An Efficient Grammatical and Semantic Sentence Embeddings Framework

Huy Thanh Vu, Thu Thi Nguyen, Supreeth Narasimhaswamy
SBU IDs: 112276108, 112344010, 112007225

Abstract

In this project we propose a method for sentence embeddings that can capture both semantic and grammatical information of a sentence in a computationally efficient way. Our project is mainly inspired by the work '*Efficient Framework for Learning Sentence Representations*' (also known as *Quick-Thought*) (5). This original paper proposes a method to reduce computational cost of Skip-Thought (4) model but still outperform Skip-Thought on semantic downstream tasks. However, one problem with this Quick-Thought model is that it can only capture a sentence's semantics but not its grammatical information. We address this problem by adding a grammatical learning task while still keeping the model computationally efficient. The experiments shows that our method yield better results than the Quick-Thought model, thanks to the encoder's ability to capture richer information about sentences. In short, our model tries to get the best of both Skip-Thought and Quick-Thought method, learning both semantic and grammatical information while keeping the model's complexity minimum.

1 Introduction

Sentence embedding is the task of mapping a sentence to a vector of real numbers such that the resulting vector captures as much as possible information about the sentence. We can evaluate the learned sentence representations on downstream tasks such as movie-review classification, paraphrase detection, etc. A good set of sentence representations are expected to perform well on these downstream tasks.

The method of learning sentence representations can be broadly classified into two categories: *supervised*, and *unsupervised* methods. Supervised methods require labelled datasets such as dataset containing sentences translated from one language to another, sentiment labels of paragraphs, etc. In supervised methods, typically there is an encoder whose sentence representations are trained to solve these specific tasks such as machine translation, sentiment classification, etc. Unsupervised learning allows us to learn useful representations from large unlabelled corpora. These are encoder-decoder models that learn to predict/reconstruct the context sentences for a given sentence. Some of the popular work in these areas include Skip-Thought (4), FastSent (3), CNN model (1), and Quick-Thought (5).

Despite the success of above methods, several modelling issues exist. The models such as Skip-Thoughts (4), FastSent (3), CNN model (1) are trained to reconstruct the surface form of a sentence. Therefore the output layer size is the size of the entire vocabulary, which can easily be as large as 50,000 words. This leads to very high computational cost. The Quick-Thought (5) model successfully solves this problem by transforming the Seq2Seq reconstruction tasks to a classification task with much less number of parameters but still tries to learn semantic information of a sentence. This method, however, still has a drawback. It fails to capture the grammatical aspects of sentence.

We observe that the Quick-Thought model could be improved significantly if we augment it with an additional task of learning the grammatical structure of a sentence. Therefore, in this project, we propose adding to the original model the task of predicting the syntactic structure of sentences. In particular, we predict the parts-of-speech tags of words in the sentence. Note that this leads to a model with slightly larger number of parameters than the Quick-Thought model. But the number of parameters will still be much smaller than the Skip-Thought model. The computational efficiency of our model is discussed next sections.

To evaluate our idea, we train the model with UMBC webbase corpus, which contains 129 millions ordered sentences. However, due to time constraints and limited computational resources , we compare our results with the results of Quick-Thought model based on a smaller subset of the data. We extract 30 million sentences to train and test for both of the models. This methodology provides an insight about how the two models behave in the similar circumstances.

In summary, in this project we propose a model that tries to get the best of both Skip-Thought and Quick-Thought method, learning both semantic and grammatical information while keeping the models complexity minimum.

2 Related Work

There are many approaches to learning sentence embeddings. One of the most popular and effective method is called Skip-Thought (4). This method uses a Seq2Seq model to reconstruct adjacent sentences from an input sentence. Although this method helps the encoder to to capture both semantic and grammatical information of the sentence, it has very high computational cost. This is because to reconstruct a sentence, the output layer must have the size of the entire vocabulary. Quick-Thought model (5), which is our baseline model, presented below addresses this problem of heavy computation. It also outperforms other methods, and is considered state-of-the-art.

2.1 Quick-Thought Model

We use the Quick-Thought model proposed by Logeswaran *et al.* (5), as our baseline model. This model is inspired by Skip-gram approach of (6). The key idea of the Quick-Thought is that it transforms the reconstruction task of Skip-Thought into a classification task which drastically reduces the computational cost. Given an input sentence and a set of candidate sentences, (this set of candidate sentences contain one ground truth context sentence of the input sentence and many other noisy irrelevant sentences), the model is trained to choose the correct target sentence. The architecture of the Quick-Thought model is given in Figure 1a.

Let s be any given sentence. Let f and g be parameterized functions that take a sentence as input and encode them into fixed length vector. Let S_{ctxt} be the set of sentences appearing in the context of s (for some context size). Let S_{cand} be the set of candidate sentences considered for a given context sentence $s_{ctxt} \in S_{ctxt}$.

For a given sentence position in the context of s, the probability that a candidate sentence $s_{cand} \in S_{cand}$ is the correct sentence for that position is given by

$$p(s_{cand} \mid s, S_{cand}) = \frac{\exp[c(f(s), g(s_{cand}))]}{\sum_{s' \in S_{cand}} \exp[c(f(s), g(s'))]} \tag{1}$$

where c is a scoring function/ classifier.

The training objective minimizes the negative log probability of identifying the correct context sentences for each sentence in the training data D:

$$\mathcal{L} = -\sum_{s \in D} \sum_{s_{ctxt} \in S_{ctxt}} \log p(s_{ctxt} \mid s, S_{cand}) \tag{2}$$

We can see that this learning task helps the encoder learn semantic information of the sentence but it does not require to reconstruct the whole sentence like the Skip-Thought method. The size of the output layer therefore is very small(much less than Skip-Thought's output layer). Hence saves significantly computational cost.

2.2 The Issues

One of the key issues with the Quick-Thought model is that by predicting the context sentences, the model can only capture semantics but fails to capture grammatical information of the sentences. Indeed, to point out the relevant context sentence, the encoder only need to focus on the sentence's semantics, it does not need to understand sentence's grammatical structure. In practice, however, having sentence representations with grammatical information is very useful, and such sentence representations can perform better in many downstream tasks. In this project, we want to address this problem of the Quick-Thought model.

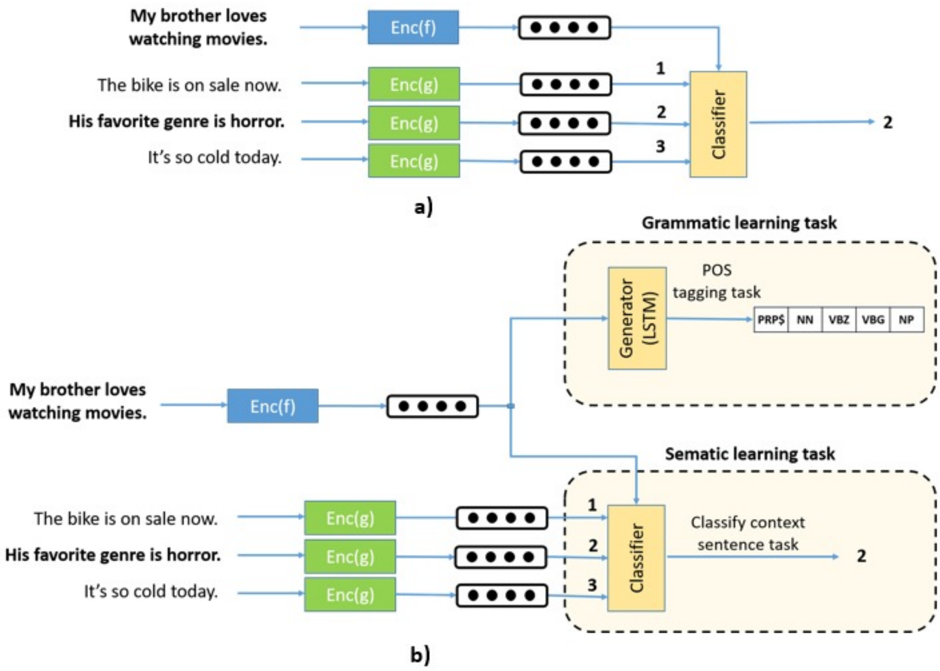

Figure 1. Architecture of Quick-Though model and our proposed model
a) Quick-Though model architecture
b) Our proposed model architecture

3 Proposed Method

We improve the existing Quick-Thought method by trying to capture the grammatical aspects of sentences alongside the contextual aspects. To do that, we augment the Quick-Thought model with an additional task of learning the grammatical aspects. More precisely, given a sentence s we train a model to predict the context sentences and the parts-of-speech tags for words in s. By learning to predict the parts-of-speech, we hope that the model learns some grammatical aspects of the sentence.

The next section describes in detail our two learning tasks. We then show that adding this additional grammatical learning tasks only slightly increases the number of parameters but helps the encoder to learn much richer information about the sentence.

3.1 Semantic Learning Task

For the semantic learning task, we use the same approach as the Quick-Thought model. Let s be any given sentence. Let f and g be parameterized functions that take a sentence as input and encode them into fixed length vector. Let S_{ctxt} be the set of sentences appearing in the context of s (for some context size). Let S_{cand} be the set of candidate sentences considered for a given context sentence $s_{ctxt} \in S_{ctxt}$. We design our model to correctly choose out the correct context sentence from the candidate set.

For a given sentence position in the context of s, the probability that a candidate sentence $s_{cand} \in S_{cand}$ is the correct sentence for that position is given by

$$p(s_{cand} \mid s, S_{cand}) = \frac{\exp[c(f(s), g(s_{cand}))]}{\sum_{s' \in S_{cand}} \exp[c(f(s), g(s'))]} \tag{3}$$

where c is a scoring function.

The semantic loss is then given by

$$\mathcal{L}_{\text{semantic}} = -\sum_{s \in D} \sum_{s_{ctxt} \in S_{ctxt}} \log p(s_{ctxt} \mid s, S_{cand}) \tag{4}$$

3.2 Grammatical Learning Task

In the grammatical learning task, we design the model to predict the parts-of-speech tags for every word in a sentence. We use the decoder in

	SICK			MSRP		TREC	CR
	r	ρ	MSE	Acc	F1		
Quick-Thought model	0.76	0.70	0.41	0.71	0.80	0.74	0.75
Our model	0.76	0.71	0.42	0.73	0.81	0.77	0.75

Table 1: Comparison between Quick-Thought model and our proposed model on 4 downstream tasks

Seq2Seq model to generate a series of parts-of-speech tags for the input sentences. Notice that because the number of parts-of-speech tags is very small (less than 50), the size of output layer in this decoder is also very small when compared to the Seq2Seq model in the Skip-Thought (which has the size of the vocabulary). Therefore our model is computational efficient.

The loss function is then given by

$$\mathcal{L}_{\text{grammar}} = \sum_{s \in D} \text{cross entropy}[k(s), \mathcal{P}(s)] \quad (5)$$

where $k(s)$ is the parts-of-speech predicting by the model.

For generating the training data containing pairs $(s, \mathcal{P}(s))$ of sentences s and corresponding sequence of parts-of-speech $\mathcal{P}(s)$, we use existing parts-of-speech tag models (such as Stanford parts-of-speech Tagger with accuracy up to 95%) to predict the parts-of-speech for all sentences in our dataset D, and use them as the true labels for the parts-of-speech during training.

3.3 Total Loss

Having defined the semantic learning task and the grammatical learning task, the full learning task is to learn both of them simultaneously. More precisely, we train a model to minimize the weighted sum of these two losses:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{semantic}} + (1 - \alpha)\mathcal{L}_{\text{grammar}} \quad (6)$$

where $\alpha \in (0, 1)$ is a hyper-parameter which acts as a regularizer that balances the semantic and grammatical learning tasks.

3.4 Implementation Details

The architecture of our model is given in Figure 1b.

3.5 Number of Parameters

The Quick-Thought model has successfully reduces the number of parameters from 57 million of Skip-Thought model down to only 19 million. Because our model has the additional grammatical learning task, it has 22 million parameters. Although our model has just slightly more parameters than the Quick-Thought model, it has the advantage of capturing both semantic and grammatical information of a sentence. Furthermore, 22 million parameters is much less than the 57 million parameters of Skip-Thought model.

4 Evaluation

To evaluate our model, we first train two models (the Quick-Thought and our proposed model) using UMBC dataset. We then evaluate them on four downstream tasks presented below.

4.1 Dataset Details

We train our model and the Quick-Thought model on the UMBC webbase corpus. The UMBC corpus (2), a dataset of 100M web pages crawled from the internet, preprocessed and tokenized into paragraphs. The dataset has 129M sentences. To save the training time, however, we have used only 30M of the UMBC dataset to train our model.

Because the original model in (5) has been trained and tuned very well, we have not compared our model with the results in (5). Instead, due to the time constraints, we have trained a smaller model on a smaller dataset. Therefore, we trained both the original Quick-Thought model and our proposed model on this limited configuration to evaluate the performances of the two models.

4.2 Evaluation Measures

We evaluate our encoder on 4 semantic relating downstream tasks.

- SICK: Sentence relatedness score prediction
- CR: Movie review sentiment classification
- MSRP: Paraphrase identification

- TREC: Question classification task

A sentence encoder is considered good if it yields high accuracy on these 4 tasks.

4.3 Configurations

For the Quick-Thought model, we choose the encoder dimension to be *H=1000*, vocabulary size *V=50,000* and Glove embedding size *W=300*. The RNN cell used is GRU.

For our proposed model, we use the exact configuration of the above-described Quick-Thought model, but adding a decoder (LSTM cell) with hidden size *H'=1000*. The output layer size is *O=48* representing number of POS tags the model have to predict.

4.4 Results

Table 1 shows the results of our model against the results of Quick-Thought model on 4 downstream tasks SICK, MSRP, TREC, CR.

4.5 Analysis

We observe that our model performs as well as the Quick-Thought model in 2 tasks SICK and CR. For the other 2 tasks - MSRP and TREC, our model yield a 2-3% better. We believe that this is thanks to the ability of our encoder that can capture not only semantic but also grammatical information of the sentence.

In the future, we will test two models on probing tasks as well (such as object number prediction, word order analysis, top constituent prediction). We are pretty confident that our model will perform better than Quick-Thought model, thanks to the fact that our model is intentionally designed to learn sentence's grammatical information.

Caveat: The performance in table 1 is not as high as the the model in original paper (5) due to our limit of resources. In this experiment, we reduce the configuration of the original model as well as training dataset size to fit our resources.

4.6 Code

Google drive link:
https://drive.google.com/file/d/1nCjWJFxToXQoI3If-i032mzt7eXX344t/view?usp=sharing. Please also find our source code compressed and attached in the submitted package.

5 Conclusions

In this project, we have learned:

- Motivation of sentence embeddings and challenges we have to address.
- Main approaches to the problem of sentence embeddings.
- Understanding more thorouhg seq2seq model and how to apply it in leanring tasks.

References

[1] GAN, Z., PU, Y., HENAO, R., LI, C., HE, X., AND CARIN, L. Learning generic sentence representations using convolutional neural networks, 2016.

[2] HAN, L., KASHYAP, A. L., FININ, T., MAYFIELD, J., AND WEESE, J. Umbc_ebiquity-core: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics* (2013), Association for Computational Linguistics.

[3] HILL, F., CHO, K., AND KORHONEN, A. Learning distributed representations of sentences from unlabelled data. In *Proceedings of NAACL-HLT 2016* (2016), Association for Computational Linguistics, pp. 1367–1377.

[4] KIROS, R., ZHU, Y., SALAKHUTDINOV, R., ZEMEL, R. S., TORRALBA, A., URTASUN, R., AND FIDLER, S. Skip-thought vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2* (Cambridge, MA, USA, 2015), NIPS'15, MIT Press, pp. 3294–3302.

[5] LOGESWARAN, L., AND LEE, H. An efficient framework for learning sentence representations. In *International Conference on Learning Representations* (2018), ICLR.

[6] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G., AND DEAN, J. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (USA, 2013), NIPS'13, Curran Associates Inc., pp. 3111–3119.