Advances in Science, Technology and Engineering Systems Journal Vol. 5, No. 3, 349-354 (2020)

www.astesj.com

ASTESJ

ISSN: 2415-6698

Sentence retrieval using Stemming and Lemmatization with Different Length of the Queries

Ivan Boban[*,1], Alen Doko[2], Sven Gotovac[3]

[1]*Faculty of Mechanical Engineering, Computing and Electrical Engineering, University of Mostar, 88 000, Bosnia and Herzegovina*

[2]*Institut for Software Technology, German Aerospace Center, 28199, Germany*

[3]*Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, 21 000, Croatia*

ARTICLE INFO

Article history:
Received: 25 March, 2020
Accepted: 13 May, 2020
Online: 11 June, 2020

Keywords:
Sentence retrieval
TF-ISF
Data pre-processing
Stemming
Lemmatization

ABSTRACT

In this paper we focus on Sentence retrieval which is similar to Document retrieval but with a smaller unit of retrieval. Using data pre-processing in document retrieval is generally considered useful. When it comes to sentence retrieval the situation is not that clear. In this paper we use $TF - ISF$ (term frequency - inverse sentence frequency) method for sentence retrieval. As pre-processing steps, we use stop word removal and language modeling techniques: stemming and lemmatization. We also experiment with different query lengths. The results show that data pre-processing with stemming and lemmatization is useful with sentences retrieval as it is with document retrieval. Lemmatization produces better results with longer queries, while stemming shows worse results with longer queries. For the experiment we used data of the Text Retrieval Conference (TREC) novelty tracks.

1. Introduction

Sentence retrieval consists of retrieving relevant sentences from a document base in response to a query [1]. The main objective of the research is to present the results of sentence retrieval with $TF - ISF$ (term frequency – inverse sentence frequency) method using data pre-processing consisting of stop word removal and language modeling techniques, stemming and lemmatization. Stemming and lemmatization are data reduction methods [2].

Previous work mentions the usefulness of the pre-processing steps with document retrieval. Contrary to that when it comes to sentence retrieval the usefulness of pre-processing is not clear. Some paper mentions it vaguely without concrete results. Therefore, we will try to clarify the impact of stemming and lemmatization on sentence retrieval and present this through test results. As additional contribution we will test and discuss how pre-processing impacts sentence retrieval with different query lengths. Because sentence retrieval is similar to document retrieval and stemming and lemmatization techniques have shown a positive effect on document retrieval, we expect these procedures to have a beneficial effect on sentence retrieval as well.

In our tests we use the State of The Art $TF - ISF$ method in combination with stemming and lemmatization. For testing and evaluation, data from the TREC novelty tracks [3 - 6], were used. This paper is organised as follows. Previous work is shown in section 2., an overview of methods and techniques is shown in section 3., in section 4. data set and experiment setup are presented, result and discussion are presented in section 5 and 6, and the conclusion is given in section 7.

2. Previous research

2.1. Sentence retrieval in document retrieval

Sentence retrieval is similar to document retrieval, and document retrieval methods can be adapted for sentence retrieval [7]. When it comes to Document retrieval the State of The Art $TF - IDF$ (term frequency – inverse document frequency) method is commonly combined with preprocessing steps stemming and stop word removal. However, sentences of a document have an important role in retrieval procedures. In the paper [8], research results have shown that the traditional $TF - IDF$ algorithm has been improved with sentence-based processing on keywords helping to improve precision and recall.

2.2. Document retrieval with stemming and lemmatization

Stemming and lemmatization are language modeling techniques used to improve the document retrieval results [9]. In [10] the authors showed the impact of stemming on document retrieval, using short and long queries. The paper [10] proved that

[*] Ivan Boban, +38763484395 & ivan.boban@hotmail.com

https://dx.doi.org/10.25046/aj050345

stemming has a positive effect on IR (the ranking of retrieved documents was computed using $TF-IDF$). Paper [11] compares document retrieval precision performances based on language modeling techniques, stemming and lemmatization. In papers [9, 11] it is shown that language modeling techniques (stemming and lemmatization) can improve document retrieval.

2.3. $TF-ISF$ sentence retrieval with stemming and lemmatization

When it comes to stemming and lemmatization and their impact on the $TF-ISF$ method, the results are not clearly presented, unlike the $TF-IDF$ method, where the impact is clear. In paper [12] stemming is mentioned in context of sentence retrieval. The paper states that stemming can improve recall but can hurt precision because words with distinct meanings may be conflated to the same form (such as "army" and "arm"), and that these mistakes are costly when performing sentence retrieval. Furthermore, paper [12] states that terms from queries that are completely clear and unambiguous, can match with sentences that are not even from the same topic after the stop word removal and stemming process.

3. Using $TF-ISF$ method for sentence retrieval in combination with stemming and lemmatization

For sentence retrieval in this paper we use $TF-ISF$ method based on vector space model of information retrieval. $TF-ISF$ was also used in [13, 14].

The ranking function is as follows:

$$R(s|q)=\sum_{t\in q}\log(tf_{t,q}+1)\log(tf_{t,s}+1)\log\left(\frac{n+1}{0.5+sf_t}\right) \qquad (1)$$

Where:

- $tf_{t,q}$ – number of appearances of the term t in a query,
- $tf_{t,s}$ – number of appearances of the term t in a sentence,
- n – is the number of sentences in the collection and
- sf_t – is the number of sentences in which the term t appears,

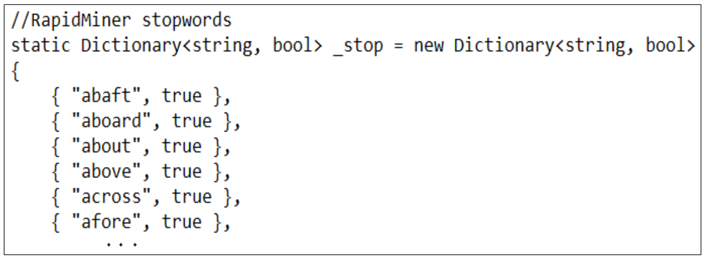
The search engine for sentence retrieval with ranking function (1) uses data pre-processing consisting of following three steps: stop word removal, stemming and lemmatization. In information retrieval, there are many words that present useless information. Such words are called stop words. Stop words are specific to each language and make the language functional but they do not carry any information (e.g. pronouns, prepositions, links, ...) [15]. For example, there are around 400 to 500 stop words in the English language [15]. Words that often appear at the collection level can be eliminated through some tools like RapidMiner or programmatically, so as not to have an impact on ranking.

There are several different methods for removing stop words presented in [16] like:

- Z-Methods;
- The Mutual Information Method (MI);
- Term Based Random Sampling (TBRS);

In this paper we used the classic method of removing stop words based on a previously compiled list of words.

Part of the list of words to be removed by pre-processing is shown in Figure 1 which is a snippet from our source code:



```
//RapidMiner stopwords
static Dictionary<string, bool> _stop = new Dictionary<string, bool>
{
    { "abaft", true },
    { "aboard", true },
    { "about", true },
    { "above", true },
    { "across", true },
    { "afore", true },
        ...
```

Figure 1: Example part of the stop words list

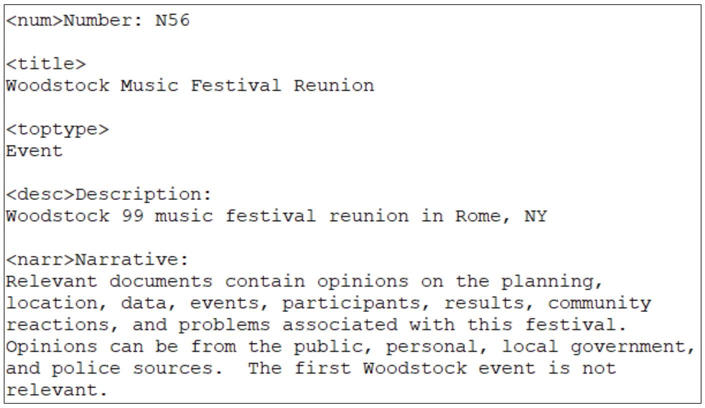
Stemming refers to the process of removing prefixes and suffixes from words. When it comes to stemming, there are many different algorithms. One of them use the so called "bag of words" that contain words that are semantically identical or similar but are written as different morphological variants. By applying stemming algorithms, words are reduced to their root, allowing documents to be represented by the stems of words instead of original words. In information retrieval, stemming is used to avoid mismatches that may undermine recall. If we have an example in English where a user searches for a document entitled "How to write" over which he raises the query "writing", it will happen that the query will not match the terms in the title. However, after the stemming process, the word "writing" will be reduced to its root (stem) "write", after which the term will match the term in the title. We use the Porter's stemmer, which is one of the most commonly used stemmers, which functions on the principle that it applies a set of rules and eliminates suffixes iteratively. Porter's stemmer has a well-documented set of constraints, so if we have the words "fisher", "fishing", "fished", etc., they get reduced to the word "fish" [17]. Porter's stemmer algorithm is divided into five steps that are executed linearly until the final word shape is obtained [18]. In paper [19] it was proposed modified version of the Porter stemmer.

Lemmatization is an important pre-processing step for many applications of text mining, and also used in natural language processing [20]. Lemmatization is similar to stemming as both of them reduce a word variant to its "stem" in stemming and to its "lemma" in lemmatizing [21]. It uses vocabulary and morphological analysis for returning words to their dictionary form [11, 20]. Lemmatization converts each word to its basic form, the lemma [22]. In the English language lemmatization and stemming often produce same results. Sometimes the normalized/basic form of the word may be different than the stem e.g. "computes", "computing", "computed" is stemmed to "comput", but the lemma of that words is "compute" [20]. Stemming and lemmatization have an important role in order to increase the recall capabilities [23, 24].

4. Data set used and experiment setup

Testing was performed on data from the TREC Novelty tracks [3]-[5]. Three Novelty Tracks were used in the experiment: TREC 2002, TREC 2003 and TREC 2004. Each of the three Novelty Tracks has 50 topics. Each topic consisting of "*titles*", "*descriptions*" and "*narratives*".

Figure 2. shows a part of the file related to one topic labeled "N56" from TREC 2004 novelty track with parts "*titles*", "*descriptions*" and "*narratives*" [25]:

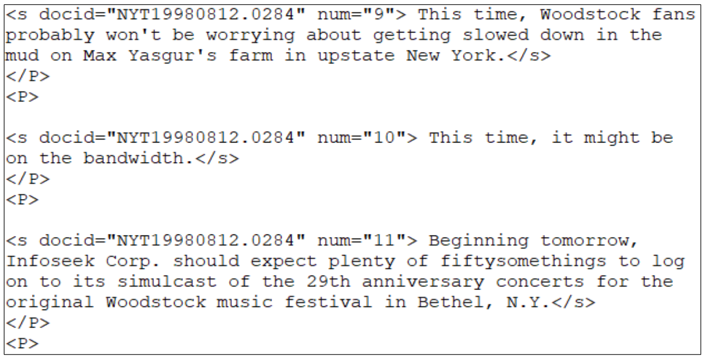


```
<num>Number: N56

<title>
Woodstock Music Festival Reunion

<toptype>
Event

<desc>Description:
Woodstock 99 music festival reunion in Rome, NY

<narr>Narrative:
Relevant documents contain opinions on the planning,
location, data, events, participants, results, community
reactions, and problems associated with this festival.
Opinions can be from the public, personal, local government,
and police sources.  The first Woodstock event is not
relevant.
```

Figure 2: Example of the topic N56 from TREC 2004 novelty track

Two types of queries were used in the experiment. The short query uses the <*title*> part and the longer query the <*desc*> part. To each of 50 topics 25 documents were assigned. Each of the 25 documents contains multiple sentences.

Figure 3 shows a snippet from one of the 25 documents assigned to topic N56, which has multiple sentences, which are in the format: *<s docid="xxx" num="x">Content of Sentence </s>*.



```
<s docid="NYT19980812.0284" num="9"> This time, Woodstock fans
probably won't be worrying about getting slowed down in the
mud on Max Yasgur's farm in upstate New York.</s>
</P>
<P>

<s docid="NYT19980812.0284" num="10"> This time, it might be
on the bandwidth.</s>
</P>
<P>

<s docid="NYT19980812.0284" num="11"> Beginning tomorrow,
Infoseek Corp. should expect plenty of fiftysomethings to log
on to its simulcast of the 29th anniversary concerts for the
original Woodstock music festival in Bethel, N.Y.</s>
</P>
<P>
```

Figure 3: Example of part within the document from TREC 2004 novelty track

In our experiment we extract single sentences from the collection. During the extraction we assign a *docid* (document identifier) and *num* (sentence identifier) to each sentence.

Three data collections were used, (Table 1 and Table 2).

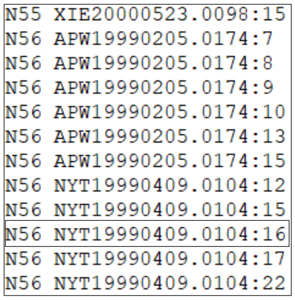
Table 1: Description of three data collections

Name of the collection	Number of topics	Number of queries (title/desc)
TREC 2002	50	50
TREC 2003	50	50
TREC 2004	50	50

Table 2: Overview of number of sentences per document

Name of the collection	Number of documents per topic	Number of sentences
TREC 2002	25	57792
TREC 2003	25	39820
TREC 2004	25	52447

For results evaluation one file is available which contain a list of relevant sentences [25]. Figure 4 shows a snippet from the relevant sentence file.



```
N55 XIE20000523.0098:15
N56 APW19990205.0174:7
N56 APW19990205.0174:8
N56 APW19990205.0174:9
N56 APW19990205.0174:10
N56 APW19990205.0174:13
N56 APW19990205.0174:15
N56 NYT19990409.0104:12
N56 NYT19990409.0104:15
N56 NYT19990409.0104:16
N56 NYT19990409.0104:17
N56 NYT19990409.0104:22
```

Figure 4: File with list of relevant sentences

The format of the list of relevant sentences shown in Figure 4 is: N56 NYT19990409.0104:16.

Where:

- N56 - indicates the topic number,
- NYT19990409.0104 - indicates a specific document,
- 16 - indicates the sentence number - identifier.

N56 NYT19990409.0104: 16 defines sentence "16" of document "NYT19990409.0104" as relevant to topic "N56".

Using the presented TREC data we test at first the $TF - ISF$ method without any pre-processing. Then we test the same $TF - ISF$ method with stemming and with lemmatization. All three tests we do twice: First time with short queries and second time with long queries. In all of our tests we use stop word removal.

We denote the baseline method as $TF - ISF$, the method with stemming we denote as $TF - ISF_{stem}$ and the method with lemmatization we denote as $TF - ISF_{lem}$.

5. Result and discussion

As already mentioned, we wanted to test if data pre-processing steps stemming and lemmatization affect the sentence retrieval. Also, we want to analyse if the effect of pre-processing is different when using different query lengths. For test evaluation we used standard measures: P@10, R-precision and Mean Average Precision (MAP) [26, 27].

Precision at x or P@x can be defined as:

$$P@x(q_j) = \frac{\begin{array}{c}number\ of\ relevant\ sentences\\ within\ top\ x\ retrieved\end{array}}{x} \qquad (2)$$

The P@10 values shown in this paper refer to average P@10 for 50 queries.

R-precision can be defined as [26]:

$$R - precision\ = \frac{r}{|Rel|} \qquad (3)$$

Where:

- $|Rel|$ is the number of relevant sentences to the query,

- *r* is the number of relevant sentences in top *|Rel|* sentences of the result.

As with P@10 we also calculate the average R-precision for 50 queries. Another well-known measure is Mean Average Precision which gives similar results to R-precision.

Mean Average Precision and R-precision is used to test high recall. High recall means: It is more important to find all of relevant sentences even if it means searching through many sentences including many non-relevant. In opposite to that P@10 is used for testing precision.

Precision in terms of information retrieval means: It is more important to get only relevant sentences than finding all of the relevant sentence.

For result comparison we used two tailed paired *t-test* with significance level $\alpha=0.05$. Statistically significant improvements in relation to the base $TF-ISF$ method (without data pre-processing) are marked with a (*). The results of our tests on different data sets are presented below in tabular form. Table 3 shows the results of our tests on TREC 2002 collection with short queries presented on Figure 2 and labeled with *<title>*.

Table 3: Test results using TREC 2002 collection with short query

TREC 2002 - title			
	$TF-ISF$	$TF-ISF_{stem}$	$TF-ISF_{lem}$
P@10	0,304	0,328	0,34
MAP	0,1965	*0,2171	*0,2149
R-prec.	0,2457	0,2629	0,2575

From Table 3 we see that the method with stemming $TF-ISF_{stem}$ and the method with lemmatization $TF-ISF_{lem}$ show statistically significant better results in comparison to the baseline method, when it comes to MAP measure.

Table 4 shows the results of our tests on TREC 2002 collection with longer queries presented on Figure 2 and labeled with *<desc>*.

Table 4: Test results using TREC 2002 collection with longer query

TREC 2002 - description			
	$TF-ISF$	$TF-ISF_{stem}$	$TF-ISF_{lem}$
P@10	0,332	0,296	0,324
MAP	0,2075	0,2157	*0,2176
R-prec.	0,2490	0,2601	0,2570

Only $TF-ISF_{lem}$ provides better results and statistically significant differences in relation to the base $TF-ISF$ method (without data pre-processing), when the MAP measure is used. We can see that stemming performs a little worse when it comes to longer queries in relation to the base $TF-ISF$ method. Table 5 and Table 6 show the results of our tests using TREC 2003 collection with short and longer queries respectively.

Table 5 show that $TF-ISF_{stem}$ and $TF-ISF_{lem}$ provide better results and statistically significant differences in relation to the base $TF-ISF$ method, when the MAP and R-prec. measure are used.

Table 5: Test results using TREC 2003 collection with short query

TREC 2003 - title			
	$TF-ISF$	$TF-ISF_{stem}$	$TF-ISF_{lem}$
P@10	0,692	0,712	0,714
MAP	0,5765	*0,5911	*0,5887
R-prec.	0,5470	*0,5611	*0,5593

Table 6: Test results using TREC 2003 collection with longer query

TREC 2003 - description			
	$TF-ISF$	$TF-ISF_{stem}$	$TF-ISF_{lem}$
P@10	0,734	0,738	0,738
MAP	0,5914	*0,6059	*0,6049
R-prec.	0,5617	0,5699	*0,5750

Table 6 is shows that lemmatization keeps showing statistically significant better results even with long queries, unlike the method with that uses stemming. Table 7 and Table 8 show the results of our tests on TREC 2004 collection with short and longer queries.

Table 7: Test results using TREC 2004 collection with short query

TREC 2004 - title			
	$TF-ISF$	$TF-ISF_{stem}$	$TF-ISF_{lem}$
P@10	0,434	0,448	0,444
MAP	0,3248	*0,3390	0,3331
R-prec.	0,3366	0,3385	0,3387

Table 8: Test results using TREC 2004 collection with longer query

TREC 2004 - description			
	$TF-ISF$	$TF-ISF_{stem}$	$TF-ISF_{lem}$
P@10	0,498	0,49	0,498
MAP	0,3540	*0,3644	*0,3635
R-prec.	0,3583	0,3688	0,3699

Table 7 shows that $TF-ISF_{stem}$ method with short queries, provides better results in comparison to the baseline, when it comes to MAP measure. Table 8 shows that with longer queries both methods shows statistically significant better results. When taking a look at all the tables above we see that stemming and lemmatization often give statistically significant better results when it comes to MAP and R-Prec. Therefore, we can assume that these pre-processing steps have similar positive effect on sentence retrieval as they have on document retrieval. Let us analyse how query length impacts our two methods ($TF-ISF_{stem}$ and $TF-ISF_{lem}$). Table 9 shows an overview of the overall number of statistically significant better results for four pairs (stem - short queries, stem - long queries, lem - short queries, lem - long queries).

As we can see $TF-ISF_{stem}$ seems to go better with short queries and $TF-ISF_{lem}$ seems to go better with long queries. At the moment we do not have enough data to examine this behaviour further. But that will be a topic for further research of us.

Table 9. Performance of $TF-ISF_{stem}$ and $TF-ISF_{lem}$ in regard to query length

Number of statistically significant better results with methods		
	$TF-ISF_{stem}$	$TF-ISF_{lem}$
Short queries	4	3
Long queries	2	4

6. Additional result analysis

To understand the reasons why we have better results using $TF-ISF_{stem}$ and $TF-ISF_{lem}$ methods in relation to the base $TF-ISF$ method, we analysed the positioning of one relevant sentence in the test results of the different methods when using the TREC 2003 collection. Table 10 and 11 analyze one sentence ("*Two of John F. Kennedy Jr.,'s cousins, David and Michael, both sons of Robert Kennedy, died young, the latter of a drug overdose in 1984, as did four Kennedys of the preceding generation*") from topic "N42" in two different scenarios using short and long query and with 3 different methods. One of them is baseline method and the remaining two use stemming and lemmatization. As already said we match the sentence with two different queries: Short query ("*John F. Kennedy, Jr. dies*") and long query ("*John F. Kennedy, Jr. was killed in a plane crash in July 1998.*").

Table 10 shows the matching of the sentence with short query with resulting sentence position for each of the three methods.

Table 10: Analysis of the sentence and the short query with the different methods

$TF-ISF$	
Short query	"**john**","**f**"," **kennedy** ","**jr**","dies"
Sentence content	"**john**","**f**","**kennedy**","**jr**","s","cousins", "david","michael","sons","robert", "**kennedy**","died","young","latter","drug", "overdose","kennedys","preceding", "generation"
Sentence position	(24)
$TF-ISF_{stem}$ *(stemming)*	
Short query	"**john**","**f**","**kennedi**","**jr**","**die**"
Sentence content	"**john**","**f**","**kennedi**","**jr**","s","cousin","david", "michael","son","robert","**kennedi**","**die**", "young","latter","drug","overdos","**kennedi**", "preced","generat"
Sentence position	(1)
$TF-ISF_{lem}$ *(lemmatization)*	
Short query	"**john**","**f**","**kennedy**","**jr**","**die**",
Sentence content	"**john**","**f**","**kennedy**","**jr**","s","cousin","david", "michael","son","robert","**kennedy**","**die**", "young","latter","drug","overdose","**kennedy**", "precede","generation"
Sentence position	(1)

Table 11 shows the same as Table 10 but with long query.

Table 11: Analysis of the sentence and the long query with the different methods

$TF-ISF$	
Long query	"**john**","**f**","**kennedy**","**jr**","killed","plane", "crash","july","1998",
Sentence content	"**john**","**f**","**kennedy**","**jr**","s","cousins", "david","michael","sons","robert","**kennedy**", "died","young","latter","drug","overdose", "kennedys", "preceding","generation"
Sentence position	(67)
$TF-ISF_{stem}$ *(stemming)*	
Long query	"**john**","**f**","**kennedi**","**jr**","kill","plane","crash" , "juli","1998"
Sentence content	"**john**","**f**","**kennedi**","**jr**","s","cousin","david", "michael","son","robert","**kennedi**","die", "young","latter","drug","overdos","**kennedi**", "preced", "generat"
Sentence position	(63)
$TF-ISF_{lem}$ *(lemmatization)*	
Long query	"**john**","**f**","**kennedy**","**jr**","kill","plane", "crash", "july","1998",
Sentence content	"**john**","**f**","**kennedy**","**jr**","s","cousin","david", "michael","son","robert","**kennedy**","die", "young","latter","drug","overdose","**kennedy**", "precede","generation"
Sentence position	(62)

Table 10 and Table 11 shows how stemming and lemmatization help to position relevant sentences closer to the top of search result.

More precisely, every match of words between query and sentence is marked bold. Matches that occurred with stemming or lemmatization but not with the baseline are marked as bold and underlined.

In Table 10 and Table 11 we clearly can see some words that could be matched thanks to stemming and lemmatization. For example, if we look at a short query and a sentence through three different methods shown in Table 10, we can see how the word "dies" and "died", in query and sentence is reduced by the stemming and lemmatization to the word form "die", through which it is possible to overlap between the query and the sentence. Also, the tables show a few more examples that show how some words could be matched thanks to stemming and lemmatization, and why a sentence has a better position in the search result.

7. Conclusion

In this paper we showed through multiple tests that pre-processing steps stemming and lemmatization have clear benefits when it comes to sentence retrieval. In most of our tests we got better results when combining $TF-ISF$ with stemming or lemmatization. However, the positive effects only appeared with the measures MAP and R-prec. which improve recall. At the same time the pre-processing steps did not show any negative effects on sentence retrieval. Therefore, we think that stemming and lemmatization is generally beneficial to sentence retrieval, we saw that stemming tends to show better result with short queries while lemmatization tends to show better results with longer queries which we will explore in more detail in the future.

References

[1] Doko, A., Stula, M., & Stipanicev, D. (2013). A recursive tf-isf based sentence retrieval method with local context. International Journal of Machine Learning and Computing, 3(2), 195.

[2] Florijn, W. J. (2019). Information retrieval by semantically grouping search query data (Master's thesis, University of Twente).

[3] Harman, D. (2002). Overview of the TREC 2002 novelty track. In Proceedings of the eleventh text retrieval conference (TREC).

[4] Soboroff, I., & Harman, D. (2003). Overview of the TREC 2003 novelty track. In Proceedings of the twelfth text retrieval conference (TREC).

[5] Soboroff, I. (2004). Overview of the TREC 2004 novelty track. In Proceedings of the thirteenth text retrieval conference (TREC).

[6] Text REtrieval Conference (TREC) Novelty Track. (2003, March 4). Retrieved March 19, 2020, from https://trec.nist.gov/data/novelty.html

[7] Doko, A., Stula, M., & Seric, L. (2015). Using TF-ISF with Local Context to Generate an Owl Document Representation for Sentence Retrieval. Computer Science & Engineering: An International Journal, 5(5), 01–15.

[8] Vetriselvi, T., Gopalan, N. P., & Kumaresan, G. (2019). Key Term Extraction using a Sentence based Weighted TF-IDF Algorithm. International Journal of Education and Management Engineering, 9(4), 11.

[9] Samir, A., & Lahbib, Z. (2018, April). Stemming and Lemmatization for Information Retrieval Systems in Amazigh Language. In International Conference on Big Data, Cloud and Applications (pp. 222-233). Springer, Cham.

[10] Kantrowitz, M., Mohit, B., & Mittal, V. (2000, July). Stemming and its effects on TFIDF ranking. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 357-359).

[11] Balakrishnan, V., & Lloyd-Yemoh, E. (2014). Stemming and lemmatization: a comparison of retrieval performances.

[12] Murdock, V. (2006). Aspects of sentence retrieval. Ph.D. thesis, University of Massachussetts.

[13] Allan, J., Wade, C., & Bolivar, A. (2003, July). Retrieval and novelty detection at the sentence level. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (pp. 314-321).

[14] Losada, D. (2008, July). A study of statistical query expansion strategies for sentence retrieval. In Proceedings of the SIGIR 2008 Workshop on Focused Retrieval (pp. 37-44).

[15] V. Srividhya, R. Anitha, Evaluating Preprocessing Techniques in Text Categorization, International Journal of Computer Science and Application, Issue 2010.

[16] Dr. S. Vijayarani, Ms. J. Ilamathi, Ms. Nithya, Preprocessing Techniques for Text Mining - An Overview, International Journal of Computer Science & Communication Networks, vol 5(1), pp. 7-16.

[17] Sandhya, N., Lalitha, Y. S., Sowmya, V., Anuradha, K., & Govardhan, A. (2011). Analysis of stemming algorithm for text clustering. International Journal of Computer Science Issues (IJCSI), 8(5), 352.

[18] Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130–137.

[19] Joshi, A., Thomas, N., & Dabhade, M. (2016). Modified porter stemming algorithm. Int. J. Comput. Sci. Inf. Technol, 7(1), 266-269.

[20] Plisson, J., Lavrac, N., & Mladenic, D. (2004). A rule based approach to word lemmatization. In Proceedings of IS (Vol. 3, pp. 83-86).

[21] Ms. Anjali Ganesh Jivani, A Comparative Study of Stemming Algorithms, Anjali Ganesh Jivani et al, Int. J. Comp. Tech. Appl., Vol 2 (6), 1930-1938, ISSN:2229-6093.

[22] Korenius, T., Laurikkala, J., Järvelin, K., & Juhola, M. (2004, November). Stemming and lemmatization in the clustering of finnish text documents. In Proceedings of the thirteenth ACM international conference on Information and knowledge management (pp. 625-633).

[23] Kettunen, K., Kunttu, T., & Järvelin, K. (2005). To stem or lemmatize a highly inflectional language in a probabilistic IR environment?. Journal of Documentation.

[24] Kanis, J., & Skorkovská, L. (2010, September). Comparison of different lemmatization approaches through the means of information retrieval performance. In International Conference on Text, Speech and Dialogue (pp. 93-100). Springer, Berlin, Heidelberg.

[25] Text REtrieval Conference (TREC) TREC 2004 Novelty Track. (2005, February 4). Retrieved March 19, 2020, from https://trec.nist.gov/data/t13_novelty.html

[26] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge university press.

[27] Fernández, R. T., Losada, D. E., & Azzopardi, L. A. (2011). Extending the language modeling framework for sentence retrieval to include local context. Information Retrieval, 14(4), 355-389.