Practical 2 - Technical Report

Jochem Brandsema
14546620
jochem.brandsema@student.uva.nl

Joan Velja
14950480
joan.velja@student.uva.nl

1 Introduction

Sentiment analysis, a key component of understanding textual data, has widespread applications in several domains such as customer feedback and social media monitoring. This study focuses on comparing various neural network models for sentence representation in sentiment analysis, specifically using the Stanford Sentiment Treebank dataset [1].

Research Questions: Our study is guided by key questions:

1. What is the role of word order in sentiment classification tasks?
2. Does incorporating tree structures enhance model accuracy?
3. How does model performance vary with sentence length?
4. What is the impact of supervising sentiment at each node in the tree?
5. How do the Child-Sum Tree-LSTM and the N-ary Tree-LSTM compare on this task?

Motivation: These questions are pivotal for a good understanding of the models we will implement. Understanding the nuances of these factors can significantly improve the accuracy and applicability of sentiment classification models in real-world scenarios.

Expectations: We hypothesize that models sensitive to word order and tree structures will demonstrate superior performance, attributing to their ability to capture more contextual and structural information in text.

Literature Review: [Review of related literature and discussion of how these questions have been previously addressed or overlooked.]

Methodology: Previous studies have explored various models for sentiment analysis. Our report includes a comprehensive evaluation of models such as BOW, CBOW, Deep CBOW, LSTM, and Tree-LSTM, under various configurations to address the posed research questions and especially in a comparative fashion. Our approach involves a detailed evaluation of these models, examining their performance with and without the usage of pre-trained word embeddings.

Summary of Findings: Our findings are expected to reveal insightful trends and dependencies in sentiment analysis models. [Findings will be detailed here...]

[Please note: Detailed findings and analysis will be added upon completion of the experiments.]

In the following sections, we delve deeper into the background of sentiment analysis models, our experimental setup, and a thorough analysis of the results, culminating in a discussion of the implications and potential directions for future research.

2 Background

This study relies on several key techniques in natural language processing, particularly Bag of Words (and variants), word embeddings, LSTM, and Tree-LSTM. Understanding these concepts is crucial for comprehending the methodology and results of our research. Let us break them down.

2.1 Bag of Words and Its Variants

The Bag of Words (BOW) model is a fundamental concept in NLP. It represents text by the frequency of words within it, ignoring grammar and word order (Harris, 1954). Although this is the classic approach, we implement a gradient based method to come up with a vector that associates words with output classes. While BOW is widely used due to its simplicity, it fails to capture the semantic relationships between words, limiting its effectiveness in complex NLP tasks.

The Continuous Bag of Words (CBOW) model utilizes word embeddings, providing a richer repre-

[1] https://nlp.stanford.edu/sentiment/

sentation of text (Mikolov et al., 2013a), and learns a linear layer to perform classification. Embeddings allow CBOW to capture semantic meanings, but like BOW, it still treats text as an unordered collection of words.

Deep CBOW extends the CBOW model by adding multiple layers and non-linear activations, allowing the model to capture more complex relationships. However, the increased complexity also raises the risk of overfitting, particularly on smaller datasets.

2.2 Word Embeddings

Word embeddings are a foundational aspect of modern NLP. They provide a way to represent words as vectors in a high-dimensional space, capturing semantic relationships between words. Techniques like GloVe (Global Vectors for Word Representation) and Word2Vec are prevalent, offering unique advantages in capturing word meanings (Pennington et al., 2014; Mikolov et al., 2013b). Embeddings are particularly beneficial in sentiment analysis for encoding complex nuances and relations in language.

2.3 LSTM and Tree-LSTM

Long Short-Term Memory (LSTM) networks, a special kind of RNN, are adept at processing sequences of data, like text (Hochreiter and Schmidhuber, 1997). Unlike standard RNNs, LSTMs can capture long-term dependencies in data, making them suitable for understanding the context in sentences or paragraphs. This ability is critical in sentiment analysis where the sentiment often depends on the broader context of a sentence.

Building on the concept of LSTMs, Tree-LSTM incorporates tree structures into the network, allowing it to process data with hierarchical relationships (Tai et al., 2015; Le and Zuidema, 2015; Zhu et al., 2015). This is particularly relevant for our project, as the Stanford Sentiment Treebank provides sentences with an associated binary tree structure.

2.4 Relation Between Techniques

BOW and its variants provide foundational techniques for text representation, but their effectiveness is dramatically influenced by the complexity of the task and the richness of the dataset. Word embeddings provide the input layer for models of both the BOW and LSTM families, and Tree-LSTMs extend LSTMs to handle semantical structure in data. Understanding these relationships is key to appreciating the advancements in sentiment analysis models.

In the following sections, we will explore how these techniques are implemented in our project and their impact on the performance of different sentiment analysis models.

3 Models

This section outlines the neural architectures of the sentiment analysis models employed in this study: BOW, CBOW, Deep CBOW, LSTM, Binary Tree-LSTM and Child-Sum Tree-LSTM. Each model has unique characteristics that make it suitable for different aspects of sentiment classification.

3.1 Bag of Words (BOW) Model

Unlike traditional BOW models which rely on word frequency, our neural BOW model associates each word with a multi-dimensional vector, representing the sentiment conveyed by the word, that can be learned using gradient based methods. Each entry in a vector corresponds to one of the output classes.

To classify a sentence, we sum the vectors of all words in the sentence along with a bias vector. This process, while straightforward, results in the loss of word order information. Hence, it's termed as a neural bag-of-words model. The resulting vector consists of the logits for each class, so the final sentiment is determined by the *argmax* of this vector.

3.2 Continuous Bag of Words (CBOW) Model

In the CBOW setting, word embeddings have an arbitrary size, allowing the model to capture a larger share of information about each word. The technique is the same as BOW, and the summed vector of these embeddings is transformed to the output class size using a learned parameter matrix W.

3.3 Deep CBOW Model

The Deep CBOW model extends the CBOW architecture by adding two linear layers (for a total of three) and tanh activation functions between them.

3.4 Deep CBOW Model with Pre-Trained Embeddings

This variant of Deep CBOW, just like all LSTM models described hereafter, uses the GloVe word representations as static pre-trained embeddings, rather than learning the embeddings from scratch.

3.5 Long Short-Term Memory (LSTM) Model

Our LSTM model processes text data sequentially, capturing long-term dependencies that are crucial for understanding context. A linear layer with dropout was added to this and the other LSTM architectures to allow classification.

3.6 Binary Tree-LSTM Model

The Binary Tree-LSTM model leverages the binary tree structure provided by the Stanford Sentiment Treebank. It extends the capabilities of the standard LSTM to handle structured data, potentially offering a more accurate understanding of the sentiment expressed in different parts of a sentence.

3.7 Child-Sum Tree-LSTM Model

The Child-Sum Tree-LSTM model is a Tree-LSTM model that can handle any number of children per node, where the Binary Tree-LSTM only functions on binary trees. A drawback of this model is that it cannot perceive the order of a node's children.

4 Experiments

4.1 Task and Data Description

The task in our experiments is sentiment classification using the Stanford Sentiment Treebank (SST) dataset. The SST dataset consists of sentences, their binary tree structure, and fine-grained sentiment scores. A review consists of a single sentence, and we have a sentiment score for each node in the binary tree that makes up the sentence, including the root node (i.e., we still have an overall sentiment score for the entire review). The sentiment scores range from 0 (very negative) to 4 (very positive). We have 8544 training sentences, 1101 validation sentences and finally 2210 test sentences. Additionally, we employ an extended training set where every node from the original training set is treated as one sample. We also split the test set in two; one with shorter sentences and one with longer sentences.

4.2 Training and hyperparameters

All models are trained twice, once on the original training set and once on the extended training set, using the Adam optimization algorithm and the cross-entropy loss. We use a batch size of 1 for the BOW variants and 25 for the LSTM variants. All BOW models are trained for 30000 iterations with learning rate 0.0005 and all LSTM models for 3000 iterations with learning rate $2e{-}4$, with models being evaluated on the validation set every 1000 and 100 iterations respectively. After training, the version with the best accuracy on the validation set is returned as final model for further evaluation. This procedure was repeated with 5 different seeds.

All models use word embeddings of size 300, except for the vanilla BOW model, where the embedding size must equal the output size of 5. The used hidden sizes were 100 for Deep CBOW, 168 for LSTM and 150 for both Tree-LSTM models.

Given the availability of two types of embeddings, we decided to implement GloVe in our research, due to its use of global co-occurrence information and overall larger vocabulary size: only 976 words in the training corpus are unknown to GloVe, compared to 2779 (roughly 3x) of Word2Vec. Comparisons between GloVe and Word2Vec are presented in the appendix.

4.3 Evaluation

For evaluating the models, we use accuracy. The performance is calculated based on the models' ability to correctly classify the sentiment of the sentences in the test set of the SST dataset and in the two splits of this test set as mentioned before. As each model is trained for every random seed, the mean test accuracy and the standard deviation across these runs is reported.

5 Results

As it can be seen in 1, there is a trend where more complex models that are able to capture more contextual information and dependencies in text (like LSTM, CLSTM, TLSTM) perform better than simpler models (like BOW and CBOW). As expected, BOW and CBOW, due do their underlying assumptions, fail to go above 26% and 35% accuracy respectively. Their approach to text representation ignores order and context, which is likely the reason for their poor performance. This is to be expected, as these two models do not even leverage non-linearities. Results get slightly better with Deep-CBOW, leveraging some degree of non-linearities, but biggest jump is provided by the implementation of **word embeddings**: going from words treated in isolation, without containing any contextual informations, to embedded words [...], we obtain a 5% increase in accuracy. This is once more to be expected: [...]. Sensible improvement is obtained also by allowing finetuning of word embedding features, although it is to be mentioned that the

runtime increases $\approx$ 9x. The tradeoff, in our opinion. is not worth the improvement, thus we would suggest probing instead of full finetuning. To our surprise, LSTMs do not yield the improvement we expected: we only obtain a roughly a 2% absolute increase in accuracy going from the PT Finetuned Deep CBOW to the Vanilla LSTM. These results allow us to draw some conclusions about our initial claims: first of all, they prove how crucial word order is in sentiment analysis. This is because the arrangement of words can completely change the meaning of a sentence. For instance, "not good" and "good" convey opposite sentiments, and this distinction is only possible to capture if the model considers the order of words. Models like LSTMs are specifically designed to address this aspect. Subsequently, tree structures allow models to capture not just the linear sequence of words, but also the hierarchical structure of language, as parsed in syntactic trees. As expected, this is beneficial for sentiment analysis because for instance, tree structures can help differentiate the scope of negation, which can be pivotal for accurate classification. Tackling the issue of sequence length, we provide the following results:

[... gotta decide which graphs ...]

As expected, BOW models show little variation in performance across different sentence lengths, with a slight increase in the medium length (21-30 words). This model does not account for word order or sentence structure, so the length has minimal impact on performance. **NO IDEA ON HOW TO MOTIVATE THE DROP IN CBOW TBH**. Deep CBOW on the other hand, still show decline in accuracy with sentence length, but less steep compared to the CBOW model, potentially hinting how depth allows the model to better handle longer contexts to some extent. The Pretrained Deep Continuous BOW with GloVe embeddings shows a surprising resilience in the longest sentence category (30+ words), which might indicate that the embeddings aid at providing contextual information that helps the classification even when the sentence gets complex. To no surprise, the TreeLSTM and Child-Sum Tree-LSTM models show less of a performance decrease in longer sentences compared to LSTMs, suggesting once more how structure is beneficial for longer sentence understanding. However, it's surprising to see that their advantage is not as large as one might expect, indicating that either other factors might also be at play or that the ambiguity in the dataset is non trivial.

Finally, we address the performance of Child-Sum Tree-LSTM versus Binary Tree LSTM. The former is designed to handle trees with a variable number of children, summing the hidden states of all children nodes. The binary Tree-LSTM, on the other hand, is structured to handle a fixed number of children (2), which makes it more restrictive but can be more efficient if the tree structure of the sentences is known to have a certain shape, which is the case in our dataset: indeed, Binary-Trees perform the best among all models, although not by a big margin! It is worth noting that Child Sum Trees have a (slightly) smaller amount of parameters (6534905 vs. 6399755), so the performance was actually quite surprising to us.

6 Conclusion

Our experiments have reinforced the critical role of word order and sentence structure in sentiment analysis, highlighting that models which account for these elements, such as LSTMs and TreeLSTMs, outperform more traditional approaches like BOW. Furthermore, the integration of pretrained word embeddings, particularly GloVe, has substantially enhanced model efficacy (and efficiency) by leveraging a pre-existing semantic framework. Fine-tuning pretrained models on the target dataset has also proven beneficial, demonstrating the value of tailoring general purpose tools to the specifics of the task at hand. Additionally, our study has revealed that tree-based models demonstrate robustness in managing variable sentence lengths, endorsing the integration of linguistic structures into model design to better reflect the hierarchical nature of human language.

Our findings align with current literature, corroborating the established understanding that deep learning models, particularly those that incorporate sequential and contextual information, are generally superior for sentiment analysis tasks. The literature also consistently highlights the effectiveness of pretrained embeddings in capturing latent semantic relationships that are not immediately evident from a surface-level analysis.

Unexpectedly, our experiments revealed a significant decline in LSTM performance with increased sentence length, suggesting a possible limitation in the model's ability to handle long-term dependencies without additional supportive mechanisms. We suspect that allowing for finetuning of the embed-

Model	Acc. (%)	Std. Dev.	Δ (%)
BOW	25.35	0.015	-
CBOW	34.53	0.018	+9.18
DCBOW	36.96	0.015	+2.43
PT	43.32	0.005	+6.36
PT Fine	44.42	0.007	+1.10
LSTM	46.13	0.005	+1.71
CLSTM	46.66	0.006	+0.53
TLSTM	47.62	0.004	+0.96

Table 1: Performance of different models on the dataset. Δ indicates the absolute percentage improvement.

dings could prove beneficial to the improvement of the model: we decided not to pursue this path, due to the huge computational and time cost of this approach.

Based on our insights, we recommend further exploration into the implementation of attention mechanisms as to address performance dips in longer sentences. [cite a paper maybe?]

References

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

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

Phong Le and Willem Zuidema. 2015. Compositional distributional semantics with long short term memory.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. Cite arxiv:1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. 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.

Xiaodan Zhu, Parinaz Sobihani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1604–1612, Lille, France. PMLR.

Model	Acc. (%)	Std. Dev.	Δ (%)
PT W	42.15	0.004	-
PT G	43.32	0.005	+1.17

Table 2: GloVe (G) vs. Word2Vec (W), PT DCBOW. Δ indicates the actual percentage improvement.

Model	Acc. (%)	Std. Dev.	Δ (%)
PT W F	43.97	0.012	-
PT G F	44.42	0.007	+0.45

Table 3: GloVe (G) vs. Word2Vec (W), Finetuned (F) PT DCBOW. Δ indicates the actual percentage improvement.

A Example Appendix

Model	Acc. (%)	Std. Dev.	Δ (%)
W-LSTM	44.25	0.009	-
G-LSTM	46.13	0.005	+1.88

Table 4: GloVe (G) vs. Word2Vec (W), LSTM. Δ indicates the actual percentage improvement.

Model	Acc. (%)	Std. Dev.	Δ (%)
W-TLSTM	46.18	0.007	-
G-TLSTM	47.62	0.004	+1.44

Table 5: GloVe (G) vs. Word2Vec (W), Tree (T) LSTM. Δ indicates the actual percentage improvement.

Model	Acc. (%)	Std. Dev.	Δ (%)
W-CLSTM	43.86	0.008	-
G-CLSTM	46.66	0.006	+2.80

Table 6: GloVe (G) vs. Word2Vec (W) in Tree (T) Child Sum LSTM. Δ indicates the actual percentage improvement from W-CLSTM to G-CLSTM.