A Simple and Efficient Framework for Sentence Similarity Measurement in Bengali Language

Maruf Ahmed Mridul
Department of Computer Science and Engineering
Shahjalal University of Science and Technology

Arnab Sen Sharma
Department of Computer Science and Engineering
Shahjalal University of Science and Technology

ABSTRACT

Sentence similarity measurement is a crucial task for the performance of several Natural Language Processing applications and it has received much attention mainly for English language. However, for low resource languages like Bengali, very few works have been done in this field. This article proposes a simple approach to measure sentence similarity score for low resource languages. Rather than relying on complex approaches that try to extract lexical information from text, here, semantic information using language-agnostic language models based on BERT is extracted. The variable length pairs of sentences are embedded into fixed length feature vectors using different language-agnostic BERT sentence encoders, then their differences are measured using some standard loss functions and finally the concatenated loss vectors are used to train a simple feed forward neural network to measure the similarity score between sentence pairs. The experiments show that this relatively simple approach gives satisfactory results when trained with Bengali sentence pairs. This approach requires almost no intricate pre-processing steps. Which means a similar architecture should work well for other low resources languages for which well performing stemmers, lemmatizers etc are scarce.

Keywords

Sentence Similarity, Feed Forward Neural Networks, Natural Language Processing, Sentence Transformers, Multilingual BERT

1. INTRODUCTION

In this new era of automation, understanding texts and retrieving information from them is a critical and fecund task for machines. A great deal of NLP tasks may be enhanced by the correct understanding of the semantic similarity between sentences or phrases. One of the examples of the applications of similarity measurement between a pair or sentences is to increase the efficiency of online Q&A. When a question is asked these sentence similarity measures allow computers to effectively scan through whether similar questions have been asked and answered before. Also, traditional plagiarism checkers usually do not consider paraphrases and only detect the exact text matches. If well-performing and efficient sentence similarity models can be implemented, it would be a great help for plagiarism checkers to detect paraphrases. Furthermore, graph-based text summarization also relies on similarity measures to weight edges.

To measure semantic similarity between sentences is not a trivial task because of the usage of ambiguous and variable linguistics to express the same idea. The more variations in structure/syntax of a language, the more difficult it gets to measure the similarities. This challenging nature attracted a lot of researchers to pay attention to it. And quite a lot of works have been done in this field for the last decade. The earlier works mainly focused on lexical similarities. But over the period, NLP-related works improved a lot and consequently, semantic similarity measures have been brought to light.

Although interests have grown in this field, the unfortunate fact is, most of the works done so far are for English language. Whereas low resource languages like Bengali has been paid little to no attention in this sector. A very few works have been done on sentence similarity measurement for Bengali language maybe because of its huge variety of expressions, intricate nature in syntax, and scarcity of data.

From the motivation of contributing in this little explored field in Bengali Language, in this work, a simple feed forward neural network classifier is proposed that leverages different language-agnostic BERT sentence encoders. Recently released multilingual language models are powerful for capturing the semantic information from words and sentence of different languages. These models are able to give reasonable performance with minimal pre-processing and have already achieved state-of-the-art performance in multiple language tasks; sometimes even achieving human-level performances. The simplicity of usage and effectiveness of such sentence level language encoders motivated us to apply them to calculate the similarity of Bengali sentence pairs.

Here, a supervised learning setup is considered where a pair of variable length sentences are transformed into fixed sized vectors using sentence embedding techniques and they are used as a training sample along with a label. Based on the assumption that the given labels reflect an underlying similarity measurement, the model tries to map variable length sentences of a general space into a structured metric space that can be applied to new examples not present in the dataset.

In this work, it is shown that, a simple Feed Forward Neural Network is capable to learn substantial semantics if it is trained with enough pair of sentences. This is also less susceptible to variations of syntax of sentences. The results surpass the earlier attempts for the same task in Bengali language by a good margin.

2. RELATED WORKS

Researchers started working on sentence similarity measurement more than fifteen years ago. During the early years, most of the works were based on lexical matching like word overlap measures, phrasal overlap measures, TF-IDF measures, etc.

Metzler et al. [11] proposed two baseline word overlap measures to compute the similarity between sentence pairs. The proportion of words that appear in both sentences normalized by the sentences length is defined as simple word overlap fraction, and the proportion of words that appear in both sentences weighted by their inverse document frequency is defined as IDF overlap. Banerjee and Pedersen [2] introduced a phrasal overlap measure that relies on the Zipfian relationship between the phrase lengths and their frequencies in a text. The fact behind their motivation is that the traditional word overlap measures simply treat sentences as a bag of words, not considering differences between single word and multi-word phrases.
Allan et al. [1] proposed simple TF-IDF measure to detect topically similar sentences in TREC novelty track experiment. Another variation of TF-IDF similarity measure is identity measure [5] which is for identifying plagiarized documents. The identity score is derived from the sum of inverse document frequency of the words that appear in both sentences normalized by the overall lengths of the sentences and the relative frequency of a word between the two sentences [12].

Later on, semantic measures got attention since only lexical features are not enough to measure similarity accurately because of the variation in sentence formation to express the same thing.
Li. et al. took into account the semantic information calculated from semantic nets like WordNet [8]. They also considered the word order information in the sentences. Liu et al. also considered the semantic information from WordNet and word order to calculate similarity. They added one more feature - the contribution of different parts of speech in a sentence [9]. They used Dynamic Time Warping as the distance measure which allows similar shapes to match even if they are out of phase.
Ming Che Lee did the analogous work taking advantage of corpus-based ontology that overcomes the problem with measuring similarity between irregular sentences [6] . His work applies to short, medium, and even long (more than 12 words) sentences while most of the other works focus on only short sentences. Lin Li et al. measured sentence similarity from four different aspects: Objects-Specified Similarity, Objects-Property Similarity, Objects-Behavior Similarity, and Overall Similarity to produce more reasonable results [7]. Sultan et al. calculated sentence similarity from word alignment and composition sentence vector that carries semantic information [15].

Recently, interests have shifted toward neural network approaches for most of the NLP tasks and sentence similarity measurement as well. Ferreira et al. assessed similarity taking into account all three of the lexical, syntactic, and semantic information [3]. They used a CNN-Corpus to evaluate the sentence similarity.
He et al. proposed a multi-perspective sentence similarity modeling with CNNs [4].
Mueller et al. proposed a very well-performing Siamese Recurrent Architectures for learning sentence similarity [13]. They trained Siamese LSTMs with MSE and L1 loss and outperform most of the previous works.

We found only two works available that incorporate Bengali language for this task.
Onkon et al. [14] tried to imitate the work of Ferreira et al. [3] for Bengali sentences, but the performance was not satisfactory.
And, Masum et al. focused on abstractive text summarization and they used sentence similarity measures to do so [10]. They measured sentence similarity by simply applying cosine similarity on vectors obtained from word2vec.

Scarcity of good datasets, lack of good embedders and complexity of the linguistics in Bengali language maybe are the reasons behind scanty work for Bengali Sentence Similarity.

3. DATASET

There is no available benchmarked dataset of similar/dissimilar tagged Bengali sentence pairs. So, This scarcity of data was needed to be handled somehow.

3.1 Dataset Preparation

There is a popular dataset called Flickr 8k Dataset [1] which is used for image captioning systems. The dataset contains $8,000$ images and five English captions per image.
The human-translated (to Bengali) version of this dataset was found. This translated data has not yet been made public by it's creators.
This image caption dataset is converted to a sentence similarity dataset for this work. There are $40,000$ captions in total in the dataset, where there are five captions for each image. The sentences were paired up to form the sentence similarity dataset. All the pairs formed from a cluster of five captions of an image are considered similar.
15 ($5C2 = 10$, and also considered pairing up each sentence with itself. So, $10+5 = 15$) pairs were generated from each five captions cluster.
In total, $120,000(8000 \times 15)$ sentence pairs were prepared that are called similar.

So, our definition of similar sentence pairs is - *if a pair of sentences describe the same scenario, no matter how differently the sentences are formed, they are called similar.*

Next comes the task of finding dissimilar pairs. In order to balance the dataset, around $120,000$ dissimilar pairs of sentences were supposed to be found. Since each sentence in the set of similar pairs has been paired with five sentences, just five dissimilar sentences for each of the sentences were looked for.
So, a total of $200,000(40,000 \times 5)$ dissimilar pairs were found.
Now, in general, *a pair of sentences are called dissimilar if they are the captions of two different images*. But, in the collection of 8,000 images, theres a high chance that there might be similar images. So, it was needed to skim through the formed pairs and discard the confusing ones, keeping only the original dissimilar pairs.
To balance the dataset, $130,000$ filtered dissimilar sentence pairs were included in the final dataset. *Figure 1* shows some samples of the dataset.

[1] https://www.kaggle.com/adityajn105/flickr8k/activity

So, the final dataset that is prepared to run the model on, contains around $250,000$ sentence pairs in total. *Figure 2* illustrates the distribution.

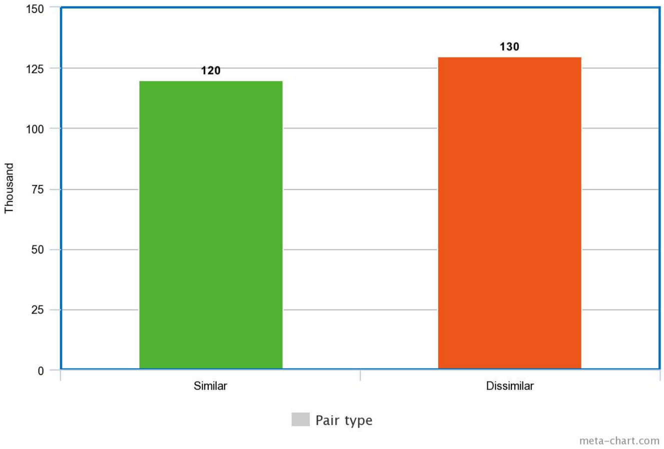

Fig. 2: **Amount of similar and dissimilar sentence pairs in the dataset**

3.2 Preprocessing

As part of the preprocessing, initially, just the punctuations were removed.

Then stopwords were removed from each sentence as in general they do not essentially contain semantic information.

But relying on the hunch that stopwords might play some roles in the measurement of similarity, to experiment the model with both stop-word removed and the original versions of the dataset, both were kept.
Figure 3 illustrates examples of sentences at each phase of the preprocessing.

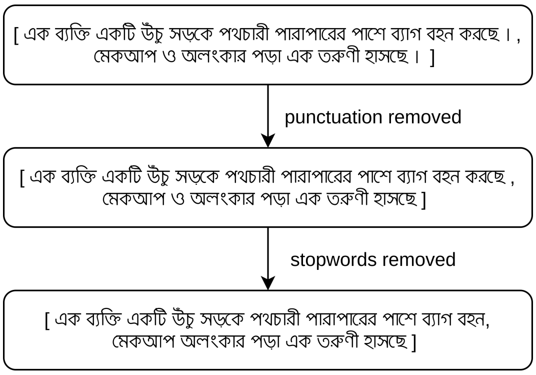
[এক ব্যক্তি একটি উঁচু সড়কে পথচারী পারাপারের পাশে ব্যাগ বহন করছে। , মেকআপ ও অলংকার পড়া এক তরুণী হাসছে।]
punctuation removed
[এক ব্যক্তি একটি উঁচু সড়কে পথচারী পারাপারের পাশে ব্যাগ বহন করছে , মেকআপ ও অলংকার পড়া এক তরুণী হাসছে]
stopwords removed
[এক ব্যক্তি একটি উঁচু সড়কে পথচারী পারাপারের পাশে ব্যাগ বহন, মেকআপ অলংকার পড়া এক তরুণী হাসছে]

Fig. 3: **Sentences after different steps of preprocessing**

Although stemming is an essential preprocessing step for most of the NLP tasks, in this case, the sentences were not stemmed. Because the whole sentences were embedded using pre-trained sentence embedding models. Since these models embed the whole sentence, changing the shapes of the words inside the sentence by stemming is not expected.

4. MODEL

The model utilizes different language-agnostic BERT sentence encoders. These sentence encoders are trained with huge amount of textual data from different languages to extract semantic information for language understanding given a sentence. It is expected that these powerful sentence embedders should encode similar sentences close to each other in a shared embedding space even if they are of different languages.

The model architecture is quite simple and straightforward. The dataset structure is depicted on Figure 1. For each row, the two sentences are simply fed to a multilingual sentence encoder. The encoder embeds these sentence in a high dimensional feature vector (the vector size varies from encoder to encoder). The similarity score of these feature vectors is calculated using 3 different similarity/loss measures.

4.1 Losses

4.1.1 Element-wise distance squared:. A loss vector is created by performing element-wise subtraction and then we square each element. The procedure is described below.
For demonstration purposes let us consider two very simple feature vectors, [1, 2, 3] and [3, 2, 1]. The following steps are performed for calculating the loss vector.

—Element-wise subtraction: After this step the vector looks like [-2, 0, 2].
—Element-wise square: After performing this step the vector is [4, 0, 4].

4.1.2 Element-squared distance:. This loss vector is created by calculating the element-wise squared distances. The calculation steps are shown below for the simple vectors [1, 2, 3] and [3, 2, 1]

—Element-wise square for both vectors: After performing this step 2 new squared vectors are found - [1, 4, 9] and [9, 4, 1]
—Element-wise subtraction: After performing this step the vector is [-8, 0, 8].

4.1.3 Cosine Similarity:. For two vector embedding A and B *Cosine similarity* is calculated using the following formula.

$$similarity(A,B) = \frac{A \cdot B}{||A|| \times ||B||} = \frac{\sum_{i=1}^{N} A_i \times B_i}{\sqrt{\sum_{i=1}^{N} A_i^2 \times} \sqrt{\sum_{i=1}^{N} B_i^2}}$$

4.2 Feature Vector

After calculating the loss values/vectors for each pair of sentences, they are just concatenated and use the resulting vector as the feature vector. The vector size will be $2 \times N + 1$, where N is the encoded vector size returned from the sentence encoder for each sentence.

4.3 Classifier Architecture

The feature vector that is formed with loss vectors is passed to a Feed Forward Neural Network classifier. Herem only 1 hidden layer is used. As the model is based upon the assumption that the sentence encoders will already encode similar sentences closer to each other in embedding space, it is decided that one hidden layer should be enough for this binary classification task. We used 100 neurons in that layer. This value was tuned with our validation dataset. We used a *Dropout* layer to randomly turn off 20% neurons in different batches of training phase to reduce overfitting. A detailed architecture of the model is depicted in Figure 4.

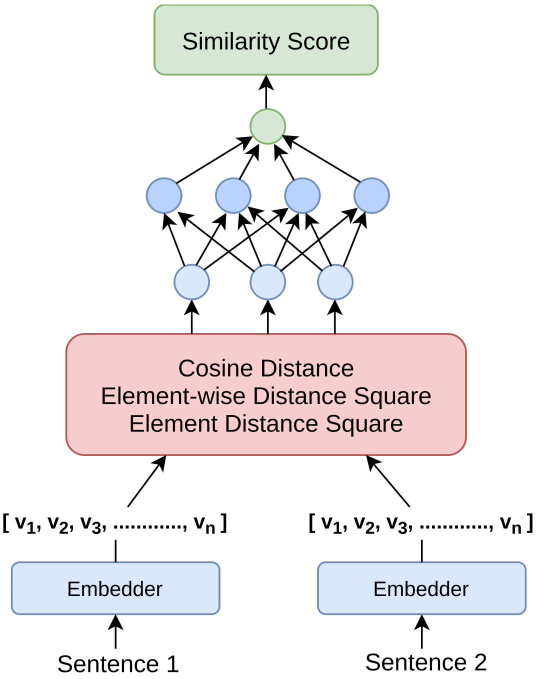
Similarity Score
Cosine Distance
Element-wise Distance Square
Element Distance Square
[v_1, v_2, v_3, ..., v_n]
[v_1, v_2, v_3, ..., v_n]
Embedder
Embedder
Sentence 1
Sentence 2

Fig. 4: **Model architecture**

4.4 Training and Testing

The data were split into 80% and 20% . The 20% data is for testing. And that 80% data were again split into 80% and 20%, using the larger portion for training and smaller portion for validation. Table 1 illustrates the amount of data using for training, validation and testing. The percentage is of the total data.

Table 1. : **Amount of data used for different purposes**

Purpose	**Percentage**	**Total Data**
Training	64%	160,000
Validation	16%	40,000
Testing	20%	50,000
Total	100%	250,000

—20 epochs were run for the training purposes.

—The model generates a similarity score in the range $[0, 1]$. But the label in the dataset is binary (1 for similar and 0 for dissimilar). So, to calculate the Test accuracy, a threshold is set that determines the binary result. A pair of sentences are said to be similar if the similarity score is over **0.6**, otherwise they are called dissimilar.

5. RESULTS AND DISCUSSION

The model was experimented with the feature vectors obtained from two different multilingual sentence encoders -

(1) paraphrase-multilingual-MiniLM-L12-v2 [2]

(2) paraphrase-multilingual-mpnet-base-v2 [2]

And, both versions of pre-processed dataset were used -

(1) Only punctuation removed sentences

(2) Punctuation and stopwords removed sentences

So, in total, there are four different combinations of feature vectors that are used for the training, validation and testing purposes.
The performances of the model for these combinations are shown in Table 2.
It can be observed that encoder(2) performed better in this particular task.
It can be inferred from the results that the removal of stopwords does not affect the result that much. But, a little proof that the previous hunch (stopwords might play some roles in the similarity measurement) is somewhat ***true*** can be seen from the results. Though the difference is very small, removal of stopwords yielded slightly worse results.

The model was tested using the test set of 20% (50,000 sentence pairs) of total data, that the model has never seen before.
The performance with some example pairs that are completely exclusive from the whole dataset was aslo observed.

Figure 5 shows five example sentence pairs that are used to test the model. First two of them are from the test set, and the last three are exclusive from the dataset.

Pair No	**SN**	**Sentence**
Pair 1	s1	একজন মহিলা গল্ফ খেলছেন (A woman is playing golf)
	s2	একটি রক আরোহী একটি শৈল আরোহণ প্রাচীর এর উপর অনুশীলন করে। (A rock climber practices on a rock climbing wall)
Pair 2	s3	দুটি কুকুর একটি ব্রীজের নিচে পানিতে খেলছে (Two dogs play in the water under a bridge)
	s4	দুটি কুকুর পানিতে খেলছে (A couple of dogs are playing in the water)
Pair 3	s5	মাঝি খুব ধীরে নৌকা চালাচ্ছে (The boatman is rowing his boat very slowly)
	s6	ছেলেটি সাঁতার কাটছে (The boy is swimming)
Pair 4	s5	মাঝি খুব ধীরে নৌকা চালাচ্ছে (The boatman is rowing his boat very slowly)
	s7	মাঝি জোরে নৌকা চালাচ্ছে (The boatman is rowing his boat fast)
Pair 5	s5	মাঝি খুব ধীরে নৌকা চালাচ্ছে (The boatman is rowing his boat very slowly)
	s8	মাঝি নৌকা চালাচ্ছে (The boatman is rowing his boat)

Fig. 5: **Example sentence pairs for testing the model**

[2] https://www.sbert.net/docs/pretrained_models.html

For the mentioned pairs, Table 3 depicts the **average** similarity scores of difference combinations of encoders and pre-processing techniques generated by the model.
It is clear that the model performed very well for *Pair 1* and *Pair 2*. *Pair 1* is clearly a pair of dissimilar sentences and *Pair 2* is consisted of similar sentences.

In the later three pairs, one sentence is fixed. This triplet was chosen to see how the model performs relatively.
For *Pair 3* the generated similarity score is 0.38, this means that they are dissimilar, but the score is not very close to 0. Here, the context is different in the sentences, but both the events are happening in water. Though not said explicitly, the model learned this underlying semantic meaning and yields a greater score.
Pair 4 and *Pair 5* are essentially similar pairs based on context. The generated scores also support this claim. The noticeable fact is, *Pair 5* has a greater score than *Pair 4*. Question should arise, why this is noticeable. In *Pair 5*, the first sentence has an adverb (very slowly) and the second sentence is the same without the adverb. On the other hand, In *Pair 4*, the first sentence is same as the first sentence of *Pair 5* and the second sentence also has an adverb (fast), which is the exact opposite of the adverb that the first sentence has. This means, though the context is similar, the opposite adverbs should decrease the similarity score a little bit, and that is what the model is doing.

Table 3. : **Similarity scores of the example sentence pairs**

Pair	Sentences	Similarity Score
Pair 1	(s1, s2)	0.07
Pair 2	(s3, s4)	0.98
Pair 3	(s5, s6)	0.38
Pair 4	(s5, s7)	0.89
Pair 5	(s5, s8)	0.93

The model is tested and observed with many such pairs outside from the dataset. In most of the cases, the model is able to maintain the expected relative discrepancy among the pairs.

However, there are some cases where the model does not seem to generate the expected similarity score. For example, if ***s4*** and ***s6*** are paired up, the generated score is ***0.08***. Based on the previously discussed examples, the score was supposed to be a bit higher. Because, both the incidents happen in water. So, it was expected that the model would find a little similarity between the sentences although the context is different. As per context, these two are completely different sentences. In that sense the score is satisfactory. But to generalize the performance on the relative scores of sentence pairs, the score should have been a little higher. However, these anomalies are scarcely found.

6. CONCLUSION AND FUTURE SCOPES

This article presents a simple Feed Forward Neural Network classifier that generates a similarity scores between a pair of sentences that are encoded by two different language-agnostic BERT sentence encoders. The architecture very well learns the semantic meaning of the sentences thus relating the similarities between them, and finally makes similarity prediction for unseen sentence pairs. The results outperform the very few works in the similar field for Bengali Language. For the dataset that is used, the results seem promising. If a well prepared benchmarked dataset could be used, it would have been more helpful to measure the performance of the proposed model. But, based on the satisfactory observations that is discussed on some examples earlier, it can be hoped that this method will guide the researchers in future to contribute more in this task.

A human annotated, balanced, good dataset is the demand of time for this specific task. Hopefully in near future, this scarcity will be lessened, and more elegant and efficient approaches will be proposed for the task. It would be great if this article helps to widen the door for the impending researches in this field.

7. REFERENCES

[1] James Allan, Courtney Wade, and Alvaro Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 314–321, 2003.

[2] Satanjeev Banerjee, Ted Pedersen, et al. Extended gloss overlaps as a measure of semantic relatedness. In *Ijcai*, volume 3, pages 805–810. Citeseer, 2003.

[3] Rafael Ferreira, Rafael Dueire Lins, Steven J Simske, Fred Freitas, and Marcelo Riss. Assessing sentence similarity through lexical, syntactic and semantic analysis. *Computer Speech & Language*, 39:1–28, 2016.

[4] Hua He, Kevin Gimpel, and Jimmy Lin. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1576–1586, 2015.

[5] Timothy C Hoad and Justin Zobel. Methods for identifying versioned and plagiarized documents. *Journal of the American society for information science and technology*, 54(3):203–215, 2003.

[6] Ming Che Lee. A novel sentence similarity measure for semantic-based expert systems. *Expert Systems with Applications*, 38(5):6392–6399, 2011.

[7] Lin Li, Xia Hu, Bi-Yun Hu, Jun Wang, and Yi-Ming Zhou. Measuring sentence similarity from different aspects. In *2009 international conference on machine learning and cybernetics*, volume 4, pages 2244–2249. IEEE, 2009.

[8] Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, 18(8):1138–1150, 2006.

[9] Xiaoying Liu, Yiming Zhou, and Ruoshi Zheng. Sentence similarity based on dynamic time warping. In *International Conference on Semantic Computing (ICSC 2007)*, pages 250–256. IEEE, 2007.

[10] Abu Kaisar Mohammad Masum, Sheikh Abujar, Raja Tariqul Hasan Tusher, Fahad Faisal, and Syed Akhter Hossain. Sentence similarity measurement for bengali abstractive text summarization. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–5. IEEE, 2019.

[11] Donald Metzler, Yaniv Bernstein, W Bruce Croft, Alistair Moffat, and Justin Zobel. Similarity measures for tracking information flow. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 517–524, 2005.

[12] Donald Metzler, Yaniv Bernstein, W Bruce Croft, Alistair Moffat, and Justin Zobel. Similarity measures for tracking information flow. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 517–524, 2005.

[13] Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[14] Abujar Onkon, Md Shahidul Islam, and Abu Abed Md Shohaeb. Assessing sentence similarity using lexical and semantic analysis for text summarization using neural network. *Assessing Sentence Similarity using Lexical and semantic Analysis for Text Summarization using Neural Network.*, 4(1):5–5, 2018.

[15] Md Arafat Sultan, Steven Bethard, and Tamara Sumner. Dls@ cu: Sentence similarity from word alignment and semantic vector composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 148–153, 2015.

Sentence 1	Sentence 2	Is Similar
একটি কালো কুকুর জলের মধ্যে লাফালাফি করছে (A black dog jumps through the water)	কালও কুকুর টি পানির মধ্যে দিয়ে দৌড়াচ্ছে । (A black dog is running through the water.)	1
স্বর্ণকেশী মেয়েটি একটি সাদা জামা পরে আছে । (The blonde girl is wearing a white dress.)	পাঁচজন একসাথে তুষারে বসে আছেন (Five people are sitting in the snow together)	0
এক ব্যক্তি একটি উঁচু সড়কে পথচারী পারাপারের পাশে ব্যাগ বহন করছে । (A man is carrying bags nearby an elevated road-way and crosswalk .)	একটি ধূসর পাখি একটি সৈকতে মহিমান্বিতভাবে দাঁড়িয়ে আছে যখন ঢেউ আছড়ে পরে (A grey bird stands majestically on a beach whilewaves roll in)	0
একটি শিশু বালক একদল মানুষের সামনে দৌড়াচ্ছে (A little boy is running towards a group of people)	ঘাসের উপর এক বাচ্চা দৌড়াচ্ছে (A little boy is running around in the grass)	1
একটি কুকুত রশির সাথে বাঁধা একটি প্রাণীকে ধাওয়া করছে (A dog chases a stuffed animal attached to a string)	একটি সাদা কুকুর সুতার ফাঁদে পড়া একটি পশুকে তাড়া করে (A white and black dog chases after a decoy-animal on a string)	1
মেকআপ ও অলংকার পড়া এক তরুণী হাসছে (A young girl with makeup and jewelry smiles .)	একটি বিশাল পাখি সমুদ্র সৈকতে জলে দাঁড়িয়ে আছে । (A large bird stands in the water on the beach .)	0

Fig. 1: **Example of sentence pairs with their English translation. *Is Similar* shows the similarity (1 for similar and 0 for dissimilar)**

Table 2. : **Performances of the model for different combinations of encoders and preprocessing**

Encoder	Preprocessing	Validation Acc.	Testing Acc.
paraphrase-multilingual-MiniLM-L12-v2	Only punctuation removed	84.98%	85.80%
paraphrase-multilingual-MiniLM-L12-v2	Punctuation and stopwords removed	84.20%	84.04%
paraphrase-multilingual-mpnet-base-v2	Only punctuation removed	90.93%	90.84%
paraphrase-multilingual-mpnet-base-v2	Punctuation and stopwords removed	89.70%	89.64%