Sentence-to-Sentence Semantic Similarity

Neelkumar P. Patel (1101357), Jay Sukhadiya (1110211), Priya Patel (1093483), Darshak Mota (1101352)
Department of Computer Science
Lakehead University
Thunder Bay, Ontario
Email: {npatel46, jsukhadi, patelp, dmota}@lakeheadu.ca

Abstract—**Semantic similarity of sentences is based on the meanings of the words and the syntax of the sentence. The semantic analysis plays an important role in the research related to text analytics. The sentences are preprocessed by performing tokenization, removal of stop-words and punctuations, and lemmatization. The similarity of sentences are calculated based on semantic vectors and word order vectors along with considering the Part-of-Speech (POS). The semantic vectors are computed using a Word2Vec model, which is trained and saved upon the corpus. The word order similarity is computed using word order vectors of both the sentences. Hence, the overall semantic similarity between sentences is the summation of the sentence similarity and word order similarity. The performance of the algorithm is evaluated against the SICK dataset which correctly predicts 81.30% of sentence pairs of SICK dataset.**

I. Introduction

Sentence is a sequence of words arranged in a combinational structure to convey a specific meaning. By using different combinations of words or phrases, the structure of the sentence can be altered while preserving the overall meaning. Such sentences are either similar or equivalent. The metric used to compute distance between such sentences based on their likeliness is called semantic similarity. An application of natural language processing named machine translation, requires to find the similarity of the user's input with each sentence present in the corpus and return the translation of the closet corpus sentence as final result. This application presents an effective need of sentence-to-sentence semantic similarity. Also, it can be useful in plagiarism checking, document classification system, query recommendation and information retrieval system.

Most of the conventional algorithms used for analysis of the sentences are based on the word frequency or word property, which indirectly eliminates the semantic meaning of the sentence. It becomes necessary to consider Part-Of-Speech (POS), word order along with the basic conventions. Hence, to compute the closeness between words, a dataset can be used to create a corpus which consists of all words present in the sentences.

Sentences Involving Compositional Knowledge (SICK) dataset is utilized. The dataset consists of 10,000 English sentence pairs having 'pair_ID', 'sentence_A', 'sentence_B', 'relatedness_score' and 'entailment_judgment' columns. The pair_ID columns consists of unique integer values. Sentence pair are provided in column sentence_A and sentence_B columns respectively. The relatedness_score represents the degree of semantic relatedness between sentences. Category such as entailment, contradiction, and neutral represents the type of sentence which are specified in entailment_judgment column [1].

This paper describes an algorithm developed to find similarity between two given sentences. Initially, the sentences are preprocessed for the removal of tokenization, stop-words, punctuations and lemmatization. A Word2Vec model is generated based on the specified corpus using 'Gensim' library of Python. This model generates relatedness of words based on their semantic and is used to compute the word similarity between the words in the sentences. Word order of sentences also impact the overall sentence similarity upto certain extent. Thus, the result is a combination of the semantic and word order similarity.

II. Literature Review

Existing methods processes the sentences in a very high-dimensional space and are not efficient and adaptable to some application domains. Initially, the methods focus on the similarity between long sentences because long texts usually contain a certain degree of shared words. Initially, the methods focus on the similarity between long sentences because long texts usually contain a certain degree of shared words. Compared to this, in short texts word occurrence is rare. Thus, the model aims for finding semantic similarity between short sentences. It uses How-net, a lexical database, to add human common-sense knowledge to the model.This approach computes similarity based on semantic of words, structure of sentence and words occurrence order in the sentence. The proposed method derives the sentence similarity from semantic and syntactic information present in sentences, which are to be compared. Method is divided into three steps: obtain word semantic similarity, obtain semantic similarity of sentences based on word semantic similarity and structure of sentences and lastly, calculating the word order similarity between sentences and combine this with semantic similarity from second step to generate final similarity between sentences [2]. Bag of Words is the most common method to calculate sentence similarity. The main reason that this method is not that reliable is that it only takes into account the word level information, ignoring its meaning in the sentence and word ambiguity. A method to calculate sentence similarity which solves the problem of word ambiguity using the machine learning method of word embedding to express the word and semantic problem using the dependency parser to analyze the internal grammatical structure. The model was implemented using Word Embedding generation, dependency relation generation, dependency matching and sentences similarity analysis [3]. Another research used a Unified Framework for Semantic Processing and Evaluation”. They measure the similarity between two sentences using three different types of features, including word alignment-based similarity, sentence vector-based similarity and sentence constituent similarity.It used some methods and it could be roughly divided into three categories: alignment approaches, vector space approaches and machine learning approaches. In alignment approach, it align words or phrases in a sentence pair, and then take the quality or coverage of alignments as similarity measure. Vector space approaches represent sentences as bag-of-words vectors and take vector similarity as their similarity measure. And in last method which is Machine learning approach, it combines different similarity measures and features using Support Vector Regression (SVR) model [4]. Generating similarity between sentences across multiple domains by leveraging corpora-based statistics into a standardized algorithm. requires the need of lexical database, which follows an edge-based approach to calculate the semantic similarity between words and sentences.The methodology presents an unsupervised approach to calculate the semantic similarity between two words, sentences and paragraphs which can also be applied to multiple domains. It divided into two passes. First pass, maximizes the similarity by following steps: Word similarity, sentence semantic similarity and word order similarity. The second pass bounds the similarity by considering recurrence of words and POS.

III. Proposed Model

SICK dataset is utilized for generating the corpus which can be used in Word2Vec model to fetch word semantic similarities. For simplicity of the table, the relatedness_score and entailment_judgment are represented using score and type respectively. The first 5 rows of the dataset are shown in Table I.

pair_ID	sentence_A	sentence_B	score	type
1	A group of kids is playing...	A group of boys in a yard...	4.5	NEUTRAL
2	A group of children...	A group of kids...	3.2	NEUTRAL
3	The young boys are...	The kids are playing...	4.7	ENTAILMENT
4	The kids are playing...	A group of kids...	3.4	NEUTRAL
5	The young boys are...	A group of kids...	3.7	NEUTRAL

TABLE I: First 5 rows of dataset.

A corpus named “myCorpus” is created by fetching and tokenizing all the sentences from the dataset. This corpus is applied as an input to the Word2Vec model named “customModelWord2Vec”. Word2Vec is a model which embeds word in a lower-dimensional vector space using a neural network. For each word, a word vector is generated and kept close to one another having similar meaning based on their context. Word2Vec accept parameters such as min_count, size, window and iter. Min_count is initialized to remove noise, i.e. least frequent occurring word. The size specifies the dimensionality of the vectors. The window value represents the number of neighboring words to be considered. The iter parameter refers to the number of epochs. The model is initialized with min_count = 3, size = 200, window = 3 and iter = 90 and saved as “8_project_2_TT.pt”.

The flow of the algorithm is shown in Fig. 1. The Pre-processing steps performs tokenization, removal of stop-words and punctuations and lemmatization along with considering the POS. POS helps in providing context of word and structure of the sentence. Semantic vectors and word order vectors are generated using the preprocessed sentences.

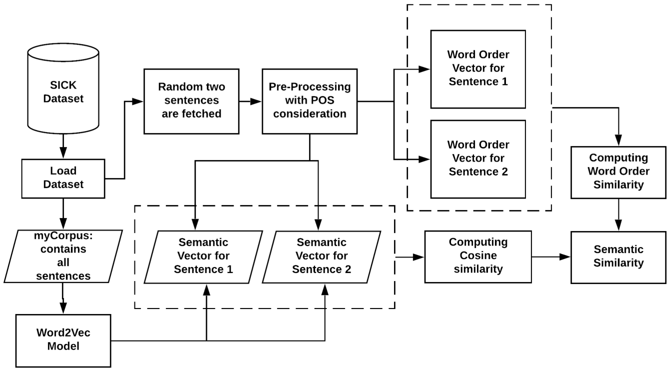
SICK Dataset
Load Dataset
myCorpus: contains all sentences
Word2Vec Model
Random two sentences are fetched
Pre-Processing with POS consideration
Word Order Vector for Sentence 1
Word Order Vector for Sentence 2
Computing Word Order Similarity
Semantic Vector for Sentence 1
Semantic Vector for Sentence 2
Computing Cosine similarity
Semantic Similarity

Fig. 1: Algorithm Flow.

The sentence similarity is calculated using cosine similarity, i.e. Eq. 1 and the word order similarity is calculated using Eq. 2.

$$Sim_s = \frac{S_1 \times S_2}{|S_1| \times |S_2|} \tag{1}$$

where S_1 and S_2 are semantic vectors of sentence 1 and 2 respectively.

$$Sim_o = 1 - \frac{||r_1 - r_2||}{||r_1 + r_2||} \tag{2}$$

where r_1 and r_2 are semantic vectors of sentence 1 and 2 respectively.

The overall sentence semantic similarity is computed by combining cosine and word order similarity as shown in Eq. 3.

$$Sim = \epsilon \times Sim_s + (1 - \epsilon) \times Sim_o \tag{3}$$

where ϵ decides the relative contributions of semantic and word order information to the overall similarity computation. The value of ϵ was set to 0.85 [2].

IV. Experimental Analysis

The Table II shows the semantic similarity rate of first 15 sentences based on the sentence similarity and word order similarity. Here, the similarity rate is in terms of percentages. Whereas, the SICK dataset has evaluated similarity of sentences between a scale 1 to 5. Thus, the similarity rate obtained by the algorithm is rounded off to one decimal point. Later, Accuracy is obtained by taking mean of the closeness of the predicted value with respect to ground truth is computed. The accuracy obtained against the SICK dataset is shown in Fig. 2. Various pre-trained models were considered such as Google sentence encoder to generate vectors for each sentence, which can be used for computing word similarity. But, it requires more memory and takes more time to fetch the word vectors. Also, the Word2Vec model performs well than the Google sentence encoder. The Word2Vec model is trained on the corpus, rather than using a pre-trained model.

Sentence 1	**Sentence 2**	**Similarity Rate (%)**
A group of kids is playing in a yard and an old man is standing in the background	A group of boys in a yard is playing and a man is standing in the background	94.3859
A group of children is playing in the house and there is no man standing in the background	A group of kids is playing in a yard and an old man is standing in the background	87.3583
The young boys are playing outdoors and the man is smiling nearby	The kids are playing outdoors near a man with a smile	84.3613
The kids are playing outdoors near a man with a smile	A group of kids is playing in a yard and an old man is standing in the background	73.0814
The young boys are playing outdoors and the man is smiling nearby	A group of kids is playing in a yard and an old man is standing in the background	69.7327
Two dogs are fighting	Two dogs are wrestling and hugging	63.3421
A brown dog is attacking another animal in front of the man in pants	Two dogs are fighting	46.7545
A brown dog is attacking another animal in front of the man in pants	Two dogs are wrestling and hugging	41.1800
Nobody is riding the bicycle on one wheel	A person in a black jacket is doing tricks on a motorbike	48.1291
A person is riding the bicycle on one wheel	A man in a black jacket is doing tricks on a motorbike	52.0978
A person on a black motorbike is doing tricks with a jacket	A person is riding the bicycle on one wheel	52.9122
A man with a jersey is dunking the ball at a basketball game	The ball is being dunked by a man with a jersey at a basketball game	96.5787
A man with a jersey is dunking the ball at a basketball game	A man who is playing dunks the basketball into the net and a crowd is in background	71.6391
The player is dunking the basketball into the net and a crowd is in background	A man with a jersey is dunking the ball at a basketball game	70.7182
Two young women are sparring in a kickboxing fight	Two women are sparring in a kickboxing match	90.7168

TABLE II: Semantic Similarity results.

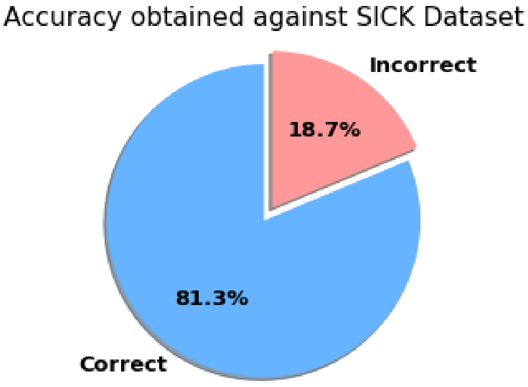

Fig. 2: Accuracy.

V. Conclusion

The algorithm developed for computation of semantic similarity between sentences takes POS, word order and semantic meaning of words into consideration. Based on the human sensing knowledge, the similarity rate obtained is closer to the relatedness_score of SICK dataset. POS adds context of words during the lemmatization of word to the root form. Also, the word order impacts the syntactic structure of the sentence. Hence, computing word order adds weight to the overall semantic similarity measure to a smaller extent. Our algorithm performance is computed against the SICK dataset resulting 0.812973 accuracy.

VI. Contributions

The project “Sentence-to-Sentence Semantic Similarity” is developed by Neelkumar P. Patel, Darshak Mota, Jay Sukhadiya, Priya Patel. The overall project development includes various task such as presentation, literature review, report writing, the idea of developing a new algorithm, dataset selection and implementation. Each person is best in their own way. For the successfully completion of the project and it’s related tasks, the contribution of each member was equally important and valuable.

References

[1] https://zenodo.org/record/2787612#.XoENd4hKjIU
[2] Zhao Jingling, Zhang Huiyun, Cui Baojiang, ”Sentence Similarity Based on Semantic Vector Model”, Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, IEEE, 2014
[3] Xinchen Xu, Feiyue Ye, ”Sentences Similarity Analysis Based on Word Embedding and Syntax Analysis”, 17th IEEE International Conference on Communication Technology, IEEE, 2017.
[4] Fu Cheng, An Bo, Han Xianpei & Sun Le, ”ISCAS_NLP at SemEval-2016 Task 1: Sentence Similarity Based on Support Vector Regression using Multiple Features.”, Association for Computational Linguistics, 2016.
[5] Atish Pawar, Vijay Mago, ”Challenging the Boundaries of Unsupervised Learning for Semantic Similarity”, IEEE Access, IEEE, 2019.