GRANTEE SUBMISSION REQUIRED FIELDS

Title of article, paper, or other content

Extractive Summarization using Cohesion Network Analysis and Submodular Set Functions

All author name(s) and affiliations on PDF. If more than 6 names, ERIC will complete the list from the submitted PDF.

Last Name, First Name	Academic/Organizational Affiliation	ORCID ID
Cioaca, Valentin Sergiu	Computer Science Department, University Pol	
Dascalu, Mihai	Computer Science Department, University Pol	
McNamara, Danielle S.	Psychology Department, Arizona State Univer:	

Publication/Completion Date—(if *In Press,* enter year accepted or completed) 24 February 2021

Check type of content being submitted and complete one of the following in the box below:

- □ If article: Name of journal, volume, and issue number if available
- ☑ If paper: Name of conference, date of conference, and place of conference
- □ If book chapter: Title of book, page range, publisher name and location
- □ If book: Publisher name and location
- □ If dissertation: Name of institution, type of degree, and department granting degree

2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC),

DOI or URL to published work (if available) https://doi.org/10.1109/SYNASC51798.2020.00035

Acknowledgement of Funding— Grantees should check with their grant officer for the preferred wording to acknowledge funding. If the grant officer does not have a preference, grantees can use this suggested wording (adjust wording if multiple grants are to be acknowledged). Fill in Department of Education funding office, grant number, and name of grant recipient institution or organization.

"This work was supported by U.S. Department of Education **[Office name]** National Center on Educat through **[Grant number]** R305A1900 to **Institution]** Arizona State U.The opinions expressed are those of the authors and do not represent views of the **[Office name]** National Center on Educa or the U.S. Department of Education.

2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) | 978-1-7281-7628-4/20/$31.00 ©2020 IEEE | DOI: 10.1109/SYNASC51798.2020.00035

Extractive Summarization using Cohesion Network Analysis and Submodular Set Functions

Valentin Sergiu Cioaca
Computer Science Department
University Politehnica of Bucharest
Bucharest, Romania
valentin.sergiu.cioaca@gmail.com

Mihai Dascalu
Computer Science Department
University Politehnica of Bucharest
Bucharest, Romania
mihai.dascalu@upb.ro

Danielle S. McNamara
Psychology Department
Arizona State University
Tempe, Arizona, USA
dsmcnama@asu.edu

Abstract—**Numerous approaches have been introduced to automate the process of text summarization, but only few can be easily adapted to multiple languages. This paper introduces a multilingual text processing pipeline integrated in the open-source *ReaderBench* framework, which can be retrofit to cover more than 50 languages. While considering the extensibility of the approach and the problem of missing labeled data for training in various languages besides English, an unsupervised algorithm was preferred to perform extractive summarization (i.e., select the most representative sentences from the original document). Specifically, two different approaches relying on text cohesion were implemented: a) a graph-based text representation derived from Cohesion Network Analysis that extends TextRank, and b) a class of submodular set functions. Evaluations were performed on the DUC dataset and use as baseline the implementation of TextRank from Gensim. Our results using the submodular set functions outperform the baseline. In addition, two use cases on English and Romanian languages are presented, with corresponding graphical representations for the two methods.**

Keywords-extractive summarization; spaCy framework; submodular functions; TextRank; Cohesion Network Analysis; Word Mover's Distance

I. Introduction

Text summarization is the process of compressing one or more documents to about 20-30% of the original text(s), while maintaining the main ideas and without significantly modifying the main points and the fidelity of the content. As online data is constantly increasing in size, information indexing and searching has become more costly in terms of performance and complexity [1]. Moreover, people tend to easily lose their interest when searching for key information in long text. Thus, making available shorter, compressed text containing the most relevant ideas considerably reduces reading time and increases success rates for identifying the desired information [2]. Increased time to search for information is crucially important in the commercial sector wherein information on the Internet can take time (and thus money) to search and digest. Web page summarization can lower costs by reducing the time to understand and search for information [3].

Despite the benefits of summaries, it is impossible to manually create summaries – there are simply far too many texts online to summarize. Automated summarization offers a viable solution [4]. Extractive summarization is the process of choosing the most important sentences from a target text, which collectively *summarize* the text. Most of these automated programs, however, solely target the English language. As such, our objective is to develop an automated program that can be retrofit to cover multiple languages – in essence, any language, any text, any time.

II. State of the art

Text summarization methods are generally classified into two different types based on the output: *extractive* and *abstractive*. The extractive method is the easiest and most frequently employed summarization approach. The common practice of extractive summarization involves compiling the most important sentences of the text. The *topic words* method [5] was one of the first approaches of this type. Its principal tactic is to identify the topic words (words best describing a subject, usually identified by counting) and to classify the sentences in a text by quantifying the presence of these words. It was subsequently improved with probabilistic patterns [6], and more complex formulas for score computation such as Tf-Idf (Term Frequency – Inverse Document Frequency).

An alternative approach for summarization consists of selecting the most representative sentences from the document. Latent Semantic Analysis (LSA) [7] provides the means to perform such a selection of representative sentences [8]. LSA is an unsupervised method widely used in Natural Language Processing, originally introduced for information retrieval systems (i.e., Latent Semantic Indexing – LSI). LSA creates a vector space that maps the texts and words in a high-dimensional semantic space using a matrix factorization known as Single Value Decomposition (SVD), followed by a projection on the most representative *k* dimensions. The summarization is usually constructed by choosing the most relevant sentence for each concept/topic enriched with different heuristics.

Further methods of extractive summarization rely on graph representations. Most of underlying algorithms for graph-based summarization are derived from PageRank [9]. The central idea of graph-based ranking is bringing the text to a graph representation. Depending on context, text units of

978-1-7281-7628-4/20/$31.00 ©2020 IEEE
DOI 10.1109/SYNASC51798.2020.00035

various sizes can form the vertex set. The connections between any pair of vertices are weighted by similarity scores. In such a way, the extractive summary is assembled by identifying the most important units in a text. TextRank [10] is one of the most renowned unsupervised extractive summarization algorithms that relies on graph-based model.

Abstractive methods are more sophisticated and resemble to a higher degree human approaches of building summaries. The obtained summary consists of phrases and sentences that are generated in whole or in part based on the model's understanding and representation of the original text. The key aspect of abstractive summarization is to remember previous information. In this respect, Recurrent Neural Networks (RNNs, including LSTMs or Long Short-Term Memory cells [11]) are renowned for their capability of learning on data with long temporal dependencies. Encoder-decoder architectures can be designed to address the challenges of abstractive summarization; however, large datasets of manually created summaries are required. Notably, however, the process of manual summarization is not only time consuming, but also subjective from the human point of view, as two summaries are unlikely to be completely similar when they are written by different individuals. It is challenging to reliably discriminate whether a given summary is better or worse, more or less comprehensive, or more or less comprehensible than another one. Moreover, and most important here, these methods do not generalize well on new types of input documents or on larger texts, and cannot be applied on texts in different languages.

III. The Current Study

Our solution was inspired from the presumption that extractive summarization should be facilely applied on various kinds of documents, including those written in languages other than English. Moreover, we set out to methodically meet the requirements of extractive summarization, without using a very large volume of input data or computational power, while still preserving the characteristics of a well composed summary.

Taking into consideration these constraints, we introduce two different approaches, wherein the primary focus is to capture the semantic relations between sentences. Thus, we rely on Cohesion Network Analysis (CNA) [12] to maximize the extent to which the central text elements from a semantic point of view are captured within the extractive summary. CNA combines text cohesion computed using various semantic models with Social Network Analysis. CNA generates a cohesion graph that reflects links between various text units (e.g. word, sentences, paragraphs, etc.) [13]. In our case, the sentence selection task for summarization is reduced to defining a set of rules to be followed for retrieving the most relevant vertices in the cohesion graph.

IV. Method

We introduce two multilingual approaches: a) CNA TextRank, an extension of the initial implementation [10], which now uses newer semantic models and the CNA graph representation, as well as b) summarization with submodular set functions. Both models are implemented using Python v3.7 and are integrated in the open-source *ReaderBench* framework [14].

A. Corpus

Although the proposed method can be adjusted to any language, it requires pre-trained word embeddings for the prespecified language and an annotated dataset to fine tune the hyper-parameters. Finetuning the hyper-parameters is important due to its large impact on the results; this process should be performed on all languages for which annotated datasets exist. The evaluations in this study were performed on two DUC datasets, namely: DUC 2001 [15] and DUC 2002 [16], both written in English. The hyper-parameters from the English model were also used for Romanian since, to our knowledge, there are no annotated datasets currently available.

B. Text cleaning and preprocessing

Our algorithm requires cleaned, pre-processed inputs. First, every document is split into sentences (i.e., a process known as sentence tokenization), and each sentence is processed independently using the following common steps:

- remove section labels, if applicable;
- remove punctuation marks;
- remove non-alphanumeric characters;
- remove sequential spaces;
- remove digits.

The second text preprocessing stage is more complex and language oriented. The central elements of this phase consist of reducing inflected words to their dictionary forms (i.e., lemmatization), followed by computing the sentence embeddings. Stop words and lemmas having fewer than two characters are eliminated. In addition, a good heuristic that improved our results was to retain sentences with more than three distinct content words. Taking into consideration the multilingual aspect of the summarization process, it was mandatory to find a versatile solution for language manipulation.

spaCy (https://spacy.io) is a free open-source library written in Python and Cython used mainly for Natural Language Processing (NLP). In addition to the wide variety of functionalities, it also provides support for many languages (53+ languages including Romanian). Text processing in spaCy is done using a standardized pipeline (see Fig. 1). The text is passed through several sequential stages of pre-defined operations until the final result is obtained. By default, spaCy uses a restricted model to perform faster computations, but the larger models should be used for additional features, such as word embeddings. In case of a new language, spaCy can provide a pre-trained model if the language is supported by the developers; alternatively, a custom user-trained model can be integrated.

The desired outcome of this initial step is the removal of less relevant words so that the text's overall coherence is not greatly impacted. Therefore, stop words are eliminated, and lemmatization is applied to increase the similarity score between structures containing words with different inflectional and derivative forms.

C. Text cohesion

In contrast to the simple similarity function from the TextRank algorithm that relies on common token overlap [10], we use word embeddings. Word embeddings translate words into a vector representation within a semantic space, in which similar terms are in the same proximity. For example, word2vec [17] assigns a vector of real values to each word as an output of a neural network. Besides cosine similarity, the Word Mover's Distance (WMD) [18] can be used as a weighted semantic distance function. WMD is a semantic editing distance that calculates the minimum distance to be traveled from one set of word embeddings to another, given a pre-trained semantic space. Accordingly, the distance between two sentences is the minimum cumulative distance that all words from the first sentence need to "travel" to match the second sentence.

D. Summarization with CNA TextRank

Derived from the original TextRank [10] algorithm, the proposed method replaces the standard similarity function with estimates of text cohesion derived from the CNA graph. Starting from the execution schema of the TextRank algorithm in Figure 2, the major difference for our semantic version lies in the computation of sentence vectors and the construction of the similarity matrix. The preprocessing stage remains the same, as it was described in a previous section. The WMD pair distances between sentences are used to adjust the weights of specific links from the CNA graph, more specifically intra-textual links between sentences. WMD distances are implemented by adding a new component in the *spaCy* pipeline, while the resulting scores are used to populate the similarity matrix. The NetworkX v2.2 package [19] was used for graph generation and PageRank computation.

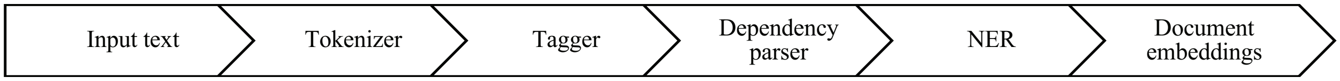


Figure 1. Text processing pipeline from spaCy.

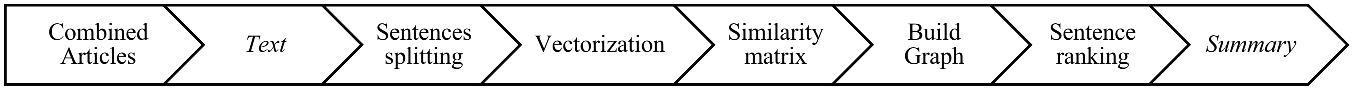


Figure 2. TextRank execution steps.

E. Summarization with submodular set functions

Submodular set functions have been successfully applied in game theory, economy or graph theory, and more recently there is an increased interest in using these functions in domains such as artificial intelligence and machine learning. Some important benefits of using submodular functions reside in their capability of solving difficult computational problems, as well as their property of diminishing returns (i.e., a decrease in the marginal/incremental output of the function).

In the context of document summarization, the text needs to be perceived as a set of separate sentences. Having this setup, the summarization problem is reduced to a submodular function maximization, under the constraint of a given summary size, which is known to be an NP-hard problem. Fortunately, there is a polynomial time solution with a good approximation of $(1 - 1/e)OPT$, where e is the base of the natural logarithm and OPT the value of the optimal solution, if the function is monotone [20]. In view of previous definitions and arguments, the solution to the summarization problem consists of finding a submodular function that also satisfies the property of monotony.

A good summary considers two central properties of the target text sentences: relevance and non-redundancy [21]. Usually, these two properties are calculated individually and gathered under the same function, by promoting the first and penalizing the second. Lin and Bilmes [21] proposed a solution that no longer penalizes redundancy, but encourages diversity. With this approach, the monotonicity of the objective functions can be better preserved, as the redundancy function usually violates it. Thus, the quality of a summary is described by the following linear combination:

$$F(S) = L(S) + \lambda R(S) \quad (1)$$

where $L(S)$ calculates the similarity or the fidelity between the summary and the original text, $R(S)$ is the diversity reward function, and λ is a trade-off coefficient.

Definitions of these functions may vary, as long as their monotony property is met. Lin and Bilmes [21] also propose several variants of submodular set functions for text summarization. For example, the coverage and the diversity reward function of summary S, in respect to text V, are defined as follows:

$$L(S) = \sum_{i\in V, j\in S} sim_{i,j} \quad (2)$$

$$R(S) = \sum_{k=1}^{K} \sqrt{\sum_{j\in P_k \cap S} \frac{1}{N} \sum_{i\in V} sim_{i,j}} \quad (3)$$

where $sim_{i,j}$ is the sentence similarity function between i and j and $P_k, k = 1,..K$ is a partition of V.

Our summarization algorithm implements the above submodular functions, as they were defined in the study of Lin and Bilmes [21]. One unique aspect of our implementation resides in hyperparameters tuning, namely λ – the trade-off coefficient and K – the number of partitions or clusters. The text partitioning was achieved using the k-means clustering algorithm. The most ideal summaries for our dataset were obtained by having $\lambda = 4$ and $K = 10\% \times N$, where N is the number of sentences from V. These hyper-parameters were experimentally determined using a grid search technique on the DUC 2001 dataset, with $2 \le \lambda \le 7$ and $0.1N \le K \le 0.3N$. The same tuned values from English were maintained across the experiments performed in other languages. Moreover, a saturation margin was also considered when computing L in order to avoid overfitting.

Having defined the submodular monotone function F, the next step involves extracting the final summary. Mathematically, this translates into solving a maximization problem under the size constraint. In other words, the generated summary considers the following equation:

$$\max_S F(S), \text{ such that } |S| \le \mathcal{B} \qquad (4)$$

with $\mathcal{B}$ being the length of the expected summary. This problem can be solved in polynomial time with an improved approximation of $1 - \frac{1}{\sqrt{e}}$ by using a *greedy* heuristic approach where at each step the greatest ratio of function gain to scaled cost is being selected [22].

V. Results

A. DUC Validation

The assessment of a summary is complex because it must consider multiple issues such as coherence, cohesion, grammar, intelligibility, and text content. The principal method of evaluating a system summary is by comparing it with a human reference. Since most dedicated datasets for automated summarization contain reference models, the quality assessment of a summary reduces to finding the appropriate comparison criteria.

To this end, ROUGE-N scores [23] reflect the similarity between the automatically generated and the genuine summaries in the dataset (see Tables I and II for recall – R, precision – P and F-score – F). Simply put, in the context of ROUGE, recall refers to how much the reference summary is covered by the system generated summary, whereas the precision measures how much of the system summary was necessary to capture the content. In this case, the most relevant score was ROUGE-1 which refers to the overlap of unigrams (individual words).

This study focused on the DUC 2001 and DUC 2002 datasets. The model summaries were generated by three algorithms: *CNA TextRank*, the algorithm based on *submodular functions*, and a baseline TextRank implementation from *Gensim* v3.4.0 library [24].

The results obtained for the two datasets are summarized in Tables I and II. These results indicate that the submodular functions provide slightly better scores than the other two methods while the CNA TextRank surpasses the *Gensim* version of summarization in terms of recall. A diminished F-score is noticeable for each method resulting from lower precision, which may be explained by differences in the size of the summaries, but also by an incorrect partitioning of the text that specifically influences the diversity function in the case of submodular functions. The partitioning is important for the diversity function because as soon as a sentence is selected from a cluster, the other candidates from the same cluster stat having diminished gain [21]. The diversity is amplified by the trade-off coefficient $\lambda = 4$, whose value was exhaustively searched through the DUC'01 dataset.

TABLE I. ROUGE-N SCORES FOR DUC 2001

	L(S)			***R(S)***			***L(S) + λR(S)***			***CNA TextRank***			***Gensim TextRank***		
	P	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
ROUGE-1	**.334**	**.589**	**.401**	.319	.563	.381	**.325**	**.574**	**.388**	.318	.563	.380	.339	.556	.385
ROUGE-2	.149	.255	.176	.136	.226	.158	.140	.233	.162	.137	.230	.160	.150	.231	.165
ROUGE-3	.095	.159	.111	.086	.136	.098	.088	.140	.100	.087	.139	.099	.096	.141	.104
ROUGE-4	.071	.117	.082	.063	.098	.071	.065	.101	.073	.064	.100	.072	.072	.102	.076

TABLE II. ROUGE-N SCORES FOR DUC 2002

	L(S)			**R(S)**			**L(S) + λR(S)**			***CNA TextRank***			***Gensim TextRank***		
	P	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
ROUGE-1	**.408**	**.539**	**.441**	.393	.523	.425	**.400**	**.531**	**.432**	.389	.523	.423	.424	.505	.427
ROUGE-2	.188	.241	.199	.176	.225	.187	.181	.230	.191	.176	.227	.188	.197	.223	.193
ROUGE-3	.119	.150	.125	.111	.138	.116	.115	.142	.119	.113	.141	.119	.126	.138	.121
ROUGE-4	.087	.108	.091	.081	.099	.084	.083	.102	.086	.083	.102	.086	.092	.099	.087

Table III includes the scores of the submodular algorithm when the semantic distance formula WMD was used to compute semantic similarity. Although recall is lowered, the harmonic mean (i.e., the F-score) is similar or even better than the previous results due to a higher precision; this increase in precision can be justified by the reduction in the number of words in the system summaries.

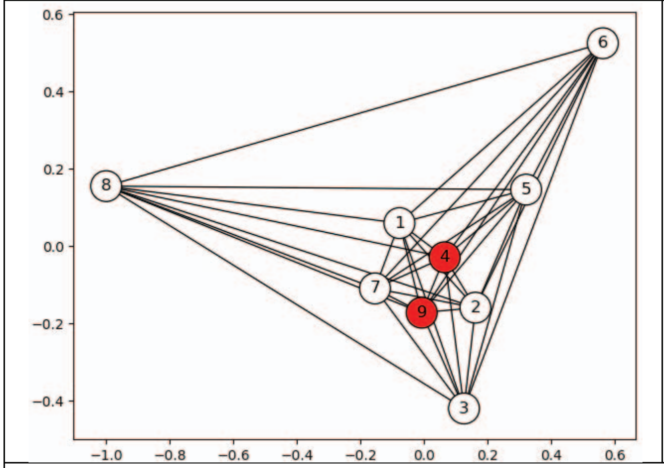
TABLE III. ROUGE-N SCORES FOR CNA TEXTRANK (WMD).

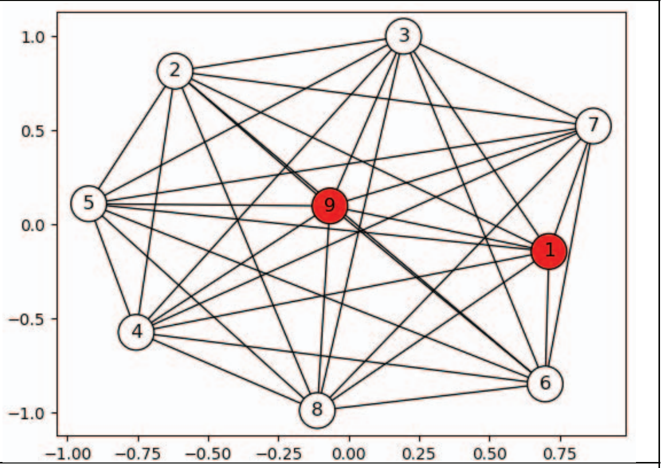
	DUC 2001			**DUC 2002**		
	P	***R***	***F***	***P***	***R***	***F***
ROUGE-1	.343	.542	.395	.409	.487	.422
ROUGE-2	.149	.229	.169	.180	.212	.185
ROUGE-3	.095	.143	.107	.114	.132	.116
ROUGE-4	.070	.105	.079	.083	.095	.084

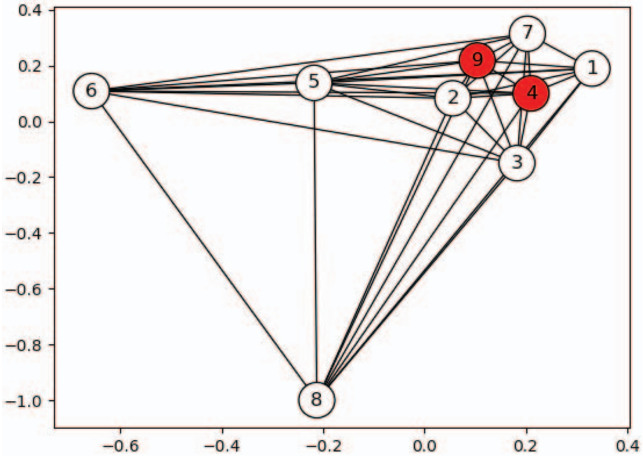
B. Sample multi-lingual use cases

Tables IV and V introduce two running examples of our summarization algorithm with bilingual support (English and Romanian languages), where the proportion of the number of sentences to be selected from the original text is set to 25%. Differences in the graph representations of texts can be observed in either case. This can be explained by the fact that cosine distance applied on the Tf-Idf representation generates a matrix with widely distributed values across the interval. By contrast, the semantic distances using WMD and word2vec return matrices with smaller variations, which corresponds to a tight interval of similarity scores. The small variation is expected in this case because the integrated transformation reveals some hidden semantic relations between sentences, which cannot be outlined by a formula based on mere words counts. Hence, the visual representation of the text using the Fruchterman and Reingold [25] force-directed algorithm is more similar with a complete graph where the selected candidates reside in center (i.e., centered vertices have the smallest cumulative distance to the other candidates), which is the case of CNA TextRank.

TABLE IV. ENGLISH EXTRACTIVE SUMMARIZATION

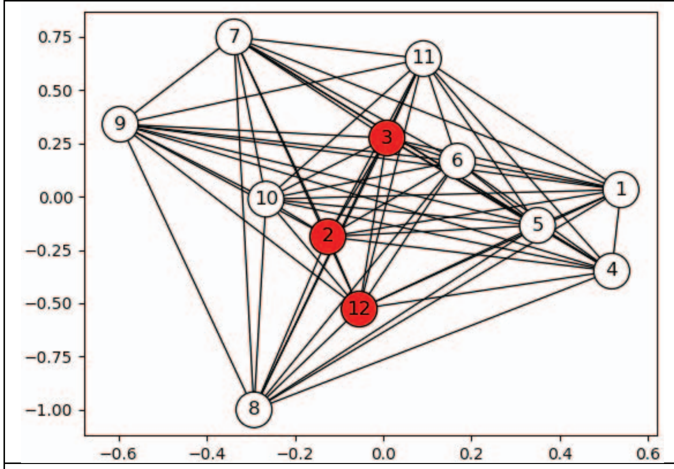
<table>
<tr><th colspan="2">Input</th></tr>
<tr><td colspan="2">S1: Elephants are large mammals of the family Elephantidae and the order Proboscidea.
S2: Two species are traditionally recognized, the African elephant and the Asian elephant.
S3: Elephants are scattered throughout sub-Saharan Africa, South Asia, and Southeast Asia.
S4: Male African elephants are the largest extant terrestrial animals.
S5: All elephants have a long trunk used for many purposes, particularly breathing, lifting water and grasping objects.
S6: Their incisors grow into tusks, which can serve as weapons and as tools for moving objects and digging.
S7: Elephants' large ear flaps help to control their body temperature.
S8: Their pillar-like legs can carry their great weight.
S9: African elephants have larger ears and concave backs while Asian elephants have smaller ears and convex or level backs.</td></tr>
<tr><th><em>Gensim TextRank</em></th><th><em>CNA TextRank</em></th></tr>
<tr><td>


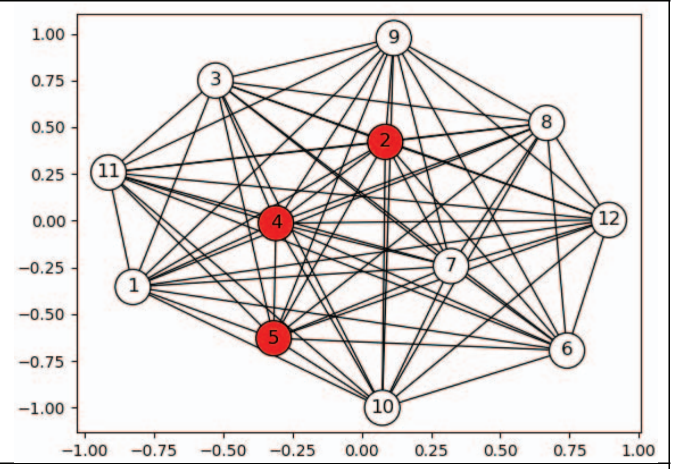
</td><td>


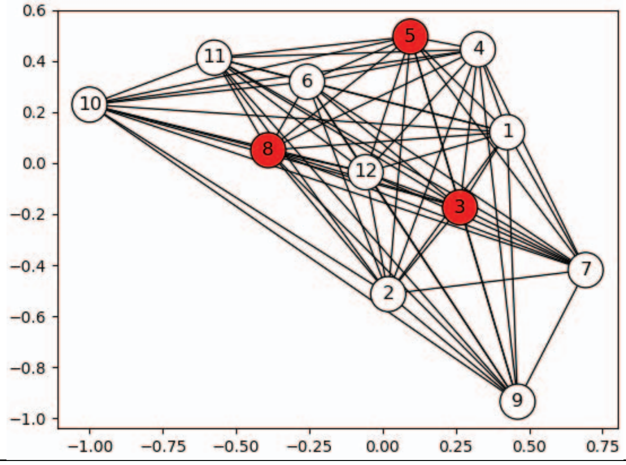
</td></tr>
<tr><th colspan="2"><em>Submodular function sets</em></th></tr>
<tr><td colspan="2">


</td></tr>
</table>

TABLE V. ROMANIAN EXTRACTIVE SUMMARIZATION

<table>
<tr><th colspan="2">Input</th></tr>
<tr><td colspan="2">S1: Fenomenul de încălzire globală nu este nicidecum ceva nou pentru planeta noastră.
S2: Perioadele de temperaturi foarte ridicate și cele glaciare alternează conform unui ciclu pe care oamenii de știință sunt încă departe de a-l înțelege, dar a cărui existentă e sprijinită de dovezi din ce în ce mai numeroase.
S3: Ca idee, în ultima jumătate de milion de ani planeta a trecut prin nu mai puțin de șapte perioade glaciare, iar acum 7.000 de ani am intrat într-o nouă perioadă de încălzire a atmosferei.
S4: Nu e ușor de acceptat ideea că societatea bazată pe utilizarea cărbunilor, petrolului și gazelor naturale e de vină pentru încălzirea globală, care anunță efecte dezastruoase din cauza schimbărilor climatice.
S5: Prin urmare, scepticii schimbărilor climatice contraatacă permanent, aducând în discuție o mulțime de alte posibile cauze care ar putea influența încălzirea globală.
S6: Unul dintre factorii naturali majori care pot influența radical temperatura Pământului este, evident, Soarele.
S7: Există posibilitatea ca o explozie solară să facă „ferfeniță” atmosfera planetei, iar radiațiile letale să treacă de bariera câmpului magnetic al planetei.
S8: Variația activității Soarelui poate duce la o creștere bruscă a temperaturii globale.
S9: Niciun om de știință nu neagă că aceste riscuri există.
S10: Însă aceste scenarii, care pot suna apocaliptic în filmele SF, nu se întâmplă brusc, ci în decursul unor decenii sau, mai degrabă, al unor secole.
S11: Iar, în ultimul secol, măsurătorile NASA și ale altor instituții arată că Soarele nostru a fost „cuminte”, iar influența asupra Pământului constantă.
S12: Doar pe la mijlocul anilor ’50 a avut o activitate ceva mai intensă, dar care, culmea, a coincis cu o scădere temporară a temperaturii medii globale.</td></tr>
<tr><th><em>Gensim TextRank</em></th><th><em>CNA TextRank</em></th></tr>
<tr><td>
7
11
9
3
6
1
10
5
2
4
12
8
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6

</td><td>
9
3
8
2
11
4
12
7
1
5
6
10
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

</td></tr>
<tr><th colspan="2"><em>Submodular function sets</em></th></tr>
<tr><td colspan="2">
5
11
4
6
10
1
8
12
3
7
2
9
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75

</td></tr>
</table>

In addition, most of the selected sentences were considered important in two out of three models. Nevertheless, aside from different underlying graphs as representations, the resulting summaries are also quite different. The most striking difference is between CNA TextRank and submodular summaries because both have the same representation and distance formula (i.e., both models rely on the same similarity scores). This is caused by the diversity reward implemented in the submodular algorithm, which forces the selection of centroids from different clusters of sentences. This behavior is more common when summarizing longer texts (see Table V).

VI. Conclusions and Future Work

In summary, we introduced a multilingual summarization framework exemplified on two different languages, namely English and Romanian. Two extractive summarization algorithms were implemented, a CNA TextRank and an algorithm relying on submodular set functions. Text cleaning and preprocessing phases were performed using *spaCy*, which provides an easily adaptable pipeline for multiple languages. Our evaluation considers the comparison of the ROUGE scores between our implemented solution and the reference summarization algorithm from the *Gensim* library.

Intrinsically, there are similarities in the generated summaries, but the end-goals are different. CNA TextRank summarization maximizes the selection of central sentences from a cohesion point of view, while submodular function sets also account for diversity in sentence selection.

Both algorithms perform text preprocessing and their complexity is reflected in pair-wise comparisons between sentence embeddings which are obtained as average word embeddings. Since the models use pretrained word embeddings, summarizations for texts of normal size can be easily performed on personal computers. Longer texts, for example entire books, should be segmented, and the extractive summarization algorithms should be run independently on coherent segments of texts (e.g., sections or subsections).

The envisioned future of work from the methodological point of view is shaped by two aspects: a) a multi-document summarization, where the current solution can be extrapolated from one document to a cluster of documents, and b) an improved similarity metric with different combined methods. For example, word2vec embeddings may not yield desired results if the model is not trained with enough or relevant data for the domain of the documents. Other techniques such as FastText [26] and BERT contextual embeddings [27] may also be fruitful. In addition, the multilingual aspect must be better put into perspective by using diversified corpora from different languages; where possible, cross-lingual comparisons should be performed.

In sum, exploring the use of different similarity metrics is worth studying more in-depth because each approach has its own advantages and disadvantages, while results after hyper-parameter tuning migh change from one language to another. The particularities of each available model must be considered in the context of the desired application.

Acknowledgment

This research was partially supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0689, Lib2Life - Revitalizing Libraries and Cultural Heritage through Advanced Technologies" within PNCDI III, the Institute of Education Sciences (R305A180144, R305A180261, R305A190063, R305A190050), and the Office of Naval Research (N00014-17-1-2300, N00014-19-1-2424). The opinions expressed are those of the authors and do not represent views of the IES or ONR.

References

[1] A. Das and A. Jain, "Indexing the world wide web: The journey so far," in Next Generation Search Engines: Advanced Models for Information Retrieval, ed: IGI Global, 2012, pp. 1-28.

[2] A. M. Barbosa, "Overview of text summarization in the context of information retrieval and interpretation: Applications for web pages summarization," Pompeu Fabra University - Audiovisual Institute, Barcelona, Spain 2001.

[3] A. Anagnostopoulos, A. Z. Broder, E. Gabrilovich, V. Josifovski, and L. Riedel, "Just-in-time contextual advertising," in 6th ACM Conference on Information and Knowledge Management, Lisboa, Portugal, 2007, pp. 331–340.

[4] E. Hovy and C.-Y. Lin, "Automated text summarization in SUMMARIST," Advances in automatic text summarization, vol. 14, pp. 197–214, 1999.

[5] H. P. Luhn, "The automatic creation of literature abstracts," IBM Journal of research and development, vol. 2, pp. 159–165, 1958.

[6] T. Dunning, "Accurate methods for the statistics of surprise and coincidence," Computational linguistics, vol. 19, pp. 61–74, 1993.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," Journal of the American Society for Information Science, vol. 41, pp. 391–407, 1990.

[8] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in 24th Annual Int. ACM SIGIR Conf. on Research and development in information retrieval, New Orleans, Louisiana, USA, 2001, pp. 19–25.

[9] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," Computer Networks and ISDN Systems, vol. 30, pp. 1–7, 1998.

[10] R. Mihalcea and P. Tarau, "TextRank: Bringing Order into Texts," in Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain, 2004, pp. 404–411.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, pp. 1735–1780, 1997.

[12] M. Dascalu, D. S. McNamara, S. Trausan-Matu, and L. K. Allen, "Cohesion Network Analysis of CSCL Participation," Behavior Research Methods, vol. 50, pp. 604–619, 2018.

[13] M.-D. Dascalu, S. Ruseti, M. Dascalu, D. S. McNamara, and S. Trausan-Matu, "Multi-Document Cohesion Network Analysis: Visualizing Intratextual and Intertextual Links," in 21st Int. Conf. on Artificial Intelligence in Education (AIED 2020), Online, 2020.

[14] M. Dascalu, S. A. Crossley, D. S. McNamara, P. Dessus, and S. Trausan-Matu, "Please ReaderBench this Text: A Multi-Dimensional Textual Complexity Assessment Framework," in Tutoring and Intelligent Tutoring Systems, S. Craig, Ed., ed Hauppauge, NY, USA: Nova Science Publishers, Inc., 2018, pp. 251–271.

[15] J. Yen and P. Over, "Introduction to DUC-2001: an intrinsic evaluation of generic news text summarization systems," Document Understanding Conference 2001.

[16] W. Liggett and P. Over, "Introduction to DUC: an intrinsic evaluation of generic news text summarization systems," Document Understanding Conference 2002.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representation in Vector Space," in Workshop at ICLR, Scottsdale, AZ, 2013.

[18] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in International Conference on Machine Learning, Lille, France, 2015, pp. 957–966.

[19] A. Hagberg, D. Schult, P. Swart, D. Conway, L. Séguin-Charbonneau, C. Ellison, B. Edwards, and J. Torrents, "Networkx. High productivity software for complex networks", 2013. [Online]. Available: https://networkx.github.io/. [Accessed: July 20th 2020]

[20] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," Mathematical programming, vol. 14, pp. 265–294, 1978.

[21] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, 2011, pp. 510–520.

[22] H. Lin and J. Bilmes, "Multi-document summarization via budgeted maximization of submodular functions," in Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, California, USA, 2010, pp. 912–920.

[23] C.-Y. Lin, "Looking for a few good metrics: ROUGE and its evaluation," in Working Notes of NTCIR-4, Tokyo, 2004.

[24] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 2010, pp. 46–50.

[25] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force ‐ directed placement," Software: Practice and experience, vol. 21, pp. 1129–1164, 1991.

[26] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," Transactions of the Association for Computational Linguistics, vol. 5, pp. 135–146, 2017.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint vol. arXiv:1810.04805, 2018.