KNOWLEDGE – International Journal
Vol. 28.7
December, 2018

A STUDY OF SUMMARIZATION TECHNIQUES IN ALBANIAN LANGUAGE

Roland Vasili
University of Gjirokastra, Albania, rvasili@uogj.edu.al
Endri Xhina
University of Tirana, Albania, endri.xhina@fshn.edu.al
Ilia Ninka
University of Tirana, Albania, ilia.ninka@fshn.edu.al
Thomas Souliotis
University of Edinburgh, United Kingdom, s1778881@sms.ed.ac.uk

Abstract: In recent years, technology has developed a lot and has revolutionized our perspective of the world. Technology and more precisely digital technology has created amazing tools, giving immediate access to anyone interested to any information he may need. This digital revolution of all media like computers, smartphones, etc. has produced a huge amount of digital data to be handled. In our research we care about one aspect of this data, the text data, and the way we can efficiently handle text and produce meaningful summaries. Thus, it is only until recently that text mining has become an interesting research field due to this vast increase of text volume on the web. However, because of its size, this text volume should be summarized so as to get all the useful information efficiently and without trying to deal with all of the initial text, which could be impractical in many cases. Therefore, text summarization systems are among the most attractive research areas nowadays. Text summarization is the process of finding the main source of information, extracting the main important contents and presenting them as a concise text in the predefined template. The two main summarization techniques available are Extractive and Abstractive, with a lot of research being carried out in these areas, especially in extractive summarization. However, meaningful summaries are obtained using abstractive techniques which are more complex, due to the nature of this technique which requires the summary to be constructed in an abstract way without using sentences from the original text, while in the extractive case the summary consists of sentences from the original text. In this paper there is a theoretical approach where the widely used summarization techniques are described at a first level. Moreover, these techniques are then put into practice focusing only on the Albanian language, since the language is an important factor which might lead to different outcomes for each algorithm, due to its structure, its form and its rules. This is the first attempt in the field of summarization in Albanian language and there is a high need for future research works in this area. This paper investigates various proposed text summarization methods which are usually used in English (and possibly other widely used) languages, comparing them and concluding which method is suitable for summarizing documents in the Albanian language. We analyze various summarization algorithms and provide a formal way of verifying the correctness of our results, by using different metrics (e.g. ROUGE) to evaluate the summaries' accuracy of each technique, by utilizing some gold standard summaries, which have been produced by linguistic experts. Finally, we will also provide the whole practical implementation of this work either by uploading it to a github repository so as to be publicly accessible by anyone or by providing our services as micro-services through a web-page.

Keywords: Text mining, text summarization, extractive summarization, summary evaluation, Python NLTK.

1. INTRODUCTION

The amount of information available today is tremendous and the problem of finding the relevant pieces and making sense of these is becoming more and more essential. The digital revolution of all media like computers, smart phones, etc. has produced a huge amount of digital data to be handled. Because of its size, this text volume should be summarized so as to get all the useful information efficiently and without trying to deal with all of the initial text, which could be impractical in many cases. Automatic text summarization is part of information retrieval (IR), machine learning (ML), natural language processing (NLP), data mining (DM) and text mining (TM). The aim is to find the core of the given text set and reduce the size while covering the key concepts and overall meaning and avoiding repetition.

1.1 Definition of Automatic Text Summarization

According to the American National Standards Institute [3], a summary is "a brief and objective representation of a document or an oral presentation" that allows quickly identification of the basic content of the document by readers and access services.

1.2 Summarization and Text Mining

Summaries contribute to text mining since any information analyst or any specialist who deals with vast amount of text that could not be read normally, now they could access the data that they care more just by reading a concise summary of everything [4]. Moreover, summaries reveal potential similarities in a big collection of documents that allow the categorization of similar documents to groups [4]. This kind of clustering techniques find useful applications to many cases, while even the documents that end up into the same cluster might produce some useful results for the documents' connection. Generally, summaries intend to provide the document's author intent and a more general abstract idea of the document, irrespective of how it is produced (abstractive generated by a human or extractive by a theoretical algorithm). However, these 'neutral summaries' can provide meaningful result if the author had a clear intent, or more precisely, if his ideas are clearly expressed and the text is well written and structured. Therefore, it is important to identify what happens in that case with the summary the algorithmic techniques produce. Furthermore, in a large pool of texts the author's intent should matter a lot since there is not a single intent and that is why an automated summarizer can help a lot to understand the true topic of a document and may reveal connections and information that were previously unknown and would not be detectable due to the volume of the documents. So finally, the issue that becomes apparent is how to evaluate a system that produces automatically a summary, since it is also fundamentally difficult to determine what a good summary is [6]. As a result the 4th section explores different ways that this evaluation is performed and various possible ways we can evaluate the precision and the effectiveness of a summarization system.

1.3 Types of Summary

There are multiple classifications for summaries, since they could be categorized based on different criteria, as it is investigated in various papers [1][2]. However, in this paper we tried to cover the most comprehensive criteria for classifying summaries [19]. Based on the different approaches of analyzing the texts and generation of the summary, text summarization systems are divided to extract and abstract systems. The extract summary is formed by reusing the portions of the main text like words and sentences. In this type of summary the most important information of the main text which is usually the first sentence of each paragraph, special names, italic or bold phrases are copied to the final summary. Unfortunately, extracts suffer from inconsistencies, lack of balance, and lack of cohesion [16]. One example of a system which use extract summary is Summ-It applet which is designed by Surrey University. In an abstract summary, the summarized text is an interpretation of an original text. The process of producing involves rewriting the original text in a shorter version by replacing wordy concept with shorter ones. At first, the system analyses the main text and then it presents its comprehension from the text in a human understandable form. For example SUMMARIST [9] includes modules to perform topic interpretation and summary generation which enables it to produce abstract summaries. Other than that summaries can be categorized based either on their *details* to *indicative* and *informative*, or on their *content* to *generic* and *query-based summaries*, or on their *limitation* on the input text to *domain dependant*, *genre specific*, and *independent*, or on the *number of input documents*, or finally on the *language* to *mono lingual* and *multi lingual* [19].

1.4 Albanian Language Structure

The Albanian language is an Indo-European language, mostly spoken in Albania, Kosovo and in other parts of the Balkans. There are two main dialects Gheg and Tosk, and the official Albania language is written in Roman alphabet [10]. In our case we consider the official Albanian language, which has 7 vowels and 29 consonants. The vowels are represented by single Latin letters (a, e, ë, i, o, u, y), and the consonants by single letters (b, c, ç, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, x, z), and combination of different letters (dh, gj, ll, nj, rr, sh, th, xh, zh). Similarly to other well-known language Albanian also include stop-words like: "dhe", "sepse", "kur", etc. Regarding the Albanian grammatical units, nouns are gender, number and case specific while suffixes may be added to them. Some noun plurals may have an irregular form and not follow the usual rules. Adjectives are similar to nouns and follow the noun they describe. Finally, verbs have many forms and irregularities. The grammar & formal distinctions of Albanian are inherited by the Romance languages and of Modern Greek [10].

1.5 Summarization Workflow

Automatic summarization is a multistage process. Before describing each of the subtasks in the workflow it is important to clarify that the term context, which appears in the workflow and further in this report quite frequently, refers to a piece of text smaller than a document but larger than a single word.

The workflow in the diagram includes six subtasks:

- ***Preprocessing*** takes a raw text as an input and applies some basic routines to transform or eliminate textual elements that are not useful in further processing of textual data. In extraction based summarization the output of this step is often a set of sentences where non-content words and punctuations are removed and content words are reduced to their base forms.
- ***Feature selection*** identifies words, phrases or some other cues that appear in a preprocessed text and may serve as valuable characteristics of textual contexts. In automatic summarization these cues are often referred to as features, topic terms or topic signatures.
- ***Context representation*** makes use of features obtained from the feature selection subtask to represent contexts in a form suitable for further processing. Contexts are often represented as context vectors.
- ***Content selection*** identifies contexts that should be included in a summary. A variety of approaches can be used for this subtask. Some of them require to compute the similarities between textual contexts.
- ***Context ordering*** arranges contexts selected in the content selection subtask to construct a coherent and readable text.
- ***Sentence realization*** applies techniques that operate on a sub-sentence level with the goal to improve the readability and clarity of a text. Sentence simplification is one of these techniques.

1.6 Sentence Features

Next to the Preprocessing step, by using sentence feature each sentence in the document is represented by a vector attribute. Each attribute represents the data used for their task. These features were used, gives a value between "0" to "1". Below are listed some of these features [19]:

Content word (Keyword) , Indicators/Cue Phrases, Legal vocabulary, Paragraph Structure, Citation, Term Weight, Named Entity Recognition, Similarity to Neighboring Sentences, Absolute Location, Sentence-to-Sentence Cohesion, Proper Noun and Sentence Position.

1.7 Approaches to Sentence Extraction

In order to generate a high quality summary different NLP techniques must be used. The generated summary is a collection of original sentences, and most of summarization systems produce summary based on key sentence selection. There are three different approaches for scoring and selecting sentences [18]:

(a) *Statistical approaches,* ***(b)*** *Linguistic approaches* (Lexical chain, WordNet, Graph theory, Clustering) and ***(c)*** *Rhetorical approaches.*

2. RELATED WORK

Previous work includes the pioneering work by Luhn in 1958 [12], and Edmundson in 1969 [11], while more recent work is of [20] that uses a recursive neural network, which operates on a parsing tree of a sentence, to rank sentences for summarization. Cheng and Lapata [13] successfully used a neural-network-based sentence extractor, which considered the document encodings and the previously selected sentences, as well as the current sentence in making its decisions. Parveen et al. [8] used a graph-based approach, modeling a document as "a bipartite graph consisting of sentence and entity nodes". Ren et al. [17] achieved state-of-the-art results through a regression-based approach. A variety of engineered features are used as inputs into a regression model. The single highest rated sentence is selected, and the process is repeated.

3. TEXT SUMMARIZERS FOR ALBANIAN LANGUAGE - EXPERIMENTAL STUDY

We aim to provide an initial experimental study on text summarization with actual methods and tools for texts written in Albanian language. Of course we must explain that are studding the text summarization as a technique of text mining.

3.1 Specific Objective

- To study linguistic aspect of Albanian language.
- To conduct experiments to choose better techniques and methods for Albanian text summarization.
- To evaluate and test the performance of the constructed summarizers.
- To design, adopt and develop a suitable algorithm based on the identified techniques and defined equations.

- To develop a prototype of automatic Albanian news text summarizer based on the best performed algorithm.

3.2 Methodology

We have started our work by developing a news aggregator for Albanian news using Scrapy[29]. In order to gain more context details (i.e. latest stories, important news) by the location of the page where the news is present, we do not use RSS feeds. News are stored using a NoSQL database (CouchDB[30]) that support android devices. In addition, we will be utilizing the NLTK Python library to build our pipeline, constructing four single document summarizers with respective algorithms [7] given below:

Auto Summarizer 1:	***Auto Summarizer 2:***
A simple sentence scoring to rank the sentences: ***Step 1:*** **Calculate total words within the document.** ***Step 2:*** **Remove the stop words from the input text** ***Step 3:*** **Calculate the content words in each sentence.** ***Content words=Total words-Stop words*** ***Step 4:*** **Calculate the sentence weight with type 1:** ***Sentence weight= (Content words/Total words) *100*** **Sort the sentences & finally generate the summary**	**Simple sentence weight learning method:** ***Step 1:*** Split the text into sentences and words. ***Step 2:*** Find the number of words in each sentence ***Step 3:*** Find the number of words in maximum lengthy sentence ***Step 4:*** Calculate the sentence score *Sentence score=Nr of words /Nr of words in maximum length sentence* ***Step 5:*** Rank the sentences and highest ranking sentences as summary
Auto Summarizer 3:	***Auto Summarizer 4:***
Algorithm for Sentence Weighting: **This system uses adjectives, adverbs and nouns as key terms** ***Step 1:*** **Split the text into sentences and words.** ***Step 2:*** **Calculate the position score of each sentence.** **The first sentences in each document got highest score.** ***Step 3:*** **Add additional score to numeric held sentences.** ***Step 4:*** **Assign score to the keywords** ***Step 5:*** **Calculate sentence score. Sentence score is the sum of words in the sentences.** ***Step 6:*** **Assign feature weight to the sentences. Sum all the feature score and extract the important sentences**	**Graph Theoretic Method:** ***Step 1:*** Split the text into words and sentences. ***Step 2:*** Construct graph and represents each vertex as sentences and edges shows the occurrence of words in the sentences. ***Step 3:*** Calculate the total number of words. ***Step 4:*** Find the affinity weight (***aw***) of word *aw=document frequency of a word/total words*. ***Step 5:*** Calculate the sentence weight. It is the sum of affinity weight. ***Step 6:*** Calculate the Levenshtein similarity weight. It is difference between maximum length of two sentences and *Levenshtein Distance* (LD) of two sentences then it is divided by maximum length of two sentences. *Levenshtein Distance* is the distance between two words. ***Step 7:*** Calculate the vertex weight. ***Step 8:*** Rank the sentences on the basis of similarity weight and vertex weight.

Our experiments are structured as follows: given the original news text, we generate a ten of sentences long summary and compare it to the given summary. In addition we constructed a basic Albanian stemmer based on work of Sadiku and Biba [15] and tested this effect on summaries. Each summary will be generated by four summarizers. At the end they are evaluated with ROUGE-1, ROUGE-2 and ROUGE-L method.

4. EVALUATION OF SUMMARIZATION SYSTEMS

Evaluating a summarization system is to identify how well the system fulfills the given requirements. Normally, evaluation is done by comparing a gold standard (usually produced by a human) to the results of the summarization

[29] https://scrapy.org/
[30] http://couchdb.apache.org/

technique. Therefore, an automatic evaluation is required since it is impossible to manually evaluate many summaries quickly and consistently without bias. Although the use of models summaries (normally human ones) is quite common, some authors have been working toward the automatic evaluation of summaries without using references, which is one of the most challenging strategies nowadays [1][14].

There are different methods that could be used so as to evaluate the summaries as a*utomatic vs. manual evaluation, intrinsic vs. extrinsic evaluation, inter-textual vs. intra-textual and the ROUGE evaluation method* we use and explain below. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [6] compares summaries by counting the number of n-gram overlaps for automatic generated summaries and human written summaries. It is based on recall and measures how well automatically generated summary covers the content in human-generated summary. ROUGE includes many techniques like: ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-N (the one we use in practice):

$$\mathbf{ROUGE-N} = \frac{\sum \mathbf{S} \in \{\mathbf{ReferenceSummaries}\} \sum_{\mathbf{gram_n} \in \mathbf{S}} \mathbf{Count_{match}}(\mathbf{gram_n})}{\sum \mathbf{S} \in \{\mathbf{ReferenceSummaries}\} \sum_{\mathbf{gram_n} \in \mathbf{S}} \mathbf{Count}(\mathbf{gram_n})} \quad (1)$$

where n is the length of the n-gram (*gramn*), and *Countmatch(gramn)*is the maximum number of n-grams that occur in both automatic summary and gold-standard summary [14]. We also use ROUGE-L which calculates the longest common subsequence (LCS). N-gram co-occurrence statistics is a good automatic scoring metric in single-document summarization task as shown in [5]. Lin [5] compared the different ROUGE methods with single document DUC data and found out that all of the methods correspond with human evaluations (ROUGE-1 or ROUGE-2 where the most promising). This correspondence appeared to be dependent to the length of the summaries, stemming and stop word removal which also increased the recall and the precision of the summary. ROUGE does not care about the grammar, the readability or the flow of the text, which resembles to BLEU measure, but BLEU is based on precision which prefers accuracy [5].

$$\mathbf{Recall} = \frac{\mathbf{TruePositives}}{\mathbf{TruePositives+FalseNegatives}} \quad (2) \qquad \mathbf{Precision} = \frac{\mathbf{TruePositives}}{\mathbf{TruePositives+FalsePositives}} \quad (3)$$

Recall and Precision are defined above. Recall is a ratio that shows how well the gold standard summary is covered from the generated summary, while Precision is the quality of the automatic summary as it decreases as the number of false positives grows. Finally, F-Measure is given by

$$\mathbf{F-Measure} = \frac{\mathbf{2*Precision*Recall}}{\mathbf{Precision+Recall}} \quad (4)$$

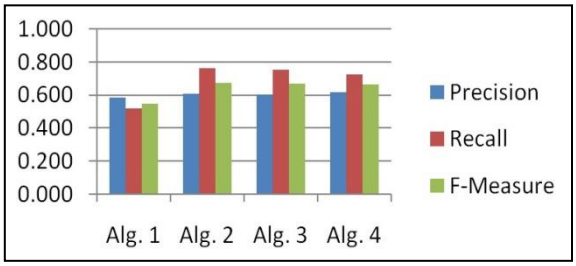
5.RESULT

The comparative result of the reviewed algorithms for ***ROUGE-1***, ***ROUGE-2*** and ***ROUGE-L*** is illustrated in the following tables and respective charts:

Table.1 Results for the comparison of average Precision, Recall & F-Measure Score for the 4 summarizers (on the left with stemming) for ROUGE-1.

Algorithm	Precision	Recall	F-Measure
Alg. 1	0.585	0.521	0.545
Alg. 2	0.609	**0.762**	**0.673**
Alg. 3	0.604	0.754	0.667
Alg. 4	**0.616**	0.726	0.663

Algorithm	Precision	Recall	F-Measure
Alg. 1	0.595	0.511	0.539
Alg. 2	0.604	**0.759**	**0.669**
Alg. 3	0.597	0.712	0.647
Alg. 4	**0.605**	0.742	0.663



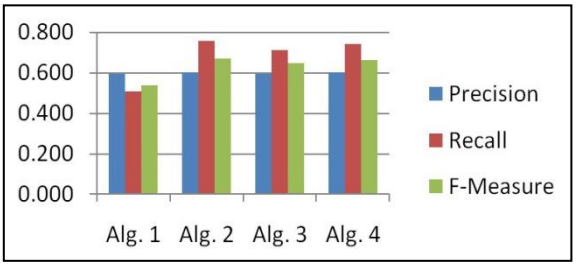


Figure 1. Graphical representation for the comparison of average Precision, Recall & F-Measure Score for the 4 summarizers (on the left with stemming) for ROUGE-1.

Table 2. Results for the comparison of average Precision, Recall & F-Measure Score for the 4 summarizers (on the left with stemming) for ROUGE-2.

Algorithm	Precision	Recall	F-Measure
Alg. 1	0.464	0.418	0.435
Alg. 2	**0.532**	**0.686**	**0.596**
Alg. 3	0.527	0.676	0.589
Alg. 4	0.531	0.651	0.582

Algorithm	Precision	Recall	F-Measure
Alg. 1	0.483	0.420	0.441
Alg. 2	**0.525**	**0.682**	**0.590**
Alg. 3	0.515	0.625	0.562
Alg. 4	0.524	0.668	0.583

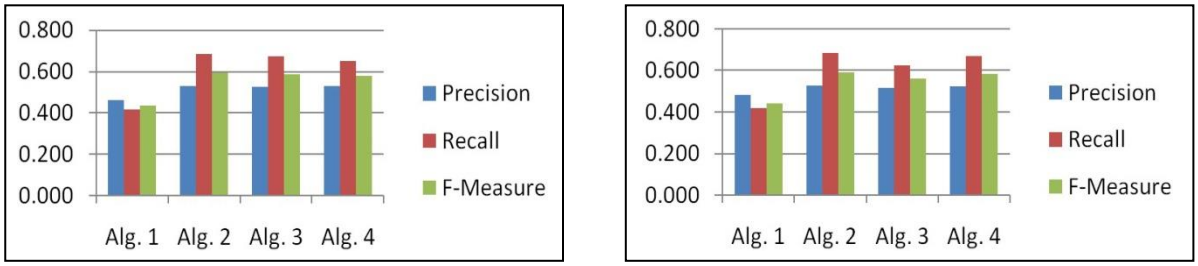


Figure 2. Graphical representation for the comparison of average Precision, Recall & F-Measure Score for the 4 summarizers (on the left with stemming) for ROUGE-2

Table 3. Results for the comparison of average Precision, Recall & F-Measure Score for the 4 summarizers (on the left with stemming) for ROUGE-L

Algorithm	Precision	Recall	F-Measure
Alg. 1	0.571	0.510	0.521
Alg. 2	0.603	**0.754**	**0.647**
Alg. 3	0.599	0.746	0.641
Alg. 4	**0.609**	0.717	0.643

Algorithm	Precision	Recall	F-Measure
Alg. 1	0.585	0.504	0.515
Alg. 2	0.597	**0.749**	**0.641**
Alg. 3	0.589	0.702	0.624
Alg. 4	**0.598**	0.733	0.639

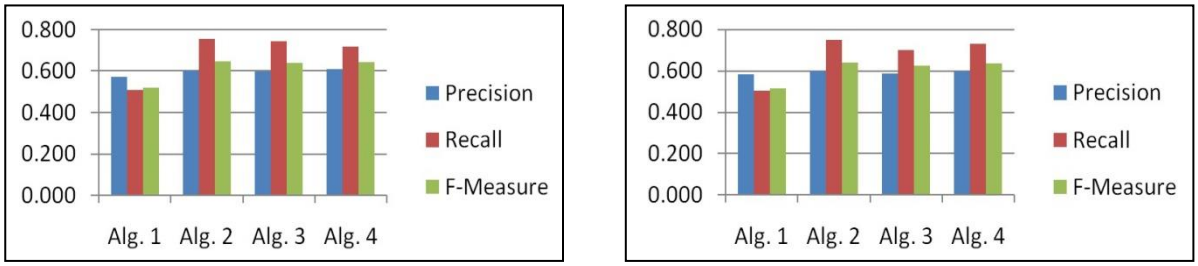


Figure 3. Graphical representation for the comparison of average Precision, Recall & F-Measure Score for the 4 summarizers (on the left with stemming) for ROUGE-L

6. CONCLUSION AND FUTURE WORK

In this paper we perform a thorough experimental evaluation of four algorithms for text summarization in Albanian language. At the beginning of our work we reviewed the state of art and the recent work done in this field. We created a text corpus of 65 Albanian news articles collected by well-known online Albanian newspapers, focusing mainly on quality authors. The corpus has six subjects related to nowadays discussion in Albania. Each subject is used as a corpus to evaluate the performance of the algorithms. Firstly, we cleaned the dataset passing it in a preprocessing phase composed by a stop-word removal and a basic Albanian stemmer, depending on the certain algorithm. We constructed two set of experiments, one without stemming and next one with stemming. The experimental results show that there almost all algorithms performed well, whereas the *summarizer_2* performed better. The stemming increase slightly the accuracy. The results seems a bit better than current state of the art, but it is not reflecting the reality. It has to do with the way the golden summaries were made, not in the standard form (abstractive) but in extractive. We evaluated the performance using ROUGE.

In the future, it would be of a great interest to evaluate the performance of the abstractive summarization algorithms in a bigger corpus and in a cross-domain corpus in Albanian language. Also, it would be interesting to construct a POS tagging system and to test more algorithms, focusing in deep learning algorithms.

REFERENCES

A. Louis & A. Nenkova, Automatically evaluating content selection in summarization without human models, Proceedings of the Conference on Empirical Methods in Natural Language Processing, Singapore, ACL Volume 1, pp. 306–314, 2009.

A. Nenkova & K. McKeown, Automatic Summarization, Foundations and Trends in Information Retrieval, Vol. 5, No 2-3, pp. 103-233, 2011.

ANSI Guidelines for Abstracts ANSI/NISO Z39.14–1997 (R2015), Baltimore, Maryland, U.S.A.,NISO Press, 2015.

C.E. Crangle, Text Summarization in Data Mining, Proceedings of the First International Conference on Computing in an Imperfect World, D. Bustard, W. Liu, and R. Sterritt (Eds.): Soft-Ware 2002, LNCS 2311, pp. 332–347, 2002.

C.Y. Lin and E. Hovy, Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics, Proceedings of HLT-NAACL, Main Papers, Edmonton, pp. 71-78, 2003.

C.Y. Lin, ROUGE: A Package for Automatic Evaluation of Summaries, Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, Association for Computational Linguistics, pp. 74–81, 2004.

D. K. Kanitha, D. M. Noorul Mubarak & S. A. Shanavas, Comparison of Text Summarizer in Indian Languages, International Journal of Advanced Trends in Engineering and Technology, Volume 3, Issue 1, pp. 79-82, 2018.

D. Parveen and M. Strube. Integrating importance, non-redundancy and coherence in graph-based extractive summarization, IJCAI, pp. 1298–1304, 2015.

E. Hovy and C. Y. Lin, Automated text summarization in SUMMARIST, MIT Press, pp. 81–94, 1999.

E.P. Hamp, Albanian Language, Encyclopedia Britannica, 2016.

H.P. Edmundson, New methods in automatic extracting, Journal of the ACM, 16(2), pp. 264-285, 1969.

H.P. Luhn, The Automatic Creation of Literature Abstracts, IBM Journal, pp. 159-165, 1958.

J. Cheng and M. Lapata, Neural summarization by extracting sentences and words, arXiv preprint arXiv:1603.07252, 2016.

J.M. Torres-Moreno, H. Saggion, I. da Cunha & E. SanJuan, Summary evaluation with and without references, Polibits: Research Journal on Computer Science and Computer Engineering with Applications 42, pp. 13–19, 2010.

J. Sadiku and M. Biba, Automatic stemming of Albanian through a rule-based approach, Journal of International Scientific Publications: Language, Individual Society, Volume 6, Part 1, pp. 173-190, 2012.

K. Jezek, and J. Steinberger, Automatic Text Summarization (the state of the art 2007 & new challenges), Znalosti, pp. 1-12, 2008.

P. Ren, F. Wei, Z. Chen, J. Ma, and M. Zhou. A redundancy-aware sentence regression framework for extractive summarization, Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 33–43, 2016.

S. Gholamrezazadeh, M. A. Salehi & B. Gholamzadeh, A comprehensive survey on text summarization systems, Proceedings of the 2nd International Conference on Computer Science and its Applications, IEEE, pp. 1-6, 2009.

V. Gupta & G. S. Lehal, A Survey of Text Summarization Extractive Techniques, Journal of Emerging Technologies In Web Intelligence, Vol. 2, No. 3, pp. 258-268, 2010.

Z. Cao, F. Wei, L. Dong, S. Li, and M. Zhou, Ranking with recursive neural networks and its application to multi-document summarization, AAAI, pp. 2153–2159, 2015.