Predicting Verbs Using Feed-forward CBOW Model

Rawan Abdulelsadig 35324987

Abstract

In this work, a simple language model was built using a two-layer feed-forward CBOW architecture then examined for its abilities when predicting verbs in sentences extracted from BBC news articles from 5 categories: business, entertainment, politics, sport and technology, The size of the context window was investegated for its effect on the model performance in addition to word lemmatization. It was found that a context window of 6, 7 or 8 words achieved the best models in all categories both with and without lemmatization resulting in predicting the verbs in the top 10 $\approx 75\% - 80\%$ and in the top 5 $\approx 67\% - 77\%$. While lemmatization was found to slightly reduce the model generalization abilities which was unexpected.

1 Introduction

Understanding written text is one of the most challenging tasks in natural language processing, it requires the model to ”understand” the context of the text and encode that understanding internally. This internal understanding is used to produce reasonable outputs for a variety of NLP problems such as text summarizing, part-of-speech tagging and machine translation.

The challenge lies in the question of ”how to encode this understanding of the context into the model?”, a trending approach is to use learning models (e.g. deep learning) to construct vector representations of words that somehow encode the common characteristics between them, which in tern helps to learn to ”understand” the context of the text when certain words co-occur in a sentence or a document.

A model that is able to successfully encode the context of written text is a model that is capable of predicting a missing word given a set of words occurring in that sentence. Therefore, predicting a missing word or a verb in a sentence can be thought of as a simplified task in the massive field of language models.

2 Related Work

Obtaining word representations that allow for building language models and thereafter achieving good context prediction results have been an active area of research for a long time. Bengio et al. proposed a probabilistic language model using a simple neural network, where a set of one-hot encoded vocabulary words are fed to a feed-forward neural network as ”context” and a probability distribution across the vocabulary is produced. Mikolov et al. introduced the continuous bag-of-words model CBOW which was inspired by both the bag-of-words approach and the feed-forward NN language model proposed by Bengio et al., the main objective of this model is predicting a mid-word given its surrounding words as a bag-of-words, and those words are represented using a distributed continuous vector representation such as word embeddings. Sundermeyer et al. suggested using the special type of recurrent neural networks (RNNs): long Short-Term Memory (LSTM) for building better language models. An LSTM network aids some of the problems of a classic RNN helping the optimization process, in addition to the gated mechanisms in its architecture that allows for keeping some key information about the sequence of data in its ”long-term memory” and ”short-term memory”.

This work takes a simpler attempt to the problem where sentences are built using the CBOW approach and words are defined as one-hot encoded vectors, with the objective of predicting verbs in the sentence which helps narrow down the probability space and makes the problem more suitable for using the simple feed-forward CBOW model.

3 Data

The BBC articles dataset (Greene and Cunningham, 2006) was used to construct the sentences for the model, it contains 2225 documents from the BBC news website corresponding to stories in five main areas: business, entertainment, politics, sport and tech from the time period 2004 - 2005.

Each news category was examined separately,then all categories were bind together to examine the model performance in that case as well. The text files were used to build the vocabulary corpus (which contains the words that are used as context and fed to the model) and the verbs corpus (which is used as the target space).

The dataset was then formed using a fixed context window that traverses each sentence in each text file and constructs a set of pre-processed words forming the ”context” of the sentences for the model to learn from, while also grapping a verb in that window (or randomly if there was multiple verbs in the same window) and storing it as the target word.

4 Methodology

The general approach of this work was transforming the words to standard form while keeping track of the verbs in the sentences, then attempting to learn the context in each sentence using a CBOWs model and predicting a verb in that sentence.

4.1 Text Preparation

From every article, paragraphs were split into sentences ending with ”.”, each sentence was then stripped from the numerical values since they are not of interest for this particular language model. *ftfy.fix_text()* was then used to fix whatever encoding / decoding problems in Unicode text that could exist.

Each pre-processed sentence is then tokenized to lower-cased words, then their corresponding part-of-speech was obtained using a POS tagger in order to filter out the verbs and add them to the verbs corpus $\mathbf{V}_B$ while all of the tokens were added to the vocabulary corpus $\mathbf{V}_C$. Each word is then either lemmatized (stemmed) in order to get the root form of the inflected words (e.g. read instead of reading) before adding to the vocabulary and verbs corpuses, or it could be added in its infected form directly depending on the experiment. Words are then represented using a one-hot encoded vector of size $|\mathbf{V}_C|$ if the word is a part of the sentence input vector, and a one-hot encoded vector of size $|\mathbf{V}_B|$ if it is the target verb.

A moving window of size *w* was swiped across each sentence to form several sets of context vectors and target verbs. The encoded context vector was constructed by adding the one-hot encoded vectors (element-wise addition) of the words in the window except for the target verb. When having multiple verbs in a window, the target verb was chosen randomly.

4.2 Model Building

The feed-forward CBOW model was made of a fully connected layer of dimensions $|\mathbf{V}_C| \times 500$, followed by a dropout layer with $probability = 0.2$ which is followed by another fully connected layer of dimensions $500 \times |\mathbf{V}_B|$, the last layer of the network was a log-softmax layer that outputs log-probabilities across all the possible target dimensions.

$$-\log p(t|C) = -\log \mathit{Softmax}(A_2(\mathbf{L}_1 + b_2) \quad (1)$$

where

$$\mathbf{L}_1 = A_1(\sum_{w \in C} \mathbf{v}_w) + b_1)$$

Equation 1 shows the main objective of the CBOW model, where C is the input set of context words, t is the target verb to be predicted. $\mathbf{v}_w$ is the one-hot encoded vector of word w. A_1, b_1 and A_2, b_2 are the parameters of the model's fully connected layer 1 and layer 2, respectively.

The CBOW attempts to minimize this objective function to maximize the probability of getting the target word *t* when shown the set of context words *C*, this was optimized using adaptive moment estimation (Adam optimizer (Kingma and Ba, 2014)). A well trained model produces a probability distribution across all verbs in the verbs corpus $\mathbf{V}_B$ where the most reasonable verbs having high probabilities and the least reasonable verbs having probabilities close to zero.

4.3 Experiments

For each one of the BBC news category, the data instances were constructed as specified in 4.1, in addition to a final experiment with all categories combined together.

A K-fold cross validation process was then made by splitting the data to train-valid-test sets K-times so that all folds are used for training and testing in order to have better model performance estimation. In each fold of the cross validation process, the training set was used in the optimization process of the model parameters while the validation set is used to evaluate the model at each epoch and allow for saving the model when the validation loss is at its minimum (as an early stopping technique), then the testing set was used to evaluate the saved model after the training epochs are done in that cross validation iteration.

When training the model, the size of batches used was 512, and the learning rate used to update the model parameters was set to 0.001, each model was then trained for 20 epochs (it was observed that the model starts to over-fit afterwards in all experiment trials).

As a measure of accuracy of the predicted probabilities across the verbs in the verbs corpus $\mathbf{V}_B$, the testing phase of the cross validation process consists of counting the number of times the true target verbs were found in the top 5 and top 10 most probable verbs suggested by the model then divided by the total number of testing instances in that experiment trial.

5 Results

The following sub-sections show the results obtained when training and evaluating the model using the process described in section 4.3, those results are then discussed in section 6.

5.1 Learning Verbs in Business Articles:

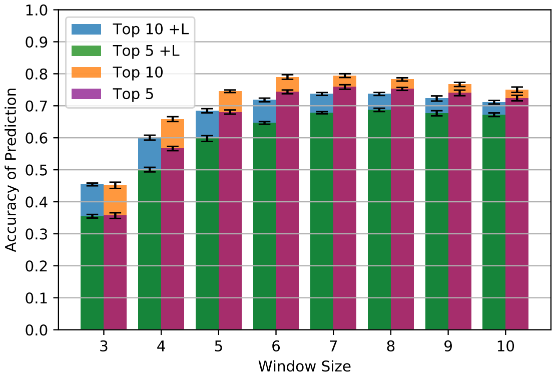

Figure 1: The accuracy of predicting the target verb within the top 5 and top 10 probabilities as the window size ranges from 3 to 10, +L indicates that the words were lemmatized in the trials.

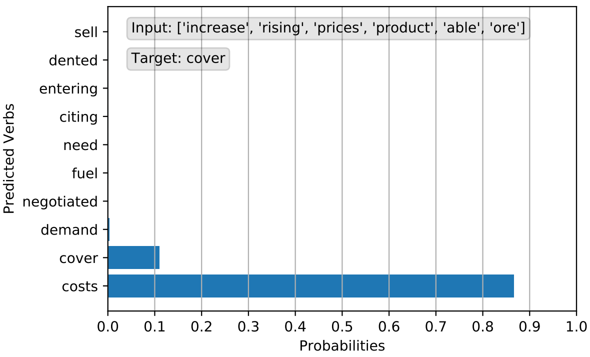

Figure 2: A sample of the top 10 predicted verbs and their probabilities produced by the model when trained using business articles with window = 7 and no lemmatisation.

5.2 Learning Verbs in Entertainment Articles:

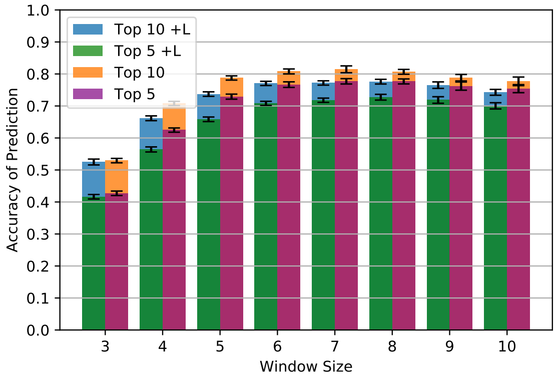

Figure 3: The accuracy of predicting the target verb within the top 5 and top 10 probabilities as the window size ranges from 3 to 10, +L indicates that the words were lemmatized in the trials.

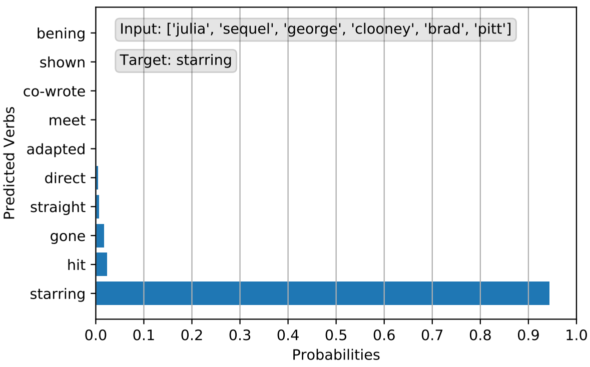

Figure 4: A sample of the top 10 predicted verbs and their probabilities produced by the model when trained using entertainment articles with window = 7 and no lemmatisation.

5.3 Learning Verbs in Politics Articles:

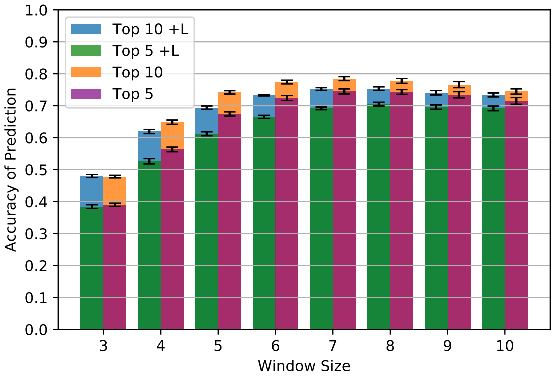

Figure 5: The accuracy of predicting the target verb within the top 5 and top 10 probabilities as the window size ranges from 3 to 10, +L indicates that the words were lemmatized in the trials.

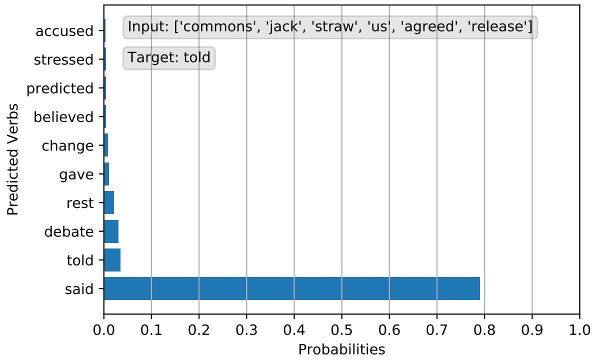

Figure 6: A sample of the top 10 predicted verbs and their probabilities produced by the model when trained using politics articles with window = 7 and no lemmatisation.

5.4 Learning Verbs in Sport Articles:

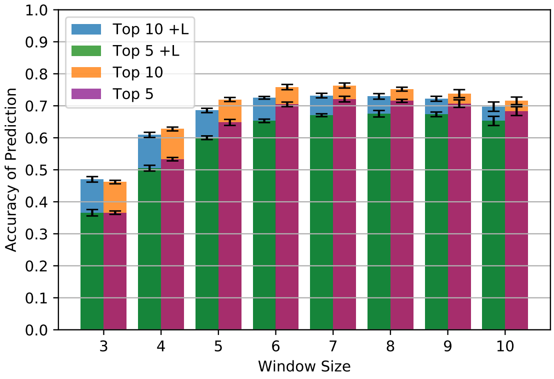

Figure 7: The accuracy of predicting the target verb within the top 5 and top 10 probabilities as the window size ranges from 3 to 10, +L indicates that the words were lemmatized in the trials.

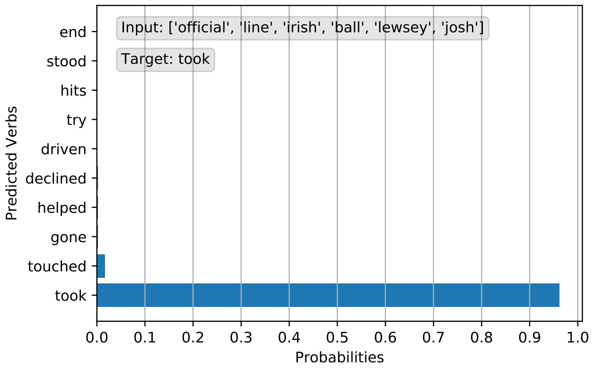

Figure 8: A sample of the top 10 predicted verbs and their probabilities produced by the model when trained using sports articles with window = 7 and no lemmatisation.

5.5 Learning Verbs in Technology Articles:

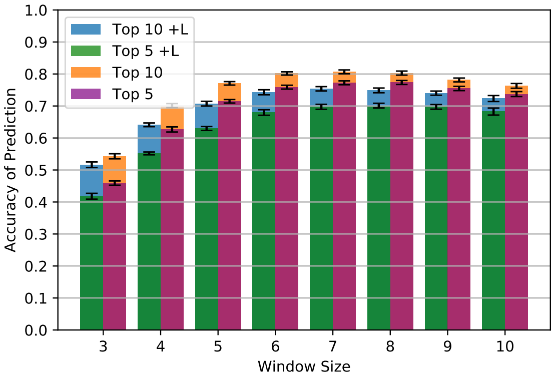

Figure 9: The accuracy of predicting the target verb within the top 5 and top 10 probabilities as the window size ranges from 3 to 10, +L indicates that the words were lemmatized in the trials.

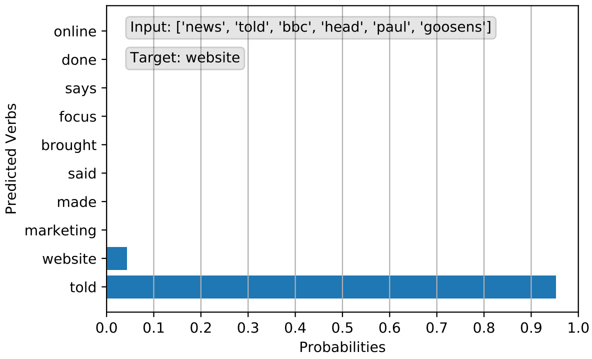

Figure 10: A sample of the top 10 predicted verbs and their probabilities produced by the model when trained using technology articles with window = 7 and no lemmatisation.

6 Findings

Figures 1, 3, 5, 7 and 9 show bar plots of the accuracy of predictions measured by the existence of the target verb in the top 5 and top 10 predicted probabilities when words were lemmatized (+L) and when they were used as they appeared in the text, those accuracy measures were recorded as the window size ranged from 3 to 10. It was found that generally the accuracy increases as the window size gets wider up to a window size of 8 then it drops down slightly since further away words were captured and included in the context input while they might have no relation to the target verb, a window size in the range 6 to 8 was found to produce the best accuracies with and without lemmatization (top 10 $\approx 75\% - 80\%$ and top 5 $\approx 67\% - 77\%$). Another observation can be that the gab between the top 5 and top 10 accuracies was reduces as the window size increases. On the other hand, lemmatization was found to hurt the accuracy of the predicted probabilities especially in the range 5-10, this can be due to the fact that when words are stemmed to their base form they appear in a variety of contexts which makes it harder for the model to learn general representations of the context and produce good probability distributions across the verbs.

Figures 2, 4, 6, 8 and 10 show the probabilities of the top 10 predicted words when the models were introduces to a sample of input context words in each of the five categories, it can be observed that the models were able to assign higher probabilities to verbs that can fit somewhere within the input words, however the predicted words were not linguistically very similar in most cases, this is due to the internal representation of the words being entirely learnt from the articles within that particular category, therefore the linguistic properties and semantics of verbs were not captured efficiently by the models.

7 Conclusion and Future Work

7.1 Conclusion

Building a language model that is able to learn a good internal representation of text which allows for producing reliable results is a challenging task. When building a language model for the task of predicting words with good accuracy the task becomes harder and therefore requires a complex model architecture, narrowing down the scope of predictions could allow for using simpler models such as a feed-forward CBOW model.

In this work, a simple two-layer feed-forward CBOW model was trained to predict verbs given sets of context words extracted from 5 types of BBC news articles: business, entertainment, politics, sports and technology, the number of context words used as input was determined using a sliding window with variable size, the words used to train and to be predicted by the model were either lemmatized (stemmed) or used as they appear in the text for examination.

The key findings of the results (section 5) of this work were that in all 5 news categories; increasing the context window size tends to increase the ability of the model to learn better associations that generalizes well to unseen data, and therefore assign higher probabilities to the true target verbs. It also shows that lemmatization actually slightly decreases the qualities of the predictions despite it shrinking down the target space.

7.2 Future Work

When trying to learn associations between words, word embeddings can be used as inputs to the CBOW model in order to increase the quality of the probability distribution produces by the model. A more complex model architecture such as bidirectional LSTMs or transformers can also increase the learning abilities of the model due to the attention mechanisms and the deeper structures of the networks.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.*

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association.*