IJCSI International Journal of Computer Science Issues, Volume 17, Issue 6, November 2020
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.5281/zenodo.4431057

SED: An Algorithm for Automatic Identification of Section and Subsection Headings in Text Documents

Muhammad Bello Aliyu[1], Rahat Iqbal[2], Anne James[3] and Dianabasi Nkantah[4]

[1] School of Computing and Mathematics, Coventry University, Coventry, West Midlands, United Kingdom

[2] School of Computing and Mathematics, Coventry University, Coventry, West Midlands, United Kingdom

[3] School of Computing and Mathematics, Coventry University, Coventry, West Midlands, United Kingdom

[4] School of Computing and Mathematics, Coventry University, Coventry, West Midlands, United Kingdom

Abstract

The word processing applications, such as the Microsoft Word Office, have advanced features like the automatic table of contents (ToC) feature. The ToC is a representation of the headings of both sections and subsections that are within the document. Currently, there is no computational procedure to transverse the document and identify section and subsections to extract this information needed for ToC and other text analytics purposes. All the applications rely on the users to identify and highlights the texts (headings and subheadings) within the document that are to appear in the ToC. Text documents are organised into sections and subsections each with a named heading and subheading.

This paper presents a novel algorithm for identifying the headings and subheadings within text documents. The automatic identification of the headings and subheadings (of all the sections) in the document. By leveraging this novel algorithm, the generation of the table of contents can be fully automated such that users do not have to identify/select the headings and subheadings manually.

The algorithm is simple, rule-based and unsupervised. This improves the process and saves a great deal of time as there is no training involved. The algorithm has been tested on several documents (papers) and achieved an accuracy of over 82%. The algorithm also improves the computational capabilities of the current natural language processing approaches. It is also useful for automating some tasks in systematic literature reviews and would speed up the analysis and evaluation of the natural language resources and text analytics in general

Keywords: Natural language processing, big data, text mining, information retrieval, algorithm.

1. Introduction

The natural language processing (NLP) involves identification, extraction and processing of data from text documents (Nelson 2018). It also involves the application of NLP techniques for analysing and processing documents to obtain the relevant and useful data (Rahija and Katiyar 2014). These include basic NLP techniques such as tokenization, lemmatization, stemming etc. which are the building blocks for NLP analytics. More sophisticated techniques were however, developed to address the complexities of the natural languages to deduce meaning and extract relevant information (Muhammad *et al.,* 2019). Due to the overwhelming volume of data produced daily, the NLP techniques are required now more than ever to address the data deluge.

An estimated 2.5 quintillion bytes of data is generated each day (Marr 2018), with about 80% of such data being unstructured. Unstructured data includes scientific research publications, reports, online article, memorandum etc. These text documents are unstructured (text-heavy), not organised in any pre-defined model and not organised in any pre-defined model. They also have no special structures for retrieving data from the various sections of the documents. Text documents are structurally organised into entities or units such as sections, subsection, paragraphs and sentences (Muhammad *et. al.,* 2018). This typical structure of a text document is shown in the fig. 1 below.

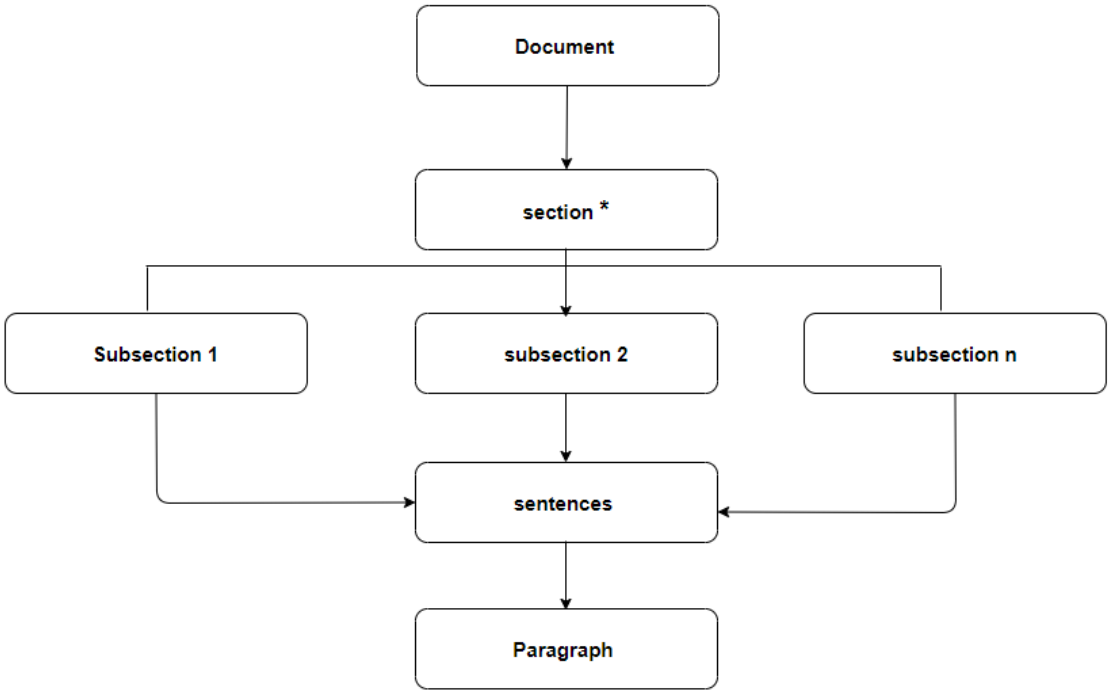

Fig. 1 Hieratical structure of text document (Muhammad *et al.,* 2018)

As shown in fig. 1, a text document is organised in a hierarchical structure in a top-down passion, consisting of sections and subsections. Each section/subsection in turn consists of paragraphs. And finally, each paragraph consisting of several sentences. Sections are named entities which represents a new topic within the document.
Word processing packages such as the Microsoft Word are efficient for text processing, providing both basic and advanced features. The Table of contents (ToC) is an advanced that feature heavily rely text mining techniques to extracts the headings and subheadings to be used for the constructing the ToC.
To the best of our knowledge, there is not any computational procedure to automatically identify all the headings and subheadings within the text documents. To generate the table of contents therefore, users must manually label all the headings and subheadings that would appear in the ToC (Gunnell 2019). Similarly, the automatic extraction of information from unstructured document such as in systematic literature reviews (SLR) depends on the ability to identify the different sections from the documents. From the sections, a section could be targeted for extracting the relevant information.

This paper presents a simple and unsupervised approach that could identify/extracts headings of sections as well as the associated the subsections within a structured document such as scientific research publications, reports, online article, memorandum etc. It can also extract the text within those sections. The algorithm, being a rule-based and unsupervised, means that it does not involved any training, as in the case of the machine learning nor does it require any special computational needs. Hence, it is faster and without any computational overhead. The algorithm works by identifying the underlying features of the sections and headings. Areas that could potentially take advantage of this research (method) include text summarisation, text-to-text generation, text-to-speech etc. Similarly, the ability of word processors to automatically identify headings and subheadings from documents to generate the automatic table content (TOC) feature would be greatly enhanced. Hence, the ToC feature would be fully automated removing the manual need to identify the headings and subheadings to be included in the ToC.
An effective natural language text processing involves the ability to develop robust computational methods that could transverse this structure for further processing. This means that the methods should have the intelligence to identify and, possibly, extracts each of the above entities in the document structure shown in fig. 1.0 below. Automatic processing of these documents, therefore, requires effective utilisation of the robust and NLP based

automated methods. Our novel approach (algorithm) would also improve the computational capabilities of the current NLP approaches.

2. Background and Related Work

Data mining involves text analytics to extracts value from unstructured and semi-structured textual documents (Oliverio 2018). Several approaches have been developed to enhance the mining of relevant information from unstructured text.

The scientific research documents, which are text documents containing unstructured data, are organised into hierarchical structure, represented by hierarchical constituents like sections, paragraphs, sentences etc. as depicted in the fig. 1 (Power, Scott and Bouayad-Agha 2003). Identification of the desired information from these structured documents is a challenging task. This is because the document structure, depicted in fig.1, must be navigated through to identify the desired elements. Therefore, to effectively process the structured documents, effective techniques for processing the above identified constituents also require advanced techniques. This pushes the need for research in this direction.

Muhammad et al., (2018) produced a canonical model of structure as a framework for data extraction in scientific research articles. The canonical model is depicted in fig. 2 below. The canonical model is a representation of the Introduction, Method, Result and Discussion (IMRaD) components of the research articles.

The work of Sporleder and Lapata (2004) has used the machine learning methods for paragraph identification within a document. Similar works include method for paragraph boundary identification (Filippova and Strube 2006), the pragmatics of paragraphing in English language (McGee 2014) etc. Most of these works focus on identifying and working with paragraphs as the basis for text processing. The paragraphs are important units in text processing but are limited in the amount of information they contain and are not a structural unit for documents such as a scientific research publication (document). In addition, complex documents such as the scientific articles, reports, news articles etc. requires processing beyond paragraphs level. A section, however, contains a general viewpoint or information which may be represented by several paragraphs. Linking such paragraphs to build the main idea expressed by a section generates a computational overhead. Therefore, building methods that could identify and process a section rather a paragraph would remove such computational overhead.

Edward (2018) used rule-based heuristics for sentence identification from a document using the 'punctuation' approach. Using this approach, sentence is split using the punctuations such as period (.), question mark (?) and exclamation mark (!). However, there are lots of exception when splitting sentences using punctuations only.

Tomanek, Wermter and Hahn (2007) used a machine learning based annotation framework for sentence splitting. Sentence boundary annotation was the main feature for classifying the sentences. Since they used a biomedical dataset, the potential sentence boundary symbols (SBS) for biomedical language texts, such as those from the PUBMED literature database, include the 'classical' sentence boundary symbols. Conditional Random field was used, and a good accuracy was reported. After the sentence, the next higher-level unit of organisation for structured document is paragraph.

Rasekh and Toluei (2009) performed paragraph identification using the Pongsiriwet's discourse scale (2001) and Cheng's multi-trait assessment scale (2003). However, these do not apply to any structured documents. Sporleder and Lapata (2004) developed a supervised machine learning algorithm that identifies paragraphs from documents which uses textual and discourse cues as features for the classification and/or identification. The paragraph boundaries are usually unambiguously marked in texts. Hence, they used supervised methods for this task. This required training, testing and validation.

Hearst (1997) produced the text tilting algorithm that splits text into multi-paragraph units that represents subtopics using the term overlap in the neighbouring text blocks. He argued that the subtopic structure is marked in technical context by heading and subheadings. Hence, the importance of a technique that identifies the heading as well as the subheading of the structured document is of paramount importance.

The highest level (in the hierarchy of document structure) is a 'section'. A section contains one or more paragraphs and is usually reported under a named heading and or subheading. The ability to identify as well as extract and analyse the sections in a structured document will take the NLP analytics to a new level.

Sections are put together in a sequence to create a text document. To extract the text that lies within a section, the algorithm extracts the text that lies between the first encountered heading until the next heading. The algorithm is also efficient in detecting subheadings for the respective headings. This way, the headings and the subheadings, as well as their associated text are put together to make up a section.

Our novel approach would be useful in realising the canonical structure developed by Muhammad et al., (2018). This is because it would recognise the headings, subheadings as well as the associated text within. These could be used for further analysis. Similarly, the ToC feature in word processors would greatly be improved by removing the overhead of manual identification of headings and subheadings needed for inclusion in the ToC.

3. Algorithm Design

For any unstructured text such as the text in scientific research articles, new articles etc., every section is reported under a named heading. This research proposes a novel algorithm for automated identification of sections heading and subheading within the text document. The algorithm was designed after assessment and analysis of the documents (papers). The documents used in the experiment consist of two (2) different document formats: PDF and Docx, each converted to raw text (.txt) but retaining the original formatting. The algorithm is rule-based and unsupervised. The algorithm is as follows:

1. Pull out the entire texts from the PDF/Docx document.
2. Divide the extracted texts into paragraphs (sections).
3. Identify sections that begin with numbers (either Arabic or Roman). n=0
(a) Get (n+1)th paragraph. If section begin with numbers, go to (5). Else n=n+1, loop through.
(b) Else go to (4)
4. Break the entire text into sentences using sentence tokenization.
5. Process the texts
(a) Tokenise the text into sentences.
(b) Tokenise the sentence into words/numbers/characters go to 5(c)
(c) get the length of the first sentence. If length <50 then go to 5(c.) else go to 5(d.)
(c.) Check the number of special symbols. If number >3 then go 5(d.). Else go to (8)
(d). Get the next sentence. Go to 5(b)
(e) if last sentence, go to (6)
6. Analyse the text font style
7. Extract and store the headings.
8. End.

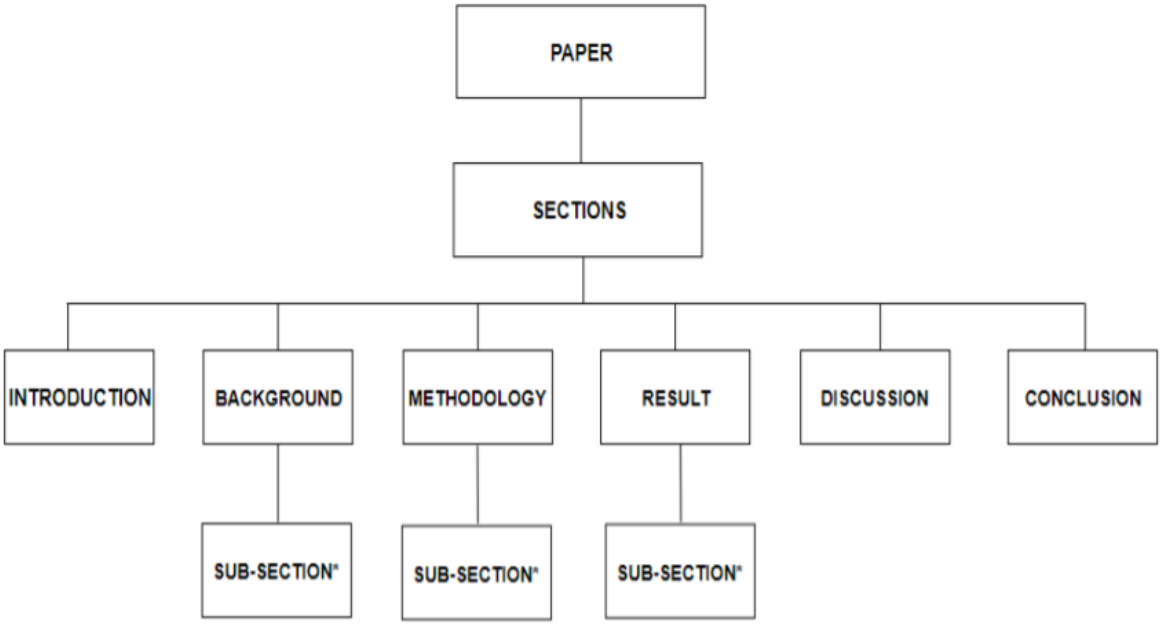

Fig. 2 The canonical structure

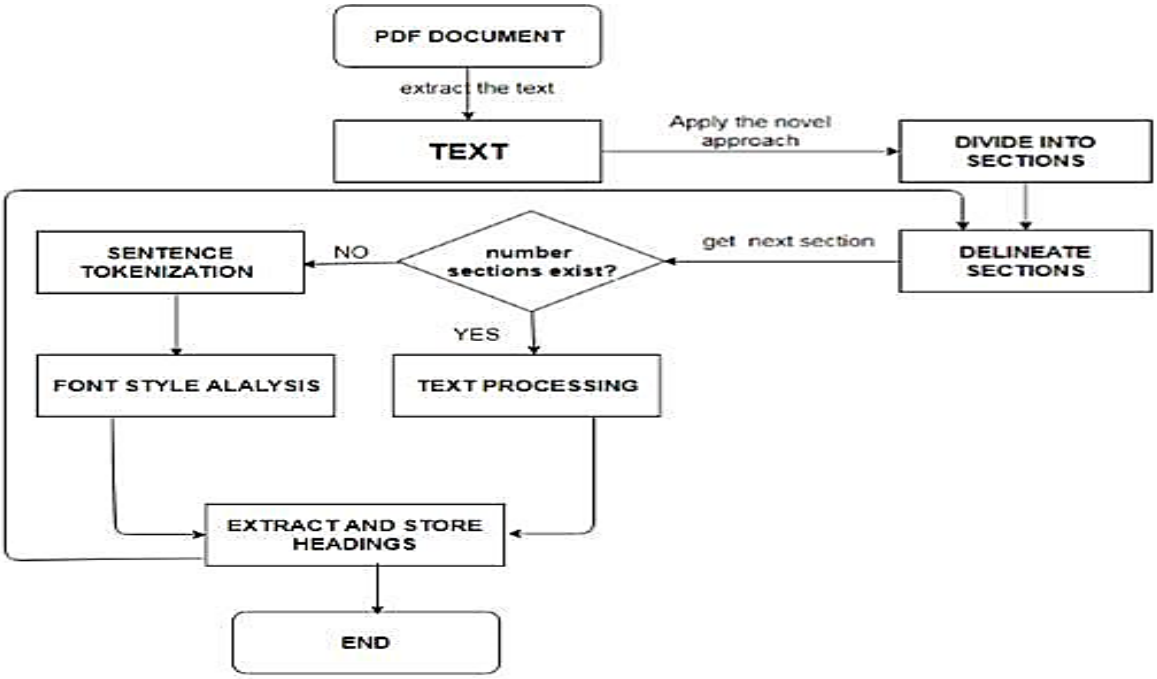

Fig. 3 Diagrammatic representation for the algorithm

3.1 Dataset

The dataset used for the experiment has been collected as part of the research project at Coventry University. Each data item (document) is a full-text research article from a software engineering domain. The data was obtained from the reputable online databases including the IEEEXplore, Science Direct, ACM digital library, etc. These sources have very rich software engineering subjects. In total, there were two hundred full text documents (scientific research publications). The data was manually analysed for the features and ingredients to the algorithm.

4. Experimentation

A tool was specifically developed and used for this experiment. It was built in Python. This is because the Python's NLTK module provides all the necessary support for the NLP tasks. The process followed to experiment and assess the 'SED' includes, tokenization, stop-words removal and analysis. The steps are highlighted in the subsequent sections below. The fig. 4 below shows the diagrammatic representation of the stages for the implementation process.

The documents were read and tokenized including removing the stop words. This provided the features needed for the training. Each document was individually read, experimented and analysed for its structure. The above designed (our proposed) algorithm was applied along with experimental settings. This experiment used all the default settings that come with the Python's NLTK module.

4.1 Analysis

The analysis of the structure of the documents include how different sections are organised and reported. The numbering format (Roman or Arabic), font style, length of text and character encoding were the features that were carefully observed from each of the document for building the algorithm.

From the analysis, the 100% of the papers have structural sections with a named heading and subheading. 67% of the papers have the sections numbered. Of the 67%, about two-third (2/3) are numbered in Arabic number while the one-third (1/3) in Ro-man numbers. For Arabic numbered sections, the sub-sections use decimal number subsection (e.g 2.1) like the format used for this paper) while for Roman number section, the subsections are alphabetically numbered. 33% of the papers have sections which are not numbered. However, the font styles and length of the heading text were different from the rest of the document. Also, 100% of the sections do not have special characters such as £, $, & etc. in the heading names. Also, all the heading names, from all the document have characters of short length. 95% of the headings from the all the documents have character length of not more than fifty

(50). The remaining 5% also do not have more than 70 length of the characters in the names.
The result of this analysis was unified and built into formal procedures called 'SED' the algorithm for section heading identification and text extraction. It was also used on a different set of papers for effective assessment of the proposed algorithm. The full experiment to assess the algorithm is reported in the section below.

The experiment also includes the application of the algorithm on the papers for evaluation. The fig. 2 above shows the output of the algorithm on the papers. For every paper in portable document format (PDF), the associated headings from that paper is displayed next to the paper. The algorithm has been successful on most papers.

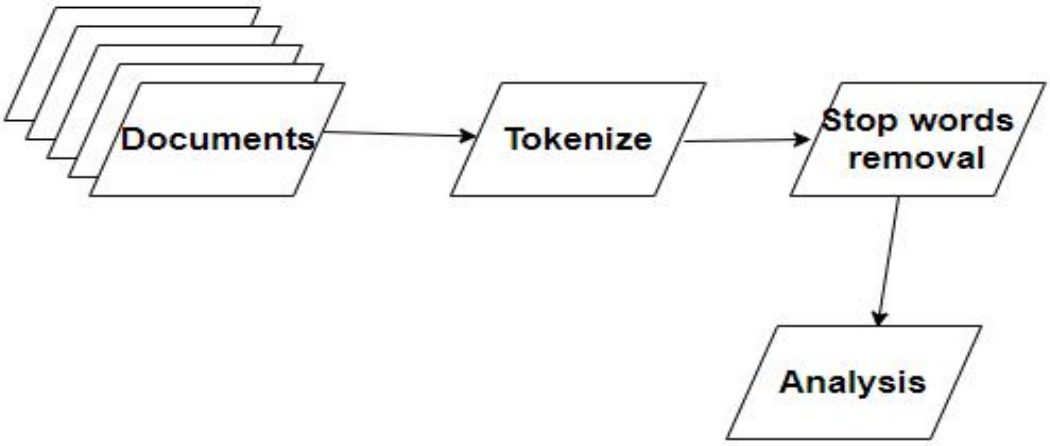

Fig. 4 Implementation Process

However, fewer papers have defied the algorithm. For some papers however, it was later discovered that binary coding was the reason for the algorithm's failure. When copied or formatted to a different format other than PDF, the algorithm later succeeded on some of the papers in identifying some or all the section headings.
Since the intention of the algorithm was to be used automatically on the papers, which are mostly available in PDF format. The scientific research publications are mostly available in PDF across most online databases because PDF format for presentation of contents and contents are mostly not editable in PDFs. This helps to preserve the content against alterations and possible manipulations (Muhammad et al., 2018).
This is algorithm was thus, implemented on PDF available documents and the result is outstanding. The details of the algorithms' performance are captured in the evaluation in section 5.

5. Evaluation

As highlighted in section 3.1, the data used for the experiment involves a collection 500 full text research articles. 500 hundred were used for the analysis described in section 4.3. After the algorithm was successfully formalised, all the five hundred (500) papers were tested on the formalised algorithm, the SED, for practical use. The intended evaluation purpose was to extract as much headings as possible and the associated text with that heading. It was observed that the algorithm picked all the headings from some papers, picked none from some and picked some headings from some. For evaluation purposes, we assumed that the algorithm is effective any paper where, at least, majority of the headings were identified by the algorithm. Where the algorithm identified very few (less than 50%) of the total headings from the paper, we recorded no success (score) for that algorithm on that paper. Finally, the following formula in equation 1 was used to compute the efficiency score for the SED algorithm.

$$\frac{Nx}{Ny} \quad (1)$$

Where:

Nx = Number of Papers in the collection
Ny: Number of papers where the SED succeeds. For simplicity purposes, papers where the algorithm succeeded are recorded with 1 point and 0 point for papers which defied the algorithm. The table 1.0 below shows the tally scores for the papers.

Table 1: Scores Tally

Successful Papers	164
Unsuccessful Papers	36
Total	200

Using the criteria highlighted above, the algorithm achieved 82% success rate. The efficiency is calculated using the formula in equation (1) as follows:

$$\frac{164}{36} \times 100$$

$$82\%$$

6. Conclusions

This paper presented an algorithm called, SED, for the automatic extraction of the section heading and subheadings as well as the text within that section. It is automatic, natural language bases, rule-based and unsupervised algorithm, giving no computational overhead and without any need for training. From the result of the algorithm's efficiency in in section five (5), the algorithm has been successful for the task it was designed for. This means that it can identify the headings from at least, 8 out of 10 research articles. It can also do so automatically without any computational overhead since it is unsupervised i.e. required no training at all. Therefore, it will make the useful in several natural language processing tasks such as systematic review, language modelling and extraction of certain information from the unstructured documents such as scientific research publications.

7. Future Work

This work has produced an algorithm for the purpose of identifying and extracting the headings as well as the subheadings from the text documents, as described previously. The algorithm produced is efficient and works for the default format of the papers (the PDF format). However, there were challenges such as the binary coding described in the analysis section. In addition, the same set papers were used. This means that there was no training/testing separation. Therefore, implementing the same problem using the supervised approaches particularly the machine learning approaches may improve the result.

References

[1] A. Angela, (November 2015). Popular document files & their differences. Retrieved from https://blog.online-convert.com/popular-document-files-their-differences/

[2] M. Bernad (2019). How much data do we create every day? The mind-blowing stats every-one should read. Retrieved from https://www.forbes.com/sites/bernard-marr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-every-one-should-read/#6ef2836660ba

[3] F. Cheng (2003). *The Effects of Rhetorical Specification in Writing Assignments on EFL (Eng-lish as a Foreign Language) Writing.*

[4] M., Edward (2018). NLP pipeline: Sentence tokenization (part 6). Retrieved from https://me-dium.com/@makcedward/nlp-pipeline-sentence-tokenization-part-6-86ed55b185e6

[5] K. Filippova & M., Strube (2006). Using linguistically motivated features for paragraph bound-ary identification. Paper presented at the *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing,* 267-274.

[6] M. A. Hearst (1997). TextTiling: Segmenting text into multi-paragraph subtopic passages. *Com-putational Linguistics, 23*(1), 33-64.

[7] I. McGee (2014). The pragmatics of paragraphing English argumentative text. *Journal of Prag-matics, 68*, 40-72.

[8] B.A Muhammad, R. Iqbal, R., A. James, A. & D. Nkantah (2019). Convolutional neural network for core sections identification in scientific research publications. Paper presented at the *International Conference on Intelligent Data Engineering and Automated Learning,* 265-273.

[9] P. Nelson (2018). *Natural Language Processing (NLP) Techniques for Extracting Information,*

[10] Paltoglou, G., & Thelwall, M. (2013). More than bag-of-words: Sentence-based document rep-resentation for sentiment analysis. Paper presented at the *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013,* 546-552.

[11] R. Power, D. Scott & N. Bouayad-Agha. (2003). Document structure. *Computational Linguis-tics, 29*(2), 211-260.

[13] A. Rajput (2019). Natural language processing, sentiment analysis and clinical analytics. *ArXiv Preprint arXiv:1902.00679,*

[14] A.E. Rasekh & B. Toluei (2009). Paragraph boundaries: Examining identification and produc-
tion performance of iranian EFL learners. *English Language Teaching, 2*(2), 30-38.

[16] N. Raheja, and V.K Katiyar (2014). Efficient Web Data Extraction Using Clusteringapproach In Web Usage

Mining. *International Journal of Computer Science Issues (IJCSI)*, *11*(1), p.216.

[17] C. Sporleder & M. Lapata (2004). Automatic paragraph identification: A study across lan-guages and domains. Paper presented at the *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing,* 72-79.

[18] K. Tomanek, J. Wermter & U. Hahn (2007). Sentence and token splitting based on conditional random fields. Paper presented at the *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics, 49* 57.\