Deep Ordered and Unordered Syntactic Composition for Sentiment Analysis

Anish Acharya

1 Introduction

With the success of Deep Learning NLP research has moved from shallow-models on sparse-feature-space to deep-models on dense word-embedding-space. In this work we explore two popular deep learning approaches on a sentiment classification task. We evaluate our models on Rotten Tomatoes movie reviews dataset, which assigns a binary label to each movie review.

2 Experiments and Results

2.1 Deep Unordered Composition

This class of models give up the sequence information in a sense that it tries to learn a sigle representation of the input without encoding any information about the order of words. Deep Averaging Networks (DAN) are the most common variant of unordered composition, where the sentence representation is an average (composition function) of individual word representations, thus dropping any order information. This averaging is followed by a few layers of linear or nonlinear transformations. However, these networks worked surprisingly well on most sentence classification tasks almost at per with sequential models we tried. We tried two particular variants of DAN. **DAN-RAND** Here the word embeddings were xaviar initialized. In the other variant **DAN** we used pretrained glove embedding to initialize the embedding layer. DAN seemed to perform slightly better as compared to the RAND version. We also tried word dropout 3.1 which improve the performance in both variants. [1]

2.2 Deep Ordered Composition

This is by far the most popular variant of deep models used in text classification, where the sentence representation is computed through a series of states in a sequential fashion. We particularly tried LSTM which has special gated architecture in addition to recurrence logic to handle long term context. We tried different settings of LSTM where the best performace was seen for the bi-directional variant of the architecture where the context is computed in both direction. Similar to the DAN setting we tried xaviar embedding initialization (**BiLSTM-RAND**) and glove initialized version(**BiLSTM**). Glove initialized BiLSTM seems to perform slightly better than the RAND variant.[1]

Model	Accuracy	Embedding
DAN-RAND	74.10	Trained
DAN	75.80	Frozen
	76.26	Trained
BiLSTM-RAND	76.82	Trained
BiLSTM	76.55	Frozen
	77.49	Trained

Table 1: Performance on RT Sentiment Classification Dataset. RAND: randomly initialized embedding layer (Xavier initialization), otherwise pre-trained embedding vector(glove) was used to initialize embed weights. Frozen refers to keeping the embedding layer fixed throughout training. Trained implied the embedding parameters were finetuned on the task

3 Additional Experimental Setup

3.1 Word Dropout

For DAN settings, instead of taking an average of all words in the sentence we randomly drop some word from the average composition for sufficiently longer sentences. This seems to improve the performance possibly by acting as an additional regularizer along with network dropout.

3.2 Embedding Inititalization

When creating the embedding weights its common to set embedding of **UNK token** or **PAD token** as a zero vector. In our experiements instead of setting them to zero we tried setting them to a random sample from normal distribution which improved the performance and robustness slightly.

3.3 Learning Rate Schedule

We tried different heuristics to schedule learning rate. The best performance comes with clubbing Adam with a cyclically annealed learning schedule.