Adding Convolution Operations to an LSTM for WSD

1st Lisa Ewen
Department of Computer Science
Lakehead University
SN: 0655603
Thunder Bay, Canada
lewen@lakeheadu.ca

2nd Tim Heydich
Department of Computer Science
Lakehead University
SN: 1121117
Thunder Bay, Canada
theydric@lakeheadu.ca

3rd Bleau Moores
Department of Computer Science
Lakehead University
SN:0011548
Thunder Bay, Canada
bdmoores@lakeheadu.ca

***Abstract*—Word Sense Disambiguation is a very challenging task in the field of natural language processing(NLP). Most commonly supervised machine learning methods are used to approach this problem due to their high accuracy in comparison with other techniques. However, supervised learning methods still have flaws and require analysis to determine what would improve these models to achieve better accuracy. The proposed amendment to the bidirectional long short term memory(LSTM) approach given by Mikael Kågebäck and Hans Salomonsson, modifies the original structure to include convolutional layers. This produced two amended models, the first of which includes convolution operations between the embedding and BiLSTM layers. The second model includes convolution operations between the embedding and BiLSTM layers, as well as after the BiLSTM layer. While the results given by the simulation testing are unusually high, the first model slightly outperforms the original model in terms of its cosine similarity among the stochastic gradient descent(SGD), Adam and NAdam optimizers. The second model, unfortunately, significantly under-performs.**

***Index Terms*—WSD, LSTM, BLSTM, CNN LSTM**

I. Introduction

While much progress has been made in the area of word-sense disambiguation (WSD), it is still defined as an open problem in NLP. WSD is the task of identifying the meaning of a given word in the specific context it exists in. For example, the word "port" has many meanings: a type of alcoholic beverage, a harbour town, an opening in machinery, the left side of a ship, the process of transferring a phone number between carriers and the communication endpoint in a network. Deciphering ambiguity is often a simple task for the human brain, which is capable of understanding the sophisticated nuances that exist in human language. As is the case with all NLP tasks, a computer has difficulty understanding language, which often has ambiguous meanings and definitions. This is extremely present in the case of WSD, where determining the meaning of a word is heavily reliant on its context, which is sometimes a difficult task for even humans to complete.

There has been significant progress in the field of WSD with supervised learning approaches, especially in comparison to the performance of unsupervised learning methods. The most notable among the supervised learning methods are support vector models (SVM), neural networks, naive Bayesian models, decision trees, and decision lists [4]. While over time these methods have been able to achieve increasingly better results, these methods have not yet been perfected. It is then imperative to analyze the work that has been done and make a determination on what improvements can be made to the existing work to generate new outcomes.

This paper contains a discussion on a bidirectional LSTM model proposed by Mikael Kågebäck and Hans Salomonsson [1] and an improvement to the original work to produce better results is suggested. Included in the following sections shall be a discussion of the work by Kågebäck and Salomonsson as well as a few other relevant authors, our proposed amendment to the original work, the simulation results, and discussions for future research.

II. Related Works

The developed model is based on the model proposed by Kågebäck and Salomonsson, we first must discuss their approach and subsequent results. Their model features a softmax layer, a hidden layer, and BiLSTM, which shares parameters with the hidden layer over all word types and senses. The softmax layer selects the weight matrix and bias vector for each word type in the model. The inputs to the model are one-hot representations of a word type, and Glove vectors are used for word embeddings. The model achieved F1 Scores of 66.9 and 73.4 on Senseval 2 and 3 English lexical sample task, respectively.

Following the work by Kågebäck and Salomonsson, we see more authors approach WSD using an LSTM model such as Yuan et al. [7] Le et al. [2]. Yuan et al. note the work by Kågebäck and Salomonsson, stating that their approach differs as they do not use a bidirectional LSTM, and is trained on large training set in comparison to the limited number of word sense-labelled examples featured in the model by Kågebäck and Salomonsson. Yuan et al. also propose a semi-supervised LSTM model in addition to the supervised LSTM using a large number of unlabeled sentences from the web. In all simulation tests, their model outperforms other algorithms from authors such as Taghipour and Ng, Chen et al. and Weissenborn et al. with the exception of Sem-Eval 2013.

Le et al. also use an LSTM-based language model and cite the work done by Yuan et al. as a basis for their approach.

The authors criticize that the model created by Yuan et a was not available to the community, and therefore resulte in a standpoint for further research on the method. In thi case, the authors attempt to reconstruct and reproduce th model and results given to better understand their approac to enable further improvements to this type of method. Afte approaching this reconstruction, the authors noted simila results to Yuan et al. while using significantly less data tha was used in the original work. While others have take the work by Kågebäck and Salomonsson and approached i differently, the other authors mostly made changes to the wa the model was being trained or more minor changes to th LSTM structure that still resemble a traditional LSTM. Wha has yet to be attempted is combining convolutional layers t the typical LSTM structure both in reference to Kågebäck an Salomonsson, or in the field of WSD as a whole. Relativel recently, we have seen some researchers approach other NL tasks such as sentiment analysis [6] and emotion recognitio [8] using CNN LSTM models with promising results.

When working with text as data, which is the case for WSD then the text needs to be transferred into number form. Thi process is referred to as vectorization. Vectorization is neede since the computer can only work with numbers. There ar vectorization approaches that produce sparse data vectors suc as Term Frequency - Inverse Document Frequency (TF-IDF where each word has a single value. The other approache produce dense data such as Word2Vec or GloVe [5]. Thes approaches are referred to as embedding, and they produc a vector per word instead of just a single value. The vec torization used in this project is the GloVe embedding fror Stanford. Both GloVe and Word2Vec preserve the correlatio between different words in the corpus, and they can be use to determine words based on other vectors. This means that when one, for example, has the vectors for 'king', 'men' and 'women', then 'king'-'men'+'women' would give the vector for the word 'queen'. Word2Vec achieves this through either a continuous bag of words or skip-gram. GloVe tries to be more transparent and achieve higher accuracy by only relying on word occurrence statistics in the corpus.

III. Proposed Model

Two different modifications to the original model architecture by Kågebäck and Salomonsson are composed. Both models maintain the originally defined layers but with some convolutional layers input in varying orders. It must be noted that the base code used is not from the authors Kågebäck and Salomonsson; instead, a modified version of the code from Jeff09 [3] was used. The code was further modified, with the additional layers elevated to a more functional level.

A. Model 1

The first model features the convolution operations taking place between the embedding layer and the BiLSTM layer, with the remaining layers being the same as the original structure by Kågebäck and Salomonsson. Figure 1 displays a visual representation of model 1.

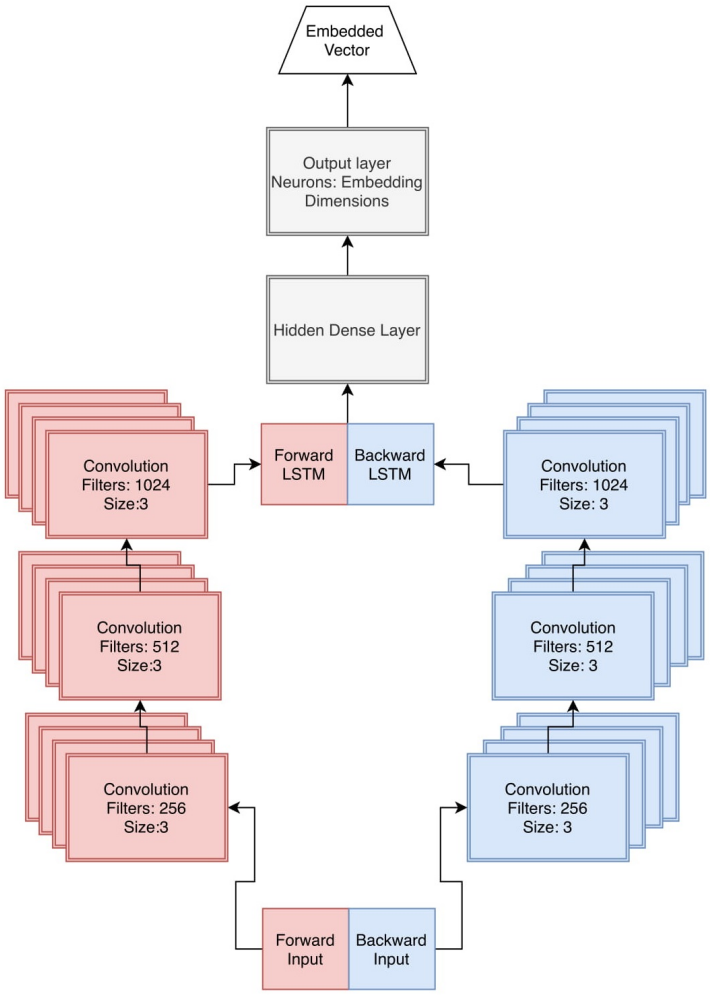

Fig. 1. Model 1 Architecture

B. Model 2

The second model structure is identical to the previously discussed model until the BiLSTM layer. However, an additional three convolution operations take place following the BiLSTM layer. The remaining layers follow in the same order as the previously discussed model. A visual representation of model 1 is shown in figure 2.

IV. Evaluation Metric

In order to compare the predicted vector and goal vector, we use cosine similarity given by

$$\cos(\mathbf{t}, \mathbf{e}) = \frac{\mathbf{te}}{\|\mathbf{t}\|\|\mathbf{e}\|} = \frac{\sum_{i=1}^{n} \mathbf{t}_i \mathbf{e}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{t}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{e}_i)^2}} \quad (1)$$

We apply the cosine similarity to the vectors given by three different optimizers: SGD, Adam, and NAdam. These optimizers are evaluated over three train/test runs in which the average of these runs is taken.

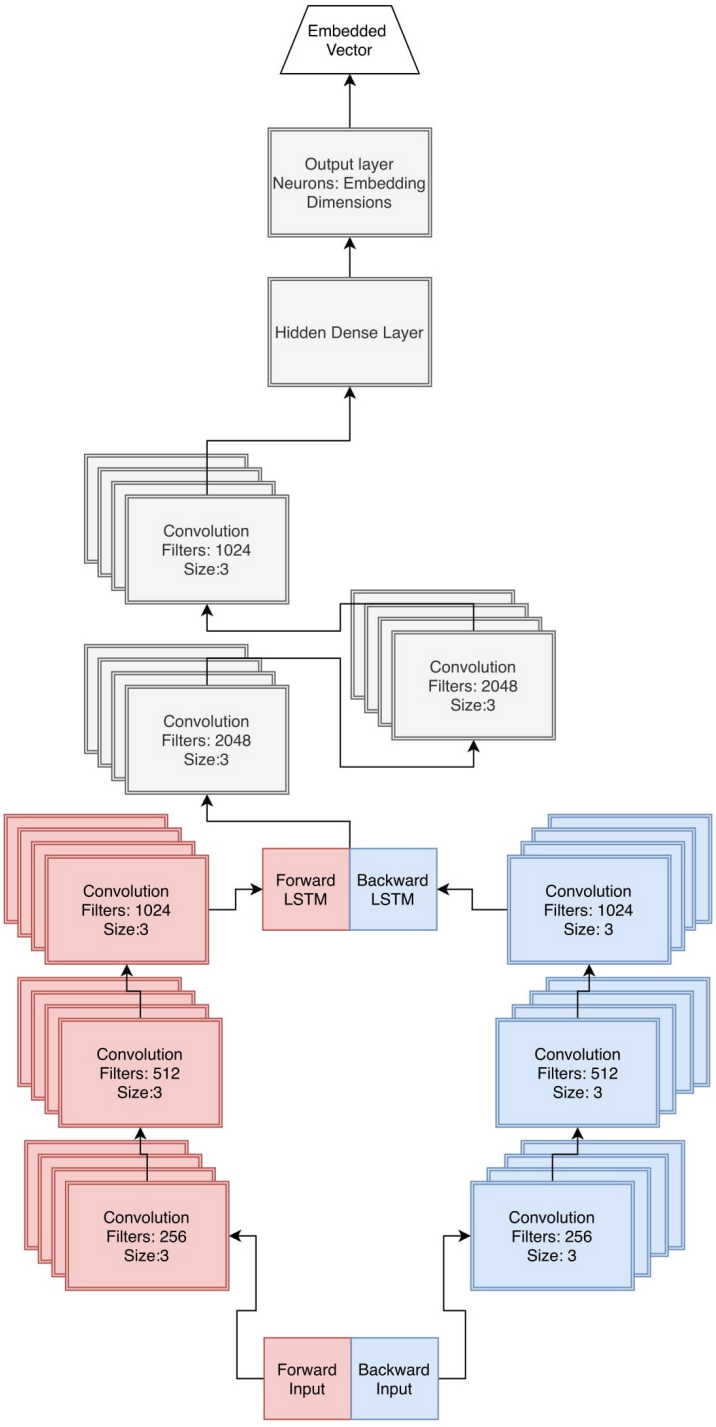

Fig. 2. Model 2 Architecture

V. Simulation Results

Both models 1 and 2 are tested alongside an implementation of the model proposed by Kågebäck and Salomonsson. The average results of each model are given in table 1.

TABLE I
Average Simulation Results

Model	SGD	Adam	NAdam
Original Model	2.83	3.01	95.23
Proposed Model 1	2.96	2.87	97.16
Proposed Model 2	1.43	1.67	61.11

From these results, it was determined that model 2 performs significantly worse than the other models and that model 1 performs slightly better than the original model by Kågebäck and Salomonsson.

That being said, it is essential to acknowledge that the results obtained are unusually high. Different attempts for training and testing were tried to achieve more reasonable results, including using training data that is split into training and validation data, creating separate sense embeddings for the test data, and using the test target sense ID to get the proper embedding from the training embedding. Each of these efforts resulted in the same unusually high accuracies.

VI. Conclusion and Future Work

Historically, supervised learning models have been the most accurate and popular choice when building a WSD model. Despite this progress using supervised learning models, there remains room for improvement among them. This requires us to look closely at the methods being used and analyzing what improvements can be made. The first to implement an LSTM approach was the authors Mikael Kågebäck and Hans Salomonsson, though a few attempts were made to make improvements upon their original work. Despite the positive results of the work of these other authors, it seemed imperative to utilize another technique that has been utilized elsewhere in the field of NLP: adding convolution operations in combination with an LSTM.

During this approach, two models are introduced. The first model having convolution operations added before the BiLSTM layer and the second model with convolution operations both before and after the BiLSTM layer. In comparison between those two models, and the original model by Kågebäck and Salomonsson, it was noted that the first model performed better than the other models, despite the testing outputs of all models being unusually high. Unusually high meaning that the accuracy achieved does not seem likely and is unexpected. These high accuracy values are believed to be caused by the way the test data is being processed and compared. Multiple ways to achieve a workaround were attempted. However, these attempts all had the same result when it came to the testing accuracies. The code base used, from Jeff09, was not functioning correctly, and several adjustments had to be implemented to achieve a working and testable pipeline. This means that

the initial testing set up utilized by Jeff09 was not working; this led to the different attempts to test the model. The first attempt to evaluate the model was splitting the training data into training and validation data; this resulted in the first appearance of the high accuracy values. Accuracies that high seemed unlikely, which led to the second attempt where a separate embedding matrix was created for the test data and then the predicted key was compared to the embedding in the matrix. As mentioned above, this also produced the same high cosine similarity score. The third and final attempt utilizes the embedding matrix created from the training set. The test data is passed through the model, and then the goal sense id is used to retrieve the embedding of said id from the training set. This embedding is then compared with the predicted one, but this attempt also yielded the same cosine similarity score. This begs the question of whether the models do perform exceptionally or if more changes need to be made to the evaluation process.

In addition to the different model structures, it needs to be noted that the optimizer had a significant impact on the performance of the models. This can be observed in table I, which clearly shows that both SGD and Adam optimizers did not produce good results. The optimizer that produced the best results is NAdam. This displays that there could promising value in utilizing convolution operations with LSTM models for a WSD application, as well as other models where convolution operations may not have been previously used. Some other future work may aim to modify the structure we have proposed or employ other pre-processing techniques to provide more reasonable testing outcomes.

REFERENCES

[1] Mikael Kågebäck and Hans Salomonsson. "Word Sense Disambiguation using a Bidirectional LSTM". In: *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex - V)*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 51–56. URL: https://www.aclweb.org/anthology/W16-5307.

[2] Minh Le et al. "A deep dive into word sense disambiguation with lstm". In: *Proceedings of the 27th international conference on computational linguistics*. 2018, pp. 354–365.

[3] Jeff09 (Kun Li). *Word Sense Disambiguation using Bidirectional LSTM*. https://github.com/Jeff09/Word-Sense-Disambiguation-using-Bidirectional-LSTM. 2018.

[4] Lokesh Nandanwar and Kalyani Mamulkar. "Supervised , Semi-Supervised and Unsupervised WSD Approaches : An Overview". In: 2015.

[5] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

[6] Jin Wang et al. "Dimensional sentiment analysis using a regional CNN-LSTM model". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2016, pp. 225–230.

[7] Dayu Yuan et al. "Semi-supervised word sense disambiguation with neural models". In: *arXiv preprint arXiv:1603.07012* (2016).

[8] Jianfeng Zhao, Xia Mao, and Lijiang Chen. "Speech emotion recognition using deep 1D & 2D CNN LSTM networks". In: *Biomedical Signal Processing and Control* 47 (2019), pp. 312–323.