Sentiment Analysis on Movie Reviews using Recursive and Recurrent Neural Network Architectures

Aditya Timmaraju
Department of Electrical Engineering
Stanford University
Stanford, CA - 94305
adityast@stanford.edu

Vikesh Khanna
Department of Computer Science
Stanford University
Stanford, CA - 94305
vikesh@stanford.edu

Abstract

In this project, we introduce a method to tackle the problem of sentiment classification of large movie reviews, each consisting of multiple sentences. We leverage a new architecture that encapsulates previous work applied to sentence-level sentiment classification, using Recursive Neural Networks and use it in unison with a Recurrent Neural Network (where each time step is a sentence in the review). In this report, we describe our results with this architecture as well as four other intermediate approaches, one of which involves hand-crafted features. We demonstrate through our experiments that hand-crafted features carry complementary discriminative information in addition to that present in the learned neural network models.

1 Introduction

The problem of sentiment classification has received a lot of attention of late. It finds applications in the fields of business intelligence, recommender systems (so as to be able to understand the sentiment in a user's feedback), in online surveys that have natural language sentences as responses, message filtering, assessing movie review polarity, among others.

There have also been a spate of developments in the field of natural language processing using methods from deep neural networks, for various tasks including simpler ones like POS tagging and Named Entity Recognition [6].

We wish to investigate ways in which these recent advances can be applied to the problem of sentiment classification for paragraphs of text (i.e., with multiple sentences, as is the case typically in moview reviews).

2 Related Work

Sentiment Classification has traditionally been solved using linear classification methods, such as Supper Vector Machines (SVM) and logistic regression [1]. In early work in [3], methods such as Naive Bayes classification and Maximum Entropy were studied. Advances in deep learning have also recently been applied to sentiment classifcation. For instance, in [1], Maas et. al propose a strategy to learn word vectors specifically for the task of sentiment analysis. They use an unsupervised model to learn the semantic similarities between words, and a supervised component that is able to capture nuanced sentimental information. They show very promising results - their semantic + sentiment model captures sentimental similarity between words extremely well. For instance, 'melancholy' is closest to 'bittersweet' and 'heartbreaking' (sentiment) as opposed to 'thoughtul' and 'warm' (semantic). However, our intent is to propose an architecture that performs well with off-the-shelf word vector representations (like word2vec).

In [2], Richard et. al. argue that semantic word spaces cannot express the meaning of longer phrases in a principled way. They introduce Recursive Neural Tensor Network that pushes the state of the art in single sentence positive/negative classification. They also introduced 'Stanford Sentiment Treebank', a dataset that contains over 215,154 phrases with fine-grained sentiment lables over parse trees of 11,855 sentences. Their results clearly outperform bag-of-words models, since they are able to capture phrase-level sentiment information in a recursive way.

3 Technical Approaches

3.1 Using Semantic Word Vectors and Recurrent Architecture

In each review, we first separate the sentences from one another. We then represent each sentence in the review with the average of word vectors of each word in the sentence. We then feed each of these averaged word vectors into a Recurrent Neural Network (RecNN). This RecNN is designed in such a way that the loss (cross entropy) only propagates from the last time step, all the way back to the first time step, based on the predicted and actual sentiments of the full review.

3.2 Recursive Neural Network with mean likelihood

We split the review into sentences and feed the parse tree of each sentence into a recursive neural network. The class probabilities output by the RNN are averaged to decide the most likely class.

3.3 Recursive Neural Network with Affine NN

We split the review into sentences and feed the parse tree of each sentence into a recursive neural network. The hidden vectors output by the RNNs are averaged into a single vector and fed into a 2-layer neural network (affine NN).

3.4 Averaged Semantic Word Vectors

We represent each review by the average of the word vectors of all the words in the review. We then use this as the feature as classify the sentiment of the review using an affine Neural Network/a Support Vector Machine.

3.5 Semantic Word Vectors and Bag-of-Words Features

In addition to the averaged word vector representation of all the words in the review, we also include bag-of-words features with IDF weighting and then use a Support Vector Machine classifier on this concatenated feature vector. We experimented with a variety of loss functions and regularization strengths for the SVM classifier, as this was our best performing model.

3.6 Recursive-Recurrent Neural Network Architecture

In this approach, we use the idea of recursively learning phrase-level sentiments [2] for each sentence and apply that to longer documents the way humans interpret languages - forming sentiment opinion from left to right, one setnence at a time. Specifically, we split the movie review into sentences. Then, each sentence is fed into a recursive neural network (RNN) which outputs a hidden vector and a sentiment for the sentence. We pass the RNN's hidden vector of each sentence as an input to a recurrent neural network (i.e., we consider each sentence as a time step for the RecNN). It is certainly conceivable that sentences that occur early on in a review need to be considered differently from those in the last part, to evaluate their importance for sentiment classification. We explore this idea later in the report. In this way, we are able to capture phrase-level sentiment for each sentence and sentence-level sentiments for the whole document. This architecture is illustrated in Figure 1.

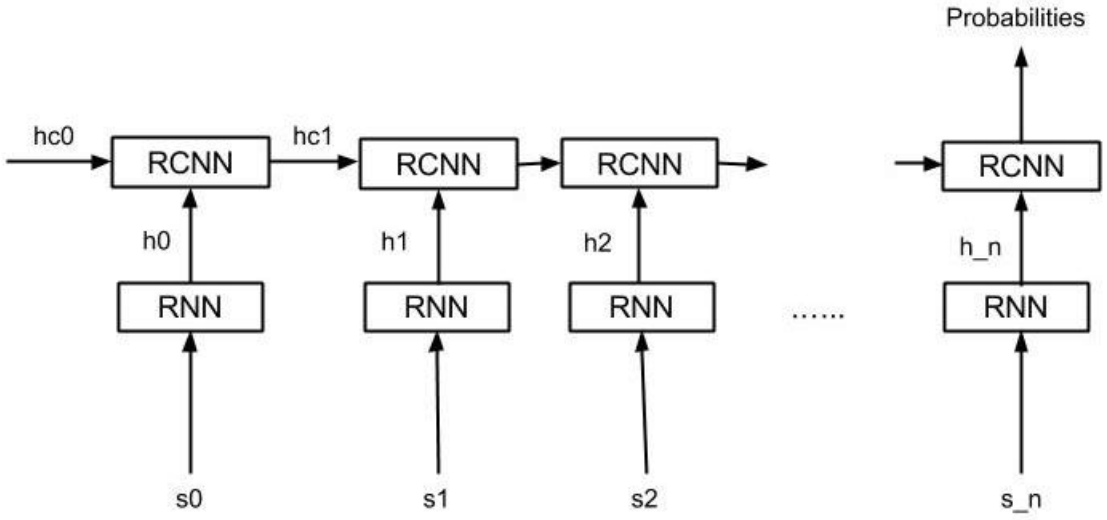

Figure 1: Approach 6: Classification Architecture.

4 Data

We use the IMDB movie review dataset provided by Maas et. al. [1]. We train the word vectors on this corpus using the skip-gram architecture. Note that [1] is specifically about learning word vectors for sentiment analysis. As mentioned earlier, we intend to use standard, off-the-shelf vectors along with a novel architecture. Therefore, we have used word2vec to generate 100 dimensional vectors for each word in our corpus (using skip-gram).

5 Transfer learning with RNNs

We trained a recursive neural network using the dataset provided by Stanford Sentiment Treebank [7]. The dataset contains fine-grained phrase-level sentiment labels (5 classes, ranging from Very Negative to Neutral to Very Positive). We used this trained model on the classification task on the IMDB movie review dataset. Since we do not have fine-grained labels for this dataset, we do not update the RNN parameters during training. Additionally, we used the NLTK tokenizer (`punkt` module) to split the movie reviews into sentences and Stanford Parser to generate its parse tree. While our implementation of the RNN (which uses a ReLU non-linearity as opposed to the one in [2]) yields an accuracy of 80.8% on all nodes (fine-grained), it only yields an accuracy of 45% (fine-grained) and 85.1% (binary) on root node classification.

6 Experimental Results

We experimented extensively with the various approaches and tuned the hyper-parameters to extract the best performance from these models.

For the approaches that use semantic word vectors, we experimented with using both the 300-dimensional GloVe vectors [4] pre-trained on Wikipedia 2014 and Gigaword 5 data and also a self-trained word2vec model. We learnt 100-dimensional vectors using the word2vec model [5], with our moview review training data as the corpus. Apart from these, in the bag-of-words model, we used idf weighting and removed stop words. We also performed L2 normalization. We explored using bigrams but it proved to be too computationally expensive.

For our baseline approach (method 3.4), we tokenize the movie review to extract the words and then average the word vectors to create a fixed dimensional input vector for each of our classifiers

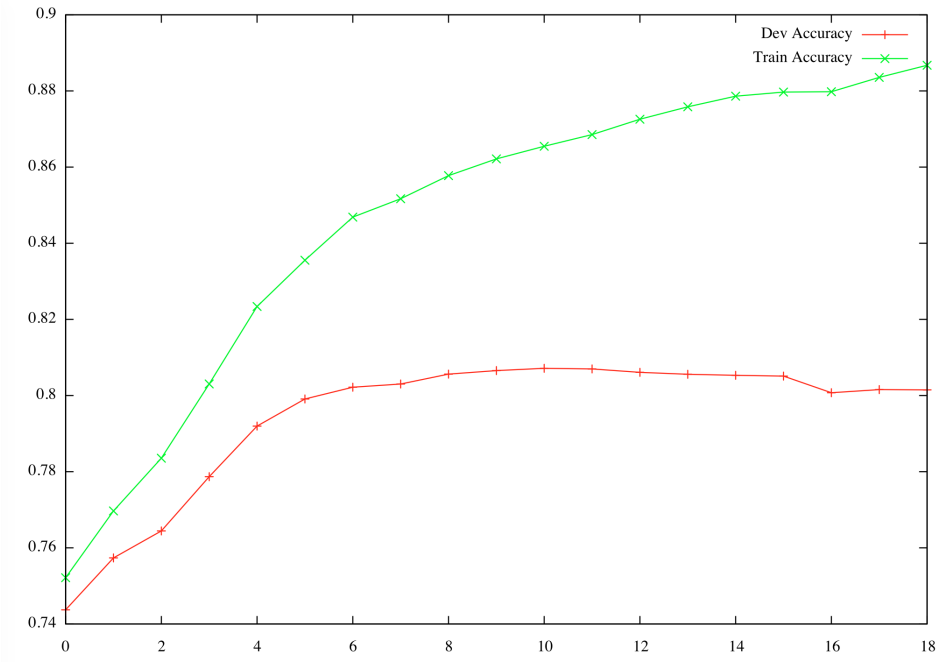

Figure 2: Transfer Learning RNN: Training and Dev Accuracy vs. Number of epochs.

Approach	Test Accuracy
Method 3.1, Mean Word - RecNN	82.35%
Method 3.2, Mean Prob - RNN	81.8%
Method 3.3, RNN-Affine	81.42%
Method 3.4, SVM-RBF	76.69%
Method 3.4, SVM-Linear	83.28%
Method 3.5, SVM-Linear	**86.496%**
Method 3.5, 2-layer NN	83.94 %
Method 3.6, RecNN-RNN	83.88%

Table 1: Summary of results

(SVM/NN). NN model performs the better of the two. The NN uses a sigmoid activation for the hidden layer and cross-entropy loss as the objective. Figure 9 plots the loss and accuracy of the NN.

In the Recursive-Recurrent approach, we experimented with reversing the order of the sentences input to the Recurrent architecture but that did not have any significant impact on the test accuracy (83.76 % vs 83.88 %). The learning curves for both these are plotted in Fig 3 and Fig 4 respectively. We separated 5000 reviews from the train set and constituted our dev set, for tuning the hyperparameters. We finally tested on the test set of 25000 reviews. For the normal order (Fig. 3), we used a learning rate of 0.001 and a regularization strength of 0.00001. For the reversed order, we used a learning rate of 0.001 and a regularization strength of 0.00003.

7 Conclusion

Our objective in this project was to apply the advances in deep learning, including more intuitive model architectures to the sentiment classification problem. We performed several experiments with approaches that have traditionally been used for sentiment analysis, like SVM/Affine neural networks. We also extensively experimented with the proposed architecture - Recursive Neural Network for sentence-level analysis and a recurrent neural network on top for passage analysis.

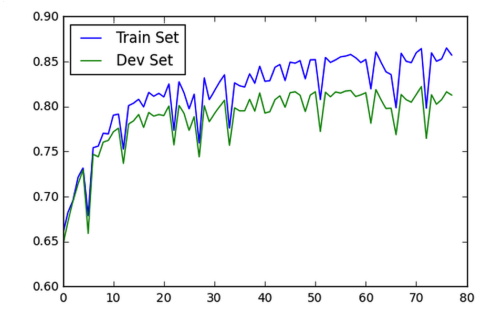

Figure 3: Approach 3.1, Normal order of sentences to the Recurrent Architecture

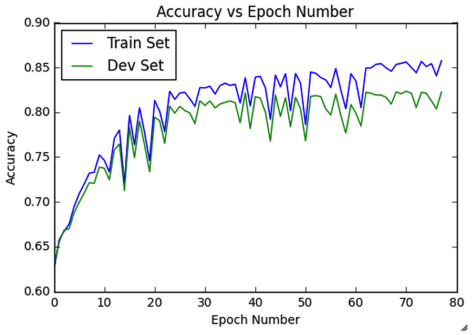

Figure 4: Approach 3.1, Reversed order of sentences to the Recurrent Architecture

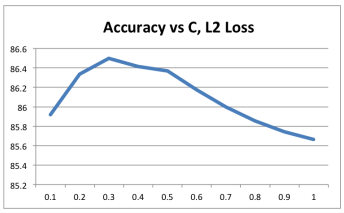

Figure 5: Approach 3.5, Test Accuracy vs Regularization strength

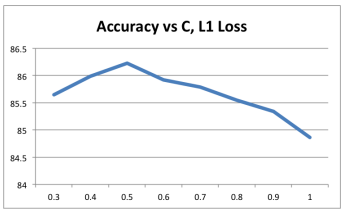

Figure 6: Approach 3.5, Test Accuracy vs Regularization strength

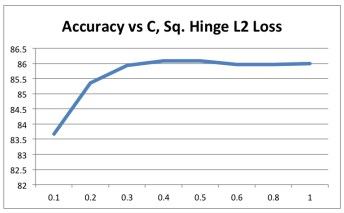

Figure 7: Approach 3.5, Test Accuracy vs Regularization strength

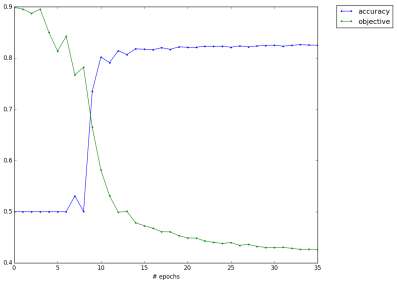

Figure 8: Approach 3.3, Loss and Test Accuracy vs Epoch Number for the 2-layer NN on top of sentence-level RNN

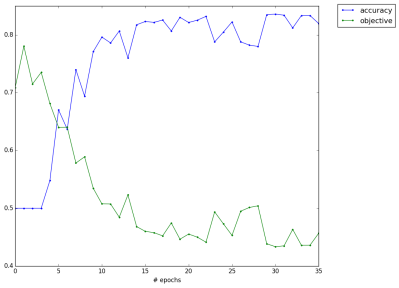

Figure 9: Approach 3.5, Loss and Test Accuracy vs Epoch Number for the 2-layer NN

We considered this to be a more intuitive model since it maps naturally to how humans interpret passages.

We observed that the information in the semantic word vectors is complementary to that contained in the tf-idf bag-of-words representation, since addition of these features to the averaged semantic vectors yields an increment of 3.2% in the test accuracy, as seen in Table 1.

We also observed that a simple model with handcrafted features (like Bag of words with TF-IDF, bigrams etc.) performs better than other methods. However, our proposed architecture (RNN+RecNN) is able to achieve a test-set accuracy of **83.88%** without any handcrafted features at all. We think its an important observation that the model can perform almost as well as featured-engineered models, even with a tranfer-learned RNN and no handcrafted features. This serves as an optimistic baseline for what the model can achieve - With more architectural exploration and fine-tuning, the model can possibly outperform the state-of-the-art.

References

[1] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011, June). Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 142-150). Association for Computational Linguistics.

[2] Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the conference on empirical methods in natural language processing (EMNLP) (Vol. 1631, p. 1642).

[3] Pang, B., Lee, L., and Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10 (pp. 79-86). Association for Computational Linguistics.

[4] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014) 12 (2014).

[5] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In Advances in Neural Information Processing Systems, pp. 3111-3119. 2013.

[6] Collobert, Ronan, Jason Weston, Lon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch." The Journal of Machine Learning Research 12 (2011): 2493-2537.

[7] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning and Andrew Ng. "Parsing with Compositional Vector Grammars" (EMNLP 2013)