Evaluating the Addition of Syntactic Information in Deep Learning Models for Sentiment Analysis

Andrei Mircea (260585208), Stefan Wapnick (260461342)

School of Computer Science, McGill University
andrei.mircea@mail.mcgill.ca
stefan.wapnick@mail.mcgill.ca

Abstract

This document contains instructions for preparing NAACL HLT 2016 submissions and camera-ready manuscripts. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. Papers are required to conform to all the directions reported in this document. In this Word template, the required formatting is preformatted for author use and further instructions are provided for how to use Word formatting.

1 Introduction

Sentiment analysis is an important application of Natural Language Processing (NLP) with widespread use in industry, where distilled information on opinion can be a significant source of value. Typically, this task is framed as a text classification problem whereby a text document is categorized within predefined sentiment labels (Zhang, Wang, & Liu, 2018).

Over the past decade, deep learning (DL) has become a popular machine learning approach, yielding state-of-the-art results on such text classification tasks. The input features to these DL models are generally word embeddings which encode word co-occurrence information; and by extension, some degree of semantic information (Senel, Utlu, Yucesoy, Koc, & Cukur, 2017).

However, this approach ignores explicit syntactic information traditionally used in NLP, such as dependency parse trees and part-of-speech (POS) tags. There is conflicting information as to whether this information can improve performance of DL models on NLP tasks such as text classification for sentiment analysis as mentioned in our related works section.

We address this unresolved question by isolating and evaluating the impact of syntactic information on deep learning models for sentiment analysis. Specifically, we propose a novel architecture for automatically generating and encoding POS-tag and parse tree information into the input feature space of three different DL models (CNN, BiLSTM and simple feed-forward) used for polarity classification in the IMDB review dataset (Maas, et al., 2011).

2 Related Work

While the use of parse tree and POS pattern information has been successfully used to improve sentiment analysis with linguistic-based approaches (Biagioni, 2016) as well as traditional machine learning approaches (Das & Balabantaray, 2014; Nicholls & Song, 2009); there is limited research on using this information to augment the input features of DL models. Moreover, it was found that word embeddings already carry syntactic information (Andreas & Klein, 2014), and that DL models can learn internal representations that capture syntactic information (Blevins, Levy, & Zettlemoyer, 2018).

This suggests that adding syntactic information to word-embeddings as input features to deep learning models would have a limited impact on model performance. Despite this, there are findings that suggest the contrary. For example, Rezaeinia et al. (2018) created constant vectors for POS tags and appended them to pre-trained word-embeddings for sentiment analysis. However, their experiment does not isolate the impact of POS tag information on performance. Conversely, Liu et al. (2018) used a

Baseline parameter	Value
Maximum text sequence length	300
Batch size	256
Dropout	0.5

Table 1: Test accuracy without word-embedding training

Bi-directional LSTM to convert a variable length sentence to a fixed-length vector representation of its parse tree structure; and found it to improve question answering on the SQuAD dataset.

We extend this research by (1) encoding POS-tags as a trainable embedding layer instead of constant vectors; (2) encoding dependency parse tree information as a filter which multiplies vectors for dependent words; (3) isolating and evaluating the impact of augmenting word embeddings with this syntactic information on a sentiment analysis task with different DL model architectures.

3 Method

3.1 Data

The large movie (imdb) review dataset (Maas, et al., 2011) was used through the Tensorflow Keras API with a train/dev/test split of 22,500/2,500/25,000 reviews. SpaCy's en_core_web_md model (Honnibal & Montani, 2017) was used to extract lemmatized word embeddings, POS tags, and dependency parse trees from reviews. To increase training speed, glove-wiki-gigaword-100 word embeddings were used through the Gensim API; as we found no significant loss in performance compared to SpaCy's larger word embeddings. Furthermore, both embeddings had a 23% out-of-vocabulary rate on the dataset lexicon.

3.2 Model Architectures

Using the Tensorflow Keras API, we created three baseline model architectures to test our hypothesis: Bidirectional LSTM, CNN, and feed forward. Hyper-parameter tuning was done to select values for dropout, sequence truncation length, and batch size that maximized dev-set accuracy while preventing out-of-memory errors and overfitting. Furthermore, we evaluated the addition of a trainable word embedding layer with randomly initialized weights for out-of-vocabulary words for each experiment.

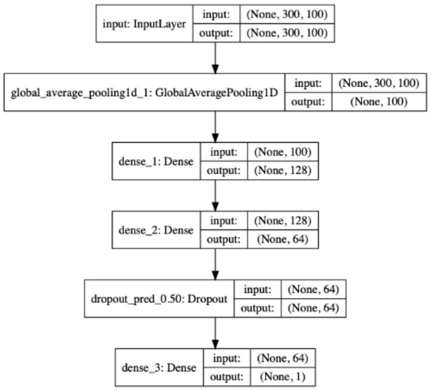
3.3 Input Features

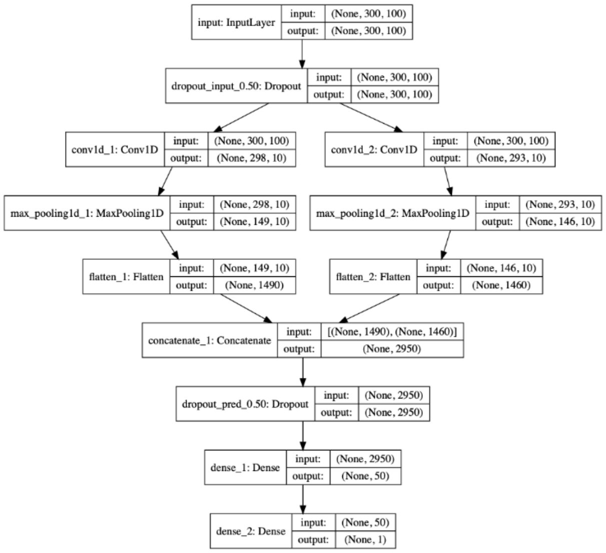
To augment pre-trained word embeddings with POS tag information, POS tags were converted to indices using the Universal POS tags schema (Universal Dependencies Contributors, 2014) and encoded as vectors with two approaches: (1) one-hot encoded vectors; (2) 10-d trainable embedding layer. The resulting POS vectors were then combined with word embeddings using a concatenation layer. Different embedding layer dimensions were tested with no observed change in performance

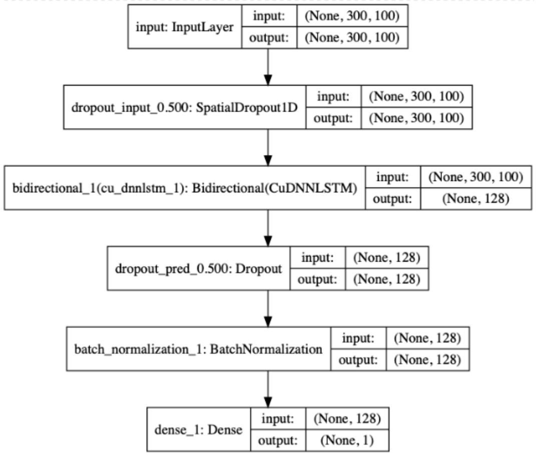
Dependency parse tree information was encoded as a filter matrix which rearranges a sequence of word embeddings such that the head of each word takes the place of its child. The resulting tensor is then element-wise multiplied with the original tensor such that each word embedding is multiplied by the word embedding of its head. The intuition behind this approach was to amplify "relevant" dimensions in word embeddings, where relevance is defined as co-occurrence with the head word. A similar approach to subtract head word vectors from child word vectors to encode paradigmatic relations was attempted. However, this decreased performance, likely because paradigmatic relations captured by vector differences (Senel et al, 2017) do not correspond to the syntactic relations captured by a dependency parse tree.

3.4 Training and Evaluation

The models were trained until dev-set accuracy stopped improving to prevent overfitting. Models were then evaluated using test set accuracy, f1-score, precision, and recall. For each model architecture, a difference to the baseline was calculated to isolate and evaluate the impact of syntactic information on performance.







	LSTM		**CNN**		**FF**	
	Acc.	Δ	Acc.	Δ	Acc.	Δ
baseline	84.30	0	83.06	0	78.31	0
pos	84.17	**-0.13**	82.77	**-0.28**	78.82	**0.51**
pos-embed	84.66	**0.36**	82.48	**-0.57**	78.82	**0.51**
parse	79.90	**-4.40**	76.85	**-6.20**	73.32	**-4.99**
pos + parse	79.49	**-4.80**	76.50	**-6.56**	73.19	**-5.12**
pos-embed + parse	80.97	**-3.33**	78.04	**-5.02**	74.71	**-3.60**

Table 2: Test accuracy without word-embedding training

	LSTM		**CNN**		**FF**	
	Acc.	Δ	Acc.	Δ	Acc.	Δ
baseline	86.21	0	87.25	0	86.91	0
pos	86.78	**0.57**	87.90	**0.67**	86.83	**-0.08**
pos-embed	86.70	**0.49**	87.92	**0.64**	86.85	**-0.06**
parse	83.85	**-2.36**	85.28	**-1.97**	85.4	**-1.50**
pos+ parse	84.23	**-1.98**	85.46	**-1.79**	85.51	**-1.40**
pos-embed + parse	83.18	**-3.03**	85.7	**-1.55**	85.52	**-1.39**

Table 2: Test accuracy with word-embedding training

4 Results

The NAACL HLT 2016 main conference accepts long and short paper submissions. Long papers may consist of up to eight (8) pages of content, plus unlimited pages for references. Accepted long papers will be given one additional page (up to 9 pages with unlimited pages for references) so that reviewers' comments can be taken into account. Short papers may consist of up to four (4) pages of content, plus unlimited pages for references. Accepted short papers will be given five (5) pages in the proceedings and unlimited pages for references. For both long and short papers, all illustrations and appendices must be accommodated within these page limits, observing the formatting instructions given in the present document.

5 Statement of Contributions

5.1 Literature Review

Both team members worked jointly during the literature review process and contributed equally.

5.2 Implementation

Andrei: Data preprocessing, extraction of syntactic data, word index processing, sequence class data generator, experiment wrapper, result analysis.

Stefan: Data visualization, train/dev split, word vector embeddings cache, file system helper methods, keras neural network model generation, result analysis.

5.3 Documentation

Andrei: Draft of introduction, related world, discussion and conclusion.
Stefan: Draft of method and result sections.

Final review and formatting were done jointly by both team members.

6 Double-blind review process

As the reviewing will be blind, the paper must not include the authors' names and affiliations. Furthermore, self-references that reveal the author's identity, e.g., "We previously showed (Smith, 1991) ..." must be avoided. Instead, use citations such as "Smith previously showed (Smith, 1991) ..." Papers that do not conform to these requirements will be rejected without review. In addition, please do not post your submissions on the web until after the review process is complete (in special cases this is permitted: see the multiple submission policy below).

We will reject without review any papers that do not follow the official style guidelines, anonymity conditions and page limits.

7 Multiple Submission Policy

Papers that have been or will be submitted to other meetings or publications must indicate this at submission time. Authors of papers accepted for presentation at NAACL HLT 2016 must notify the program chairs by the camera-ready deadline as to whether the paper will be presented. All accepted papers must be presented at the conference to appear in the proceedings. We will not accept for publication or presentation papers that overlap significantly in content or results with papers that will be (or have been) published elsewhere.

Preprint servers such as arXiv.org and ACL-related workshops that do not have published proceedings in the ACL Anthology are not considered archival for purposes of submission. Authors must state in the online submission form the name of the workshop or preprint server and title of the non-archival version. The submitted version should be suitably anonymized and not contain references to the prior non-archival version. Reviewers will be told: "The author(s) have notified us that there exists a non-archival previous version of this paper with significantly overlapping text. We have approved submission under these circumstances, but to preserve the spirit of blind review, the current submission does not reference the non-archival version." Reviewers are free to do what they like with this information.

Authors submitting more than one paper to NAACL HLT must ensure that submissions do not overlap significantly (> 25%) with each other in content or results. Authors should not submit short and long versions of papers with substantial overlap in their original contributions.

8 STREAM Tools

This Microsoft Word file has been preset for compatible use with the STREAM Tools template designed for creating well-formatted reports and papers with Microsoft Word. The principles behind this template and others STREAM templates are explained in (Mamishev, 2010; Mamishev, 2013).

Acknowledgments

Do not number the acknowledgment section.

References

Alfred. V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling, volume 1*. Prentice-Hall, Englewood Cliffs, NJ.

American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.

Association for Computing Machinery. 1983. *Computing Reviews*, 24(11):503-512.

Ashok K. Chandra, Dexter C. Kozen, and Larry J.Stockmeyer. 1981. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114-133.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Alexander V. Mamishev and Murray Sargent. 2013. *Creating Research and Scientific Documents Using Microsoft Word*. Microsoft Press, Redmond, WA.

Alexander V. Mamishev and Sean D. Williams. 2010. *Technical Writing for Teams: The STREAM Tools Handbook*. Wiley-IEEE Press, Hoboken, NJ.

References

Andreas, J., & Klein, D. (2014). How much do word embeddings encode about syntax? *ACL*.

Biagioni, R. (2016). *The SenticNet Sentiment Lexicon: Exploring Semantic Richness in Multi-Word Concepts.* Springer.

Blevins, T., Levy, O., & Zettlemoyer, L. (2018). Deep RNNs Encode Soft Hierarchical Syntax. *ACL*.

Das, O., & Balabantaray, R. C. (2014). Sentiment Analysis of Movie Reviews using POS tags. *International Journal of Computer Applications*, 36-41.

Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Liu, R., Hu, J., Yang, Z., & Nyberg, E. (2018). *Structural Embedding of Syntactic Trees for Machine Comprehension.* Carnegie Mellon University.

Maas, A., Daly, R., Pham, P., . Huang, D., Ng, A., & Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 142--150). Portland: Association for Computational Linguistics.

Nicholls, C., & Song, F. (2009). Improving Sentiment Anaysis With Part-Of-Speech Weighting. *Eighth International Conference on Machine Learning and Cybernetics.* Baoding.

Rezaeinia, S. M., Rahmani, R., & Ghodsi, A. (2018). *Text Classification based on Multiple Block.*

Senel, L. K., Utlu, ˙., Yucesoy, V., Koc, A., & Cukur, T. (2017). Semantic Structure and Interpretability of Word Embeddings. *arXiv e-prints*.

Universal Dependencies Contributors. (2014). *Universal POS tags*. Retrieved from http://universaldependencies.org: http://universaldependencies.org/u/pos/

Zhang, L., Wang, S., & Liu, B. (2018). Deep Learning for Sentiment Analysis: A Survey. *arXiv e-prints*.