Target based sentiment analysis based on syntactic parsing

Kelvin Ting

kelvin.ting@student.adelaide.edu.au

Abstract

Aspect-based sentiment classification is a task that involve modelling the semantic relatedness of an aspect term with its context words in a sentence. Due to complex grammatic structures and ambiguous in meaning, different context words have different influences on determining the sentiment polarity of a sentence towards the aspect term. To examine the sentiment polarity, syntactic parsing is useful, as it breaks down the structure of the sentence into part of speech, tagging and dependency between context word and aspect term.

1 Introduction

Sentiment analysis is a fundamental task in natural language processing. Due to the popularity of social media, sentiment analysis is important to understand user generated comments in social network. By understanding these comments, conducting a review on certain products and services maybe useful for business.

"I loved their **fajitas**"	→ {fajitas: *positive*}
"I hated their **fajitas**, but their **salads** were great"	→ {fajitas: *negative*, salads: *positive*}
"The **fajitas** are their first plate"	→ {fajitas: *neutral*}

2 Approach

2.1 Long Short-term Memory (LSTM)

In this section, I would like to discuss a long short-term memory (Hochreiter and Schmidhuber, 1997) approach for aspect term sentiment classification. This LSTM approach models the relationship of an aspect term with its context words and weight will be adjusted for each context word to infer the sentiment polarity towards the aspect term. The model could be trained in an end-to-end way with standard backpropagation, where the loss function is cross-entropy error of supervised sentiment classification.

Vectors are used to represent each word, this is known as word embedding. To maximize performance, pre-trained word embedding trained by twitter corpus, consists of 27 billion vocabularies and each word presented by 100 dimensions (Pennington et al., 2014) is chosen to examine the semantic and grammatical association of words.

LSTM is a kind of recurrent neural network (RNN), which is able to transfer the previous current word vector (w) from previous information (h-1). However, traditional RNN suffer gradient vanishing or exploding, where previous information is not well transformed to next layer. Hence, additional neural gates: an input gate, a forget gate and an output gate are introduced to compose LSTM (Hochreiter and Schmidhuber, 1997). Each gate is adjusted by weight to adaptively remember input vector, previous information and generate output (Tang et al., 2016).

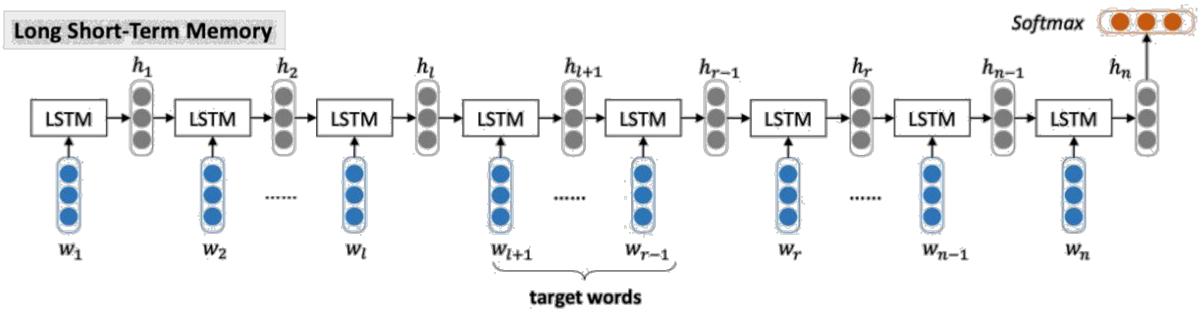


Figure 1: The basic long short-term memory (LSTM)

The last layer of the LSTM model regards as the sentence representation. In order to calculate the 3 polarities: negative, neutral and positive. SoftMax is used to assign probability of each classification. The category with the highest probability will be assigned with the corresponding polarity.

2.2 Target-Dependent-Connection LSTM (TD-TC-LSTM)

The previous LSTM model examine the sentiment classification in a target-independent, which means target words (aspect term) is irrelevant regards to the features that determine the sentiment (Tang et al., 2016). For example,

"I hated their **fajitas**, but their **salads** →{fajitas: *negative*, salads: were great" *positive*}

By the basic LSTM model, each word will pass to the model disregard position of the aspect term. This means both **fajitas** and **salads** will be assigned with the same polarity as the sequence of the inputs is the same. In fact, not all the sentences commit the same polarity as we can see in the example, **fajitas** have negative sentiment while **salads** towards to positive sentiment.

To take into account of the target information, the previous LSTM is adjusted to counter this problem. Instead of modelling from left to right, I model the preceding and following contexts surrounding the target words, so that this will be a better capturing in meaning of the aspect term. For example,

"I hated their **fajitas**" →SoftMax ← "their **fajitas** but their **salads** were great" → {fajitas: *negative*}

"I hated their **fajitas** but their **salads** were great" →SoftMax ← "were great" → {fajitas: *positive*}

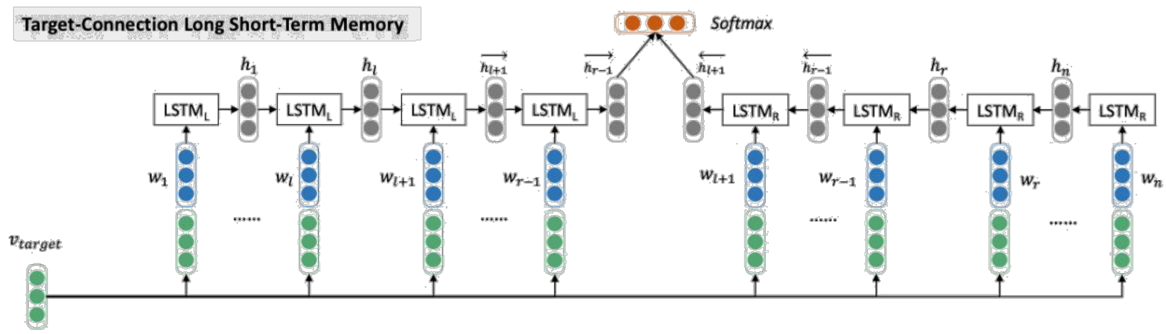


Figure 2: The target-connection long short-term memory (TC-LSTM)

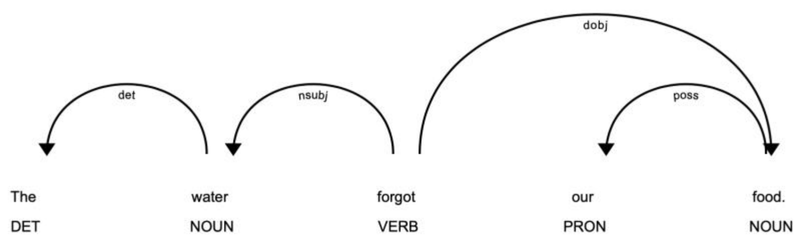
To have a better emphasis on connection between aspect term and context words, Tang et al. (2016) recommend that stacking the aspect term under the context word will lead to improved performance.

2.3 Dependency Parsing Target-Connection LSTM (DP -TC-LSTM)

To further improve the accuracy of the model, dependency parsing provides important information about the structure of the sentences such as part of speech (POS), part of speech tag (TAG) and dependencies (DEP).

In traditional grammar, a part of speech (POS) is a category of words that have similar grammatical properties. Words that are assigned to the same part of speech generally display similar syntactic behaviour—they play similar roles within the grammatical structure of sentences. Similarly, a POS tag (TAG) is a special label assigned to each token (word) in a text corpus to indicate the part of speech and often also other grammatical categories such as tense, number (plural/singular).

Moreover, dependency is the notion that linguistic units, e.g., words, are connected to each other by directed links. The (finite) verb is taken to be the structural centre of clause structure. All other syntactic units (words) are either directly or indirectly connected to the verb in terms of the directed links, which are called dependencies. For example,



In this model, the pos, tag and dependency of each context word are extracted. Then I encode these features as one hot vectors. To get a good features representation and reduce the sparseness, the one hot vector is passed to an embedding layer. Finally, the embedding composite of pos, tagging and dependency will concatenate with the aspect term and context word into a single matrix.

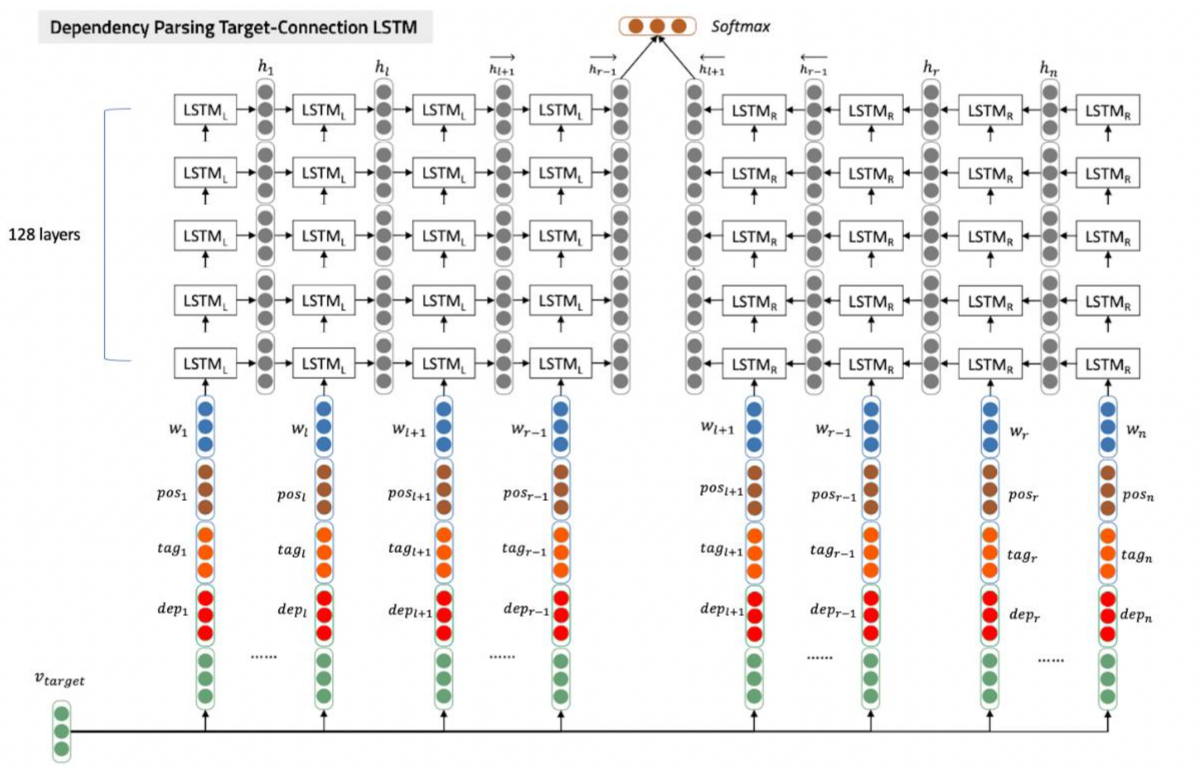


Figure 3: The target-connection long short-term memory (DP-TC-LSTM) model for target-dependent sentiment classification.

The input consists of context word's global vector, part of speech, tag, dependency and aspect term's global vector. Then it will feed to the LSTM with 128 hidden layers. The output of the last layer will feed to a SoftMax for sentimental classification.

3 Result

3.1 Experimental setting

The data set is from a restaurant review. The data consist of 3602 aspect terms and its polarity, and the testing set consist of 1102 terms. The evaluation is based the number of correct labels in during testing.

From all the LSTM based models introduced in this report, the basic LSTM performs the worst due to this is an aspect term sentiment analysis. It is necessary to understand the between the relationship of surrounding context word and aspect term, while the basic LSTM does not examine any aspect term related information.

Method	Accuracy	Epoch
LSTM	75.63%	6
TC_LSTM	76.25%	**2**
TC_LSTM + POS	76.16%	4
TC_LSTM + TAG	77.14%	4
TC_LSTM + DEP	78.12%	4
TC_LSTM + POS & TAG	78.21%	**2**
TC_LSTM + POS & DEP	78.04%	3
TC_LSTM + TAG & DEP	**79.11%**	5
TC_LSTM + POS & TAG & DEP	78.48%	5

Table 2: Comparison of LSTM method on sentiment classification.

Evaluation metrics are accuracy and epoch. Best scores are in bold.

Among all the LSTM models, target-connection LSTM with part of speech tagging combined with dependency perform the best. This is because part-of-speech tags describe the characteristic structure of lexical terms within a sentence or text, therefore, I can use them for making assumptions about semantics and dependency explore the relationship between context word and aspect term.

3.4 Effects of Word Embedding

It is expected that higher dimension vector representation of word would lead to better performance. In general, an increasing accuracy trend is found when the model used single depending parsing embedding such basic LSTM + POS. The accuracy does not vary much with 2 or more dependency parsing embedding combined.

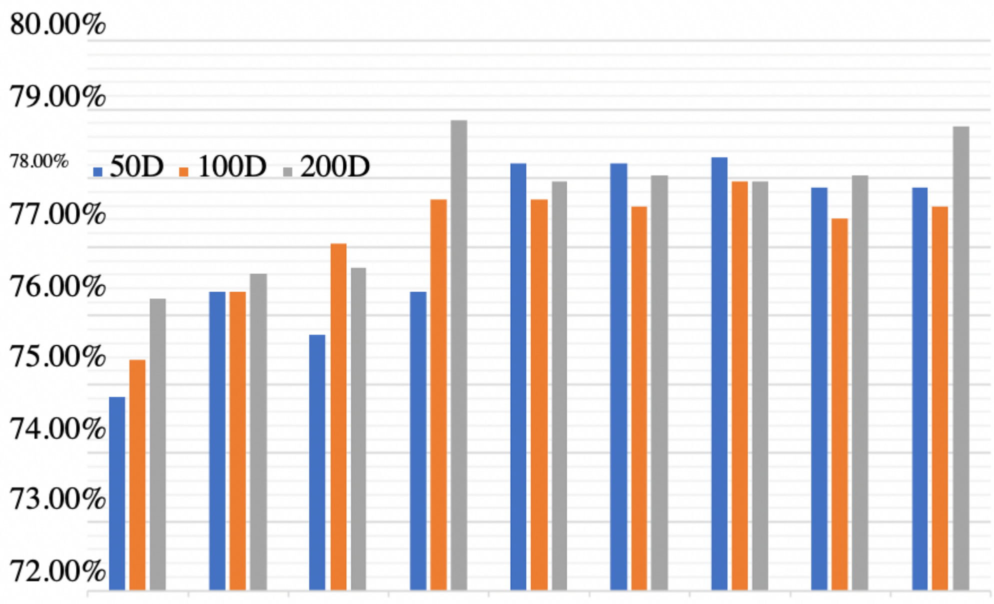


Figure 3: Classification accuracy of LSTM, TD-LSTM and TC-LSTM with different word embeddings.

However, a higher dimension does not significantly improve the models but the size of file of 200 dimension (~2GB) might slow down the training process. The optimal size of this model is 100 dimension.

3.5 Effects of Post Padding vs Prep Padding

Since the length of sentence is different, in order to use batching for training. It is recommended to pad the sequence. There is significant improvement if prep padding is chosen, roughly 2-3% on every model.

Though post padding model peaked its efficiency at 6 epochs and started to overfit after that, its accuracy is way less than pre-padding. Artificial Neural Networks are inspired from biological neural networks. This can be explained by the gradient vanishing. The maximum length of input is 69, and average sentence length 30 words, hence the previous information is not able to pass though such long distance.

4. Conclusion

A sentimental analysis based on aspect term is developed. Different models were examined. Among all the LSTM model, I found that Target Connection with part of speech tagging embedding perform the best, achieved 79.19% in accuracy.

Reference

1. Adam: A Method for Stochastic Optimization by Diederik P. Kingma, Jimmy Ba
2. Effective LSTMs for Target-Dependent Sentiment Classification by Duyu Tang, Bing Qin, Xiaocheng Feng, Ting Liu
3. Long short-term memory by Hochreiter