**Reproducing Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs**

Track: 3, Paper: T3-03 (<https://arxiv.org/abs/1801.05453>)

Team members:

Olivier Simard-Morissette. ID: 260563480,

Email: olivier.morissette@gmail.com

Ruo Yu Tao. ID: 260706629,

Email: ruo.tao@mail.mcgill.ca

Irene Qizhen Zhang. ID: 260671588,

Email: qizhen.zhang@mail.mcgill.ca

### Introduction

This reproducibility study aims to reproduce the findings of the paper *Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs.* The original paper outlines a method called Contextual Decomposition (CD) of LSTMs, which is an algorithm for producing interpretable results from the outputs of an LSTM, without changing the functionality of the underlying model.

The algorithm uses linearizing activation functions, so that contributions in determining the output of the LSTM from the phrase itself and the contributions from everything around the phrase may be separated. The paper claims that by mathematically separating the LSTM outputs, the contributions made at every step by different parts of a review may be easier to interpret.

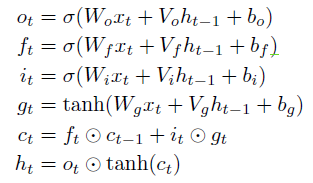
The focus of the reproducibility study was on re-producing the CD scores of the Stanford Sentiment Treebank (SST) dataset by training an LSTM network in sentiment analysis. CD scores were computed and visualized on various tasks in sentiment classification. The results from the study reproduce how CD is able to identify phrases and words of differing (and often negating) sentiment - something that the baselines described in the paper were not able to accomplish. The CD method was also validated by comparing CD scores with the “gold-standard of interpretability” - logistic regression coefficients - by reproducing and visualizing the correlation between CD scores and logistic regression coefficients.

The reproducibility study successfully reproduced word-level importance scores using CD and also was able to show that there exists a significant correlation between CD scores and logistic regression coefficients but not as high of a correlation described in the original paper. The reproducibility study was not able to fully reproduce the separation of the distributions between positive/negative negating phrases and positive/negative dissenting phrases.

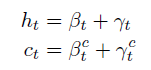
### Implementation Goals

The first goal was to implement the CD algorithm. Given a sample such as a movie review and the location of the phrase or word of interest within the review, the algorithm computes the CD score. The CD score quantitatively measures the degree with which the phrase or word contributes towards the LSTM sentiment prediction for the review.

First, an LSTM must be trained to obtain the learnt weights. Recall that in LSTM. the weights learned are W, V, and b shown below.



After the model is trained, the algorithm decomposes the cell state and hidden state of the last time step of LSTM each into two parts: contributions made by the word or phrase of interest (Beta), and contributions made by the others in the text (Gamma).



To decompose the cell state and hidden state, two linearizing activation functions derived in the paper are used. The two functions allow to map products between gates to products over linear sums of contribution terms. Using these functions, the derived formulas from the paper are used to compute Beta, Beta\_c, Gamma, Gamma\_c recursively, starting from the first time step.

Finally, the score of Beta at the last time step multiplied by W is returned, and this is referred to as the CD score for the rest of this report.

In addition to implementing the CD algorithm, reproducing the paper requires demonstrating that the algorithm is capable of capturing phrase and word contributions as well as being interpretable. This is accomplished by conducting the following experiments.

1. **Show that CD scores are highly interpretable:** For each word in the validation set, the CD scores as well as the learnt coefficient in logistic regression is computed. Logistic regression was chosen by the authors for comparison because its coefficient value provides a good quantity measure of importance in sentiment analysis, hence it is often seen as the gold standard for interpretability. We expect a high pearson correlation score between the CD score and logistic coefficient if the algorithm can be considered interpretable. The pearson correlation obtained by the paper is 0.76 on the SST dataset when both logistic regression and the LSTM are performing the task of determining word importance. Having a high correlation would suggest that there is a strong linear relationship between the two, which can allow us to conclude that CD scores are highly interpretable.
2. **Show that CD scores can recognize sub-phrases with differing sentiments:** In existing methods, adjacent sub-phrases with opposing sentiments are often classified as both having negative or positive sentiments. The paper demonstrated that using CD produces the correct result for classifying sub-phrases. Positive dissenting sub-phrases are phrases contained in a negative review which are labeled as positive. Negative dissenting sub-phrases are phrases contained in a positive review which are labeled as being negative. The goal of this experiment is to successfully differentiate the two types of dissenting sub-phrases in the training set using the CD method. The expected distribution of CD scores for positive and negative dissenting sub phrases should be significantly separate when plotting the two distribution curves. If these results are obtained, then it’s possible to conclude that the CD algorithm is indeed capable of capturing dissenting sub-phrases.
3. **Show that CD scores can capture negations.** Reviews containing negation phrases are found in the training set by checking if the review contains a phrase in a list of negation phrases provided in the paper. The negation interaction is extracted by computing the CD score of the entire phrase, then subtracting the CD score of the negated phrase and the negation term itself. If CD can capture negations, there should be a distinct separation of the distribution curves between the positive and negation negation classes. If these results are obtained, it’s possible to conclude that the CD algorithm is capable of capturing negation interactions, and hence an LSTM learns a negation mechanism.

The team decided to not reproduce the following in the paper:

1. **Interpretation Baselines**

The paper mentions baselines in section 4.1.3 and runs these interpretation baselines for various sentiment classification tasks and word importance tasks. The baselines were not reproduced to focus on reproducing the novel or new findings in the paper.

1. **Identifying Similar Phrases**

The paper mentions in section 4.6 this application of CD scores anecdotally and doesn’t emphasize the result in the abstract, introduction or conclusion. Focus was put on reproducing results that author emphasized as important or significant and due to limited resources, reproducing this part was not completed.

1. **Yelp Dataset Experiments**

Focus was put on SST dataset because of it’s phrase level labeling and it being used throughout the experiment portion of the paper on various sentiment / word importance tasks. Validating the key results of the paper could be done by focusing on SST for all experiments. Also the scatter plot of logistic regression coefficients and CD scores had a much higher correlation using CD than baselines on the SST dataset. Yelp didn’t provide significantly better performance in this case then the presented baselines.

Filtering to get the Yelp dataset used in experiments was not reproducible. In particular the paper states: “For computational reasons, we report interpretation results on a random subset of sentences of length at most 40 words.”. The report doesn’t mention the seed used to obtain the random sentences. The searches done on SST were deterministic and the focus of the team’s efforts.

### Results and discussion

Overall, we managed to achieve most of our reproducibility goals in the time span of this project.

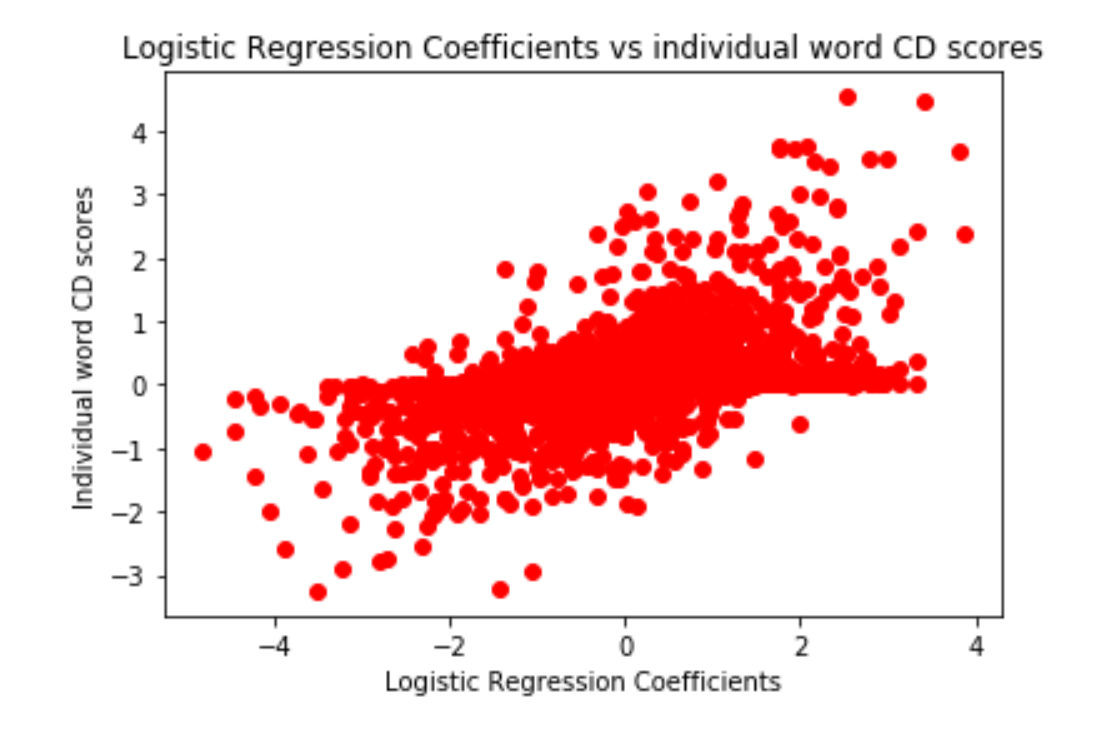
**LSTM Training**

The first major hurdle was actually training an LSTM that achieves the same results as they cite in the paper. The LSTM trained fell short by a margin of 7% in accuracy compared to what was achieved in the original pape. The exact same arguments, configuration and libraries described in the paper were used to train the LSTM model.. One possible explanation could be that the “default” hyper parameters they cite could be different than the ones used in the reproducibility study. Another reason could be that the computational power and the time span of the original group’s project could have been significantly higher, preventing the reproducibility study from achieving as accurate of a model.

**Logistic Regression Coefficients**

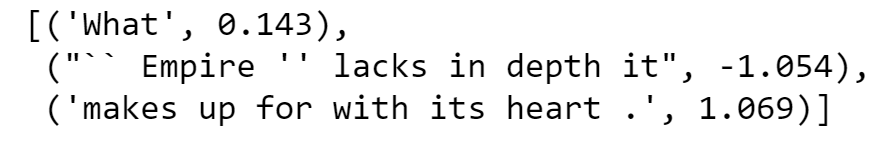
After successfully training an LSTM, the team focused on training a logistic regression model. The methods in which the paper went about this were difficult to understand. A great deal of effort was placed in understanding what the original authors had done. Eventually a consensus was reached that the authors had trained a logistic regression model to classify sentiment using a bag of words input over the binary SST dataset. The coeffiencts from this logistic model corresponded to the word importance of each word in classifying a review as positive or negative.

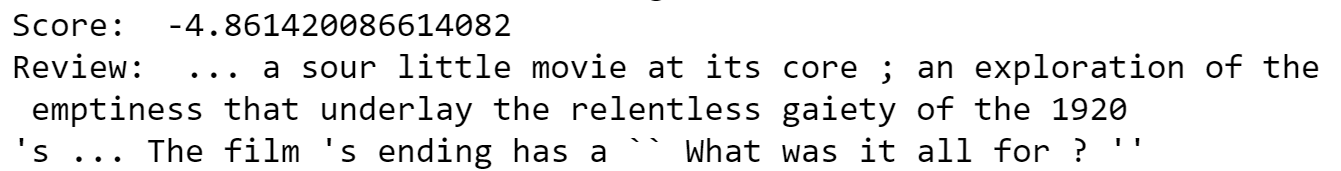
**Dataset parsing**

The biggest challenge involved parsing subsets of the SST dataset to reproduce experiments from the original paper. Olivier used the tree-structured version of the dataset to find dissenting subphrases, whereas Irene and David were working on implementing CD and functions to calculate the CD scores. To implement CD, the dataset was loaded with a pre-built in function from the Python library torchtext, that returned batch iterators for the training and validation set, already loaded into the GPU. Finding the dissenting sub-phrases on the other hand, required the use of the raw tree-structure of the SST dataset - this resulted in a disparity when trying to find CD scores of dissenting sub-phrases. The CD class was instantiated with data from the pre-built SST dataset from Torchtext, and wasn’t compatible with the tree-shaped data from SST. To match reviews from the two datasets, we created a hashmap to map data to the correct index in the CD data.

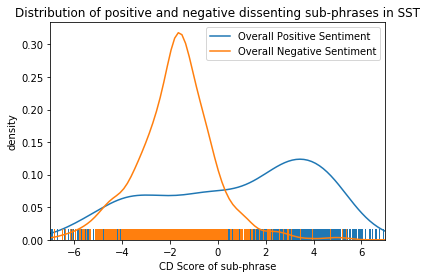
**Raw CD scores**

The following interpretable results were produced for identifying dissenting sub-phrases:



This is an example of calculated CD scores for dissenting sub-phrases. As shown above, the CD method is able to identify the conflicting CD scores of negative and positive dissenting sub-phrases. Similarly, CD can produce similar results at the review-level decomposition:

We also produced word-level CD scores (figure 1). The word level CD scores reproduced the results of 4.3 quite well, except there was not sufficient time to create the visually appealing heat map similar to the one available in Table 1 of the original report. Individual scores are not reflected linearly in the overall CD score, but instead reflect scores/contributions for individual phrases.



**Correlation between Logistic Regression Coefficients and CD Scores**

The correlation plots reproduce the correlation plots as shown in 4.2 (figure 4 of the original paper) - we calculated logistic regression coefficients from the logistic regression model, and also the individual CD score for each word. Doing this brought up an important issue, - CD scores of words are calculated in the context of the review that they appear in - what if they appear multiple times in the validation set that we’re trying to plot? A 1 to 1 mapping between the CD score of the first instance of the word and the logistic regression coefficient achieved the following results:

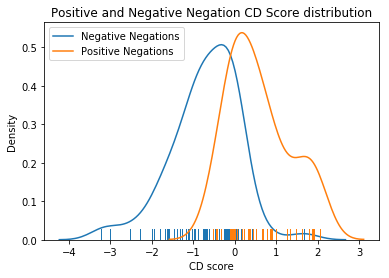
This resulted in a pearson-correlation score of 0.507 - a ~0.2 score decrease from the value described in the original paper. If the mapping between individual CD scores and the logistic regression scores was described in the paper, it may have been possible to dramatically improve the correlation.

**CD Scores and Dissenting Sub-Phrases**

After finding the dissenting sub-phrases, their CD scores and aggregating them in two different datasets (one for positive overall sentiment and one for negative), the result in section 4.4 of the original (figure 3 in the paper) was reproduced with the following results:

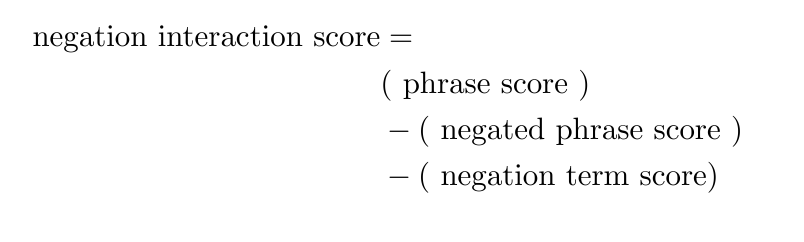
The negative and positive sentiments are clearly split in this case. Previously, with lower accuracies from the LSTM, the two distributions were much less separated. The separation between the distribution is clearly due to the differences in the robustness and performance of our model compared to their model. This leads to the conclusion that the performance of CD is highly correlated to how well the model generalizes and fits the data - a well fit means a clear and highly interpretable decomposition, whereas a sub-optimal model yields scores not as “accurate”.

**CD Scores and Negation Interactions**

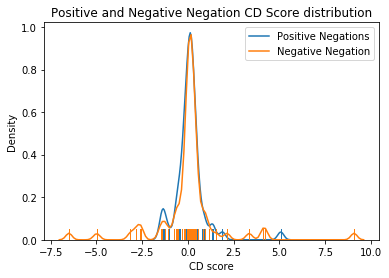
A similar distribution plot for negation sub-phrases was also generated:

Looking at the figure above, there is a decent separation between positive and negative negation. But this is with resulting negation sub-phrases that only have less than 10 words. This results to a distribution with only around 100 results. This puts into question why the paper decided to do this filtering.

It should also be considered why the paper use this particular formula to capture negation interactions:



The paper does not justify the use of this equation. The paper also does not justify why only phrases with less than 5 words are included in the resulting distribution. As sentences become longer the separation between classes in CD becomes worse. Below is a class separation with 10 words:

This result shows that with longer reviews, the CD scores do not allow for separability.

A few examples of negation interactions are included in the appendix with less than 10 words (Figure 2) and more than 10 words (Figure 3). With longer sentences, CD doesn’t capture negation as well.

Overall, the study did not obtain the same level of results as the paper for all three experiments. One fundamental reason might be that the weights learnt from LSTM did not match up to the authors. During the implementation process, the team tried using weights learnt from different iterations of LSTM to calculate the CD scores of the same example, and the resulting scored varied significantly. The LSTM model achieved accuracy of 7% less than the model in the paper using default parameters as they suggested. This suggests that the weights we learnt are different from theirs as well. Hence, we should expect different CD scores and different experiment results, as the calculation entirely depends on these weights. Given more time and computation resources, it could be possible to perform an exhaustive search to tune the LSTM parameters and obtain results similar to the paper but was deemed unfeasible given the time constraints of the project.

### Challenges

1. Replicating the datasets used to reproduce the distributions in Figure 2 and Figure 3 of the paper require writing complicated search queries that are tricky to get right. To reproduce the search queries, the team was required to read the source code of the torchtext library. This helped the team understand how a review is stored in an NLTK tree and then write the search queries using the NLTK tree API. The paper also doesn’t mention if the search was conducted over the training, validation or test set of SST. The only way to determine this was by looking at the examples shown in the tables and searching the datasets for those specific examples. Some solutions could be:
   1. Provide the subset of reviews and sub-phrases parsed for experiments in section 4.4 and 4.5 available online.
   2. Provide pseudocode for the search queries. Algorithm implementation for search query was very complex and wording wasn’t sufficient to be 100% sure that it was correct.
   3. Specify which set of samples the distributions are generated from.
2. Training the examples took a bit of time at first. We were running the training for the number of iterations to achieve our highest validation accuracy, then stopping without considering training accuracy. This turned out to have somewhat of an effect on our later CD predictions, as we use our training set to visualize the distribution of dissenting sub-phrases (4.4) - so some measure of overfitting here actually helped our visualization result. This result is somewhat skewed, as in this case CD might not be able to generalize very well in comparison to a model that generalizes well. Whether they used the training/validation set for section 4.4 was ambiguous in the paper.
3. The wording of the phrases that were filtered using an absolute score of 1.5 in section 4.3 was vague. Language says to filter out phrases with “absolute score is over 1.5”, but not specifying what this absolute score is referencing. We spoke with both Ryan Lowe and Ali Emami about what this meant and both were unsure. As a proxy, we used an upper bound for confidence score for our logistic regression, with decent results.
4. In section 4.1.1, the authors mention the accuracy of their LSTM and Logistic Regression classifiers. They do not mention if this accuracy is the training, validation or test accuracy. This brought ambiguity when trying to reproduce these classifiers and has an impact on CD scores as well as logistic regression coefficients.
5. Authors mention that all models were trained using Torch and default hyperparameters were used. While this level of detail is helpful, the paper does not mention:
   1. how many epochs the classifiers were trained for
   2. The original random seed selected to choose the order in which samples are trained on each epoch.
   3. What version of Torch was used. Default parameters may change between versions.
   4. If the default training/validation/test splits were used for training and validation.
6. It was challenging to implement a Logistic Regression classifier in PyTorch. There were issues with getting the correct dimensions. Decision was made to implement in sklearn for simplicity.
7. For the first experiment of obtaining the word level correlation between logistic regression coefficient and CD scores, the author did not specify the mapping between the two numbers. If a word appears multiple times throughout the dataset, it will have a CD score for each appearance that represents its contribution to review it belongs to. However, logistic regression learnt coefficients are always the same. Hence there may be many CD scores for each word, as opposed to only one logistic coefficient score. To obtain the correlation between the two, we attempted the two following methods.
8. Produce a one to one mapping between CD score and logistic regression coefficient. If word occurs multiple times throughout the text, the score of that word is assigned to be the cd score for the its first appearance.
9. Produce a many to one mapping. For each word’s appearance, produce its CD score. This would create a many to one mapping.

In the end, we decided to go with the one to one mapping, as it produced the higher correlation between the two.

1. Lastly, when implementing the method for calculating the CD score, authors did not provide the exact equations of the linearization activation functions. We had to look up the source code by the authors to find the math equation and see how it was used .

Conclusion

For this project, the CD algorithm was implemented to interpret individual predictions made by LSTM without modifying the underlying model. In addition, three key experiments from the paper were reproduced to demonstrate the capability of CD. In the first experiment, team reproduced the pearson correlation between word level logistic regression coefficient and CD scores described in the paper. Although the correlation score obtained is 0.2 lower than the results in the paper, the method does achieve results that is comparable to the existing baselines for interpretability described in the paper. In addition, the group plotted the distribution curve for positive and negative dissenting sub-phrases, in order to show that CD scores capture the difference between the two. The two distributions that were obtained were not as separating as the results in the paper. In the last experiment the team plotted the distribution curve for positive and negative negations. Again, the distribution showed some separation, but was not as obvious as the results in the paper. Overall, the reproducibility study achieved reasonable results showing that CD gives an interpretable view of an LSTM and can capture dissenting sub phrases and negations.

### References

https://github.com/jamie-murdoch/ContextualDecomposition

### Contributions

Olivier

* Preprocessing to obtain a bag of words representation.
* Implemented binary bag of words logistic regression model in sklearn.
* Implemented visualization of logistic regression coefficients and CD scores.
* Implemented search queries for finding reviews with negating sub-phrases and reviews with dissenting sub-phrases.
* Obtained CD scores for dissenting subphrase and negating subphrase reviews and did visualization of positive/negative distributions.
* Data analysis for KS2 test on positive/negative distributions for dissenting sub-phrases.
* Helped write the report by outlining challenges and justifying parts of the report that were not reproduced.

Irene

* Preparing dataset as batches to be fed into CD score method.
* Understanding math behind CD scores, and implementing the main CD Score method and linearization activation function methods
* Troubleshooting CD score accuracy, trying out different ways (1:1 mapping, many:1 mapping, and getting CD score of word relative to word itself) to plot CD score against logistic regression coefficients made by Olivier
* Help write the report

Ruo Yu

* Loading SST dataset and generating iterators for training/validation
* Implemented main LSTM architecture
* Implemented data helpers and main LSTM model training function
* Combined CD methods to main CD class
* Implemented CD querying functions in the CD class, including by-word querying and subphrase querying
* Created visualization function in CD class for all CD scores with the help of Olivier’s negating sub-phrases and reviews with dissenting sub-phrases
* Organized main github repository for project
* Refactored all code from collaboratory notebook to separate files and demo notebook
* Helped write the introduction to the report, outline challenges in the project and write up results/discussion of the project.

We hereby state that all the work presented in this report is that of the authors.

### 

### Appendix

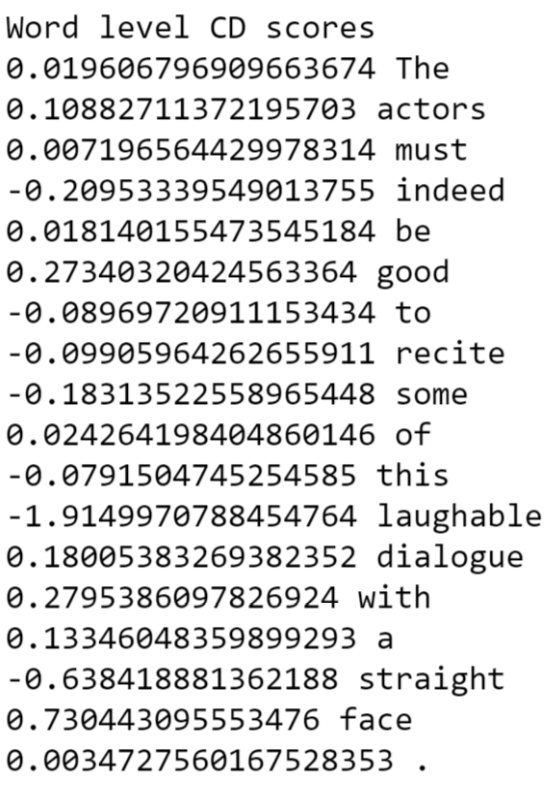


Figure 1: Word level CD scores. CD does a very good job in interpreting individual phrases.

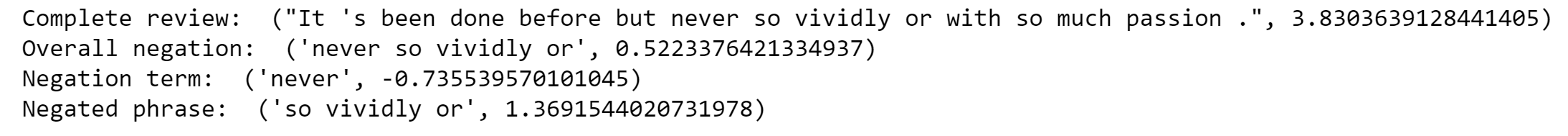


Figure 2: Negating sub-phrases of length less than 10. CD captures negation very well in comparison to it’s phrase score.

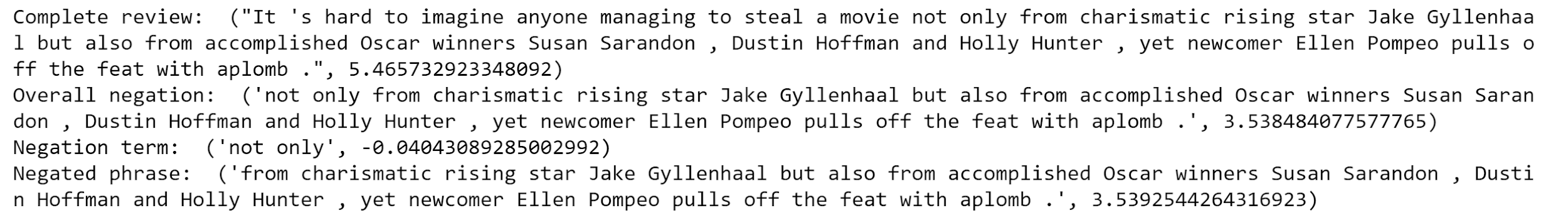


Figure 3: Negating sub-phrases of length more than 10. While the negation term is still identified correctly by it’s CD score, the overall negation score does not reflect the negation interaction.