Similar Content Retrieval From Wikipedia Citation

Rahul Gupta MT20065 | Deepti Gupta MT20085 | Palak Tiwari MT20103 | Ravi Rathee MT20066 | Mohit Ghai MT20308

1. PROBLEM DESCRIPTION

The drastic growth in the content available on the web has been observed in the past few years. Wikipedia saw exponential growth since 2006, and the total number of articles on the site reached 55 million, out of which 6.2 million are in English. This massive collection of articles poses new kinds of difficulties to users, such as retrieving semantically similar information. As lexically similar texts may not relate semantically like food and bread are semantically equivalent however lexically different. It is time-consuming to visit various cited links and read content manually when the data is vast and diverse. Even if we try reading content manually, not all cited text is relevant, leading to wastage of time. Sometimes the user needs context to the quoted text and requires to search manually for the cited text and may end up getting some relevant data or may deviate from the context. This problem needs the system to be more imaginative and efficient and increase the user readability and experience.

We aim to propose an information retrieval system that would retrieve semantically similar data from the noted link, noted link here means Wikipedia's highlighted internal links and external citations. We propose to make a chrome extension plugin which will return the most relevant article from citations based on user's query. Our approach will save users precious time by not surfing over and over for the cited context. It helps the user to extract the vital information more engagingly. It also avoids the situation when the user visits multiple articles for the cited text and may often deviate from the context. This will enhance the overall knowledge of the context of the user as well.

2. LITERATURE REVIEW

[1] **Semantic Cosine Similarity** Cosine similarity is a widely implemented metric in information retrieval and related studies. Metric tries to model the document as a vector of terms and similarity between documents can be calculated by finding cosine values between the document term vectors. Even in search engines, similarity between documents and user query are stored in descending order of similarity. Paper summarises related work done in three sections mainly, A) Cosine Similarity B) WordNet C) Semantic Similarity between Words and finally discusses about disadvantages of using Cosine Similarity. Paper also proposes two enhancements over previously used cosine formula. The two enhancements were A) Dimension equalization of both term vectors. B) If there exists a synonym pair in both vectors choose the first one as dimension name for both vectors. For example, rodent is chosen as dimension's name from a pair of "rodent-mouse".

[1] **Automatic extraction of semantic relations from Wikipedia, -** Recently Patrick Arnold and Erhard Rahm presented Automatic Extraction of Semantic Relations patterns from voluminous unstructured Wikipedia articles using a finite state machine. The machine is capable of parsing the first few lines of the wiki article. The proposed system only parses the first 750 words of the article to get the abstract information. The paper describes the system where input is a sentence with a list of predefined anchor terms. If the sentence contains an anchor term, FSM discovers the semantic pattern starting with the anchor term and ending in the final state. The system can capture at least two semantic ways. After the extraction of object terms, subject terms, and field terms, semantic relations are found. These relations can be used to make huge updated thesaurus as a knowledge for tasks such as retrieving semantic ontology mappings. The paper also discusses that method is capable of detecting entity articles from concept articles with the use of categories. For this, regular expressions are used to cover few types to discard pieces under such categories.
As discussed in the paper, FSM faced parsing errors due to the sentences being too complex. Recall problem in the term extraction due to the removal of complex expressions with parenthesis which had relevant terms.

[2] **Hybrid Geometric Approach for Measuring Similarity Level Among Documents and Document Clustering -** Propose a method for measuring similarity levels among documents and document clustering. This involves a few steps, which in order are: 1) Carefully choosing features to reflect underlying semantics. 2) Creating a Vector Space Model. 3) Comparing algorithms used to measure similarities between documents like Non Geometric measures and Geometric measures - cosine similarity and Euclidean Distance 4) Drawbacks of above algorithms are discussed.
In this paper, the authors discuss a novel approach to overcome the drawbacks of methods discussed earlier: the Novel geometric similarity measurement method.
This algorithm proposed by authors called the "TS-SS Algorithm" computes between vectors from two diverse perspectives. It generates the similarity value between two vectors from the angel and ED between them and the difference be-

tween their magnitudes.

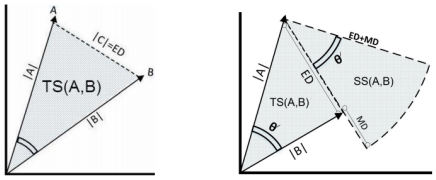

Figure 1: TS and SS Algorithm

This algorithm has two parts:-
A) TS Algorithm - used to generate different similarity values.
B) SS Algorithm (Sector's Area Similarity) TS alone is not sufficient to interpolate vector's differentiation precisely, and hence we need an SS algorithm. TS and SS complete each other, and that is why we combine them by multiplying them together.

[3] **A New Approach for Implicit Citation Extraction -** In this paper, the problem is divided into two and then conquered. Two subproblems are topic modeling and implicit citation detection (similarity detection between the text and the citation)and then proposed techniques are applied.LDA(Latent Dirichlet Allocation) is used for topic modeling to get the semantically similar context as lexical similarity might not always give appropriate contexts. For implicit citation detection, two-word embeddings(vectors) are determined. Sentence2vec and Topic2vec are used. However, both are determined using the concepts of Doc2vec. Unsupervised learning techniques are used as the Wikipedia dataset is not annotated. The paper also shows a real-world example to demonstrate the proposed method.
The limitation of this paper is, the dataset used is not multilingual, and also the accuracy of the proposed system is not determined.

[4] **Semantic Analysis of Wikipedia documents using Ontology -** With the drastic growth in web content, there is the need for better query results from the server-side. The authors proposed a system where they fetched the user query, extracted the critical term using an algorithm like KPE, and ranked it. For the next step, they applied Cosine Similarity. The Semantic Web will offer a way for solving the problem at the architecture level. Each page possesses semantic metadata that records additional detail concerning the Web page itself. For evaluation, they used precision, recall, f-measure, and accuracy. The results achieved by them were 0.73 Accuracy, 0.6 Precision, 0.87 recall, and 0.78 F-measure. However, the techniques used are unsupervised. Adding supervised methods for ontology learning may add to the results

[5] **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding -**BERT or Bidirectional Encoder Representations from Transformers is an encoder only based Transformer network used for Natural Language Processing tasks. The major application of Bert is to extract features i.e. convert sentence into its vector representation. BERT uses attention mechanisms and considers both left and right context of the given sentence. It is trained on two phases MLM (Masked Language Model) & NSP (Next Sentence Prediction). There are two types of BERT models BERT BASE (110 million parameters) & BERT LARGE (340 million parameters). We can either use pre-trained model of BERT or do fine-tuning based on our data. Other applications of BERT include Classification, Summarization, Question-Answer tasks etc.

[6] **Annotating documents by Wikipedia concepts -**This paper demonstrates what all Wikipedia pages in the target Wikipedia document are the most relevant. For this, they rank the pages as well. There are four types of Wikipedia pages connected to a single Wikipedia page denoted by the target document. Authors propose an algorithm that will retrieve the most semantically relevant pages in the link structures of pages. According to the algorithm, firstly title of the target document is cleaned. These concepts are currently ranked as per the formula stated in the paper, and the top-ranked pages are considered the most relevant.

[7] **Identifying Citing Sentences in Research Papers Using Supervised Learning -** Sugiyama et al. designed a method to predict whether a sentence requires citation or not using a supervised learning approach. They used ACL Anthology Reference Corpus.Extracting the features like Bigram, Unigram, Previous and Next sentence, Position, Proper Nouns, and Orthographies. Constructing the classifier Maximum Entropy and Support Vector Machine and classify the sentence whether it needs to be cited or not. They achieved an accuracy of 0.82 using both Maximum Entropy and SVM with C = 0.9. The results showed that for training accurate models in both ME and SVM, next sentence and proper nouns are effective features. The limitation is that the system doesn't consider any similarity between the sentences. We try to explore similarities using different techniques.

[8] **Extractive Text Summarization from Web pages using Selenium and TF-IDF algorithm -** Paper describes Extractive method of text summarisation, where sentences are ranked according the importance of the sentence in order to get the summary. Proposed method uses extractive method to generate summary of the context based on the users's query by scrapping data from website. Urls are scrapped related to the query using google, bing etc. Data is scrapped from the fetched urls. Fetched data is summarised using TF-IDF algorithm. Sentences are tokenised, term frequency is calculated, inverse document frequency is calculated. Sentences are scored based on Tf-idf score. Threshold is found and based on that sentences greater than equal to threshold are shortlisted to make summary.

[9] **Implementation of Textrank Algorithm in Product Review Summarization** Paper proposes textRank algorithm to generate product review summarization. TextRank algorithm is graph based extractive method of text summarization. Where sentences are as vertex. Most important vertex are ones which are connected to many other vertex. Proposed method collects data, preproceses it for get only important information. Similarity between sentences are calculated, weights of sentences are calculated and finally sentences are ranked. Hence summary is calculated.

Evaluation of the method is done using Rouge-N method.

[10] **Method Of Text Summarization Using Lsa And Sentence Based Topic Modelling With Bert** Paper introduces Extractive method of summarisation of text using topic modeling and BERT for sentence embedding. Latent semantic analysis is used to reduce dimension of matrix to extract topics from documents efficiently. BERT pre-trained deep learning model which includes contextual encoding of sentences. Proposed method applies topic modelling using LSA. TF-IDF keyword extractor is used to get relevant keywords. BERT encoder used for sentence embedding which captures positional embedding of keyword extracted in previous step. Positional embedding for LSA topics and keyword is extracted. Cosine similarity is used to find the difference between vectors. Finally summary is created using maximum cosine score.

3. METHODOLOGY

3.1 Extracting Similar Articles

- Extracting most contextually similar paragraphs from the cited articles from the query.
- Recommending Contextually similar articles from the current article.

For the given input query we fetched the citations present in the query, both external and internal links, here internal means links that are from some wikipedia page itself and external means those from internet that are cited by the authors. We scrap document id's for the internal links. Then we used Doc2vec model for the documents to get the embeddings. For query we used Word2vec to get the vector embeddings. For finding the similarity between the document vectors and the input query we used Cosine Similarity measure and WMD (Word Mover's Distance).

For Recommending Contextually similar articles to the user for the query or for the current page depending on users choice, for this topic modeling is used. This can also be done by finding the similarity of the introductions paragraphs of different articles and the given query or the introduction paragraph of the current page and thereby ranking those articles to get top five most relevant articles for users to read. For this, scrapping of the topic and extraction of different topics from the the Wikipedia dumps. Creating vectors of the current and other articles using above models that is Word2Vec and Doc2Vec. Finally finding similarity between the vectors and ranking them to get the most similar topics to recommend.

3.2 Text Summarization

Text Summarization is a process of creating a summary of a document. Summarization refers to presenting data in a concise form, highlighting the part that convey fact and information while preserving the meaning. It is divided into two classes [11]:-

1. Extractive Summarization
2. Abstractive Summarization

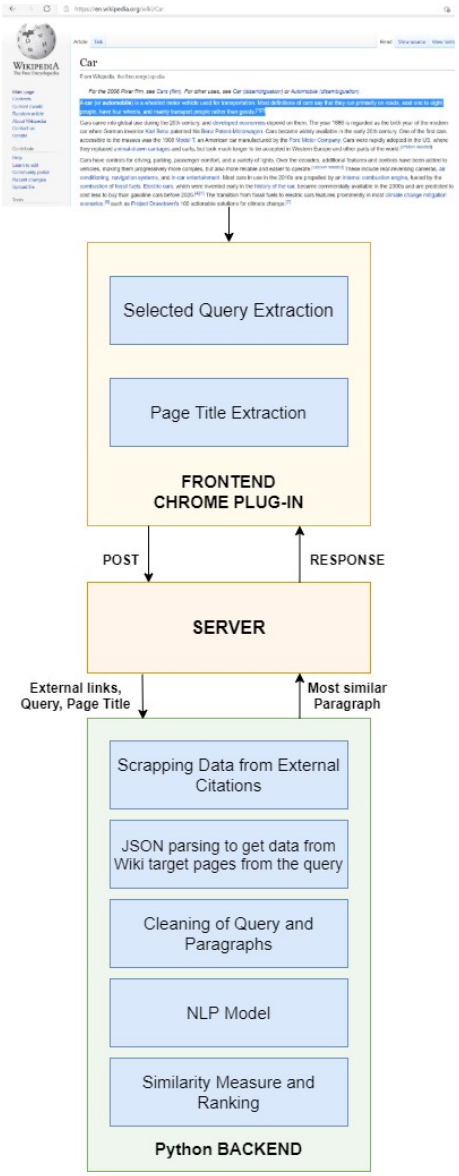

Figure 2: Pipeline Flow

Extractive summarization picks up sentences directly from the original document depending on their importance. [11]

Abstractive summarization tries to produce a bottom-up summary using sentences or verbal annotations that might not be a part of the original document. [11]

We have used the following approaches for Extractive summarization :-

1. **TF-IDF** - Term frequency-inverse document frequency is a numeric measure that is used to score the importance of a word in a document based on it's appearace in the document and a given collection of documents. If a word appears frequently in a document then it should be important and that word should be given high score. But if a word appears in too many other documents, it's probably not a unique identifier and should be assigned low scores. [12]

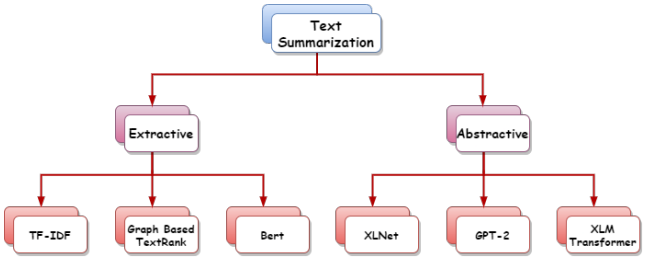

Figure 3: Text Summarization

Steps :-

(a) The sentences are converted into tokens.

(b) A frequency table is created for each sentence which stores a term and it's frequency.

(c) The Term frequency is calculated for each term of the sentence. A term frequency is the number of times term t appears in a document by total number of terms in the document.

(d) The number of sentences containing a word are calculated. It will be useful in creating Inverse Document Frequency (IDF).

(e) The inverse document is calculated for each term. IDF is log of total number of documents by number of documents with term t in it.

(f) TF-IDF scores are calculated by multiplying tf and idf scores for each term.

(g) Each sentence is given a score by averaging the scores of the terms present in the sentence.

(h) Summary is generated by selecting the top scored sentences based on certain threshold.

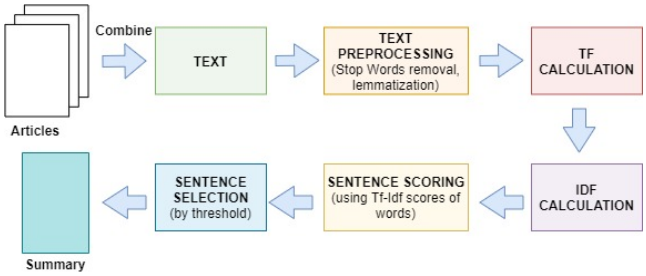

Figure 4: TF-IDF Approach

2. **TextRank** - It is an unsupervised graph-based technique. It is based on PageRank algorithm. It is used for ranking text sentences. Each sentence in the document is represented by a node in the graph. The edges denotes the similarity between the two nodes i.e. sentences. The algorithm assigns scores to each sentence and picks the top-ranked sentences to make summary.

   Steps :-

   (a) A graph is created where nodes denote the sentences. The edges in the graph denote the similarity between the two nodes i.e. sentences.

   (b) PageRank algorithm is applied on the weighted graph.

   (c) The highest score nodes are added to the summary.

   We have used Gensim TextRank module.

3. **BERT** - BERT stands for Bidirectional Encoder Representations from Transformers. It works on mechanism that learns contextual relations between words (or sub words) in a text. The transformer includes two parts - an encoder that reads the text input and a decoder that produces a prediction for the task. By biderectional it means that the encoder reads the entire sequence of words at once. [13] We have used BERT module from HuggingFace Transformers.

We have used the following approaches for Abstractive summarization :-

1. **XLNet** - It is a bidirectional transformer where the next tokens are predicted in random order. [14] It is an extension of the transformer-XL model pre trained using an autoregressive method to learn bidirectional contexts by maximizing the expected likelihood over all permutations of the input sequence factorization order. It leverages the advantages of both, Auto-regressive and Auto-encoding methods for its pretraining which helps it to overcome pretrain-finetune discrepancy. [15] We have used XLNet module from HuggingFace Transformers for summarization.

2. **GPT-2** - It is Generative Pre-trained Transformer 2. It a variant of the transformer model which only has the decoder part of the Transformer network. It uses multi-headed masked self-attention, which allows it to look at only the first i tokens at time step t, and enables them to work like a uni-directional language models. The models process tokens in parallel i.e. by predicting tokens for all time steps at once. [16] We have used GPT-2 module from HuggingFace Transformers.

3. **XLM Transformer** - It is a transformer based architecture that is pre-trained using one of the three language modelling objectives [17] -

   - Causal Language Modeling - models the probability of a word given the previous words in a sentence.
   - Masked Language Modeling - the masked language modeling objective of BERT.
   - Translation Language Modeling - a (new) translation language modeling objective for improving cross-lingual pre-training. [17]

   We have used XML module from HuggingFace Transformers.

4. BASELINE RESULTS

We developed a chrome extension which fetches the query highlighted by the user and returns the most relevant paragraph from the citations present in the given query. For finding the relevant paragraph from the citations we use

the Doc2vec model for the document and Word2vec for the query as mentioned in [4] For finding the similarity between the paragraph and the given query we use different similarity measures like Cosine Similarity, Word Movers distance

The system will get a cited text from Wikipedia as an input and will return the semantically relevant Paragraph from the cited articles as an output.

- **Input from FrontEnd** - User can highlight some part of the wikipedia text, and our plugin will take that highlighted text as input and other required details of the articles along with citations and hyperlinks in the articles as input.

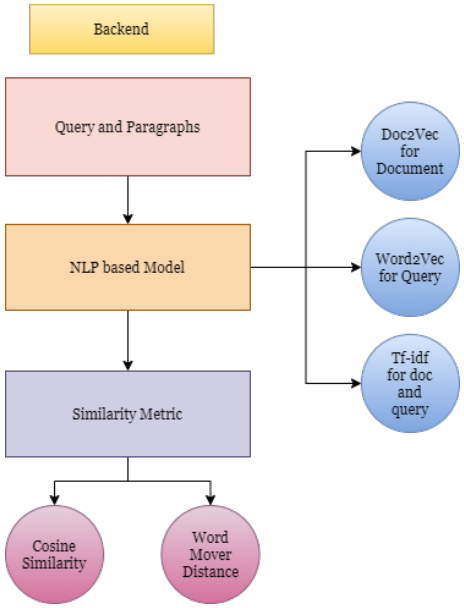

Figure 5: Backend Flow

- **Scraping data** - Query and title name is fetched using javascript and sends a post request to the flask server. Now using page title citations are scrapped from the current page. Citation occurring in the query is extracted. Using the extracted citations from the query, documents are scrapped. Each extracted articles are appended together which will further be broken into paragraphs.
- **Cleaning** - Now the appended articles is broken down into paragraphs and we apply some pre-processing to clean data like, removal of junk symbols, tokenization, stop words, removing punctuation and applied lemmatization.
- **Embeddings** - We used three types of word embeddings as mentioned below:-
  - **Word2Vec** - Word2Vec is a combination of 2 techniques - CBOW (Continous Bag of Words) and Skip-Gram (Skip -Gram model). It helps to keep semantic information and to preserve relation between different words. We get 300 dimensional embeddings for each word. Then we take mean for all the vectors and resultant query vector of 300 dimensions.
  - **Document2Vec** - It is used to create embeddings for group of words taken collectively. It represents each document as a vector. The document is splitted into words. Then tokens are generated. This token list is passed as input to doc2vec model. It gives a 300 length dimensions embeddings for each paragraph.
  - **Tf-idf** - We applied Tf-idf on both query and document after building the corpus from various wikipedia documents. After building the corpus we vectorized the input query and the document where for each paragraph present in the document different vectors are made which later on will be checked for similarity againts the query vector.
- **Similarity Measures** - We applied similarity measuring technique like Cosine similarity , that is a metric to measure the similarity betweeen document vector and query vector by determing the cosine angle and Word Mover Distance to find the similarity between the document and the query.
- **Output to FrontEnd** - Most relevant results by cosine similarity are sorted in descending order and top results are returned, which potentially helps user in finding similar paragraphs. On using the Word Mover Distance also we get the most relevant result after sorting the results in descending order accordint to the scores.

Query :'an American public organization that focuses on non-profit activities and creative media ventures.'

Embedding-distance metric	Top first Result	Top second Result	Top third Result
Pretrained Doc2Vec and word2Vec with cosine similarity	Archewell press secretary Toya Holness told Town & Country, "Founded earlier this year by The Duke and Duchess of Sussex, Archewell uplifts communities through non-profit partnerships and creative activations. It's a place where compassion matters, communities gather, and storytelling is the engine. The website has been updated to reflect the work Archewell has undertaken throughout 2020 and to create a place for people and communities around the world to share their stories. USA railway vehicle having its own means of propulsion	On the Archewell Foundation page, Harry and Meghan have listed a series of partnerships and projects the non-profit is undertaking, including those with James R. Doty's Centre for Compassion and Altruism Research and Education (CCARE), Tristan Harris' Centre for Humane Technology, Rachel Cargle's Loveland Foundation, Dr. Safiya Noble and Dr. Sarah T. Roberts' UCLA Centre for Critical Internet Inquiry, and José Andrés' World Central Kitchen.	And finally, the Archewell Productions page contains a description of the couple's newly-creation video production company, which was launched after Harry and Meghan inked a landmark deal with Netflix earlier this year. "Archewell Productions will utilize the power of storytelling to embrace our shared humanity and duty to truth through a compassionate lens," the page reads.
Pretrained Doc2Vec and word2Vec with wm distance	On the Archewell Foundation page, Harry and Meghan have listed a series of partnerships and projects the non-profit is undertaking, including those with James R. Doty's Centre for Compassion and Altruism Research and Education (CCARE), Tristan Harris' Centre for Humane Technology, Rachel Cargle's Loveland Foundation, Dr. Safiya Noble and Dr. Sarah T. Roberts' UCLA Centre for Critical Internet Inquiry, and José Andrés' World Central Kitchen.",	And finally, the Archewell Productions page contains a description of the couple\'s newly-creation video production company, which was launched after Harry and Meghan inked a landmark deal with Netflix earlier this year. "Archewell Productions will utilize the power of storytelling to embrace our shared humanity and duty to truth through a compassionate lens," the page reads.',	The Archewell Audio page features the first episode of Archewell Audio, Harry and Meghan\'s new podcast with Spotify, as well as a short description of the production company\'s aims: to "produce programming that uplifts and entertains audiences around the world," and "spotlight diverse perspectives and voices," as well as "build community through shared experiences, powerful narratives, and universal values:',
Tfidf with cosine similarity	Just ahead of the new year, Prince Harry and Meghan Markle have updated the website for their new non-profit, Archewell. The site was launched in October with just a landing page, but now, the couple has fleshed it out, adding pages for the Archewell Foundation, Archewell Audio, and Archewell Productions:.,	I am my mother's son. And I am our son's mother. Together we bring you Archewell," the couple wrote on the homepage for their new non-profit.	Archewell press secretary Toya Holness told Town & Country, "Founded earlier this year by The Duke and Duchess of Sussex, Archewell uplifts communities through non-profit partnerships and creative activations. It's a place where compassion matters, communities gather, and storytelling is the engine. The website has been updated to reflect the work Archewell has undertaken throughout 2020 and to create a place for people and communities around the world to share their stories.",

Figure 6: Baseline results

Figure 3 shows the results from different variations on inputing the same query, the first result is from the Doc2vec on document and Word2vec on query with Cosine similarity as metric the next is with WM distance metric followed by Tf-idf Vectors on both query and document with Cosine Similarity Metric.
To validate our result input document itself contained query as one of the paragraph. When this input set was fed to the model , query text was output(most relevant paragraph).

5. EVALUATION

5.1 Metrics Used

Rouge-N : ROUGE-1 refers to the overlap of unigram (each word) between the system(output results from various algorithms) and reference summaries(ground truth).

ROUGE-L: Longest Common Subsequence (LCS) based statistics. Longest common subsequence problem takes into account sentence level structure similarity naturally and identifies longest co-occurring in sequence n-grams automatically.

5.2 Results

Evaluation Metric	Summarization Algorithm	Recall	Precision	F Measure
Rouge-1	TF-IDF	0.2068	0.9619	0.3405
Rouge-1	TextRank	0.6454	1.0	0.1228
Rouge-1	Bert	0.1698	1.0	0.2904
Rouge-1	Gpt-2	0.1941	1.0	0.3251
Rouge-1	XLNet	0.1941	1.0	0.3251
Rouge- L	TF-IDF	0.2061	0.9586	0.3393
Rouge- L	TextRank	0.6454	1.0	0.1228
Rouge- L	Bert	0.1698	1.0	0.2904
Rouge- L	Gpt-2	0.1941	1.0	0.3251
Rouge- L	XLNet	0.1941	1.0	0.3251

Figure 7: Rouge Score of Algorithms

6. CONCLUSION

6.1 Extracting Similar Articles

We have solved the solved the problem of extracting similar content from cited sites from wikipedia and displayed them on the plugin application.

6.2 Text Summarization

We have generated summary using various models such as TF-IDF, BERT, TextRank, GPT-2, XLNet, XLM Transformer. Each model generates slightly different summary for the same article 'Car'. Summarization based on TF-IDF outperformed other methods with F-Measure as 0.3405 with Rouge-1 and 0.3393 with Rouge-L evaluation metric. The another way to judge the best summary generated by asking human evaluator. We also tried to implement question answer module using these summaries.

7. REFERENCES

[1] P. Arnold and E. Rahm, "Automatic extraction of semantic relations from wikipedia," *International Journal on Artificial Intelligence Tools*, vol. 24, p. 1540010, 04 2015.

[2] A. Heidarian and M. J. Dinneen, "A hybrid geometric approach for measuring similarity level among documents and document clustering," in *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, 2016, pp. 142–151.

[3] C. Jebari, M. J. Cobo, and E. Herrera-Viedma, "A new approach for implicit citation extraction," in *Intelligent Data Engineering and Automated Learning – IDEAL 2018*, H. Yin, D. Camacho, P. Novais, and A. J. Tallón-Ballesteros, Eds. Cham: Springer International Publishing, 2018, pp. 121–129.

[4] P. Banik, S. Gaikwad, A. Awate, S. Shaikh, P. Gunjgur, and P. Padiya, "Semantic analysis of wikipedia documents using ontology," in *2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA)*, 2018, pp. 1–6.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[6] P. Schönhofen, "Annotating documents by wikipedia concepts," in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1, 2008, pp. 461–467.

[7] K. Sugiyama, T. Kumar, M. Kan, and R. C. Tripathi, "Identifying citing sentences in research papers using supervised learning," in *2010 International Conference on Information Retrieval Knowledge Management (CAMP)*, 2010, pp. 67–72.

[8] K. U. Manjari, S. Rousha, D. Sumanth, and J. Sirisha Devi, "Extractive text summarization from web pages using selenium and tf-idf algorithm," in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 2020, pp. 648–652.

[9] M. R. Ramadhan, S. N. Endah, and A. B. J. Mantau, "Implementation of textrank algorithm in product review summarization," in *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, 2020, pp. 1–5.

[10] H. Gupta and M. Patel, "Method of text summarization using lsa and sentence based topic modelling with bert," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 511–517.

[11] S. Chaudhary, "Extractive text summarization using neural networks," Apr 2021. [Online]. Available: https://heartbeat.fritz.ai/extractive-text-summarization-using-neural-networks-5845804c7701

[12] A. Jain, "Automatic extractive text summarization using tfidf," Apr 2019. [Online]. Available: https://medium.com/voice-tech-podcast/automatic-extractive-text-summarization-using-tfidf-3fc9a7b26f5

[13] R. Horev, "Bert explained: State of the art language model for nlp," Nov 2018. [Online]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

[14] M. Singh, "Summarize reddit comments using t5, bart, gpt-2, xlnet models," Jan 2021. [Online]. Available: https://towardsdatascience.com/summarize-reddit-comments-using-t5-bart-gpt-2-xlnet-models-a3e78a5ab944

[15] S. Ghag, "Using xlnet for sentiment classification," Jun 2020. [Online]. Available: https://medium.com/swlh/using-xlnet-for-sentiment-classification-cfa948e65e85

[16] R. K. Singh, "Generating text summaries using gpt-2 on pytorch," Apr 2021. [Online]. Available: https://blog.paperspace.com/generating-text-summaries-gpt-2/

[17] "Papers with code - xlm explained." [Online]. Available: https://paperswithcode.com/method/xlm