Comparison of Traditional vs Context Sensitive Embeddings for the Task of Text Summarization

Hari Vidharth
s4031180 and
Nadir Al Hamzi
s4180216
and Ashwin Vaidya
s3911888

Abstract

In this project, we present three encoder-decoder models which use different embeddings to establish whether contextual embeddings benefit over traditional embeddings. We compare the performance of these models on the task of text summarization and present the results. We experimented on the WikiHow dataset and used rouge scores as the evaluation metric. We observe that contextual embeddings do not show an increased performance over traditional embeddings.

1 Introduction

There is a growing need for summarization as the pace of information generation has increased. Doing this at scale requires systems which can successfully condense the texts with reasonable accuracy. Previous research used models with embedding layers to represent sentences during training and inference. Traditional approaches of embedding the sentences involve using Word2Vec (Mikolov et al. (2013)) and GloVe (Pennington et al. (2014)). However, newer embeddings such as BERT (Devlin et al. (2018)) and ELMO (Joshi et al. (2018)) which generate embeddings based on the context the words have been since used for various language modelling tasks such as question-answering.

Our project aims to test whether contextual embeddings offer an advantage over traditional embeddings in the task of summarization when used with an encoder-decoder model. The three embeddings used here are Word2Vec, GloVe and BERT. We conduct experiments and present the results in the Result section and interpret them in the Discussion section.

Our code is available on Github `https://github.com/ashwinvaidya17/LTP_2020_RUG`

2 Related Work

Recurrent neural networks are widely used in tasks with sequential data and have proven to be very especially effective in the field of natural language processing. A recent study (Ramesh Nallapati, 2016) proposed novel architectures for abstractive text summarization which achieves the state of the art results. The architectures are based on the encoder-decoder model with an attention mechanism. These models are designed to capture the hierarchy of sentence-word structure and to remove words that are unseen during training. Another paper by (Tian Shi, 2016), provided a comprehensive survey on the recent advances of sequence to sequence models and the common challenges with recurrent neural network architectures. This study primarily focuses on the challenges associated with model parameter inference mechanisms and summary generation procedures. The authors proposed a convolution sequence to sequence model to overcome the limitation of traditional recurrent neural networks with long sequences. Most studies, however, show that encoder-decoder based models are so far dominating the text summarization architectures.

3 Model

All three models share a similar base architecture. We used a seq2seq encoder-decoder LSTM with a global attention mechanism. The encoder contains three LSTMs on top of ane embedding layer. The decoder contains only a single LSTM layer followed by an attention layer and a dense layer. Each of the LSTMs contains 500 hidden units. We used the dropout of 0.2 for the encoder along with

the dropout of 0.1 for the decoder. Early stopping is used on the validation set with the patience of 2. The individual architectures differ only in terms of the embeddings used and the hyperparameter values.

As Keras does not have an inbuilt attention layer, we used the custom attention layer by Ganegedara (2020).

3.1 Word2Vec Embeddings

Word2Vec is an embedding model that uses a skip-gram approach to train a classifier on a large lexicon to compute the embeddings vectors for words. The intuition behind word2vec algorithm is that the meaning of a word can be inferred by its co-occurrences. For a given target word, the embedding of the target word is computed by training a classifier to distinguish context words from non-context words for a given target. The learned regression weights are the embedding of the target word.

3.2 GloVe Embeddings

Global Vectors for Word Representation (GloVe) Pennington et al. (2014) is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

There are many different pre-trained GloVe models available, but for this project, we choose the model trained on the Wikipedia 2014 + Gigaword 5 corpus. It contains 6B tokens, 400K vocab, uncased, 50d, 100d, 200d, 300d vectors, and for our project, we choose the vector dimension size of 300.

3.3 BERT Embeddings

The BERT model is similar to the previous two models except that it uses the BERT embeddings only in the encoder and has a trainable embedding layer in the decoder. It takes three inputs which are specific to the BERT layer and represent the input sentences. It uses sub-words tokenization of the input sentence. This input is converted into a list containing, token id's, sentence mask and sentence segments.

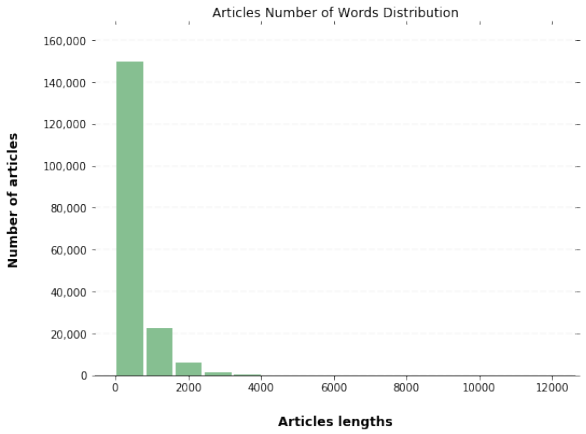

Figure 1: Number of words distribution

4 Experiments

Data The WikiHow dataset is used for the experiments. It contains 230,000 article and title pairs. Each article consists of multiple paragraphs. The paragraphs are combined to form the articles. Only the articles with valid summaries are used in this experiment. The resulting dataset contains 180,128 long-sequence pairs. The data is summarised by different people which introduces inconsistency across the summaries. However, for this experiment, we used the titles as the summaries for the articles.Sample title and article pairs are mentioned below.

How to Feed Hamsters *Most of a hamster's diet should be made up of commercial hamster food. These are usually composed either of pellets or seed mixes. They are designed to accommodate your hamster's diverse nutritional needs.*

How to Ask Your Crush for Their Cell Phone Number *Talk about something he likes - TV shows, movies, sports, his hobbies, etc. Make sure to be calm and casual, yet still, appear interested.*

	Articles	Summaries
count	180,128	B
mean	500.4	6.9
std	481.35	2.0
min	1	1
25%	210	5
50%	353.0	7.0
75%	604.0	8.0
max	12091.0	26

Table 1: Properties of the dataset used in the experiment.

Pre-processing Text collected from reviews or summaries usually contains irrelevant features which act as noise. Punctuation, numbers, dates and symbols can be considered noise and must be removed while preserving any useful features in the data. The end goal of preprocessing is to provide a clean representation that can be used by the model to learn the given task. The first step in our preprocessing is to split the sentences in each article and summary into individual words or tokens.

Text Normalization Another step for preprocessing is to discard articles that have less text than the summary or have no summary at all. Only the articles with summaries that are less than 75% of the article's length are considered valid article-summary pairs. This ensures there is always enough information for the decoder to use during the learning phase. The final total number of articles that are valid for training is 180128. The normalization pipeline is described in details below.

Removing of stop words and punctuation The articles and titles are preprocessed to remove any punctuation, symbols and numbers that are not relevant for our task. Stop words were only removed from the articles but not from the titles. The dataset is mainly written by a different number of people and actively being updated. Thus the use of punctuation may not be consistent across the dataset and hence can be removed. For stop words removal we used English stop words corpus provided by natural language toolkit or in short nltk.

Lemmatization The text is normalized using WordNetLemmatizer. Lemmatization transforms the word into its root. This further reduces the number of unique words. Different words like drank, drink, drinks share the lemma drink. Table 2 shows a sentence of an article before and after preprocessing.

Contractions Mapping Contractions are a shortened version of words or syllables. We created a contraction map to expand the words, for example, "ain't": "is not", "aren't": "are not" etc. Converting each contraction to its expanded original form helps with text standardization.

Experiment Details The baseline, word2vec and the GloVe model use the pre-processing steps mentioned above. The embedding weight matrix is used in both the encoder and decoder layer, except for the baseline model which does not have any pre-trained embeddings. The text size was set to 100 words and the summaries size was set to 10 words.

Original Raw Text	Preprocessed Text
One way to keep your hamster's cage smelling fresh is to train it to use a litter box	one way keep hamster cage smelling fresh train use litter box

Table 2: Original text vs processed text

Experiment Details: BERT Model We experimented on BERT embeddings by introducing the embeddings to both the encoder and decoder parts of the model. When using these on both the stages, the model failed to converge. Thus, BERT embeddings are used only in the encoder.

We also tested the performance on the model when the stopwords were not removed before training. As expected to remove the stopwords gave better results.

Evaluation We used word2vec as our baseline model. The base model used an embedding layer initialised to word2vec weight matrix. Then we compare Glove embedding which is not context-sensitive against BERT's contextual embeddings. We then use the Rouge score to evaluate the performance of the models and make the final comparisons. The reason we chose to work with Rouge score is that Rouge measure recall, it has more focus on context and word matches, word and sentence fluency. Hence it is more suited for summarization tasks.

5 Results

Word2Vec Results

Word2Vec model generated acceptable summaries but in many examples, the predicted words do not fit well with the input context. This was expected as word2vec is not context-sensitive.

	ROUGE-1	ROUGE-2	ROUGE-L
F	0.315	0.164	0.312
P	0.325	0.168	0.321
R	0.310	0.163	0.307

Table 3: ROUGE scores for Word2Vec model

Sample outputs: Ground Truth: *how to tutor kid*
Predicted: *how to get a good grade at school*
Ground Truth: *how to make enchilada*
Predicted: *how to make a shrimp and vinegar sauce*
Ground Truth: *how to politely stop being friend with someone*
Predicted: *how to deal with a bad friendship online*

GloVe Results

The GloVe base model was able to generate decent summaries, as shown below. In some of the sample outputs, it was able to predict the same text as the original summary and in the other cases, it was able to understand the overall meaning and predict synonyms or words with similar meanings. However, in the rest of the cases, it was unable to get the related meaning of the document by mispredicting one word which changes the meaning of the summary entirely. This is evident by the scores as shown in 4.

Sample outputs:

Original summary: *how to log out of other devices on instagram on android*
Predicted summary: *how to log out of other devices on instagram on android*
Original summary: *how to jailbreak an ipad*
Predicted summary: *how to jailbreak an ipad*
Original summary: *how to make giant cookie*
Predicted summary: *how to make chocolate chip cookie*

GloVe SCORES	ROUGE-1	ROUGE-2	ROUGE-l
P	0.5209	0.3524	0.5262
R	0.4799	0.3164	0.4814
F	0.4828	0.3164	0.4868

Table 4: ROUGE scores for GloVe model

Results for BERT

Sample outputs Original summary *how to stop racing thoughts in the middle of night*
Predicted summary *how to stop anxiety at night*
Original summary *how to get your kids to do their homework*
Predicted summary *how to teach kids to be productive*
Original summary *how to avoid an pylori bacterial infection*
Predicted summary *how to prevent bacterial infection*

	ROUGE-1	**ROUGE-2**	**ROUGE-L**
F	0.41	0.26	0.41
P	0.43	0.27	0.43
R	0.41	0.26	0.41

Table 5: ROUGE scores for BERT model

We calculate the unigram, bigram and the longest subsequence scores for each of the models. For each of these, F1, Precision and Recall are calculated. We discuss these results in the following section.

6 Discussion

We first establish a baseline Word2Vec model and present its results in Table 3. We then implement the GloVe model. As expected, it gives better results as tabulated in Table 4. However, it can be seen from Table 5, using contextual embeddings lead to lower performance.

We hypothesised that contextual embeddings would give better summarization results as the embeddings are extracted from the sentence thus eliminating the ambiguity in certain words. Additionally, recent work in language modelling has shown that contextual embeddings result in better performances in various tasks. However, our experiments show that contextual embeddings lead to lower scores.

We believe that the differences in results can be attributed to three main factors. The first factor is that summaries are subjective. The evaluating metrics like rouge use n-gram matches to score sentences. However, the results produced from a text can have multiple summaries which capture the same meaning without using any common words. One can argue that *how to get your kids to do their homework* has the same meaning when compared to *how to teach your kids to be productive*. When only the paragraph is given, the predicted sentence also seems plausible. Moreover, the answer to the question *how to stop racing thoughts in the middle of night*, can also lead to one inferring *how to stop anxiety at night* as the solution is the same. This also underscores the challenge of producing summaries as the design

of a metric also affects how accurately different models are compared.

Furthermore, we believe that the second factor responsible for GloVe having a better score is that as synonyms are used in similar contexts, word embeddings make a more suitable choice when it comes to summarization as summaries usually include synonyms. However, further experiments need to be conducted to establish this.

Finally, the success of BERT also depends on the fact that the original paper used Transformers for language modelling. In our experiment, we are using LSTMs on top of BERT inputs. Adding to this fact, our model converged only when BERT embedding was limited to the encoder. Another support to this comes from a recent work by Liu and Lapata (2019) which used BERT for summarization and involved a similar encoder-decoder architecture. However, they used Transformer stack for their decoder after extracting BERT embeddings. Thus we feel that while our model could have benefited by using Transformers, our experiment on comparing just the inclusion of contextual embeddings instead of traditional embeddings serves its intended purpose and we found that contextual embeddings confer no benefit.

7 Conclusions and Future Work

To conclude, we compare context-sensitive models to traditional models and look at their performance in the task of text summarization. From the results, we can see that the GloVe model performs the best out of all the other models. One way the results of GloVe could be improved further is to use Bidirectional LSTM instead of the regular LSTM. This has been shown to increase the performance of the model. The current pre-trained GloVe embeddings are 6B tokens, 400K vocab but, there exist other pre-trained GloVe embeddings with 42B tokens, 1.9M vocab and 840B tokens, 2.2M, using these embeddings might also add on and increase the model performance further. Additionally, we would like to experiment with Transformer layers in our architecture to measure their performance.

Appendix

Contribution of Nadir Al Hamzi, Project: Prepossessing of data, implementation of word2vec. Report sections: Related work, Dataset section, Preprocessing, Removing of stop words and punctuation, text normalization, lemmatization, relevant parts of word2vec.

Hari Vidharth, Project: Implementing the GloVe model with its related preprocessing. Report: Discussions, GloVe results, GloVe Model and embeddings preprocessing contractions mapping, model architecture, experiment details and evaluation, conclusion and future work.

Ashwin Vaidya, Project: Implementing the BERT model along with the required BERT specific preprocessing. Report: Discussions, BERT results, BERT Model and experiment details, abstract and introduction.

References

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

Ganegedara, T. (2020). Keras layer implementation of attention. `https://github.com/thushv89/attention_keras/blob/master/src/layers/attention.py`.

Joshi, V., M. Peters, and M. Hopkins (2018). Extending a parser to distant domains using a few dozen partially annotated examples.

Liu, Y. and M. Lapata (2019). Text summarization with pretrained encoders.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space.

Pennington, J., R. Socher, and C. D. Manning (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Ramesh Nallapati, Bowen Zhou, C. d. S. G. u. B. X. (2016). abstractive Text Summarization using Sequence to sequence RNNs and Beyond. In *SecarXiv:1602.06023v5*, pp. 1–12.

Tian Shi, Yaser Keneshloo, N. R. C. K. R. (2016). Neural Abstractive Text Summarization with Sequence to Sequence Models. In *arXiv:1812.02303v3*, pp. 1–12.