SENTIMENT ANALYSIS WITH RECURRENT NEURAL NETWORK AND UNSUPERVISED NEURAL LANGUAGE MODEL

Ph.D. Candidate: Abdalraouf Hassan, Advisor: Ausif Mahmood
Dep of Computer Science and Engineering, University of Bridgeport, CT, 06604, USA
abdalrah@my.bridgeport.edu, mahmood@bridgeport.edu

ABSTRACT

This paper describes a simple and efficient Neural Language Model approach for text classification that relies only on unsupervised word representation inputs. Our model employs Recurrent Neural Network Long Short-Term Memory (RNN-LSTM), on top of pre-trained word vectors for sentence-level classification tasks. In our hypothesis we argue that using word vectors obtained from an unsupervised neural language model as an extra feature with RNN-LSTM for Natural Language Processing (NLP) system can increase the performance of the system. Our technique has improved the model to capture syntactic and semantic word relationships. We show that simple RNN-LSTM with word2vec achieves excellent result on IMDB Stanford benchmark for sentiment analysis task.

***Index Terms*—** RNN, LSTM, Bag of words, N-grams.

1. INTRODUCTION

Text classification is a classic topic in natural language processing (NLP). When one need to assign predefined categories to free-text document, text classification play important roles in many applications such as, document retrieval, web search, spam filtering. Machine learning algorithms also in the core of these applications, e.g. logistic regression or K-means. These algorithms require the text input to be represented as a vector. The range of the research in text classification goes from designing the best features to choosing the best possible machine learning classifiers. [1, 2]. In NLP, many techniques deals with words as discreet atomic symbols, therefore, these methods provide no useful information to the system, and no notion of similarity detected among the words.

A popular method for fixed-length vector representations is bag-of-words, a text such as a sentence or a document is represented as multiset of its words. the frequency of each word is used as a feature for training classifier [3]. However, there is a drawback for this method, e.g. ignoring the order of words, and grammar, which means different sentences, can have the same vector representations.

Another popular method is n-grams, which typically perform the best [4]. However, n-grams take in account the word order in short sentence, but it suffers from high dimensionality and data sparsity. The simple techniques are at their limits in many NLP tasks. In this paper we propose new a methodology that capture syntactic and semantic relationships in word representations.

2. OVERVIEW AND RELATED WORK

Recursive neural network proved to be efficient in constructing sentence representations. The model has tree structure, which is able to capture semantic of sentence. However, this is a time-consuming task, due to constructing the textual tree complexity [5].

RNN has better time complexity. In this model text is analyzed word by word, then preserve the semantic of all the previous text in a fixed-sized hidden layer [6]. The ability to capture better contextual information could be beneficial to capture semantics of long text in recurrent network. However, RNN suffer from vanishing gradients, and makes it difficult to learn long-distance correlation in sequence.

Our technique is inspired by the recent work of [7], where simple model was train with one layer of convolution on top of words vectors. These vectors were obtained from unsupervised neural network language model [8] that was trained on 100 billion words of Google News.

3. METHODOLOY AND APPROACH

3.1. Long Short-Term Memory LSTM:

LSTM are a type of RNN. Prediction is sequentially in RNN, and the hidden layer from one prediction is the hidden layer of the next prediction. This will assign a memory to the network. Results from previous predictions can improve future predictions. LSTM provides RNN an extra aspect that gives it a fine-grained control over memory. This aspects control how much the current input matters in creating the new memory, and how much the previous memories matters in creating the new memory, and what parts of the memory are important in generating the output, word2vec improved the performance of the model, in the absence of a large supervised training set [5, 9].

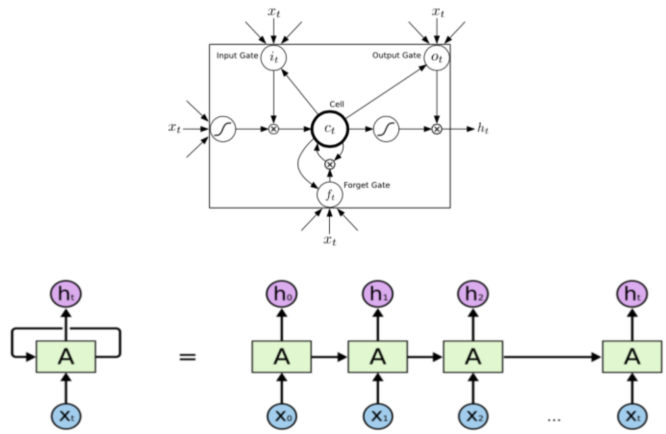

Figure 1: shows the setup for LSTM using only last hidden layer for sentiment prediction. LSTM jointly trained for language molding we experimented averaging hidden states and use softmax/cross-entropy loss associated with sentiment analysis.

3.2. Dataset and Results

The IMDB dataset was first proposed in [10], as a benchmark for sentiment analysis. We follow the basic recipes for implementing the LSTM, by extracting the cell outputs and gradients. Using available dataset for sentiment analysis IMDB. Mostly applied method on this dataset such as bag-of-words or n-grams, classically ignore long-range ordering information.

Model	IMDBA
NBSVM-bi	91.2 %
Paragraph Vector	94.5 %
LSTM-word2vec	**95.1** %

Table 1: The performance of the LSTM-word2vec compared to other approaches on the IMDB dataset binary classification task, NBSVM-bi [11] . Paragraph Vector [12].

4. CONCLUSION AND FUTURE WORK

We have introduced a language model that utilizes only word2vec inputs over LSTM. Our experiments using the IMDB sentiment analysis dataset show that the method is competitive with state-of-the-art to previous reported results. We can capture more information about the syntactic and semantic of word by using pre-trained word vectors. Hence, our proposed model has the prospective to overcome several flaws in traditional methods e.g., bag-of-words and n-gram models where order and information about word is vanished. The combination of two models yields a significant improvement. Work in progress is evaluating LSTM-word2vec on multiple benchmarks, implementing and testing several deep learning inspired models, and comparing them to traditional methods. Finally, we are using our method to build an NLP system that perform sentiment analysis on twitter data that ultimately predict movements in stock market.

5. REFERENCES

[1]. Deerwester, S., et al., *Indexing by latent semantic analysis.* Journal of the American society for information science, 1990. **41**(6): p. 391.

[2]. Pang, B., L. Lee, and S. Vaithyanathan. *Thumbs up?: sentiment classification using machine learning techniques*. in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. 2002. Association for Computational Linguistics.

[3]. Harris, Z.S., *Distributional structure.* Word, 1954. **10**(2-3): p. 146-162.

[4]. Joachims, T. *Text categorization with support vector machines: Learning with many relevant features*. in *European conference on machine learning*. 1998. Springer.

[5]. Socher, R., et al. *Parsing natural scenes and natural language with recursive neural networks*. in *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011.

[6]. Elman, J.L., *Finding structure in time.* Cognitive science, 1990. **14**(2): p. 179-211.

[7]. Kim, Y., *Convolutional neural networks for sentence classification.* arXiv preprint arXiv:1408.5882, 2014.

[8]. Mikolov, T., et al. Distributed representations of words and phrases and their compositionality. in Advances in neural information processing systems. 2013.

[9]. Collobert, R., et al., *Natural language processing (almost) from scratch.* Journal of Machine Learning Research, 2011. **12**(Aug): p. 2493-2537.

[10]. Maas, A.L., et al. *Learning word vectors for sentiment analysis*. in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. 2011. Association for Computational Linguistics.

[11]. Wang, S. and C.D. Manning. *Baselines and bigrams: Simple, good sentiment and topic classification*. in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. 2012. Association for Computational Linguistics.

[12]. Hong, J. and M. Fang, *Sentiment Analysis with Deeply Learned Distributed Representations of Variable Length Texts.*