International Journal of Intelligent Information Systems
2016; 5(6): 82-87
http://www.sciencepublishinggroup.com/j/ijiis
doi: 10.11648/j.ijiis.20160506.11
ISSN: 2328-7675 (Print); ISSN: 2328-7683 (Online)

Using Text's Terms and Syntactical Properties for Document Similarity

Mohamed Taybe Elhadi

Department of Computer Science, University of Zawia, Zawia, Libya

Email address:
mtelhadi@yahoo.com

To cite this article:
Mohamed Taybe Elhadi. Using Text's Terms and Syntactical Properties for Document Similarity. *International Journal of Intelligent Information Systems*. Vol. 5, No. 6, 2016, pp. 82-87. doi: 10.11648/j.ijiis.20160506.11

Received: October 4, 2016; **Accepted**: November 5, 2016; **Published**: December 5, 2016

Abstract: This paper reports on experiments performed to investigate the use of syntactical structures of sentences combined with sentences' terms for document similarity calculation. The document's sentences were first converted into ordered Part of Speech (POS) tags that were then fed into the Longest Common Subsequence (LCS) algorithm to determine the size and count of the LCSs found when comparing the document sentence by sentence. As a first stage, these syntactical features of the text were used as a structural representation of the document's text. However, the produced strings of tags not only work as text representative but also provide for text size reduction. This improves the processing efficiency of comparing the document's representative strings using the LCS. A score is generated by computing an accumulative value based on the number of the LCSs found. In the second stage, documents that score well in the first stage are subjected to further comparison using the actual words of the sentences (content) in a sentence by sentence fashion. An overall final is generated as a measure of similarity using the common words (accumulated for the whole document) and the total number of LCSs from the first step. Experiments were done on two different corpora. Results obtained have showed the utility of the proposed procedure in calculating similarities between written documents. The overall discrimination power was maintained while the size of the documents was reduced using only a representative of the document based on the tagged string.

Keywords: Syntactical Structures, Document Similarity, Bag-of-words, Longest Common Subsequence

1. Introduction

With the growth of the web and the emergence of digital libraries, document management, text analysis and similarity calculations have become an important text processing technique. This is in-line with the more established, prevalent and important fields of natural language processing and knowledge and Data discovery [30, 32].

Deciding on relevance, an important Information Retrieval (IR) task is mostly performed using string representations and processing [1]. Text similarity is an integral part of many such applications [22, 23, 30]. It is common task shared among various applications ranging from copy detection [16, 19], near-copy detection [16, 11] plagiarism [9, 10, 12], IR systems [1, 14] and computational biology [2, 4, 6, 21]. Many such applications [26, 27, 28, 29] employee a combination of techniques and apply to multidisciplinary fields [7, 8, 13].

The work performed here is on the investigation of how related documents can be treated as modified version of one another. Such versions are, in turn, considered as a result of edit operations similar to those used in evolutionary biological sequences [2, 21]. Those operations that can be performed on strings, text or bio-sequences, include insertion, deletion or replacement of one unit or more into the string.

In a similar fashion to the way bio-sequences are processed, one can take advantage of syntactic unites derived from POS tagging to be used instead of actual text characters in document relevance (similarity) determination. This has the advantages of representing strings using meaningful units (POS-Tags) that are better defined and are a more clearly-represented set of units. These strings capture some semantics contained in the writing style of authors, selection of vocabulary types, use of language phrases and the relationships defined by ordered text units.

As such, attempts made to modify an existing text, whether maliciously (plagiarizing) or on purpose (reducing or expanding on news article) would certainly involve one of the following text operations:

1. Total cut-and-paste, where very little is done to modify original text.

2. Insertion, where a person would, for example, insert new words into an original text to produce a partially modified version.

3. Deletion, where the opposite of the above takes place. In deletion, for example, some of the words of the original text are deleted to produce a partially modified version.

4. Substitution where the operation is a combination of a delete(s) and an addition(s). The original text would be modified by the deletion of a word and the addition of similar one or more words.

Reducing the original document to its syntactical structures greatly reduces the dimensionality of the document. Smaller strings will be dealt with instead of the whole character string in the document. At the same time, less information is lost when compared to what happens when documents are processed based on actual characters or words. This is so due to the captured structural properties of the sentences such as the order and the part of speech roles.

A procedure that uses LCS algorithm on POS-Tagged strings of the document's text as a basis of similarity calculations is defined. This combined use of the LCS on POS string functions as front-end filter to the more basic information retrieval task, namely, the bag-of-words technique [1]. Sentences of the documents are converted into ordered POS tags that are then fed to the LCS algorithm to determine the size and count of the LCS found when comparing document text sentence by sentence.

The LCSs algorithms, dynamic programming based, are slow [2, 4], especially when used to process large strings. In order to improve the efficiency of such techniques, the syntactical and structural properties of the original document's text were used as a much shorter representation for the document. Reduction in string size is achieved when using POS strings as a representatives. The produced POS-Tagged string serves as a more concise and a much shorter representative string of the original text. It can then be used in text processing rather than comparing the full text of the document. Documents that score well in the first stage as measured by an accumulative score that is a function of the number of the LCSs found, are subjected, in a second stage, to further comparison using actual content words. Content here is used to mean actual English words or terms processed sentence by sentence. A final measure of similarity, based on common words (accumulated for the whole document) and the total number of LCSs already found, is produced and used to rank the documents by their similarity.

Experimental validation of the procedure was done on two different corpora [15, 24]. The results obtained have shown the utility of the proposed approach in calculating similarities between written texts.

The rest of the paper is made up of section 2 on related work; section 3 on the proposed procedure; section 4 on the experiments conducted and document collections used; section 5 on results and their analysis and lastly section 6 on conclusions and future work.

2. Related Work

Even though text processing used as either a representative or a comparison techniques is an old and well-studied field, it is mostly based on actual text (character based) or bag of words (vector space models of IR) techniques. The combined use of syntactical POS tagging and text processing methods for the purpose of text similarity calculations and its applications is recent on most part. Literature does not seem to show much previous or similar use of such method, even though semantically and statistically motivated techniques have been around in natural language processing and other Artificial Intelligence (AI) based work [22, 23, 30, 32]. The idea is a realization of the intuition that similar (exact) documents would have similar (exact) syntactical properties. In particular, those documents that contain or reuse other documents or parts of other documents would have similar structures. This is more certain when the production or refinement of new documents is the result of reduction, expansion, plagiarism or modifications in general. Use of syntactic properties to determine similarity of text by way of comparing POS strings is briefly mentioned next. For more on this see [25].

2.1. Syntactical Structures and POS-Tagging

When comparing text, a major hurdle appears to be due to the differences on the make-up of the character strings and the lack of a theory that can be used for explaining this make-up. Mere sequences of text letters is not enough to represent the syntactic properties, let alone the semantics of the text's content. One way to load a string with content is to consider it as a lump of text made of meaningful, well defined and numerable units (alphabets). As such modification of text can be thought of as an intervention or application of edit operations on an original set of units. These operations introduce new strings that are structurally (order and POS roles) similar depending on how much operation (rephrasing) is done on the original text.

To POS-tag a text document is to annotate the text with its part-of-speech. Several approaches, mostly implementing the probabilistic methods, existed [3, 20]. Good probabilistic methods are based on first-order or second-order Markov models. However, such systems have difficulties in estimating small probabilities accurately.

For the purposes of the work reported here, TreeTtagger [3] will do the job. Differences on the number of tag sets available will manifest the level of details (syntactic and semantic) that a tagger can capture. TreeTagger applies a probabilistic method for automatic words' annotations with POS tags. It uses a decision tree to obtain more reliable tagging. It received many improvements achieving the highest accuracy in comparison to other taggers with an accuracy of up to 96.36% [3, 20]. It also features a range of tag sets that can reach 55 tags.

2.2. POS String Matching Using LCS

POS tagging is the task of assigning an appropriate and a

particular part of speech or word category into the text's sentences. This annotation process might be based on both its definition and or its context [32].

POS tagging is applied in many domains, such as, in a preprocessing stage to parsing, information retrieval, text to speech systems, and corpus linguistics, etc [33, 32].

There are many approaches –both inductive and non-inductive- have been investigated and implemented to tackle the tagging problem. The widely known types of these approaches are divided into: Rule-Based, Probability-Based and Memory-Based approaches [31, 32]. Most of the taggers that use statistical approaches are based on Markov model especially, the hidden Markov model. Such taggers use a tagged corpus to compute probabilities of co-occurrence of words. These probabilities are used later on for text tagging. Thus, taggers do not need to know anything about the rules of language [32].

Many taggers, however, implement the statistical approach [5]. TreeTagger is what is used here in this work. It uses a decision tree rather than Markov models [3, 20].

1. Taggers use variable sets of POS tags. In most part, they include the basic parts of verb, noun, pronoun, adjective, adverb, preposition, conjunction and the interjection.

2. A tagged text document can get huge reduction in size. If we assume a page of text contains 250 words where the average word length is five letters, the raw text string is made up of 1250 characters. An equivalent tag string would correspond to the total numbers of words, that is 250 in this case with some extra punctuation tags. Hence, a very huge reduction in size that can still be further refined when we extract unique sequences (work we are considering in future).

2.3. Text Similarity Calculation

Different methods and approaches have been in use to tackle the issue of similarities between documents using semantically and syntactically motivated approaches [17, 18]. Semantic approaches receive less attention due to the difficulties of representing semantics and the limitations on assessment coverage of user studies [17, 18]. Syntactic approaches, on the other hand, are more common and include fingerprinting [11], Information Retrieval [1] and hybrid techniques [7, 8, 12].

Information retrieval puts more emphasis on representing documents through their words and word frequencies. They use indexing with an appropriate model to evaluate similarities between documents [1].

Fingerprinting techniques use the text chunking where a document's text is divided into small units. Each unit is hashed to produce a list of values representing the document. These values are then compared to other documents' values to detect similarities [11].

Combining some of the above techniques has also been attempted. One such approach combines fingerprinting and information retrieval [7].

Many more techniques are discussed in the literature, many of which aim to detect overlap between documents for more specific purposes by adopting different strategies, depending on the task required [8, 12].

3. The Proposed Procedure

A brief description of the proposed procedure is shown in Fig. 1. Phases of the procedure are briefly described next:

3.1. Syntactical Processing Phase

In this phase the text is reduced into a smaller set of POS tags. It makes use of the whole documents' content without excluding any stop words, stemming or removal of numbers, punctuation and special characters.

The choice of tagger and size of a tag-set can affect the accuracy and efficiency of the system. Taggers are relatively accurate but may not be 100% error free [3, 20]. The TreeTagger [11], which was adopted for this work, is freely available, with high accuracy and a relatively large tag set.

3.2. Tagged String Optimization Phase

Since the LCS algorithm handles characters and its efficiency is a function of the length of string, each tag of each string has been replaced by a single symbol.

3.3. The LCSs Processing Phase

Tagging and optimization produces a set of strings representing the ordered tags corresponding to words in the original document. Pairs of documents are compared sentence-by-sentence using the tagged strings. This step results in two values for each pair comparison:

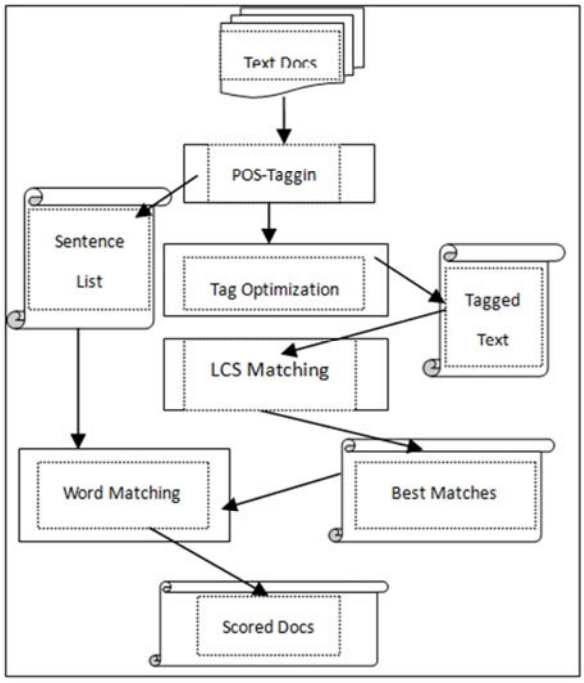

Figure 1. *Overall depiction of the proposed procedure.*

(1) the accumulated size of the LCSs (*LCSAccSize*) of each pair of sentences and (2) the count of such pairs (*LCSCount*). These values are used in the next step.

3.4. Bag-of-Words Comparison Phase

For each pair of sentences that meet some minimum value from the previous step, we further compare the relevant sentences using the actual words in the original sentences. With some experimentation, we decided to use a 6-tag string as a minimum length to process sentences any further. This phase results in an accumulated count of common words for paired sentences for the complete document (*AccWordCount*).

3.5. Final Similarity Calculation Phase

A final similarity score (*SimScore*) is calculated for each pair of documents taking into account the number of *LCSCount* from the LCSs Processing phase and the *AccWordCount* from the Final Similarity Calculation phase. The two values are combined by the following formula:

SimScore(doci,docj) = LOG(AccWordCount)/ LCSCount

Paired documents are then ranked based on the similarity score and analyzed for any discrepancies.

4. Experiments and Datasets

To evaluate the proposed approach two datasets each serving a different purpose and different context were used.

4.1. Datasets

Two freely available corpora were used. The first one created by Paul Clough and Mark Stevenson [15] that we refer to as the CLOUGH-STEVENSON-09 collection. The other corpus created by the Meter project [24] and is known as the Meter collection.

According to its creators, the CLOUGH-STEVENSON-09 collection is meant to be a corpus for plagiarism detection testing. It consists of answers to Computer Science questions in which plagiarism has been simulated representing different levels of plagiarism. The following are the four levels of plagiarism as defined by the creators of the CLOUGH-STEVENSON-09 corpus [15]:

1. Near copy: Participants were asked to answer the question by simply copying text from the relevant Wikipedia article (i.e. performing cut-and-paste actions).
2. Light revision: Participants were asked to base their answer on text found in the Wikipedia article and were instructed that they could alter the text in some basic ways including substituting words and phrases with synonyms and altering the grammatical structure (i.e. paraphrasing).
3. Heavy revision: Participants were asked to base their answer on the relevant Wikipedia article but were instructed to rephrase the text to generate an answer with the same meaning as the source text, but expressed using different words and structure.
4. Non-plagiarism: Participants were provided with learning materials that could be used to answer the question. They asked to read these materials and then attempt to answer the question using their own knowledge and told that they could look at other materials to answer the question but explicitly instructed not to look at Wikipedia.

The Meter Corpus [8] was built for the aim of investigating text reuse in the sector of newspaper journalism. It contains texts from the domains of law courts and show business. The texts were manually collected and classified by professional journalists. The articles come from the Press Association (PA) and nine other British national newspapers. All of the newspaper articles were classified at the document level based on their dependency on the PA as (1) wholly derived from PA, (2) partially derived from PA or (2) not derived from PA.

4.2. Proposed Procedure Steps

The following steps were applied to each corpus:

1. Text documents are first tagged using TreeTagger.
2. Tags are then converted into single-character tag strings giving an ID number to each string. The ID is just a sequential number for ease of association with sentences and for ease of retrieval.
3. Documents are also stored with each sentences identified with an ID equivalent tagged- string.
4. The LCS algorithm was run on each file comparing sentence by sentence. This step results in a number of LCSs and an accumulated size of the whole file.
5. Those sequences that meet an experimental cut-off value (used 6 tags or more) are further compared to find all common words.
6. The final resulting values of accumulated LCSs, their count along with common words of those sentences with a minimum sequence length of tags are further analyzed.
7. An overall similarity value is calculated using the accumulated number of common word between sentences of paired files and the total number of the LCSs with a minimum of 6 tags.
8. A final score is calculated as explained previously.

5. Results and Discussions

The results of the performed experiments are explained next:

5.1. Clough-Stevenson-09 Collection

The aim here was to use the proposed procedure to see if it can meaningfully identify the different levels of plagiarism done on the data as outlined by the corpus creators.

The calculated scores give a convenient cut-value. It is the score of the original vs. original comparison. Their score falls in the zero score conveniently dividing the scoring range into high (positive) and low (negative) regions.

This of course is expected considering that each of those original documents when compared to itself will has a complete overlap. On the other hand, the most similar documents would score in the higher range (positive region) and the less similar ones would score in the lower range (negative region). Table 1 describes the obtained results showing that in total, excluding the Non-Plagiarism level,

98.34% are in the positive or higher region with very close results of the various levels of plagiarism. It should be noted that the system uses both structural (external) features and (internal) word-based comparison.

It is expected that such a system would group similar documents due to possibly many smaller syntactically similar tag-sentences that are supported with many overlapping (matching) content words. This is evident from the results (77.42%) obtained for the no-plagiarism category as shown in Table 2. Such a result can only be verified through a more rigorous manual investigation of the documents (a step that we hope to do in the future).

5.2. The Meter Corpora

This set of data is different from the previous one. The Meter collection is all real news articles supplied by PA and used (reused) by other newspapers. The articles used in this collection are divided into two sets. One is derived from news coming from law courts while the other is derived from news coming from entertainment and show business.

It is worth noting that some articles are included from media outlets that do not base their news on what is provided by PA.

Since we are only interested in investigating and evaluating our procedure's ability to measure text similarity (due to reuse in this case and content similarity), we have selected to use the show business collection for our analysis. Some basic preparation of the documents was done on the level of collection but without affecting the documents content.

To reduce the size of the collection we have removed all PA articles and only based our analysis on the articles from the different newspapers. In doing so, we could see how our procedure ranks, classifies or groups related documents. Related documents are those that are based on the same story or news item and have been based on the PAs documents. Documents not based on the PA articles are also included and are considered related as well as they are reporting on the same news item.

Results are shown in Table 3. Clearly the procedure based on similarity score's lower (negative region) and higher (positive region) ranges gives high scores to related (similar) items with an overall positive percentage of 75.32%.

The percentage is interesting when we look at the percentages for the wholly and partially PA-derived documents.

For the Non-PA derived stories the procedure give slightly lower percentage of positive documents. This is also interesting as there is no direct reuse of PA articles but still the documents are considered similar in content since they report on the same stories.

Table 1. *Clough-Stevenson-09 Corpus Based Results.*

Category	Total	Positive	Negative	%
Original vs. Originals	5	5	0	100.00
Original vs. Cut	19	18	1	94.74
Originals vs. Light	19	19	0	100.00
Originals vs. Heavy	19	19	0	100.00
Total	62	61	1	98.39

Table 2. *Clough-Stevenson-09 Corpus'S No-Plagiarism Category.*

Category	Total	Positive	Negative	%
Original vs. None	31	24	7	77.42

Table 3. *The Meter Corpus Results.*

	Wholly PA Derived	Partially PA Derived	Non-PA Derived	Total
Positive	9.00	32.00	17.00	58.00
Negative	2.00	7.00	10.00	19.00
Total	11.00	39.00	27.00	77.00
Positive %	81.82	82.05	62.96	75.32
Negative %	18.18	17.95	37.04	24.68

One final interesting result is that documents derived from PA were given higher scores even though different newspapers did them.

6. Conclusions

A similarity calculation procedure based on the combined use of syntactical properties and sentences words has been proposed and evaluated. The proposed procedure takes advantage of the syntactical structure manifested as POS-tagged string subjected into the LCSs comparison before further processing of the document. Documents are pre-processed using a POS tagger converting them into string of tags. This constitutes a representation of documents' content on a higher level of abstraction. Document's alterations can be captured as strings enabling their processing using many slower text processing techniques such as the LCS algorithms. Encouraging results were obtained on the performed experiments addressing this issue of similarity determination using a two staged approach made of syntactical string processed using the LCS and the bag-of-words information retrieval. More analysis and fine-tuning are needed to fully understand the implications of such an approach and address issues of better tuning of the tags and efficiency of the procedure. Nonetheless, the greatly reduced string size, the obtained similarity and the discrimination power can be considered very encouraging.

References

[1] Chowdhury G. Introduction to modern information retrieval. Facet publishing; 2010 Jul 31.

[2] L. Bergroth, H. Hakonen and T. Taita, "A Survey of Longest Common Subsequence Algorithms", In String Processing and Information Retrieval, 7th. International Symposium on, 27-29 Sept. 2000., pp. 39–48.

[3] H. Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees", Intern Conf. on New Methods in Language Processing, Germany, 1994, pp. 4-9.

[4] Li H, Homer N. A survey of sequence alignment algorithms for next-generation sequencing. Briefings in bioinformatics. 2010 Sep 1; 11(5):473-83.

[5] T. Brants, “TnT -- Statistical Part-of-Speech Tagging”, in Proceedings of the 6th Applied Natural Language Processing Conference (ANLP), (Seattle, Washington, USA, April 29 - May 4 2000), 2000, pp. 224-231.

[6] Baral, C., Local Alignment: Smith-Waterman algorithm, CSE 591: Computational Molecular Biology Course, Arizona State University, 2004.

[7] Y. Liu and L. Liang, “A Dual-method Model for Copy Detection”, IEEE, IAT Workshops, 2006, pp. 634-7.

[8] K. Monostori, R. Finkel, A. Zaslavsky, G. Hodasz and M. Pataki, “Comparison of Overlap Detection Techniques”, Intern. Conference on Computational Science, Amsterdam, Holand, 21-24 Apr., 2002, pp 51-60.

[9] Clough, P., Old and new challenges in automatic plagiarism detection, Department of Information Studies, University of Sheffield, 2003.

[10] Bull, J., C. Collins, E. Coughlin and D. Sharp, Technical Review of Plagiarism Detection Software Report, Computer Assisted Assessment Centre, University of Luton, Luton, UK.

[11] Stein B, zu Eissen SM. Fingerprint-based Similarity Search and its Applications. Universität Weimar. 2007.

[12] Kang, N., A. Gelbukh and S. Han, PPChecker: Plagiarism Pattern Checker in Document Copy Detection, 2006.

[13] Steinberger, R., B. Pouliquen and J. Hagman, Cross-lingual Document Similarity Calculation Using the Multilingual Thesaurus EUROVOC, Springer-Verlag Berlin Heidelberg, 2002.

[14] Poinçot, P., S. Lesteven and F. Murtagh, Comparison of Two “Document Similarity Search Engines”, ASP Conference Series, Vol. 153, 1998.

[15] Clough, P. and Stevenson M. Developing a Corpus for Plagiarized Short Answers, Language Resources and Evaluation: 45:5-24, London: springer, 2011.

[16] Grune, D, and M, Huntjens, Detecting copied submissions in computer science workshops, Vakgroep Informatica, Faculteit Wiskunde & Informatica, Vrije Universiteit, AMSTERDAM, 1989.

[17] A, G. Maguitman, F, Menczer, H. Roinestad and A. Vespignani, “Algorithmic Detection of Semantic Similarity”, International World Wide Web Conference Committee, 2005, pp. 107-116.

[18] Mihalcea, R., C, Corley and C, Strapparava, Corpus-based and Knowledge-based Measures of Text Semantic Similarity, American Association for Artificial Intelligence, Jul, 2006.

[19] D. M. Campbell, W. R. Chen and R. D. Smith, “Copy Detection Systems for Digital Documents”, IEEE, Washington, DC, USA, May, 2000, pp. 78-88.

[20] H. Schmid, “Improvements in Part-of-Speech Tagging With an Application To German”, EACL SIGDAT workshop, in Dubai (UAE), 1995.

[21] Elzinga C, Rahmann S, Wang H. Algorithms for subsequence combinatorics. Theoretical Computer Science. 2008 Dec 28; 409(3):394-404.

[22] Sagayam R, Srinivasan S, Roshni S. A survey of text mining: Retrieval, extraction and indexing techniques. International Journal of Computational Engineering Research. 2012 Sep; 2(5).

[23] Natural Language Processing in Information Retrieval Thorsten Brants Google Inc, 2004.

[24] Clough, P. and Stevenson, M., “Developing A Corpus of Plagiarised Short Answers”, Language Resources and Evaluation: Special Issue on Plagiarism and Authorship Analysis, Volume 45(1), pp. 5-24. 2010.

[25] Elhadi, M. Al-Tobi, M. "Detection of Duplication in Documents and WebPages Based Documents Syntactical Structures through an Improved Longest Common Subsequence", IJIPM: International Journal of Information Processing and Management, Vol. 1, No. 1, pp. 138~147, 2010.

[26] Pradhan N, Gyanchandani M, Wadhvani R. A Review on Text Similarity Technique used in IR and its Application. International Journal of Computer Applications. 2015 Jan 1; 120(9).

[27] Roshdi A, Roohparvar A. Review: Information Retrieval Techniques and Applications. International Journal of Computer Networks and Communications Security VOL. 3, NO. 9, SEPTEMBER 2015, 373–377.

[28] Gomaa WH, Fahmy AA. A survey of text similarity approaches. International Journal of Computer Applications. 2013 Jan 1; 68(13).

[29] Traina AJ, Traina Jr C, Cordeiro RL, editors. Similarity Search and Applications: 7th International Conference, SISAP 2014, Los Cabos, Mexico, October 29-31, 2104, Proceedings. Springer; 2014 Oct 8.

[30] Cambria E, White B. Jumping NLP curves: a review of natural language processing research. IEEE Computational Intelligence Magazine. 2014 May;9(2):48-57

[31] W. Daelemans, J. Zavrel, P. Berck and S. Gillis, “MBT: A Memory-Based Part of Speech Tagger Generator”, in Proceedings of Fourth Workshop on Very Large Corpora (WVLC), University of Copenhagen, Copenhagen, Denmark, August 5- 9 1996), 1996, pp. 14-27.

[32] Yonghong Mao, Natural Language Processing Module (Part of Speech Tagging and Sentence Parsing), Cognitive Science in Context Laboratory, Cornell University, New York, U.S., 1997.

[33] J. Cussens, “Part-of-Speech Tagging Using Progol”, in Proceedings of the 7th International Workshop (Inductive Logic Programming), (Prague, Czech Republic, September 17-20 1997), 1997, pp. 93-108.