An Evaluation for Various Text Summarization Algorithms on Blog Summarization Dataset

Shakshi Neha[1], Amanpreet Singh[2], Ishika Raj[3], Saveta Kumari[4]

[1, 2, 3, 4]Student, Department of Inforamtion Technology,HMRITM, Delhi, (State), India

Abstract: ***This paper aims at finding the accuracy of five different text summarization algorithms when applied on blogs and finding out the most accurate algorithm for creating a highly reliable Automatic summarization tool. The most pertinent aspect of summarization is to find a representative subset of the data, which contains the information of the entire set. Summarization technologies are used in a large number of sectors in industry today. Document summarization tries to automatically create a representative summary or abstract of the entire document, by finding the most informative sentences. Document Summarization of content on the Internet is an enterprise that is widely in demand in current times. Blogs form an integral part of formulating and disseminating popular opinions. A recent estimate revealed that nearly 152 million blogs exist on the internet, creating a dynamic and powerful echo chamber. Therefore, analyzing opinions generated via blogs is integral towards determining trends regarding customer spending, political views, entertainment reviews etc. Information obtained thus can be utilized to carry out further studies across fields like Consumer Spending, Anthropology, Psychology, Politics, Economics etc. Tools for carrying out these summarizations should be in sync with the requirement of the analysis. Different algorithms based on different mathematical and computing concepts are suited for different purposes. Therefore, an analysis of the algorithms itself is imperative towards determining the right approach to take towards analyzing opinions generated via the medium of blogs.***

Keywords: Text Summarization Algorithms, Comparative Study, Blog Summarization, Extractive Summarization, KLSum, LuHN, LexRank, TextRank, LSA

Abbreviations: KL, Kullback-LeiblerDaily; LSA, Latent Semantic Analysis; LSI, Latent Semantic Indexing

1. Introduction

With nearly 152 million blogs on the Internet, the influence of blogging in shaping public opinion is undeniable. Blogs form an intense echo chamber of voices at a global level, with widespread ramifications. Recent reports have revealed that the US election between Donald Trump and Hilary Clinton was deeply affected by the opinions that key people shared via social media. Therefore, to understand and analyze various factors behind the success/failure of one candidate over the other, it is imperative to analyze content generated via the medium of blogs.

Recent studies have revealed that attention spans of humans have fallen drastically. A recent estimate evaluated that humans focus their attention on their phones and other digital media for 8 seconds at best. Therefore, it has become important to provide concise, accurate and catchy information that can sustain the attention of the users. Many applications are specifically catering to the same. Organizations such as “Inshorts”, work on providing summarized content for reading news on the go. Similarly, the computational study of opinions, sentiments, evaluations, attitudes, appraisal, affects, views, emotions, subjectivity, etc., expressed in the text will impact various organizations and their decision-making process towards their business and the clientele [13-18].

Ever since the advent of social media, people have increasingly started sharing their personal experiences and opinions about anything and everything in reviews, forums, blogs, Twitter, micro-- blogs etc. Facebook has nearly 1.57 billion active users on its website and Twitter gets its contributions from nearly 317 million users (as of 2015). These staggering statistics provide an easy source for studying and evaluating public perception about situations, people, products etc [2-5]. Carrying out studies based on Summarization of opinions obtained is a cost-effective way of carrying out Focus group studies, surveys and other mediums of assessing consumer interest [19].

Many businesses and organizations focus on creating benchmark products and services via capitalizing on market intelligence using consultants, surveys and focus groups, etc. Individuals make decisions to purchase products or to use services based on opinions and reviews of others. The sum of opinions generated before, during and after this process provide important links towards carrying out a SWOT analysis for evaluating major and minor loopholes. The computational tools for the same should be selected accordingly.

The motivation behind this project was to study various standard algorithms in practice for text summarization and analyze their applicability over the domain of blogs centered around key aspects such as Precision, Recall, F--Score, Unit Overlap and Cosine Similarity[20]. All these parameters are mathematical concepts used to compare the accuracy of a summarization algorithm against a human-generated ‘perfect’ summary that incorporates the syntax and semantics of the English language, is capable of assessing logic, the flow of arguments, factual accuracy/inaccuracy and other relevant aspects.

Table 1: Parameters for finding overall accuracy of the summarization algorithms

Parameters on which accuracy is judged
Precision
Recall
F-- Score
Unit Overlap
Cosine Similarity

The summary generated should be similar to what a human would ideally generate after going through an article on the blog. For this, what elements are to be looked into, is what is to be evaluated. Understanding the advantages and drawbacks of the algorithm upon applying them to different categories of data set, we can ascertain as to which algorithm is appropriate for which sort of classification within the domain of internet blogs [1].

Thus, this paper aims at providing details for which algorithm is the most accurate for which type of blog text, and thus provide summaries with maximized accuracy by leveraging various text summarization algorithms.

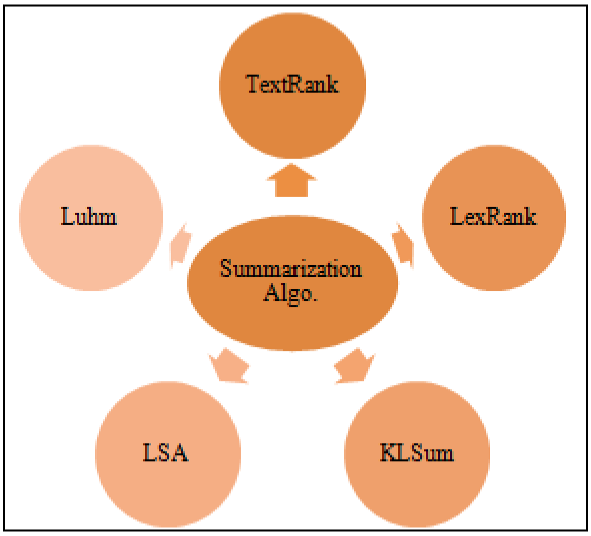


Figure 1: Visualization of the summarization algorithms being used for this study

2. Materials and Methods

The general methodology and process by which the experimental study has been performed is shown Fig. 2.

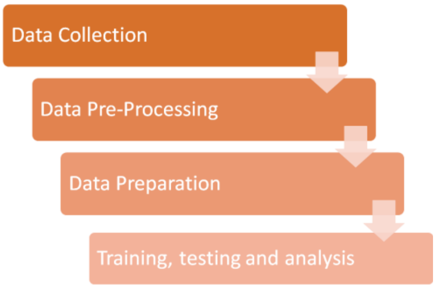


Figure 2: Major Steps of the Proposed Method

2.1 Collection of Data

In this study, we are using a dataset [1] which has been gathered from individually for the for various topics and it is provided with human generated and algorithm generated summaries for the training and testing of our models. The dataset contains of text files, 2 for each topic one with human generated summary and the other with algorithm generated summary and then perform feature extraction and reduction. tri-axial, meaning that they consider the 3-D coordinates of the surrounding.

2.2 Description of Data

The dataset contains human generated and algorithm generated summary for the topics of science, entertainment and sports

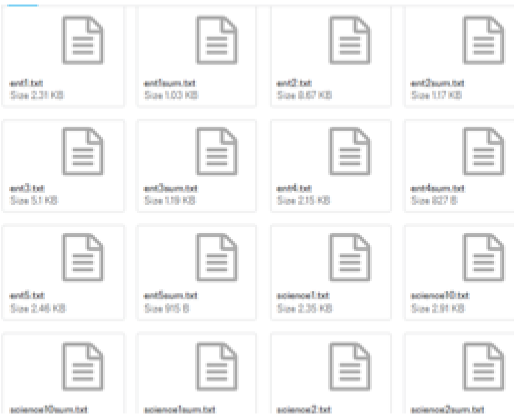


Figure 3: A small snippet of the file being used

2.3 Algorithms Used

LEXRANK: LexRank is a stochastic graph--based method for computing relative importance of textual units for Natural Language Processing. Extractive TS relies on the concept of sentence salience to identify the most important sentences in a document or set of documents. This method works firstly by generating a graph, composed of all sentences in the corpus. Every sentence represents one node, and the edges are similarity relationship between sentences in the corpus. Salience is typically defined in terms of the presence of particular important words or in terms of similarity to a centroid pseudo--sentence. In this model, a connectivity matrix based on intra--sentence cosine similarity is used as the adjacency matrix of the graph representation of sentences. Results show that degree--based methods (including LexRank) outperform both centroid--based methods and other systems in most of the cases.

TEXTRANK ALGORITHM: First, the words are assigned parts of speech, so that only nouns and adjectives (or some other combination for different applications) are considered. Then a graph of words is created. The words are the nodes/vertices (denoted V). Each word is connected to other words that are close to it in the text. In the graph, this is represented by the connections on the graph (denoted E). TextRank uses the structure of the text and the known parts of speech for words to assign a score to words that are keywords for the text. The algorithm gives more value to nodes with lots of connections, and gives more influence in steps to better connected nodes, so it reinforces itself and eventually finds its stable score. The algorithm is then run on the graph. Each node is given a weight of 1. Then the algorithm goes through the list of nodes and collects the influence of each of its inbound connections. The influence is usually just the value of the connected vertex (initially 1, but it varies) and then summed up to determine the new score for the node. Then these scores are normalized, the highest score becomes 1, and the rest are scaled from 0 to 1 based on that value. Each time through the algorithm gets closer to the actual "value" for each node, and it repeats until the values stop changing. In post-- processing, the algorithm takes the top scored words that have been identified as important and outputs them as key/important words. They can also be combined if they are used together often.

LUHN: German computer scientist H.P. Luhn came up with a method for automatically generating abstracts from scientific papers, along with many cool information theoretic ideas. As Luhn states in his paper, the algorithm really doesn't do anything fancy in terms of NLP or anything like that, it pretty much relies on frequency analysis and word spacing. The justification of measuring word significance by use frequency is based on the fact that a writer normally repeats certain words as he advances or varies his arguments and as he elaborates on an aspect of a subject. This means of emphasis is taken as an indicator of significance. The more often certain words are found in each other's company within a sentence, the more significance may be attributed to each of these words. Certain words are significant and repeated, the denser these clusters are the more valuable they become. Now this does depend upon a couple of key assumptions that Luhn made. He states that technical writers tend to refer to the same thing over and over with the same words and that even if they use alternate terms for their readers, they will eventually use very specific terms to describe their points.

KL SUM: KL (Kullback--Leibler) Divergence: KL---Divergence is a measure of the difference between two probability distributions: from a 'true' probability distribution P to an arbitrary probability 522 distribution Q. It is a measure between the unigram probability distributions learned from seen document set and new document set. The words considered to calculate KL divergence are the ones that are present in both the document sets. Since this measure is asymmetric, we considered a slight modification and calculated. This is also a sentence selection algorithm, where a target length for the summary is fixed (L words). According to this criterion, the objective of the summarizer is to find a set of sentences whose length is less than L words and whose unigram distribution is as similar as possible to the source document set. The global optimization of the criterion is exponential in the number of sentences in the document set D. As an approximation, KL--Sum uses a greedy optimization strategy.

LSA: Latent Semantic Analysis (LSA), also known as Latent Semantic Indexing (LSI) literally means analyzing documents to find the underlying meaning or concepts of those documents. If each word only meant one concept, and each concept was only described by one word, then LSA would be easy since there is a simple mapping from words to concepts as shown in Fig. 2.Latent Semantic Analysis arose from the problem of how to find relevant documents from search words. The fundamental difficulty arises when we compare words to find relevant documents, because what we really want to do is compare the meanings or concepts behind the words. LSA attempts to solve this problem by mapping both words and documents into a "concept" space and doing the comparison in this space.

2.4 Methodology

It is a very simple method in which we will start the web app and then input the blog which we want to summarize across the categories of Science, Entertainment and Sports. Then this input blog is fed to the summarization tool which will prompt you to choose the best tool for that category, on the basis of the radar graphs which shows the accuracy of the various algorithms on the basis of various accuracy determining factors such as F- Score, which you can choose or try using any other algorithm as per your choice. The algorithms which have been used here are extractive in nature. This summarization tool will generate an output and it's accuracy is determined by comparing it to a human generated output.

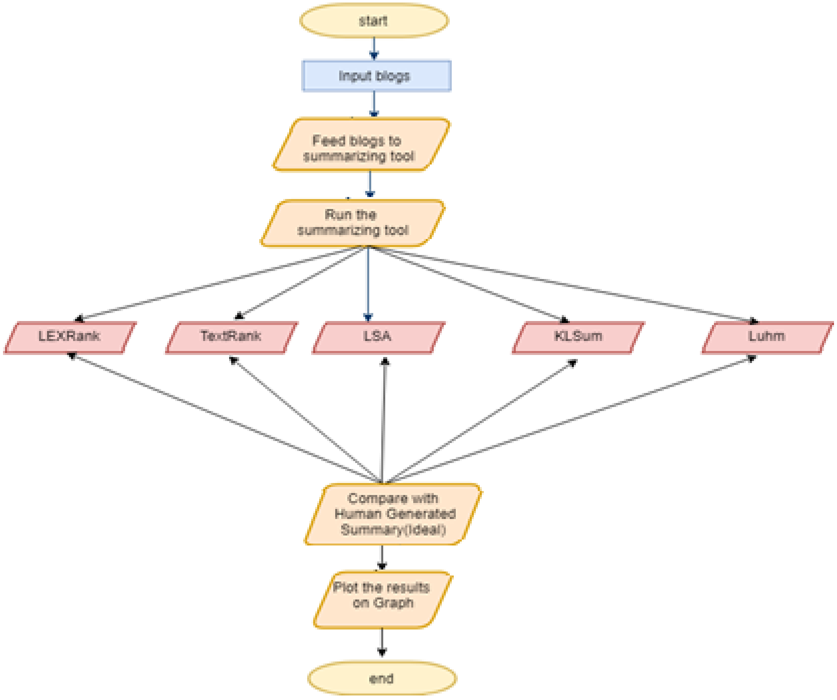

start
Input blogs
Feed blogs to summarizing tool
Run the summarizing tool
LEXRank
TextRank
LSA
KLSum
Luhm
Compare with Human Generated Summary(Ideal)
Plot the results on Graph
end

Figure 4: Methodology used for this study

This section is aimed at providing details about the extractive summarization which is being used in the summarization task. In this synopsis task, the programmed framework extricates objects from the whole assortment, without changing the items themselves. Instances of this incorporate key expression extraction, where the objective is to choose singular words or expressions to "tag" a record, and archive outline, where the objective is to choose entire sentences (without altering them) to make a short passage rundown. Extractive rundowns are figured by removing key content sections (sentences or entries) from the content, in view of measurable investigation of individual or blended surface level highlights, for example, word/state recurrence, area or sign words to find the sentences to be extricated. The "most significant" content is treated as the "most successive" or the "most well situated" content. Such a methodology hence maintains a strategic distance from any endeavors on profound content comprehension. They are adroitly basic, simple to actualize. Extractive content rundown procedure can be isolated into two stages: 1) Pre-Processing step and 2) Processing step.

Pre Processing is organized portrayal of the first content. It generally incorporates:

- Sentences limit recognizable proof. In English, sentence limit is related to nearness of spot toward the finish of sentence.
- Stop word Elimination-Common words with no semantics and which don't total pertinent data to the undertaking are killed.
- Stemming-The motivation behind stemming is to get the stem or radix of each word, which underscore its semantics. In Processing step, highlights impacting the pertinence of sentences are chosen and determined and afterward loads are doled out to these highlights utilizing weight learning technique.

Last score of each sentence is resolved utilizing Feature--weight condition. Top positioned sentences are chosen for definite outline.

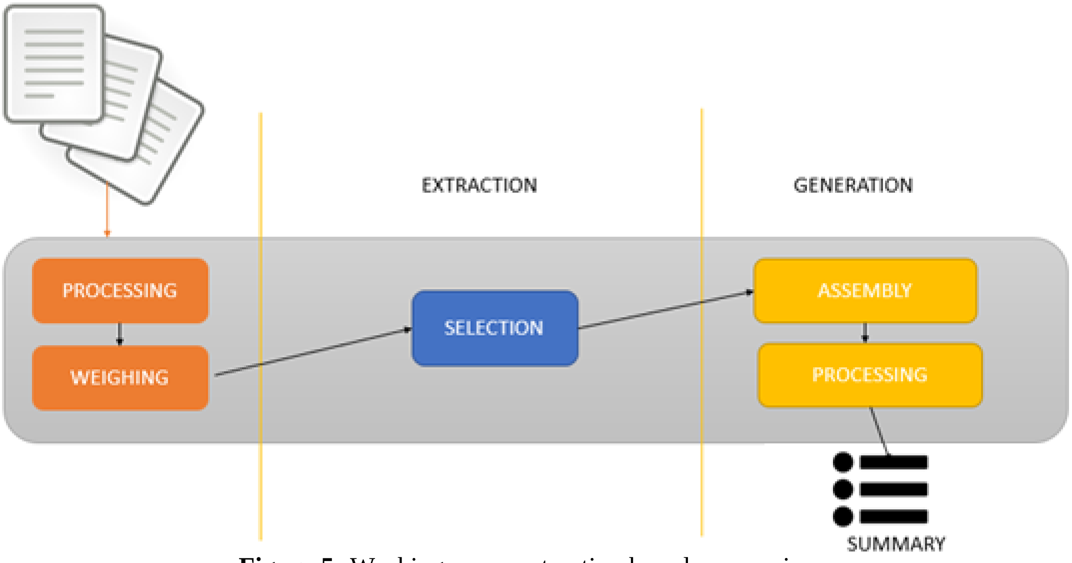


Figure 5: Working on an extraction based summarizar

2.5 Software Used

Command Prompt (CMD): Command Prompt is a command line interpreter application available in most Windows operating systems. It's used to execute entered commands. Most of those commands automate tasks via scripts and batch files, perform advanced administrative functions, and troubleshoot or solve certain kinds of Windows issues. Command Prompt is officially called Windows Command Processor, but it is also sometimes referred to as command shell or cmd prompt, or even by its filename, cmd.exe.

Python: Python is a universally useful, flexible and prominent programming language. It's extraordinary as a first language since it is brief and simple to peruse, and it is additionally a decent language to have in any developer's stack as it very well may be utilized for everything from web advancement to programming improvement and logical applications. Python is a deciphered, elevated level, universally useful programming language [9-12].

Python's structure reasoning underscores code intelligibility with its prominent utilization of noteworthy whitespace. Its language builds and item situated methodology plan to assist software engineers with composing clear, intelligent code for little and huge scale ventures. Python is progressively composed and trash gathered. Python is frequently portrayed as a "batteries included" language because of its far reaching standard library. Python utilizes whitespace space, instead of wavy sections or watchwords, to delimit squares. It has channel, map, and diminish capacities; list understandings, word references, sets and generator articulations. The standard library has two modules (itertools and functools) that execute useful apparatuses acquired from Haskell and Standard ML.

Pip Installer: Pip is a true standard bundle the board framework used to introduce and oversee programming bundles written in Python. Numerous bundles can be found in the default hotspot for bundles and their conditions — Python Package Index (PyPI). Most conveyances of Python accompany pip preinstalled. Python 2.7.9 and later (on the python2 arrangement), and Python 3.4 and later incorporate pip (pip3 for Python 3) as a matter of course.

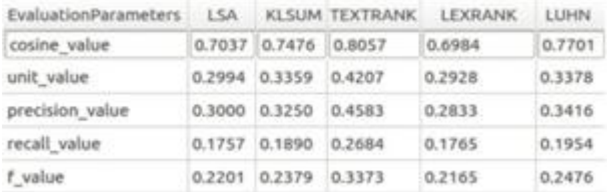
Python Flask: Flask (source code) is a Python web framework built with a small core and easy-to extend to other platforms. Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program. Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit. Flask is also easy to get started with as a beginner because there is little boilerplate code for getting a simple app up and running [7,8].

D3.js:D3.js (also known as D3, short for Data-Driven Documents) is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. It makes use of the widely implemented Scalable Vector Graphics (SVG), HTML5, and Cascading Style Sheets (CSS) standards. It is the successor to the earlier Protovis framework. In contrast to many other libraries, D3.js allows great control over the final visual result. Its development was noted in 2011, as version 2.0.0 was released in August 2011.Embedded within an HTML webpage, the JavaScript D3.js library uses pre-built JavaScript functions to select elements, create SVG objects, style them, or add transitions, dynamic effects or tooltips to them. These objects can also be widely styled using CSS. Large datasets can be easily bound to SVG objects using simple D3.js functions to generate rich text/graphic charts and diagrams [6].

3. Results and Discussion

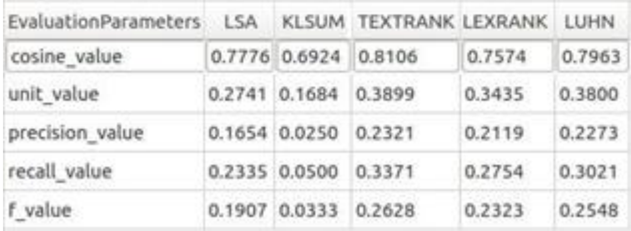
The accuracy of various parameters for the algorithms have been observed in Table II, Table III and Table IV respectively for categories of Science, Sports and Entertainment.

Table II: Results for evaluation parameters by all the five algorithms applied on Category 1: Science



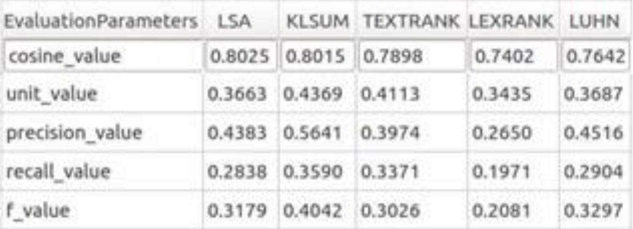
EvaluationParameters	LSA	KLSUM	TEXTRANK	LEXRANK	LUHN
cosine_value	0.7037	0.7476	0.8057	0.6984	0.7701
unit_value	0.2994	0.3359	0.4207	0.2928	0.3378
precision_value	0.3000	0.3250	0.4583	0.2833	0.3416
recall_value	0.1757	0.1890	0.2684	0.1765	0.1954
f_value	0.2201	0.2379	0.3373	0.2165	0.2476

Table III: Results for evaluation parameters by all the five algorithms applied on Category 2: Sports



EvaluationParameters	LSA	KLSUM	TEXTRANK	LEXRANK	LUHN
cosine_value	0.7776	0.6924	0.8106	0.7574	0.7963
unit_value	0.2741	0.1684	0.3899	0.3435	0.3800
precision_value	0.1654	0.0250	0.2321	0.2119	0.2273
recall_value	0.2335	0.0500	0.3371	0.2754	0.3021
f_value	0.1907	0.0333	0.2628	0.2323	0.2548

Table IV: Results for evaluation parameters by all the five algorithms applied on Category 3: Entertainment



EvaluationParameters	LSA	KLSUM	TEXTRANK	LEXRANK	LUHN
cosine_value	0.8025	0.8015	0.7898	0.7402	0.7642
unit_value	0.3663	0.4369	0.4113	0.3435	0.3687
precision_value	0.4383	0.5641	0.3974	0.2650	0.4516
recall_value	0.2838	0.3590	0.3371	0.1971	0.2904
f_value	0.3179	0.4042	0.3026	0.2081	0.3297

The comparison of the accuracy for the test and the train data of the above-mentioned algorithms have been shown through Fig 6, Fig 7 and Fig 8.

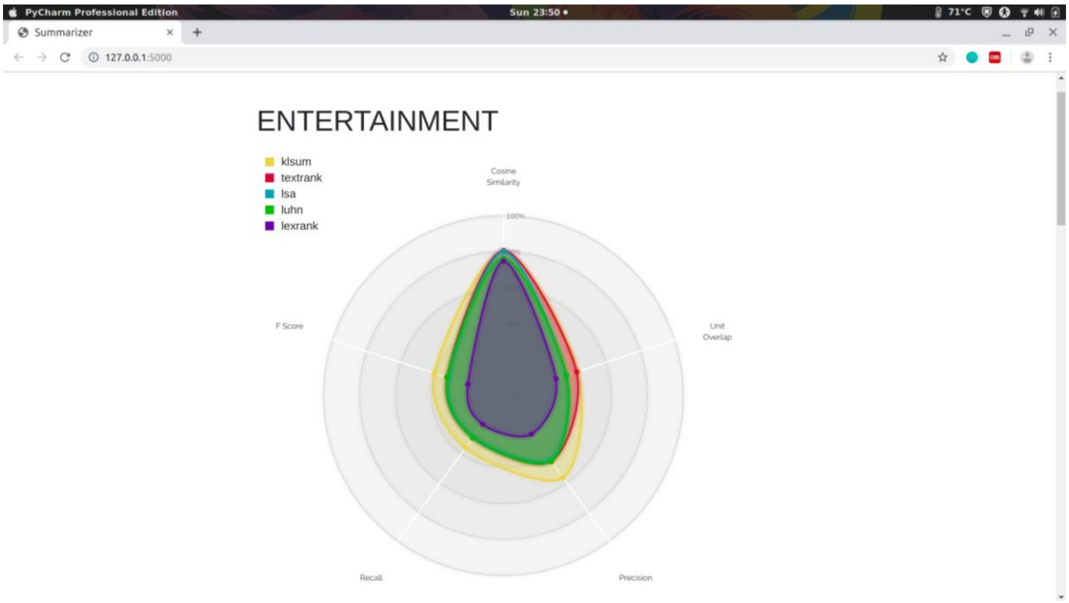


Figure 6: Graphical Representation of the accuracy of algorithms used for the algorithms using radar graph for entertainment

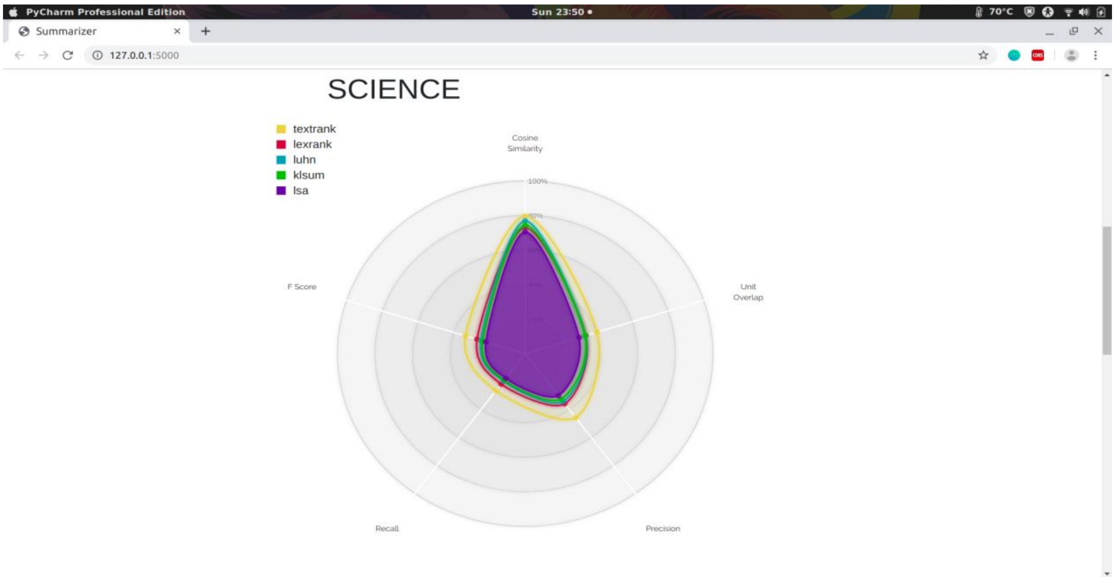


Figure 7: Graphical Representation of the accuracy of algorithms used for the algorithms using radar graph for science

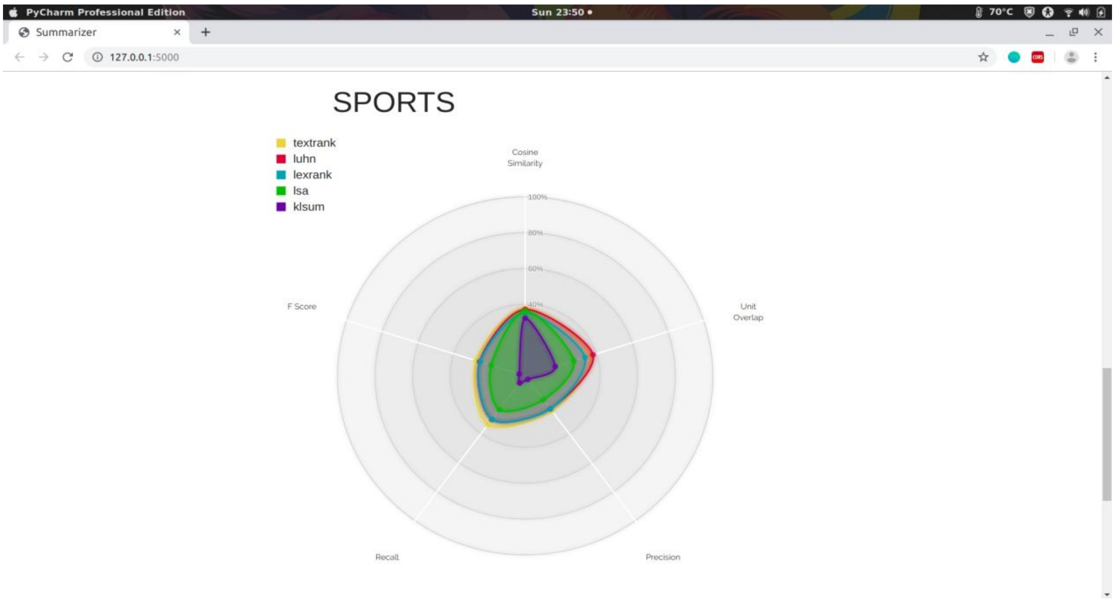


Figure 8: Graphical Representation of the accuracy of algorithms used for the algorithms using radar graph for sports

4. Conclusion

This experimental study has been conducted to find out the best suited algorithm for the purpose of blog summarization for blogs of different genres. This leads us to the conclusion that in some cases basic algorithms which give weightage to individual algorithms can lead to give us better results. This is not to discredit advanced algorithms; they don't seem to work quite as good on some types of data. We achieved what we wanted to do with the study and we have come to a conclusion that specific algorithms work good for a particular type of data.

5. Future Scope

Currently the analysis of the blogs by the algorithms has been done under three main sections namely Science, Sports and Entertainment. This can be extended to other genres as well, when we find out which algorithm will provide the best summarization for which type of algorithm. This paper provides the trial study for 3 categories which can be extended to other categories in the near future. It will help in making a generalized tool for all the blogs present online.

References

[1] Text Summarization by Machine Learning Bachelor's Thesis ,Matej GalloSpring 2016, Masaryk University

[2] Horacio Saggion, Thierry Poibeau. Automatic Text Summarization: Past,Present and Future. T. Poibeau; H. Saggion. J. Piskorski, R. Yangarber. Multi---source, Multilingual Information Extraction and summarization, Springer,pp.3---13, 2012, Theory and Applications of Natural Language Processing,978---3---642---28569---1.

[3] A Survey on Automatic Text Summarization Dipanjan Das Andr´e F.T.Martins Language Technologies Institute Carnegie Mellon University{dipanjan, afm}@cs.cmu.edu November 21, 2017

[4] Summarization Evaluation: An Overview, Inderjeet MANI The MITRECorporation, Sunset Hills Road Reston, VA 20190---5214, USA

[5] Comparing Twitter Summarization Algorithms for Multiple Post Summaries,David Inouye and Jugal K. Kalita ,School of Electrical and Computer Engineering Georgia Institute of Technology Atlanta, Georgia "d3 Releases". Github.com.

[6] Community web page for Flask https://github.com/pallets/flask/releases

[7] History". Pocoo Team. Archived from the original on 2017-11-19.

[8] Python Developers Survey 2018". www.jetbrains.com. 2018-11-01.

[9] Ronacher, Armin. "Werkzeug The Python WSGI Utility Library". palletsprojects.com. Retrieved 27 May 2018.

[10] Chowdhury, SM Mazharul Hoque. "A Review Paper on Comparison of Different Algorithm Used in Text Summarization." *Intelligent Data Communication Technologies and Internet of Things: ICICI 2019*: 114.

[11] Chen, Yen-Chun, et al. "Distilling the Knowledge of BERT for Text Generation." *arXiv preprint arXiv:1911.03829* (2019).

[12] Basak, Setu, MD Delowar Hossain Gazi, and SM Mazharul Hoque Chowdhury. "A Review Paper on Comparison of Different Algorithm Used in Text Summarization." *International Conference on Intelligent Data Communication Technologies and Internet of Things*. Springer, Cham, 2019.

[13] Iwendi, Celestine, et al. "An Efficient and Unique TF/IDF Algorithmic Model-Based Data Analysis for Handling Applications with Big Data Streaming." *Electronics* 8.11 (2019): 1331.

[14] Dash, Abhisek, et al. "Summarizing User-generated Textual Content: Motivation and Methods for Fairness in Algorithmic Summaries." *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019): 172.

[15] Gharavi, Erfaneh, Hadi Veisi, and Paolo Rosso. "Scalable and language-independent embedding-based approach for plagiarism detection considering obfuscation type: no training phase." *Neural Computing and Applications*: 1-15.

[16] Jha, Nitesh Kumar, and Arnab Mitra. "Introducing Word's Importance Level-Based Text Summarization Using Tree Structure." *International Journal of Information Retrieval Research (IJIRR)* 10.1 (2020): 13-33.

[17] Chadha, Janit. "Semantic based Automatic Text Summarization based on Soft Computing." (2019).