	61	11	368	69
	Very Positive	0	12	2	186	199

TABLE III: DISAN WITH ELMO CONFUSION MATRIX.

		Predicted Class				
		Very Negative	**Negative**	**Neutral**	**Positive**	**Very Positive**
Actual Class	**Very Negative**	76	195	4	4	0
	Negative	48	474	63	39	9
	Neutral	13	170	95	104	7
	Positive	0	63	34	305	108
	Very Positive	0	10	9	131	249

In addition, we calculated the confusion matrix of the two models. Table II shows the confusion matrix of DiSAN with Glove and Table III shows the confusion matrix of DiSAN with ELMo. We found that the percentages of a right expectation across the 5-classes (very negative, negative, neutral, positive and very positive) for *DiSAN+GloVe* is [34.1%, 70.6%, 8.7%, 72.2% and 49.9%] and for *DiSAN+ELMO* is [27.2%, 74.9%, 18.6%, 59.8% and 62.4%], respectively. Results indicate that our model is able to enhance the results of three classes (negative, neutral and very positive). However, the proposed model consumes more time to be trained. It needs 6 $times$ the time needed by the model that uses GloVe. This is due to the vector representations dimensionality is higher than that is of GloVe. This could be enhanced by reducing the number of training parameters.

V. Conclusion

Pretrained word embeddings are currently being replaced by pretrained language models in a multitude of tasks. It is just a matter time until pretrained word embeddings will be dethroned. In this paper, we demonstrated that using deep contextualized word representation, pretrained on language modeling, for Directional Self-Attention model can improve its state-of-the-art for sentiment analysis task. Our results demonstrated an enhancement of 2.53% over the Glove technique. Results also demonstrate a significant enhancement in 3 out of the examined sentiment classes. These results demonstrated the efficacy of pretrained language models in sentiment analysis.

References

[1] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[2] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," *arXiv preprint arXiv:1705.04304*, 2017.

[3] C. Xiong, V. Zhong, and R. Socher, "Dcn+: Mixed objective and deep residual coattention for question answering," *arXiv preprint arXiv:1711.00106*, 2017.

[4] T.-H. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P.-H. Su, S. Ultes, and S. Young, "A network-based end-to-end trainable task-oriented dialogue system," *arXiv preprint arXiv:1604.04562*, 2016.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[6] B. Pang, L. Lee *et al.*, "Opinion mining and sentiment analysis," *Foundations and Trends® in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[7] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *Language Resources and Evaluation conference*, vol. 10, no. 2010, 2010, pp. 1320–1326.

[8] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[9] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification." in *the American Association for Artificial Intelligence conference*, vol. 333, 2015, pp. 2267–2273.

[10] C. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 69–78.

[11] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[12] V. Mnih, N. Heess, A. Graves *et al.*, "Recurrent models of visual attention," in *Advances in neural information processing systems*, 2014, pp. 2204–2212.

[13] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[14] F. Kokkinos and A. Potamianos, "Structural attention neural networks for improved sentiment analysis," *arXiv preprint arXiv:1701.01811*, 2017.

[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[16] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv:1509.00685*, 2015.

[17] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," in *the American Association Conference on Artificial Intelligence*, 2018.

[18] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[20] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf*, 2018.

[21] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 423–430.

[22] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Advances in Neural Information Processing Systems*, 2017, pp. 6294–6305.

[23] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[24] J. Howard and S. Ruder, "Fine-tuned language models for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[25] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[26] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," *arXiv preprint arXiv:1601.06733*, 2016.

[27] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," *arXiv preprint arXiv:1606.01933*, 2016.