NLP 1 - Report Practical 2

Ankit
13608568
ankit.ankit@student.uva.nl

Bart van Vulpen
11865210
bart.vanvulpen@student.uva.nl

1 Introduction

The ability to assess the overall sentiment portrayed by the information is a key skill in text comprehension. The objective of sentiment classification is to predict the sentiment of a text; that is, the positive or negative opinion expressed towards a subject. Due to linguistic ambiguity this becomes a difficult problem to investigate.
Most models used in sentiment classification fall into two categories: word order independent and word order dependent. Bag of Words (BOW), Continuous BOW (Mikolov et al., 2013a) and Deep CBOW models fall in the order independent category, while LSTM (Hochreiter and Schmidhuber, 1997) and Tree-LSTM (Tai et al., 2015) (Le and Zuidema, 2015) (Zhu et al., 2015) fall under the category of word order dependent category. The BOW models use sentences as unordered sets, LSTM use their sequential structure and Tree-LSTM employ their complete tree structure.
In the current work we aim to investigate the following questions on various models for sentiment classification:

- How does BOW models perform in comparison to the more sophisticated models like LSTM and Tree-LSTM?
- Does a Tree-LSTM improves performance as compared to LSTM?
- How important is word order for the task of sentiment classification?
- How does the performance change with sentence length?

Investigating the importance of word order is important in the task of predicting the sentiment of a sentence since the sentiment of a sentence usually depends on the semantic meaning (or a combination of semantic meanings) of the words in a sentence and not their order. It is therefore expected that word order does not have that much influence on a models' performance.
To test our hypothesis in particular we implemented the BOW model and its variants: Continuous BOW (CBOW), Deep CBOW, and LSTM and its variants: (mini-batch) LSTM and Tree-LSTM. The models were trained on a dataset that provides sentences, their binary tree structure, and fine-grained sentiment scores for movie reviews. The sentences were classified in one of the five sentiment classes. We further investigated whether training the Tree-LSTM on a training dataset in which sentiment is supervised for each node in the tree. Finally, we compared the N-ary Tree to Child-sum Tree-LSTM and find that the performance was identical.
These models are all well known, and it's no surprise that more advanced models like Tree LSTMs perform better than the other approaches, consistent with (Tai et al., 2015). However, our results can only serve as preliminary data rather than be regarded as conclusive.

2 Background

2.1 Representing words as embeddings

Unlike the one-hot encoding technique where each word in the vocabulary represents one dimension, word embeddings map each word to a real-valued vector in a lower dimension space. By mapping the word using embeddings in a continuous vector space, we represent words with a high degree of co-occurrences as neighboring points in the vector space. In this study, we use the publicly available Word2Vec embeddings by (Mikolov et al., 2013b).

2.2 Bag of Word (BOW) models

Bag of Words (BOW) is one of the key modeling strategies employed in this study. Before being supplied into the network, the embedding vector of the input words is summed in a BOW model based on the frequency of occurrence in the dataset. The computational efficiency of BOW models is

an advantage over other types of models, however, this comes at the cost of neglecting word order and grammar. An intermediate hidden layer mapping between the initial embedding and the output can be introduced to enhance this paradigm. These models are known as Continuous Bag of Words (CBOW) models, and introducing the activation function in the linear layer yields a Deep CBOW model.

2.3 Long Short-Term Memory(LSTM)

Originally proposed in (Hochreiter and Schmidhuber, 1997), LSTM networks are a type of Recurrent Neural Networks (RNNs) in which a memory cell is used to carry information over the future time steps. This model is preferred over other models because it preserves word order although it has the drawback of being computationally more demanding. LSTM's consist of a memory cell vector along with the hidden state vector. The memory cell is connected to the hidden state through gates that regulate information of intake, output, and forgetting. The memory cell and the hidden state along with the gates help overcome the problem of vanishing gradient of the RNNs.

2.4 Tree-LSTM

A variant of the LSTM model uses tree structures, known as the N-ary Tree-LSTM (Tai et al., 2015). A Tree-LSTM preserves the intrinsic hierarchical structure of a phrase while parsing it, as opposed to LSTM's purely sequential information flow. In this study, we focus on the Binary Tree-LSTM which is a special case of N-ary Tree-LSTM, where each sentence to be classified is parsed as a binary subtree. It consists of a memory cell whose update is dependent on the hidden state vectors from the two-child units. The forget gate of the Tree-LSTM is dependent on the input from the cell's left and right child which allows each individual child to affect the parent unit independently.

3 Models

In this section, the models used in this study are discussed. All the models give an output vector of shape 5×1 of which each element corresponds to the log odds (logits) for one of the five sentiment classes. The argmax of this vector is then the predicted class label.

BOW: In this model, each word is associated with a multi-dimensional vector which expresses what sentiment this word conveys. Each vector is of size 5, the number of classes. When classifying a sentence, the sum of the vectors of each word and a bias vector is taken. The argmax of this summed vector results in the predicted class label. The model consists of one embedding layer which embeds words to an embedding size of 5.

CBOW: This Continuous Bag of Words model is the same as the BOW model, but now its embeddings can have a dimension of arbitrary size. It consists of one embedding layer which has word embeddings with size 300 and one linear layer which projects the embedding vector of size 300 down to 5 output units.

Deep CBOW: This deeper version of the CBOW model consists of one embedding layer which has word embeddings with size 300, two linear layers of 100 hidden units with Tanh activation and one final output projection layer with 5 output units. The Deep CBOW model was also trained with pretrained Word2Vec embeddings.

LSTM: This model consists of an embedding layer (of size 300) that uses Word2Vec, a single LSTM cell of size 300×168 and a linear output layer with dropout (with $p = 0.5$). For classification, the final hidden state from the cell is projected with the linear layer to 5 output units containing the logits for each sentiment class.

N-ary Tree-LSTM: This model consists of an embedding layer (of size 300) that uses Word2Vec, a single Tree-LSTM cell of size 300×150(which uses dropout with $p = 0.25$) and a linear output layer with dropout ($p = 0.5$). The classification part is the same as with the normal LSTM. In this research, we use binary trees, so $N = 2$.

Child-Sum Tree-LSTM: This model has the same architecture as the N-ary Tree-LSTM, but now uses a Child-Sum Tree-LSTM cell, where the left and right child were summed instead of concatenated.

Subtree-LSTM: This model has the same architecture as the N-ary Tree-LSTM, the only difference is in the training data that is fed into the N-ary Tree-LSTM. The train dataset at each node of the tree is supervised with the sentiment.

4 Experiments

4.1 Dataset

In order to determine how well the different models described in the previous section perform in predicting the sentiment of a sentence, the Stanford

Sentiment Treebank dataset was used (Socher et al., 2013). This dataset consists of sentences and their sentiment label. There are five sentiment labels, each defining a degree of sentiment: *very negative, negative, neutral, positive and very positive*. Additionally, a syntactic tree is provided for each sentence, where each node in that tree has its own sentiment label as well. The dataset is split into a training set (8544 samples), a development (or validation) set (1101 samples), and a test set (2210 samples).

4.2 Training

In the first experiment, all models were trained on the training set of the Stanford Sentiment Treebank dataset. For the BOW models, the model was evaluated on the validation set every 1000 iterations during training. For the LSTM models, the model was evaluated every 250 iterations, because mini-batches were used during training with a batch size of 25. A model converged when the validation accuracy did not improve for 10 evaluations on the validation set during training. Hereafter, the parameters of the model with the highest accuracy were saved into a `.pt` file. Finally, the best model was used to obtain the accuracy on the test set. The BOW models were trained with a learning rate of 0.0005 and the LSTM models were trained with a learning rate of 0.0002. For all models, the parameters were optimized using the Adam optimizer (Kingma and Ba, 2014) and Cross-Entropy was used as loss function.
After training and evaluating the performance of each model, another experiment was conducted. The trained models were used to evaluate the accuracy for each sentence length in the test set. Secondly, the Child-Sum Tree-LSTM model was trained and evaluated on the test set in order to be compared to the N-ary Tree-LSTM model. Furthermore, we attempted to train the Tree-LSTM on a modified train dataset which consists of supervised sentiments at each node of the tree. Finally, to check the importance of word order for the sentiment analysis task, all the models except Tree-LSTM were trained on sentences with the random shuffling of the words.
All experiments were done for three different random seeds. After each seed, the results (test accuracies, training accuracies, training losses) were stored in a Pickle file. This resulted in three Pickle files which were used to calculate the average accuracies and standard deviations over the three random seeds.

5 Results and Analysis

Figure 1 depicts the results of the first experiment, where the accuracies of the models on the test set of the SST dataset were obtained. The LSTM models (LSTM and N-ary Tree-LSTM) perform the best with an accuracy of 46.0% and 46.4% respectively. The normal BOW model performs the worst on the test set with an accuracy of only 28.7%. Introducing an embedding of larger size with CBOW does improve the accuracy significantly. Making the CBOW model deeper and non-linear by adding more layers and Tanh activation functions (Deep CBOW) slightly increases the accuracy. Results are displayed graphically in Figure 1.
The introduction of pretrained word embeddings with Word2Vec significantly boosts the performance of the models. As observed in Figure 1, there is a steep increase in accuracy from Deep CBOW to Pretrained Deep CBOW. This is due to the fact that pretrained word embeddings are trained on much larger corpora. The pretrained Deep CBOW model performs just slightly under the more sophisticated LSTM models.
Finally, a Tree-LSTM does not improve performance compared to the normal LSTM. Modifying

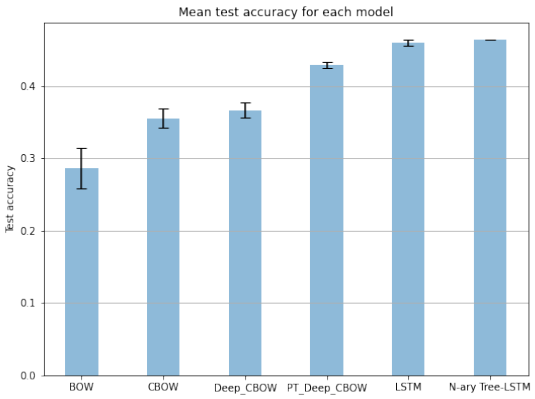

Figure 1: Mean test accuracy for each model including standard deviations.

the cell in the N-ary Tree-LSTM to a Child-Sum Tree-LSTM cell does let the model perform better in predicting the sentiment of sentences, as the accuracy slightly increases to the highest accuracy achieved in this study (46.8%) (See Figure 2). Supervising the sentiment at each node in the tree decreases the performance of the N-ary Tree-LSTM

model, reducing the accuracy to 44.5%. This could be due to the fact that subtrees do not represent full semantic and sentimental meaning of a sentence. In order to check the importance of word order,

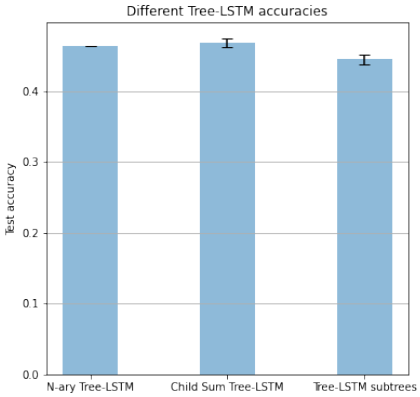

Figure 2: Mean test accuracy of three different Tree-LSTM models.

all models (except Tree-LSTMs) were trained on sentences where the words were randomly shuffled. The test set accuracy for these models did not significantly change (as depicted in Table 1), which means that word order is not very important for the task of sentiment classification.

Model	Not shuffled	Shuffled
BOW	28.7%	28.1%
CBOW	35.6%	36.2%
Deep CBOW	36.7%	37.3%
PT Deep CBOW	42.9%	42.5%
LSTM	46.0	45.5%

Table 1: Accuracy comparison when the words were randomly shuffled for all models except the Tree-LSTMs.

Finally, the influence of sentence length was investigated. Figure 3 depicts how the sentence length influences the test accuracy of the Deep CBOW model. The figures for all models can be found in Appendix A.

All models perform stable over almost all sentence lengths, for each sentence length they are all around their mean accuracy as shown in Figure 1. Compared to the LSTM models, the Deep CBOW and PT Deep CBOW model perform better on longer sentences, as can be observed in Figure 11 and Appendix A. The accuracies of all LSTM models drop and get very unstable after a sentence

Figure 3: The mean test accuracy (including standard deviation) for each sentence length for the Deep CBOW model.

length of around 45. However, the LSTM models are more stable over different seeds, since their standard deviations are smaller over all sentence lengths.

6 Conclusion

In the present study, we investigated how different models performed on the SST dataset. We found that introducing pretrained word embeddings to our models improved the performance significantly. Our findings showed that Deep CBOW and pretrained Deep CBOW models achieved the most stable performance across different sentence lengths, especially at longer sentences. Introducing the Child-Sum cell into the Tree-LSTM did not improve the general performance of the Tree-LSTM model. Lastly, the Tree-LSTM model supervised at every node of the tree, performed worse than the N-ary Tree-LSTM.

We cannot accept or reject our hypothesis totally based on our findings because there isn't enough proof. We were unable to demonstrate that word order had a significant impact on sentiment prediction. Given the limitations of our research, our conclusion should only be considered speculative. Training on larger datasets could aid in determining the true extent of these models' applicability to real-world language.

An interesting approach for future work would be to train the using Bidirectional Encoder Representations for Transformers (BERT) introduced in (Devlin et al., 2018). Word2Vec or GloVe models produce a representation to properly represent words, capturing semantic meaning. However, these models produce the same representation for a given word independent of the context. Compared to these models, pretrained BERT can create contextualized word embeddings.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Phong Le and Willem H. Zuidema. 2015. Compositional distributional semantics with long short term memory. *CoRR*, abs/1503.02510.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, abs/1503.00075.

Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015. Long short-term memory over tree structures. *CoRR*, abs/1503.04881.

A The influence of sentence length for all models

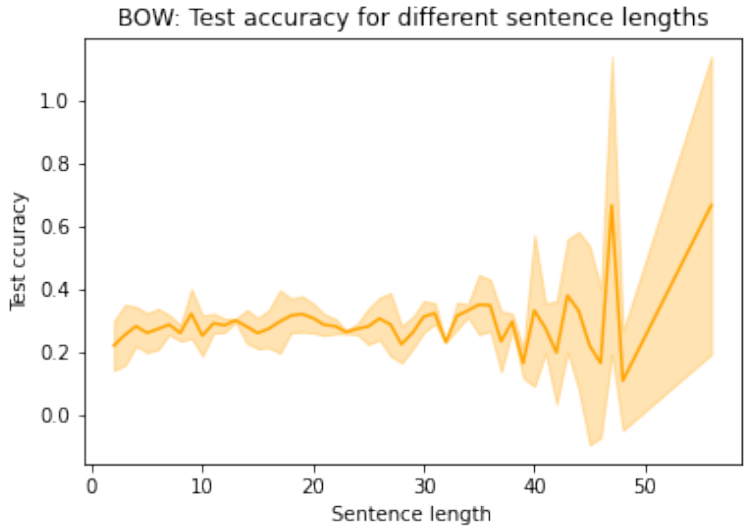

Figure 4: The test accuracy for each sentence length for the BOW model.

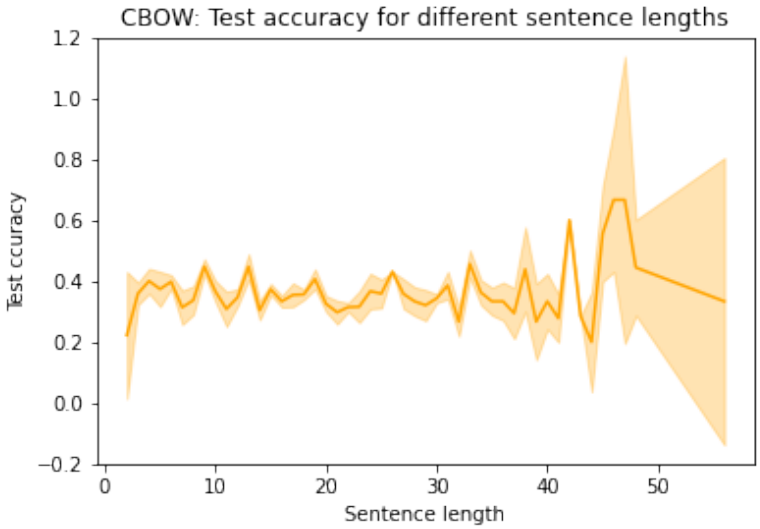

Figure 5: The test accuracy for each sentence length for the CBOW model.

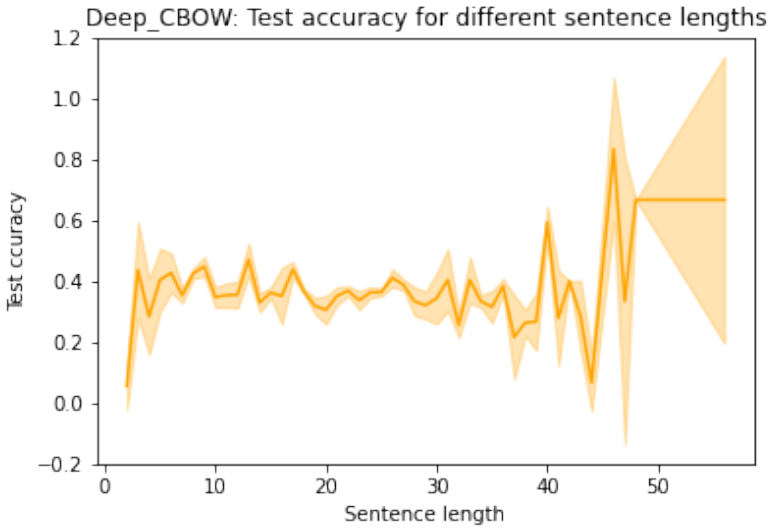

Figure 6: The test accuracy for each sentence length for the Deep CBOW model.

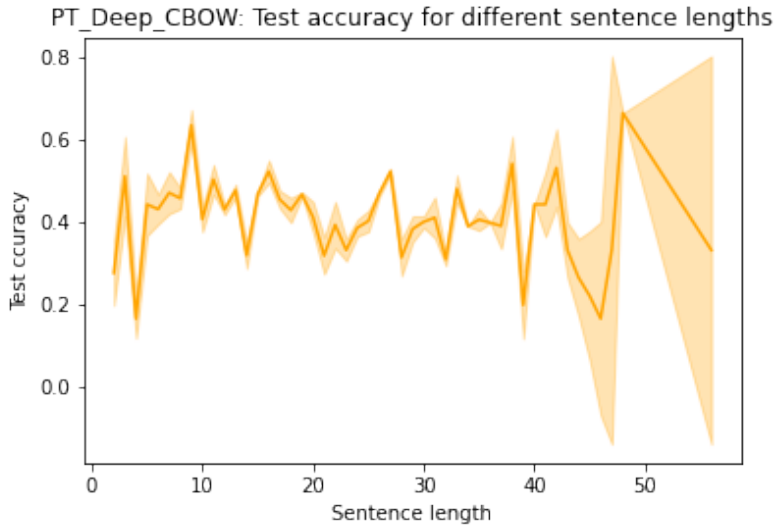

Figure 7: The test accuracy for each sentence length for the PT Deep CBOW model.

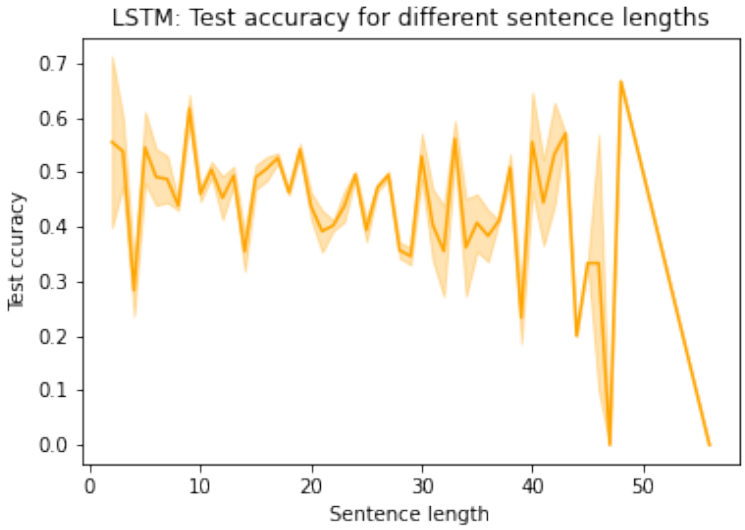

Figure 8: The test accuracy for each sentence length for the LSTM model.

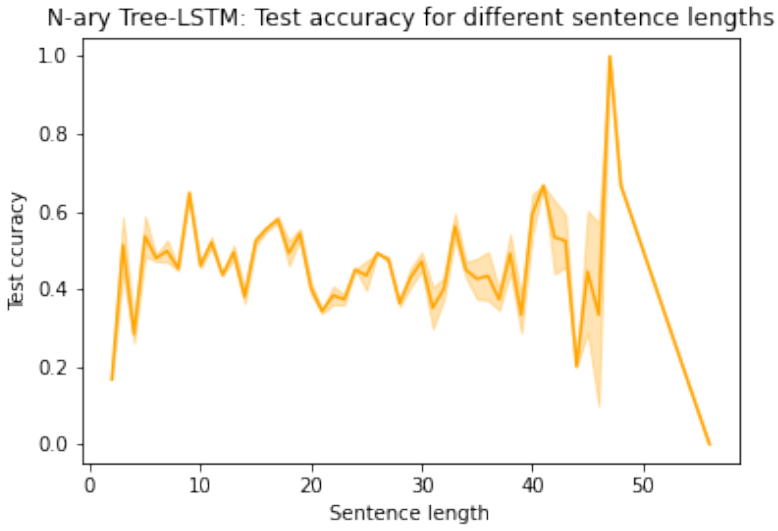

Figure 9: The test accuracy for each sentence length for the N-ary Tree-LSTM model.

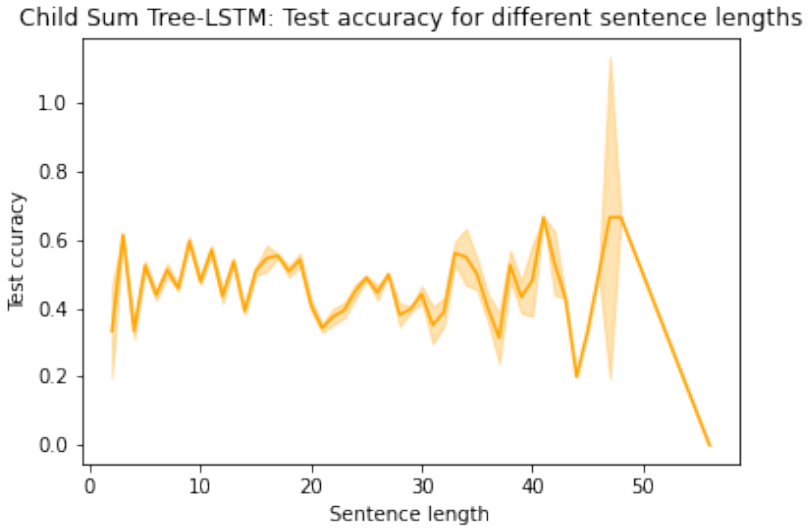

Figure 10: The test accuracy for each sentence length for the Child-Sum Tree-LSTM model.

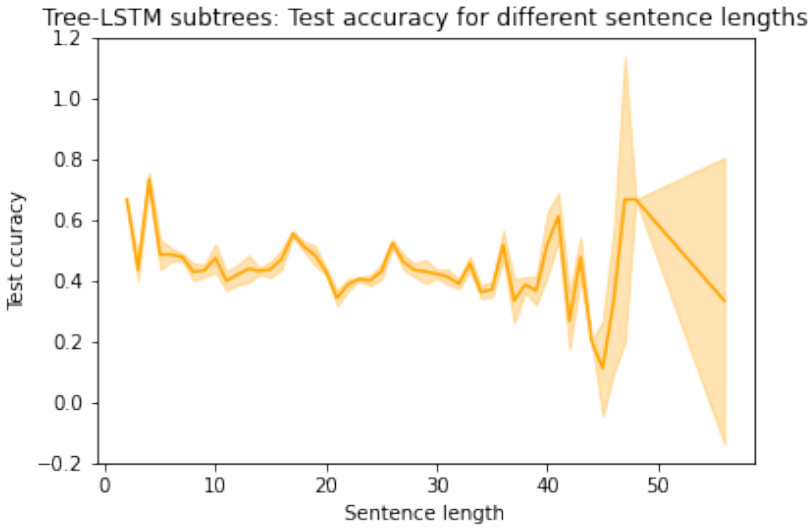

Figure 11: The test accuracy for each sentence length for the N-ary Tree-LSTM model trained on subtrees.