Core Project
(PRJ3001)

Final Report

Quantifying the text similarity between citations using NLP

By

Muddarapu Sai Anirudh Reddy

1800240C203

Imadabattuni Naveen Kumar

1800222C203

Penumarthi mohana uma sushmanth

1800248C203

Supervisor

Dr. Devanjali Relan

Assistant Professor

Dr. Kiran Sharma

Assistant Professor

BML MUNJAL UNIVERSITY™

Department of Computer Science and Engineering
School of Engineering and Technology

December 2021

Acknowledgement

We would like to express our sincere gratitude to our supervisor Dr. Devanjali Relan and Dr. Kiran Sharma for providing us a chance to do research under their guidance and giving significant direction all through this exploration. Their vision, genuineness and inspiration have inspired us a lot. They have demonstrated the techniques to make the exploration and to present this work as clearly as possible. It was an implausible advantage and regard to work under their direction. We had learnt a lot of new things, which we have not learned previously. We are amazingly thankful for what they have offered us. Their direction showed us the way in all the time of difficulties and composing of this theory. We were unable to have envisioned having a superior guide for this project.

Thanking You.

Muddarapu Sai Anirudh Reddy *1800240C203*

Imadabattuni Naveen Kumar *1800222C203*

Penumarthi mohana uma sushmanth *1800248C203*

B.Tech CSE 2018-2022

4th December 2021.

BML MUNJAL UNIVERSITY™

BML Munjal University, Gurgaon, Haryana

CANDIDATE'S DECLARATION

We, M. Sai Anirudh Reddy, I. Naveen Kumar, P. Mohana uma sushmanth, hereby declare that the work done in my core project entitled *"Quantifying the text similarity between citations using NLP*" in fulfillment of completion of 7th semester of Bachelor of Technology (B.Tech) program in the Department of Computer Science and Engineering, BML Munjal University is an authentic record of our original work carried out under the guidance of Dr. Devanjali Relan and Dr. Kiran Sharma.

Due acknowledgements have been made in the text of the project to all other materials used.

This core project was done in the full compliance with the requirements and constraints of the prescribed curriculum.

Muddarapu Sai Anirudh Reddy	*1800240C203*	
Imadabattuni Naveen Kumar	*1800222C203*	
Penumarthi mohana uma sushmanth	*1800248C203*	

Place: Gurgaon, Haryana

Date: 4th December 2021

CERTIFICATE

This is to certify that the Core Project entitled "*Quantifying the text similarity between citations using NLP*" to the best of my knowledge is a record of the Bonafede work carried out by Mr. Muddarapu Sai Anirudh Reddy, Imadabattuni Naveen Kumar, Penumarthi Mohana uma sushmanth under my guidance and/or supervision. The contents embodied in this report, to the best of my knowledge, have not been submitted anywhere else in any form for the award of any other degree or diploma. Indebtedness to other works/publications has been duly acknowledged at relevant places. The project work was carried out during *July - December 2021* as part of their 7th semester coursework for Bachelor of Technology (B.Tech) program in the Department of Computer Science and Engineering, BML Munjal University.

Name and Designation of the Supervisor:

Signature:

Date:

Place:

Contents

Abstract

Determining the relationship between the main paper and its citations in order to determine how relevant they are is a critical task because to know how much more work has been done than existing work. The main aim of this research project is to find out the similarity score between main paper and its citations. There are four steps involved this project follows data collection, data filtration, feature extraction, and text similarity check. Initially, to validate our pipeline, we have extracted the text from open-source platform dimension database and cleaned it as required. The important features are extracted from the resultant text that obtained from previous step. These summarized abstracts are used to calculate the similarity scores between main paper and its reference papers. Finally, a web application has been implemented.

Problem Definition:

When a person is willing to write a research paper in a particular topic, he goes through all the available work done on this topic in the past. They see all the methods used and the way of implementation. So, when they are creating their own methodology or work this past data will have some impact on them. So, there is a chance for high similarity of the work they cited and their own work. Our aim is to design a model that shows the similarity rate between a main paper and its references in order to know some of the insights like whether they are self-citated or not. It's not a smart idea to self-cite. It should only be used when it is absolutely required to show earlier work that is relevant to the current one.

Project Objectives

The main aim of this project is to know the similarity score of references mentioned in the main paper. The objectives of this project are:

i. To calculate the similarity rate between main paper and its citations.
ii. Categorize the similarity rate of references with main paper as self-citation, as relevant citations, as non-relevant citations etc.

Challenges

i. Extracting the sample data from the dimensions database for testing the pipeline.
ii. Preprocessing of the raw data.
iii. Generating an accurate meaning of the abstract using extractive summarization technique.
iv. Understanding the working and generating the word vectors by using the glove model.

Deliverables

We will get the similarity index of main paper and its references from three different similarity methods those are cosine, KL divergence and glove Word Embeddings methods. We can find out how many self-cited papers there are and how similar they are with the main paper.

Literature Review

- **Using Deep Learning Word Embeddings for Citations Similarity in Academic Papers:**

This paper was published in **International conference on BDCA Springer** by **Oumaima, Hourrane, Sara Mifrah, El Habib Benlahmar, Nadia Bouhriz, Mohamed Rachdi**. In this paper they have computed the similarity score between two citations cited in main paper. They have used word embeddings, here Initially they have taken whole paper data like abstract, title, authors and full text. From this they filtered only the references of that paper after all cleaning they have stored it in a csv file. Then they have tokenized into sentences, and these tokens were given to word vector model as input. For each sentence, they have used tf-idf vectorizer to assign the weights and stored it as vectors and finally used cosine similarity to find the similarity score between two citations.

- **A link-based similarity measure for scientific literature:**
  This paper was published in **19th International Conference on World Wide Web** by **Seok-Ho Yoon, Sang-Wook Kim, Sunju Park**. In this paper they have mentioned a literature retrevial service that finds a set of papers similar to the main paper. Here they are using link-based similarity measures like coupling, co-citation e.t.c to compute similarity score between two papers. Here they have fused this both techniques and established a new similarity measure called inter-connection. If we take papers A and B. Then, Coupling is based on number of papers referenced by both A and B. Co-Citation is based on number of papers which reference both A and B. Inter-connection is based on number of papers that are referenced by A reference B.

- **On computing text-based similarity in scientific literature:**
  This paper was published in **20th International Conference on World Wide Web** by **Seok-Ho Yoon, Sang-Wook Kim, Ji-Soo Kim, Won-Seok Hwang.** In this paper they have computed the similarity score between two papers by dividing the main paper into three parts as title, abstract, body and they are assigning the weights to each part that defines the percentage of content that need to be compared and after comparing the extracted content from three parts and computing the similarity score. They are tuning the ratios of weights for three parts to find the best ratio that gives highest similarity score.

Description of the Dataset

The dataset collected for this project is used for validating our pipeline. This dataset is extracted using web scraping from dimensions database, it is an open-source platform where you can find research papers, publications, and patents. We scrapped the data from this database. Our dataset contains the following eight elements.

- The column “main title” describes the title of the main research paper.
- The column “main_doi” describes the DOI number of main research paper, DOI stands for Digital Object Identifier.
- The column “main_abs” describes the abstract of main paper.
- The column “ref_title_lst” describes the list of titles that are cited in the main paper.
- The column “ref_doi_lst” describes the list of DOI numbers of cited papers.

- The column “ref_abs_lst” describes a list which has the abstracts of cited papers.
- The column “main_authors” describes the authors of main paper.
- The column “ref_authors” describes a list of lists which has the authors of cited papers.

Proposed Methodology

To validate our project pipeline, we have collected the data, preprocessed it, extracted the features from the text data, and lastly computed the similarity scores of each paper with its references .

The following flowchart represents the steps followed in the project:

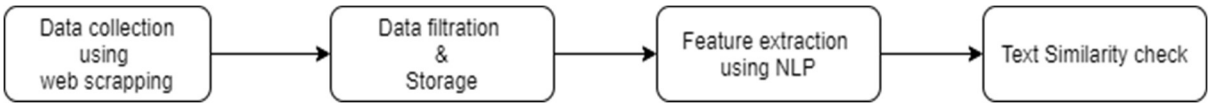

Fig 1: Schematic representation

1) Data Collection using web scraping:

The data we require here is open-source journal papers. We have collected the publications from Dimensions which are open to all. We have extracted the raw data from this dimension database using python modules. The modules we used are Requests and Beautiful soup to extract the data.

Requests: This module is used to make http requests to the specified URL. It provides the functionalities to manage both the request and the response.

Beautiful Soup: This module is used for extracting the data from any html or xml files.

Initially using requests module, we have sent the http request to that website in order to get the response object that contains the server responses like content, headers, cookies, status code and other details. Then from this response object we collected the content part and we have used beautiful soup module to extract the title, abstract, doi, references, and authors names from the whole content. This data was then stored in the lists and then the null values in the dataset were replaced with the label as ‘novalue’. Then using string indexing the html tags were cleared. This data is then replaced in the columns of the original data frame.

2) Data filtration & Storage:

After getting the partially cleaned data, this is partial because there are some labeled cells as ‘no value’. So, we should remove these cells before doing further processing. Using the basic list indexing we have removed all the respective rows that contain ‘no value’ in the reference

abstract column. Now as the data was cleaned, this data was then stored in a dataframe and the dataset after second step is shown as follows:

```
df1['ref_title_lst']=w1_title
df1['ref_doi_lst']=w1_doi
df1['ref_abs_lst']=w1_abs
df1['ref_authors']=w1_authors
```

```
df1
```

	main_title	main_doi	main_abs	ref_title_lst	ref_doi_lst	ref_abs_lst	main_authors	ref_authors
0	Enhanced Changeover Detection in Industry 4.0 ...	10.3390/s21175896	Changeover times are an important element when...	[Random Forests, Principal component analysis,...	[10.1007/978-1-4419-9326-7_5, 10.1002/wics.101...	[Random Forests were introduced by Leo Breiman...	[eddi miller, vladyslav borysenko, moritz heus...	[[Adele Cutler, D. Richard Cutler, John R. Ste...
1	Application of machine learning in the diagnos...	10.1097/bor.0000000000000612	PURPOSE OF REVIEW: In this review article, we ...	[Identifying Axial Spondyloarthritis in Electr...	[10.1002/acr.23140, 10.1007/s10067-014-2762-4,...	[OBJECTIVE: Large database research in axial s...	[jessica a. walsh, martin rozycki, esther yi, ...	[[Jessica A. Walsh, Yijun Shao, Jianwei Leng, ...
2	Designing and Comparing Performances of Image ...	10.4103/ijnm.ijnm_231_20	Objective: An image processing pipeline can ha...	[LIME: Low-Light Image Enhancement via Illumin...	[10.1109/tip.2016.2639450, 10.1109/cvpr.2018.0...	[When one captures images in low-light conditi...	[anil kumar pandey, shweta dhiman, sreedharan ...	[[Xiaojie Guo, Yu Li, Haibin Ling], [Chen Chen...
3	Machine Learning Methods for Precision Medicin...	10.18865/ed.30.s1.217	black box" models, and demonstrate the applica...	[Deep learning in neural networks: An overview...	[10.1016/j.neunet.2014.09.003, none, 10.1016/j...	[In recent years, deep artificial neural netwo...	[sanjay basu, james h faghmous, patrick doupe]	[[J\u00fcrgen Schmidhuber], [Sarah Poole, Shau...

Fig 2: Dataset

3)Feature Extraction using NLP:

After performing all the above steps, finally we will get our required input for the feature extraction step. The input for this step is abstract of the main paper and abstract of all the references of that main paper. Here we have used extractive summarization technique to extract the features from the abstract i.e, to get the summary of each abstract. We have used the nltk library to do the feature extraction process.

Extractive summarization: This technique takes the most important subset of sentences from the whole abstract. So, concatenation of all these important sentences gives the summary of each abstract.

NLTK: NLTK is known as Natural Language Toolkit. This module is used for preprocessing of unstructured data like human readable text in a way to be usable by computer programs.

Initially using regular expression commonly known as regex module in python, the brackets and all the characters that are not required for the summarization process are removed and using nltk module, initially the stop words are removed from the abstract and the abstract was made into two sets as in one of the set the abstract is tokenized into sentences and in other set it is tokenized into words. After tokenizing the relative frequency is calculated for each word and then based on these scores, for each sentence the score is calculated by summing all those word frequencies present in that sentence. Then taking the top three sentences based on the scores provides the summary of that abstract. At last, these summarized abstracts are then replaced in the respective columns of the original data frame.

4)Text Similarity check:

Finally, the similarity score of the abstract of each reference with the abstract of the main paper is calculated. Here we have used three similarity metrics i.e, Cosine similarity, Word Embeddings similarity, and Kl-divergence to show the similarity between the main paper abstract and its reference abstracts.

Cosine similarity: It measures the similarity between the documents irrespective of their size. It calculates the cosine angle between two-word vectors and as the smaller the angle the similarity will be higher.

Word Embeddings Similarity: A word embedding is a form of representing a word in the form of vectors. These vectors are formed by considering the relation between the words like the words with similar meaning will have similar type of representation. This shows that the two vectors are close to each other in the n-dimensional space. Here we have used the Glove model to obtain the word vectors. It is an unsupervised learning algorithm. It is based on the frequency of the word in some context in a large co-occurrence matrix.

Kl-divergence: This tells how much the two probability distributions differ. Here the Kl-divergence between any two distributions like let's say the two distributions as A and B, it is generally represented as KL(A || B).'||' indicates the divergence. It is calculated as follows:

$$\text{KL(A || B)} = \Sigma X * A(X) * log(A(X)/B(X))$$

TFIDF Vectorizer (Term Frequency Inverse Document Frequency Vectorizer.): Text is transformed into feature vectors, which are then fed into an estimator. Each token (word) is translated into a feature index in the matrix by vocabulary, which assigns a feature index to each unique token.

Here we have taken two different lists one for main abstracts and other for its reference abstracts. Initially we have used tfidf vectorizer to get the word vectors and inbuilt cosine similarity method is used to calculate the similarity between the main abstract and its reference abstracts and these scores are stored in a list. Then in the genism module using a glove pretrained model we have generated the word vectors and then used cosine similarity to get the similarity score. Then at last we have used Kl divergence to get the score and here the word vectors are formed using the above glove model. These three lists of scores are taken into a data frame and added a new column as a classifier one to distinguish between self-citation and other citation. At last, we have used styling methods in the pandas module to highlight and classify the self-cited references and other references.

In this way initially we have built a pipeline and validated this pipeline by using the data that we have extracted. Finally, we got the similarity scores between each main paper abstract and its respective reference paper abstracts.

5)Building the Website:

Finally, we made a user-friendly website using streamlit library in python. In this website one can check their similarity score between the main paper abstract and its reference paper abstract by using any of the above-mentioned methods. Here we have included the widgets like checkbox to select the similarity method, textboxes to enter the abstracts, slider to fix the threshold value so that the similarity scores of the reference papers will be displayed above that threshold value.

Streamlit: It is a module in python used to create interactive websites and adding the functionalities like integrating our normal python functions to the inbuilt methods of streamlit is easy and also it is easy to add the widgets like buttons, textboxes, and e.t.c

So, here initially the user enters the abstracts of the main paper and its respective citations and after selecting one of the similarity methods allows them to choose the threshold value and after selecting the threshold value and clicking on 'check' button it gives the similarity rate between the main abstract and its citations. The website looks as follows:

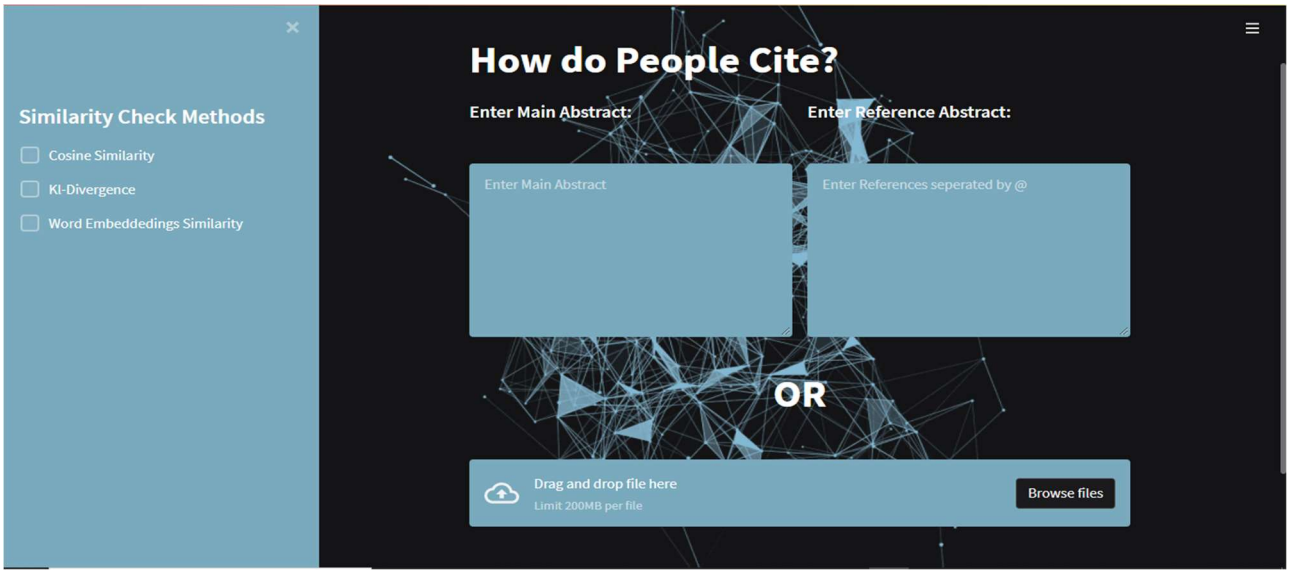

Fig 3: User interface

Experimental Results/Comparison:

Here we have achieved the similarity scores for 10 papers i.e., the 10 samples in our dataset. These scores are then stored in an excel file and we have classified the self-cited papers by highlighting the rows in the file. We have manually gone through the randomly selected papers and we found that the results which are obtained by the above methodology are appropriate.

Here the Fig 4 shows the similarity rate between a main abstract and its citations. So, we can see that there are 3 papers mentioned that was written by the main paper author that were referred to as self-citations.

Here all the scores given by the three metrics shows that the main paper abstract was not that much similar to the abstracts of self-cited papers.

	DOI	Cosine_similarity	KL-Divergence	Glove_model_similarity
0	10.1109/tip.2016.263	0.06326	-0.135044105	0.558334529
1	10.1109/cvpr.2018.0	0.18578	2.179905358	0.384078026
2	10.1109/iccv.2007.4	0.09087	6.726319717	0.544724822
3	10.1109/cvpr.2014.3	0.11909	1.257507074	0.502275825
4	10.1016/j.dsp.2013.(	0.12726	3.444593825	0.464456737
5	10.1259/bjr.2018010	0.04527	4.061175115	0.440392554
6	10.1109/cvpr.2017.2	0.05536	3.061277177	0.450984955
7	10.1109/tip.2006.88	0.22143	7.252716798	0.46958524
8	10.1007/978-3-662-0	0.20203	6.966043996	0.423003107
9	10.1016/0167-2789(9	0.2122	10.39428297	0.401585013
10	10.1109/tce.2017.01	0.10358	4.355625738	0.56033653
11	10.1109/tip.2003.818	0.13883	4.824943986	0.558960557
12	10.1109/tim.2007.91	0.11349	0.886517722	0.531319559
13	10.1109/iccv.2009.5	0.13592	4.694768842	0.436632991
14	10.1016/j.compmedi	0.13319	1.719261225	0.414194316
15	10.1109/tip.2007.90	0.12244	1.073807278	0.552453816
16	10.1093/bioinformati	0.1418	1.971506294	0.421066403
17	10.1109/tip.2017.27	0.18799	13.06803174	0.426771432
18	10.4103/wjnm.wjnm_	0.14974	6.521832919	0.422437489
19	10.1109/tip.2018.287	0.19927	-2.250582648	0.592932761
20	10.1097/mnm.00000	0.13151	6.216797995	0.487647921
21	none	0.08431	5.296308234	0.555203676
22	10.1117/1.1525793	0.12919	6.716609857	0.442013919
23	10.1117/1.2775482	0.05991	8.492229587	0.38791579
24	10.1097/mnm.00000	0.24508	8.458909965	0.575215697

Main_paper1 | Main_paper2 | Main_paper3 | Main_paper4 | Main_paper5 | Main_paper6 | Mair ...

Fig 4: Similarity scores of random sample

We designed a user-friendly web interface which allows the user to check the similarity of the main paper with its references, they can choose any of the three available similarity methods. The user can also fix a threshold value so that the similarity scores above that threshold value will be displayed.

Fig 5: Cosine similarity metric

Fig 6: Glove embedding similarity metric

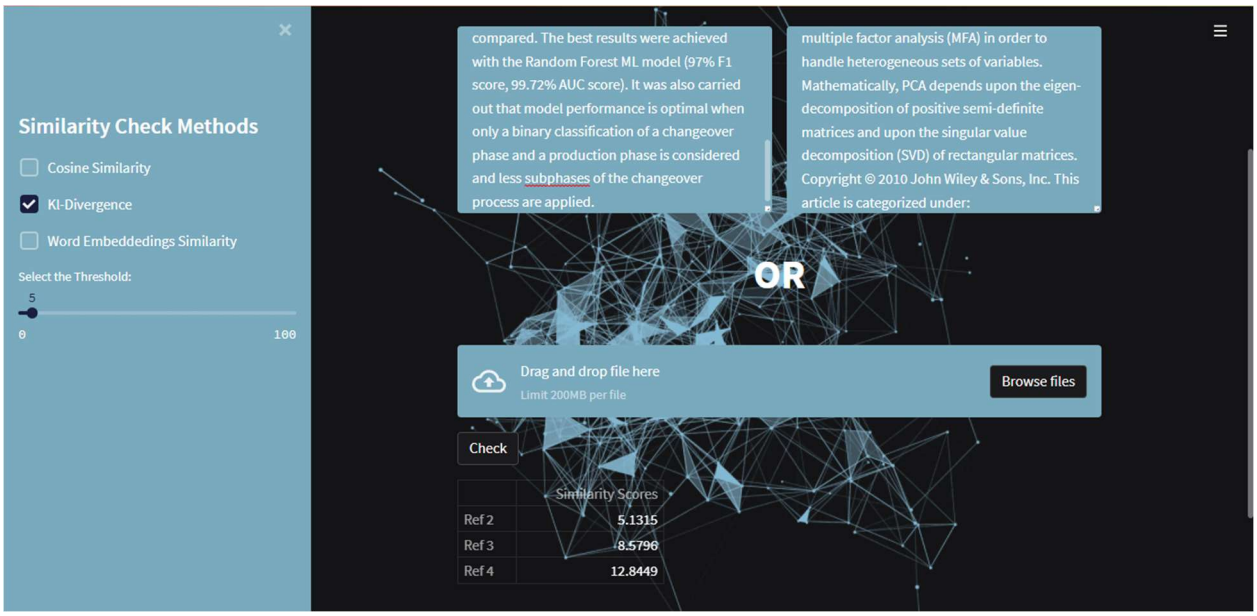

Fig 7: KI-Divergence similarity metric

Conclusions and Future Scope:

The above implementation of this research project provides the similarity rate between research paper and its respective citations. Although there are some models which check for plagiarism, they consider all available papers and data instead of citations. In this project we proposed a pipeline which has three different similarity metrics, and the user can choose any one of it to get the Relevance rate of the main paper with respect to its citations.

In future, to make our website more user-friendly, we are planning to add a new functionality where user can directly upload his research paper file (PDF/DOC) in our website. This option will make the process easy for the user. He can simply upload a pdf or word file and the result will be displayed below, based on the similarity metric he chooses and the threshold value he sets.

References

1) "**Document Plagiarism Detection Using a New Concept Similarity in Formal Concept Analysis**" by Jirapond Muangprathub, Siriwan Kajornkasirat, and Apirat Wanichsombat.

2) "**A Document Similarity Measure on Structured Heterogeneous Information Networks**" by Chenguang Wang; Yangqiu Song; Haoran Li; Ming Zhang; Jiawei Han

3) "**A fast and efficient semantic short text similarity metric**" by David Croft; Simon Coupland; Jethro Shell; Stephen Brown

4) "**An overview on extractive text summarization**" by Shohreh Rad Rahimi; Ali Toofanzadeh Mozhdehi; Mohamad Abdolahi,

5) "**An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures**" by Goldberger; Gordon; Greenspan.

6) **"GloVe: Global Vectors for Word Representation**" by Jeffrey Pennington, Richard Socher, Christopher D. Manning

7) https://www.mygreatlearning.com/blog/word-embedding/

8) https://nlp.stanford.edu/projects/glove/

9) https://pandas.pydata.org/docs/reference/api/pandas.ExcelWriter.html

10) https://machinelearningmastery.com/divergence-between-probability-distributions/

11) https://www.sciencedirect.com/topics/computer-science/extractive-summarization