REPRODUCIBILITY REPORT: BEYOND WORD IMPORTANCE: CONTEXTUAL DECOMPOSITION TO EXTRACT INTERACTIONS FROM LSTMS

Haoze Yang & Junwen Wang & Xing Xu
Department of ECS
University of Southampton
{hy2u18, jw7u18, xx1r18}@soton.ac.uk

ABSTRACT

The main purpose of this project is to reproduce the experiments described in the original paper and to assess whether the experiments are repeatable. This experiment basically reproduces most of the experiments in paper and compares the results of the experiment with the results in the original paper.

1 INTRODUCTION

In natural language processing, LSTM has become the basic building block. Although LSTM performs well, its internal working principle is still difficult to understand. The original paper introduces contextual decomposition (CD), which provides a more detailed explanation of LSTM using an evaluation of sentiment analysis cases. The original paper also provides Github repository but only with a demo of the CD algorithm. The team have extended the demo and reproduced some of the key experiments from original paper.

2 CONTEXTUAL DECOMPOSITION OF LSTMS

In section 3.1 of original paper have given the structure of LSTM. Hence, this section will give some brief explanation of Contextual decomposition algorithm which inherits the basic model from LSTM.

2.1 CONTEXTUAL DECOMPOSITION

The context decomposition algorithm decomposes the cell state and the hidden state of the last time step of the LSTM into two parts. Given the arbitrary phrase, the output of Equations 5 and 6 is divided into the sum of the two contributions, as shown below.

$$h_t = \beta_t + \gamma_t,\ c_t = \beta_t^c + \gamma_t^c \tag{1}$$

The decomposition is constructed such that β_t corresponds to the contribution made to h_t by only a given phrase, and γ_t corresponds to the contribution of the remaining parts and elements other than the phrase.

2.2 DISAMBIGUATING INTERACTIONS BETWEEN GATES

To decompose the cell state and hidden state, two linearizing activation functions derived in the paper are used, which are $'sigmoid'$ and $'tanh'$ functions. These two functions can map products between gates to products over linear sums of contribution terms. In addition, the derived functions of these functions in the paper can calculate β, β^c, γ, and γ^c by recursive methods.

3 EXPERIMENT AND RESULTS

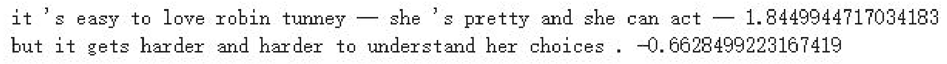
In this section, the validation of CD on the sentiment analysis task will be described. The purpose of this experiment in paper is that CD algorithm can get scores of word and phrase level importance, indicating that the CD can capture the composition of phrases of different emotions. Finally, calculating CD scores can determine instances of positive and negative negations.

3.1 Train Model

The first step is to train an LSTM model. Since the main purpose of paper is not to achieve high predict accuracy, there is not much adjustment to the best practices of parameters. This model uses Adam as the optimized function, and uses 'CrossEntropy' as the loss function. In the linear model, pre-trained Glove vectors were used for a bag of vectors model, and the same parameters as the original paper were used. Compared with the original paper, the accuracy of LSTM training is 6.2% lower. One possible reason is that the parameters of the paper may be adjusted in the experiment.

3.2 CD score of Composition

First, processing and understanding the SST data set is a challenge in this project. The tree-structured version is used as the structure of the data to find the vocabulary in the sentence and to segment the sentence or extract the phrase based on the specific vocabulary. In order to apply it to CD, the data set was loaded with a pre-built in function from the Python library torchtext, and then use the for loop to extract the phrase or words in the tree structure of the SST dataset and calculate the CD score. Following results show that CD can identify dissenting sub-phrases: Figure 1 shows a reproduction

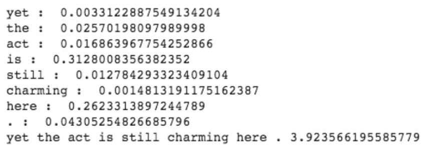


```
it 's easy to love robin tunney — she 's pretty and she can act — 1.8449944717034183
but it gets harder and harder to understand her choices . -0.6628499223167419
```

Figure 1: CD score in Phrase-Level, reproducing table II in original paper

of table 2 in original paper. It can be concluded from Figure 1 that the first half of the commentary is positive and the second half is the main part of the negative sentiment. Therefore, CD can very well distinguish the sentiment of different phrases in a review.

In addition to phrase lever of CD, Generating a score for each word in a review as shown in figure 2 can indicate the contribution of each word to the individual phrases.

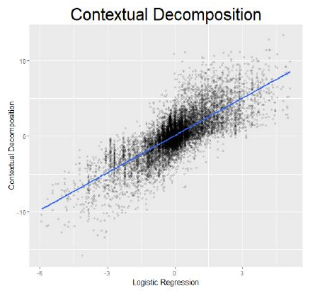


```
yet :  0.0033122887549134204
the :  0.02570198097989998
act :  0.016863967754252866
is :  0.3128008356382352
still :  0.012784293323409104
charming :  0.0014813191175162387
here :  0.2623313897244789
. :  0.04305254826685796
yet the act is still charming here . 3.923566195585779
```

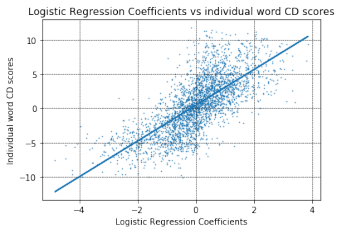
Figure 2: CD score in Word-Level

3.3 Unigram Score

The author has demonstrate three different experimental examination to compare CD approach with other related works and show some significant performances of CD. In addition, they also demonstrate CD can produce a sufficiently accurate *unigram coefficients*. Original paper states that *logistic regression coefficients* are being treated as a gold standard in many prediction task. In particular, in semantic analysis task the coefficient value can provide a qualitatively sensible measure of importance. Thus, by comparing the CD score with logistic regression coefficient we should expect there have some correlation between them. In the original paper it shows a stronger linear relationships between the logistic regression coefficients and CD score extracted from an LSTM on SST.



(a) original figure



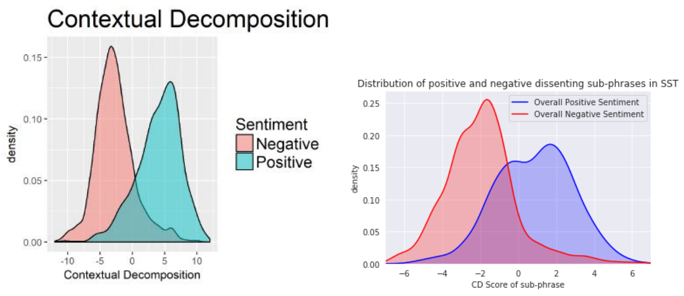
(b) experimental figure

Figure 3: Compare between original and experimental logistic figure

The paper does not clearly state how they constructed words table and how they extract coefficient from the logistic regression coefficients. We have reproduced the result in following way: first we constructed a words dictionary: in which we use CD to capture the scores of all words occur in the data set. We only measure the CD score where the word first time occurs in the data set. Then we train a Logistic Regression classifier using the training data by constructing the words look up table using *Bag of Words*. Those words were generated from the training data set, in this case we used a sklearn library, *CountVectorizer* to build a matrix of counter which counting the number of times each vocabulary occur in the document. By training the data using CountVectorizer we were able to obtain the LR coefficient of each word by calling the built in method *coef*.
Then we are finding the word intersection between our calculation of CD for each word in the data set and our LR coefficient from the look up table. Figure 3 shows a comparison. Noticing that the we also perform some scaling for each word scores for a better visual effect.

3.4 CD Scores and Dissenting Sub-Phrases

After obtaining each Dissenting Sub-Phrases and the corresponding CD scores by the above method, these scores are divided into two sets of positive and negative data sets and produce the graphs shown below. The x-axis represents the score of the sentence, the ordinate represents the frequency



(a) original figure (b) experimental figure

Figure 4: Distribution of positive and negative sentiment of SST dataset

of occurrence, the red area represents the sentence marked as a positive emotion, and the blue line represents the phrase of negative emotion. From this, it can be concluded that in this case, negative and positive sentiments are significantly different. In experiments on other models in the original paper, the separation of these two lines is much smaller, and the degree of separation can indicate the stability of the model prediction. Therefore, compared to other models, although the accuracy is relatively weak sometimes, the performance of the CD model will be more stable.

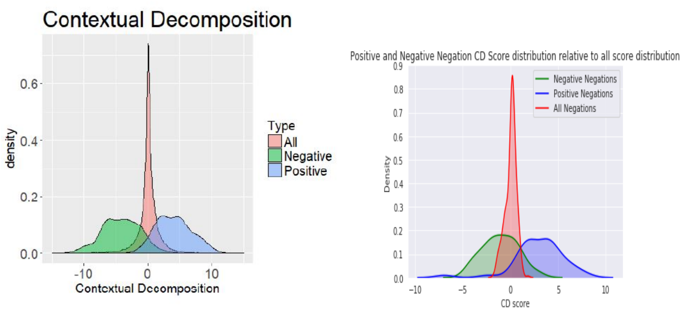
We also measured the two-sample Kolmogorov-Smirnov test statistic. However, since the *scipy* package did not support one-sided test so far, here we are implementing the KS two-sided test statistic. Our team used the function *ks_2samp* in *scipy* library. In this experiment CD produces a statistic score of 0.59 with a significantly small *p-value*. Compare to the paper section 4.4, which they obtain score 0.74, our result shows slightly smaller than theirs.

3.5 Contextual Decomposition (CD) Captures Negation

For phrases involving negation, it is now necessary to use a CD to prove that LSTM has learned a negative mechanism. First, in the entire SST data set, a phrase tag is used to search for negative phrase instances of length less than 10. In particular, a negative word phrase is included, such as not, nt, lacks, nobody, nor, nothing, neither, never, none, nowhere and remotely. Then, the negative interaction can be extracted by calculating the CD score of the entire phrase and subtracting the CD score of the negative phrase and the negative word itself, which can be expressed as

$$Negation\ Interaction = Review\ Score - Negated\ Phrase\ Score - Negation\ Term\ Score$$

Finally, the score obtained can be interpreted as an n-gram feature. The distribution of the extracted scores is shown in Figure 1. For CD, we can see that there is a clear distinction between positive and negative negation and that the negative interaction is concentrated at the outer edge of the interaction distribution. Leaving one can capture some interactions, but there is a significant overlap between positive and negative negations near zero, indicating a high false negative rate.



(a) original figure (b) experimental figure

Figure 5: Distribution of scores for positive and negative negation coefficients

4 Discussion

1. Data processing and search: Reproducing the datasets of the distributions in Figures 5 and 6 requires reading the source code of the torchtext library to see how the data is stored in the NLTK tree. The team use the NLTK Tree API to write keywords or locations in the search comments, and divide the comments into different phrases for scoring. However, the original paper does not introduce relevant knowledge.
2. The original paper did not prove that the formula for calculating the negative interactions was correct, nor did it explain why only phrases with less than 10 words were included in the final distribution. The Figure 6 is a classification test for the data set which is longer. As sentences get longer, the separation of CDs about negation interactions gets worse. The

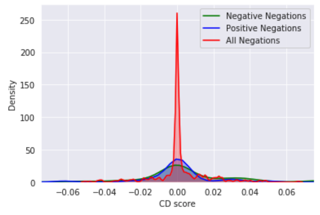


Figure 6: Distribution of scores for positive and negative negation coefficients

results show that for longer sentences, it is difficult for CDs to capture negation.

3. The paper also prove their method on Yelp polarity dataset, it can be considered to implement this for future developments.
4. Group have found that the performance of CD can be varied depends on the performance of LSTM model. There is a need for more experiments on this part.

5 Conclusion

This project implemented the CD algorithm to reproduce the key experiments in the original paper. First, the team reproduces the uni-gram scores between the lexical-level logistic regression coefficients and CD scores and obtains very close results to the original paper. In addition, the team plot the distribution of positive and negative sub-phrases to show that the CD score captured the difference between the two. However, the distribution obtained has large variance from each experiments. Similarly, the team also obtained positive and negative negation distribution, but not as obviously separate as the results in the original paper. In summary, the project reproduces the original paper and obtains reasonable results, which proves that the CD can well recognize phrases of different sentiments and can extract meaningful word interactions.

References

[1] W. James Murdoch, Peter J. Liu, Bin Yu, 'BEYOND WORD IMPORTANCE: CONTEXTUAL DE- COMPOSITION TO EXTRACT INTERACTIONS FROM LSTMS', ICLR 2018