Extraction Of Similar Semantic Sentence From Wikipedia Citation

Prashant Pathak (MT 19051), Deekshant Mamodia (MT 19119), Anchit Gupta (MT 19060)

Advisor: Dr. Tanmoy Chakraborty

Information Retrieval (CSE 508)

Problem Statement

In this project, our aim is to find a semantically similar sentence to the cited text in the cited document, in the context of Wikipedia. This task is challenging because many factors come into play. The main reason is that cited documents are of different nature, as it can be anything in the network ranging from text, image, java scripts, books,research archive, or maybe an empty document. The other goal is to provide an software solution to the new user through which user can simply input the query and get the best matched result from the cited document in minimum time.

Use Case of the Problem

- Irrelevant cited reference can be identified. If during finding the similarity we didn't find any similar document then we can easily say that the cited document is irrelevant to the sentence.
- It will also save lot of user time as user doesn't have to read the whole cited document, he/she can read just the specific parts.
- Apart from wikipedia text it also has other relevance like in Customer service. AI system should be able to understand semantically similar queries from users and provide a uniform response. The emphasis here is to create a system that mimics human conversation.

Literature Review

In-depth explanation about semantic similarity and its categories and how to measure it is explained in survey paper [1]. It also include technique such as LDA and WordNet that are used for finding the meaning of the two text and gives better similarity score. In paper [2] authors proposed a new metric WMD (Word Mover's Distance) that allows us to assess the "distance" between two documents in a meaningful way, even when they have no words in common. It uses Word2vec vector embeddings of words. It has been shown to outperform many state-of-the-art methods among k-nearest neighbors classification. In paper [3] author goes from word-level to short-text-level semantics by combining insights from methods based on external sources of semantic knowledge with word embeddings. In particular, they perform semantic matching between words in two short texts and use the matched terms to create a saliency-weighted semantic vector. In this author proposed new normalization technique named Smooth Inverse Frequency which takes inverse probability of a word in the corpus.

Dataset Description

- Wikipedia is a multilingual, web-based, free-content encyclopedia project supported by the Wikimedia Foundation and based on a model of openly editable content.
- In out project we have use only english document of wikipedia dump.
  - Log data
  - Metadata about each page
  - Text of current or all revisions of all pages
  - Short plain text abstracts of each page
- As of 1 February 2020, there are **6,023,753** articles in the English Wikipedia containing over **3.5 billion words**.
- February dumps compressed size is about 16 GB. As the dump was to big as per our resources we used a smaller dump for our model training.

System Architecture

- System architecture is divided into three small components i.e **Front-end, Server, Backend architecture** as shown in figure 1.

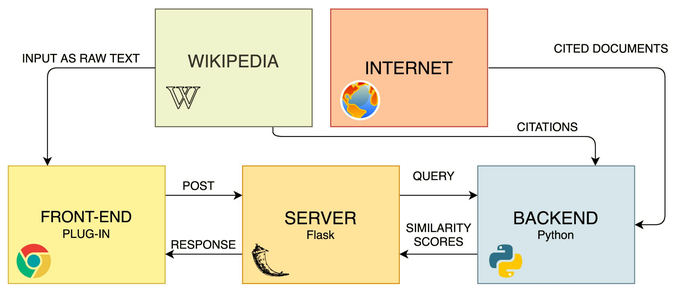


Figure 1: End to end system Architecture of similar sentence extraction model on Wikipedia

Front-end Architecture:

- This architecture is developed for end user to enter the query for which they want best match for each citation text.
- It is a **Chrome browser plugin** which works on the Wikipedia articles.
- Plugin takes raw text as input and send to server.
- Shows final result of each citation which occur in the query.

Server:

- Server is handled through API developed on **Flask framework**.
- API get raw input text from the plugin and transfer to the models.
- Get best result of each cited text from the backend and transfer to the plugin.

Backend Architecture:

- Architecture has three main components which are define below:
  - Text in different forms
  - Embedding
  - Computing Closeness Measure
- First pre-processed the raw input text with different preprocessing techniques like lemmatization, expand clitics, remove punctuation.
- Trained the Doc2Vec model on the pre-processed training data consist of 98000 wikipedia articles.
- Measure the similarity between the embedding vector of the query text and cited document text using two different approaches as defined below:
  - Cosine Similarity
  - Word Mover's Distance

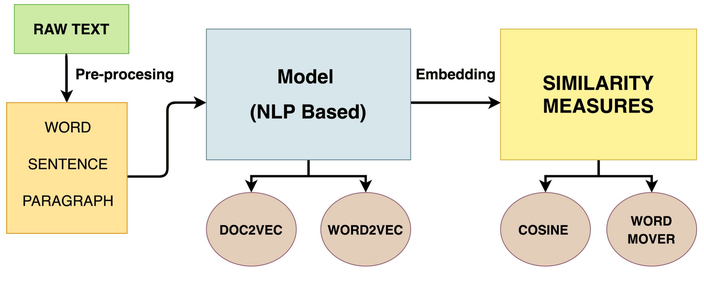
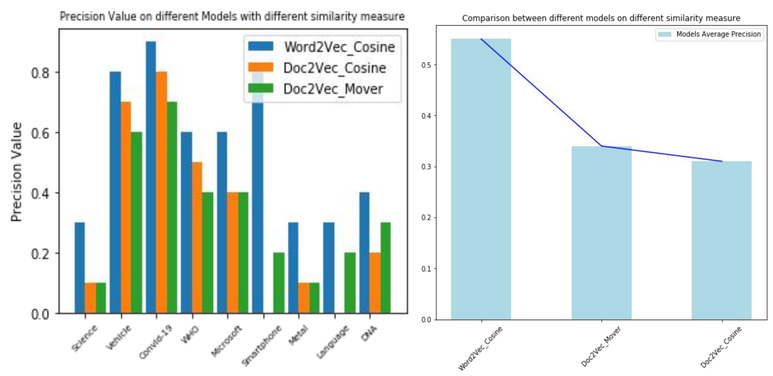


Figure 2: Backend Architecture of similar sentence extraction model on Wikipedia

Results

- To analyse the performance of our model we randomly choose 9 topics from wikipedia with different number of citation in each topic and passes it through all the models.
- On manually reading the results of the model we found that Word2Vec along with Word Mover Distance performs reasonably , so we kept this as our ground truth and compared it with other model.
- Figure 3 shows comparison between the different models with different similarity measure.



Conclusion

- From the above results we can infer that the type of embedding used plays an important role in the performance of the model.
- So to get a better precision value we can use better embedding such as Universal sentence Encoder (USC) proposed in [5] and BERT embedding proposed in [6] .
- We can also use other closeness measures such as LDA.
- In future this project can be extended to work on above advance text embedding models for better performance.

References

1. Pradhan, Nitesh & Gyanchandani, Manasi & Wadhvani, Rajesh. (2015). A Review on Text Similarity Technique used in IR and its Application. International Journal of Computer Applications. 120. 29-34. 10.5120/21257-4109.
2. Gao Huang, Chuan Quo, Matt J. Kusner, Yu Sun, Kilian Q. Weinberger, and Fei Sha. 2016. Supervised word mover's distance. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16). Curran Associates Inc., Red Hook, NY, USA, 4869–4877
3. Tom Kenter and Maarten de Rijke. 2015. Short Text Similarity with Word Embeddings. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15). Association for Computing Machinery,New York, NY, USA, 1411–1420.
4. Quoc V. Le and Tomas Mikolov. 2014b.Distributed representations of sentences and documents.
5. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.
6. Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua,Nicole Limtiaco, Rhomni St. John, Noah Constant,Mario Guajardo-Cespedes, Steve Yuan, Chris Tar,Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil.2018. Universal sentence encoder.