EAI Endorsed Transactions
on Creative Technologies

Research Article EAI.EU

From web to SMS: A text summarization of Wikipedia pages with character limitation

J.L.E.K Fendji[1],* and B.A.H. Aminatou[1]

[1]Computer Engineering, University Institute of Technology, The University of Ngaoundéré – Cameroon

Abstract

Wikipedia is one of the main sources of information on the Web. But the access to this content may be difficult especially when using a basic telephone without browsing capability and only a GSM network. The only means of text-based communication remains through SMS. Due to the limitation of the number of characters, a Wikipedia page cannot always be sent through SMS. This work raises the issue of text summarization with character limitation. To solve this issue, two extractive approaches have been combined: LSA and TextRank algorithms. Generated summaries have been evaluated using ROUGE metrics. Since ROUGE metrics do not consider character limitation, a new threshold named Threshold of Acceptability for Character-Oriented Summaries (TACOS) has been proposed to appreciate ROUGE metrics. The evaluation showed the relevance of the approach for pages of at most 2000 characters. The system has been tested using the SMS simulator of RapidSMS without a GSM gateway to simulate the deployment in a real environment. To the best of our knowledge, this is the first work tackling text summarization issue with character limitation.

Keywords: Character-limitation summarization, SMS, LSA, TextRank, ROUGE, TACOS, Wikipedia.

Received on 18 May 2020, accepted on 09 June 2020, published on 11 June 2020

Copyright © 2020 J.L.E.K Fendji *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/3.0/), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.11-6-2020.165277

1. Introduction

Wikipedia is an encyclopedia hosted by Wikimedia and it is considered as one of the main sources of information on the Web. Its freely available web-content can be used offline in remote regions that experience teacher shortage like rural areas in sub-Saharan Africa. Due to the lack of reliable power infrastructure, these areas are experiencing a high penetration rate of mobile devices. However, the bulk of those devices is composed of basic phones with no browsing capabilities[1]. The only way to send and receive text is through SMS, because of the good GSM network coverage [2]. But because of its length, a complete Wikipedia webpage cannot be always sent through SMS. The web page should therefore be summarized.

Summarizing involves condensing the most important information from a document (or multiple documents) to produce an abbreviated version [3]. An automatic summary is thus a text resulting from the reduction by a computer (or any computing system) of one (or several) text (s) that contains the same idea as the original ones.

Several systems have attempted to automatically summarize Wikipedia pages. Almost all of these employ an extractive approach, and try to extract relevant sentences from the content of the page [4]. Most of those systems produce summaries that are generally one-quarter the length of the original text. Although some of them propose word limitations, they are not dealing with the limit in terms of characters. If the generated summary contains more than 455 characters (limitation depending on the mobile carrier) it is going to be transformed into an MMS which requires adapted devices and includes a cost. Summarizing a webpage into an SMS is therefore a challenge since there is no relationship between the number of sentences or words and the number of characters.

The current work raises the issue of character-limitation text summarization. To solve this problem, a combination of

*Corresponding author. Email:lfendji@gmail.com two existing extractive summarization approaches has been used: LSA and TextRank algorithms. To the best of our knowledge, this is the first work tackling text summarization issue with character limitation.

The rest of the paper is organized as follows. Section 2 presents related works on text summarization using Wikipedia. The proposed approach for text summarization is presented in Section 3; followed by the evaluation of generated summaries in Section 4. Section 5 presents test results using the SMS simulator of RapidSMS. This paper ends with conclusions and possible directions for future developments.

2. Related works

2.1. Automatic text summarization approaches

Text summarization approaches can be classified as presented in Figure 1, adapted from [5]. There are two major approaches to creating a summary from a document: by extraction or by abstraction. Summarizing a text by extraction is extracting portions of the original text to form the summary [6]. Generally, the sentence is used as a basic unit and the challenge in this area remains the development of effective and easy techniques for reporting important passages in a text. In contrast, summarizing a text by abstraction involves reducing the length of this text by paraphrasing it while retaining the original idea [7]. These approaches use ontological information, extraction, and fusion of information as well as compression. Usually, any summarization approach that does not use extraction is considered as an abstractive approach.

When the summary is generated using only one source document, we talk about a single-document summary otherwise, it is a multi-document summary. In addition, the system may use knowledge-rich techniques to produce the summary. In this case, the system makes use of lexical resources such as VerbOcean [8] or external resources such as WordNet [9]. The summarization process may be subject to some constraints like containing the information requested by the query in case of query-focused summarization. In update summarization, the aim is to generate an updated version of the summary by identifying new pieces of information in the more recent articles. This extension supposes that the user has already read the previous versions of the summary. Finally, in guided summarization, the summarization approach is guided by a set of aspects that should be covered in a summary.

Recent surveys about automatic text summarization can be found in [10,11].

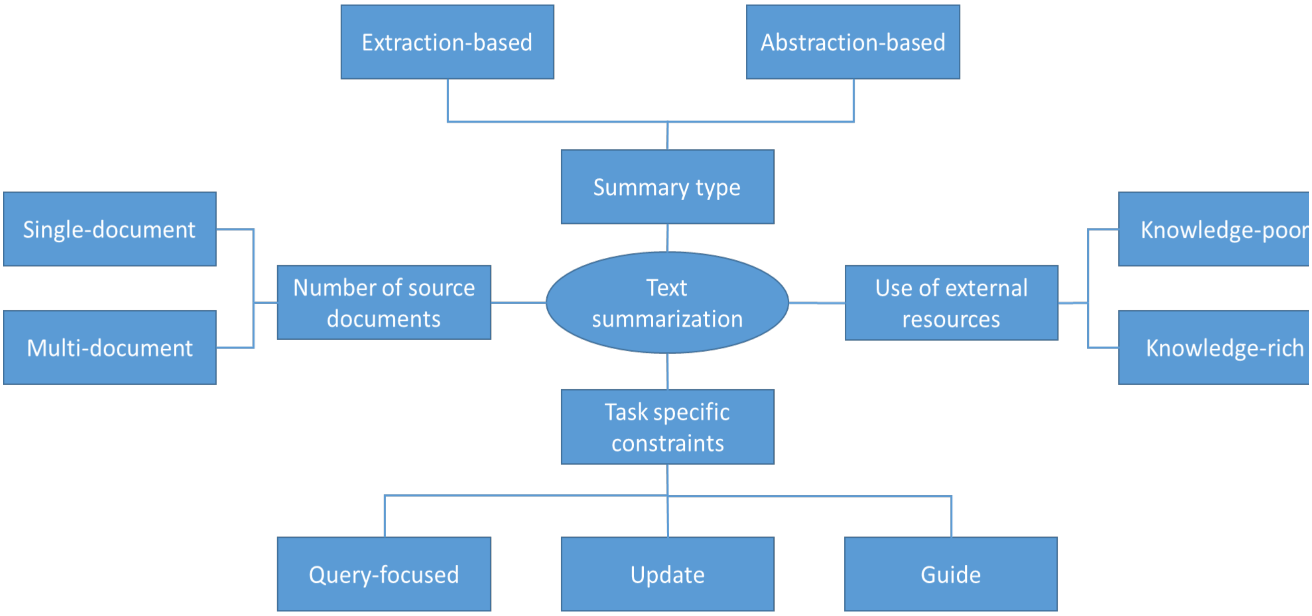


Figure 1. Classification of text summarization approaches (adapted from [5])

2.2. Wikipedia text summarization

Several works attempted to summarize Wikipedia pages. Hatipoglu and Omurca [12] have developed a mobile application for the automatic summarization of Wikipedia articles in Turkish. The system uses the extractive approach based on the structural characteristics of the Turkish language and on the semantic characteristics of the sentences. First, they score sentences from structural features such as the position of the sentence in the text, the number of words in the sentence, and the number of words from the title in the sentence. Then, a semantic analysis using the LSA (Latent Semantic Analysis) algorithm is performed before the extraction of the final summary.

Ajmera [13] built an extractive based Wikipedia summarizer in Python using the Django web framework. He extracted Wikipedia pages' content using urllib2[†], an extensible library for opening URLs. Sentence position and word similarity were used as features. The Term Frequency-Inverse Document Frequency (TF-IDF) is used to score words and the Google Search Result Count to score sections.

Hingu et al. [14] have implemented two methods for summarizing Wikipedia pages. Their methods, which extend traditional approaches, provide new features based on the citations present in the document to score sentences. In their methods, the frequency of words is adjusted according to the root form of the word. The words are stemmed with the objective to assign equal weight to words with the same root word. The length of the summary in terms of the percentage of the original text can be provided by the user.

Although those previous works are relevant, it remains that the generated summaries were not designed for limited devices such as basic phones with a limited number of characters.

One of the first works summarizing Wikipedia pages for basic phones dates back to Ramanathan et al. [15]. They designed a proxy-based approach for extractive summarization. Their approach, inspired by [16], uses the Wikipedia corpus to find the document topic. They indexed the whole Wikipedia corpus using the Lucene engine[‡]. But the generated summary is limited to 100 words which may exceed the maximum number of characters an SMS can contain.

3. Approach for text summarization with character-limitation

3.1. General idea

Our approach is based on sentence scoring methods and it is composed of 5 steps: text retrieval from Wikipedia; text pre-processing; sentence scoring; sentence selection; and summary generation. The approach is described in the flowchart in Figure 2.

3.1.1 Text retrieval

A Wikipedia page is generally divided into four parts: the top of the page, the stringcourse of the left, the body, and the page's footer. The body is the part that contains relevant information. It is subdivided into several parts including a title, an introductory summary, a table of contents, and the content itself. The first step is to retrieve the text from the requested Wikipedia page. For this, we made use of the Wikipedia tool. Given a title (name, word, group of words) the latter retrieves all the text of the corresponding Wikipedia page if it exists. Then, all sections are removed except "*content*". Afterward, the elimination of markers such as sections, subsections, and others are done. At the end, only relevant text about the page's topic remains. If the searched page does not exist, an error message is sent indicating that the requested article does not exist.

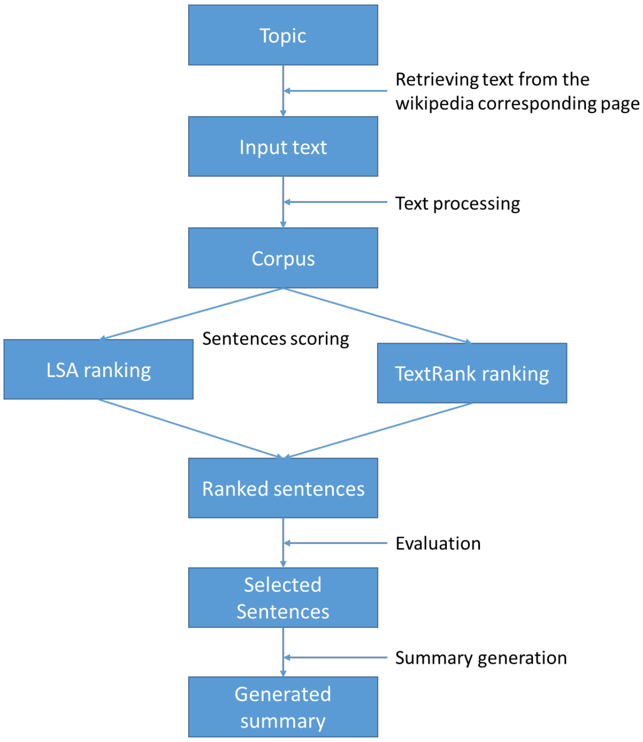


Figure 2. Proposed approach for summary generation

3.1.2 Text preprocessing

Retrieved text is pre-processed before summarizing through tokenization into sentences and then into words. Sentence tokenization is the process of splitting a paragraph into a list of sentences while word tokenization is the process of splitting a sentence into a list of words. Words are lemmatized, which means finding their basic form. Then unnecessary and special characters are removed, as well as stop words.

3.1.3 Sentence scoring

In general, sentence scoring encompasses three approaches: Word scoring that consists of assigning scores to the most important words; Sentence scoring that verifies sentence features such as its position in the document, or

[†] https://docs.python.org/2/library/urllib2.html

[‡] https://lucene.apache.org/

its similarity to the title; and Graph scoring that analyzes the relationship between sentences. Two approaches are used in this work to score sentences: LSA and TextRank.

LSA (Latent Semantic Analysis) algorithm

Inspired by the latent semantic indexing introduced by Dumais et al. [17] and improved later in [18], the LSA algorithm for text summarization was first developed by Gong et al in [19]. In this paper, we consider the more recent variant provided in [20]. LSA algorithm uses the co-occurrence of words to derive an implicit representation of the semantic of the text. The construction of the representation begins with the filling of a matrix A of size $n \times m$ with n words (one per line) and m sentences (one per column). The input of the matrix corresponds to the weight of the word i in the sentence j. The matrix A is generally sparse since sentences usually contain different words. If a sentence j does not contain a word i, the corresponding weight in matrix A is set to zero, otherwise, the weight is set to $TF \times IDF$. Afterward, Singular Value Decomposition techniques are applied to A, to obtain the product of three matrices: $A = U\Sigma V^T$. U is a $n \times m$ matrix, and each column of U can be interpreted as a subject meaning a specific combination of words of the entry, with the weight of each word in the subject given by a real number. The matrix $\sum$ is a diagonal matrix of size $m \times m$. The single entry in row i of the matrix corresponds to the weight of the "subject", which is the i^{th} column of U. The weights are sorted in reverse order, so that weight i is greater than or equal to weight j if $i < j$.

Subjects with low weight can be ignored by removing the last k columns from U, the last k rows and columns from $\sum$, and the last k rows from V^T . This procedure is called dimensionality reduction. Matrix V^T is a new representation of sentences with one sentence per line expressed in terms of subjects given in U. The matrix $D = \sum V^T$ combines the subject, the weight, and the representation of the sentence to indicate to what extent the sentence transmits the subject, with d_{ij} indicating the weight for the subject i in the sentence j. Then a sentence for each of the most important topics is selected. A reduction in dimensionality is performed, retaining as many subjects as the predefined number of sentences. The sentence with the highest weight for each selected subject is selected to form the summary. The LSA algorithm provided in [20] is given in Appendix 1.

TextRank algorithm

TextRank was introduced by Mihalcea and Tarau [21]. The internally uses the popular PageRank algorithm, which is used by Google for ranking web sites and pages and measures their importance. PageRank [22] is one of the most popular ranking algorithms and was designed as a method for web link analysis. PageRank considers the influence of incoming and outgoing links into one single model, in order to produce only one set of scores. The approach considers a directed graph represented as $G = (V, E)$, with V representing the set of pages (vertices) and E representing the set of links (edges).

Algorithm 1: WikiSMS core function

Input: *T*: a text
Output: *sum*: a summary

```
Begin
    T := Tokenize(T) ;
    T := Normalize(T) ;
    $S_0$ := TF_feature_matrix(T) ;
    $S_1$ := LSA($S_0$) ;
    $LSA_{value}$ := Get_six_Top_Value($S_1$) ;
    $S_2$ := TF_IDF_feature_matrix(T) ;
    $TextRank_{Value}$ := Get_six_Top_Value($S_2$) ;
    index = [] ;
    for i from 0 to 6 :
        index[i]:=max($LSA_{Value}$[i], $TextRank_{Value}$[i]);
    end
    len := i := 0 ;
    sum := ""
    while (len + length(summary[i]) < 456) :
        sum := sum.append(T(index[i])) ;
        len := len + length(T(index[i]));
        i := i+1;
    end
    return sum
End
```

In this paper, we consider the customized version of TextRank proposed in [23]. The TextRank algorithm uses sentences as the vertices of the algorithm based on the extractive summarization. Since they may exist multiple links between these vertices, in [23] the author modified the original PageRank algorithm to include a weight coefficient (say w_{ij}) between the edge connecting two vertices V_i and V_j such that w_{ij} indicates the strength of the connection between the vertices. The function for computing TextRank of vertices is given in (1).

$$TR(V_i) = (1-d) + d \times \sum_{V_j \in In(V_i)} \frac{w_{ji}TR(V_j)}{\sum_{V_k \in Out(V_j)} w_{jk}} \qquad (1)$$

The pseudocode of the algorithm in [23] is provided in Appendix 2.

3.1.4 Sentences selection

The six top sentences from both LSA and TextRank are temporarily selected. Their scores are compared and the sentences with the highest ranks are added to the summary if the number of characters is not exceeded.

3.1.5 Summary generation

The complete proposed algorithm to generated summaries is given in Algorithm 1.

4. Evaluation

4.1. Benchmark

For the evaluation, we used four datasets. The first contains articles with less than 500 characters. The second contains articles with a length between 500 and 1000 characters, the third with articles between 1000 and 2000 characters, and the last with articles containing over 2000 characters. The Wikipedia summary section was used as the reference summary.

4.2. Metrics

The ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric is a set of metrics used for evaluating automatic summarization [24]. The metrics compare an automatically produced summary (candidate summary) against a reference or a set of reference summaries (that are human produced). The measure is done by counting the number of matching words between a candidate summary and the reference summary. To test our system, we used two different variants of ROUGE: ROUGE-N and ROUGE-SU.

ROUGE-N (with N between 1 to 9): The summaries' texts are divided into a sequence of character of length N (N-gram). Let $N\text{-}gram_{Total}$ be the number of co-occurring in both the candidate summary and the set of reference summaries and $N\text{-}gram_{Ref}$ the number of co-occurring in the set of reference summaries, the ROUGE-N score is computed following (2).

$$ROUGE - N = \frac{\sum N - gram_{Total}}{\sum N - gram_{Ref}} \qquad (2)$$

ROUGE-SU (ROUGE Skip-bigram plus Unigram): it is an extension of ROUGE-S Skip-bigram) which measures the overlap of skip bigrams between a candidate summary and a set of reference summary. Skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps [25]. The ROUGE-SU in addition to skip bigrams counts also skip unigrams to do not assign a 0 score to a sentence just because it does not share a skip bigram when it instead has common unigrams.

As ROUGE metrics are word-oriented, to appreciate the quality of the summary, we defined a new metric named Threshold of Acceptability for Character-Oriented Summaries (TACOS) that will allow us to set the threshold of acceptability for character-limitation summaries. Let S_s be the system summary and S_r the reference summary, TACOS is defined by (3)

$$TACOS = \begin{cases} \dfrac{length(S_s)}{length(S_r)} & \text{If } length(S_r) < length(S_s) \\ 1 & \text{Otherwise} \end{cases} \qquad (3)$$

A system summary S_s is relatively good following a metric X if $X(S_s) \geq TACOS(S_s)$. TACOS is seen as the proportion of S_r in S_s. If S_s is greater than S_r, the proportion should be 1. Meaning that S_r should be part of S_s, to consider the summary as a relatively good one. On the contrary, if S_s is smaller than S_r, and if a metric provides a result at least equal to the ratio S_s/S_r, then the summary is considered as a relatively good one according to this metric.

4.3. Results of the evaluation

The results of the evaluation are presented in Table 1 to 4. From Table 1, related to the dataset of articles with a length of less than 500 characters, we can observe that all metrics give a result greater than the threshold TACOS. This means that the system provides relatively good results, irrespective of the metric used for the evaluation. From Table 2, related to the second dataset, we can observe that in the four first columns all evaluations give a result of one. This means that the entire reference summary is included in the system summary. In addition, in column nine (Mofu), even though the reference summary length is larger than the system summary length the evaluations give a result of one. This means that the system summary is a subset of the reference summary and all the bigrams in the reference summary are in the set of bigrams of the system summary. Here, we have seven relatively good results with all the metrics.

According to Table 3, related to the third dataset, we can see that we have also seven relatively good results for all the metrics. From Table 4, related to the last dataset, we can note that there are only three relatively good results for ROUGE-1 and two for both ROUGE-2 and ROUGE-SU4.

According to Figure 3, all curves are confounded because the purpose of the system is to generate summaries whose lengths are as close as possible to 455 characters. Since the pages are less than 500 characters, the system returns the entire article which is why the result of the evaluation is 1.

Figure 4 shows the results of the evaluation of the second dataset. We find that the smaller the size of the article, the less the result of the evaluation is good. This is due to the disproportion between the size of the summary generated and that of the reference summary.

From Figures 5 and 6, it appears that the summary generated is not only affected by the size of the reference summary but also by that of the article.

In light of these observations, we can state that the longer the reference summary, the worse the results for all ROUGE metrics. In other terms, the ROUGE metrics are not suitable for short generated summaries with long reference summaries. We must now think about how to evaluate character-based summaries of long texts.

Table 1. Wikipedia articles with length less than 500 characters

Title of the page	Lagdo	Yagoua	Guider	Ngaoundal	Kaele	Kolofata	Belabo	Diamare	Benoue	Mora
LA	44	117	270	267	344	380	389	418	463	483
LSS	0	0	265	244	368	358	333	376	368	483
LRS	40	118	268	168	347	43	157	223	196	483
TACOS	1	1	0.988	1	1	1	1	1	1	1
ROUGE-1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
ROUGE-2	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
ROUGE-SU4	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Table 2. Wikipedia articles with length between 500 and 1000 characters

Title of the page	Eseka	Obala	Sadou Hayatou	Tupuri people	Tibati	Madagali	Ebolowa	Kirdi	Mofu	Nanga eboko
LA	510	527	586	598	729	799	869	910	913	947
LSS	431	416	382	448	439	431	304	321	348	278
LRS	43	99	150	595	726	792	154	907	910	235
TACOS	1	1	1	0.752	0.604	0.544	1	0.353	0.382	1
ROUGE-1	**1**	**1**	**1**	**1**	0.56	**0.626**	**0.333**	**0.571**	**1**	0.68
ROUGE-2	**1**	**1**	**1**	**1**	0.5	**0.678**	**0.285**	**0.5**	**1**	0.625
ROUGE-SU4	**1**	**1**	**1**	**1**	0.38	**0.626**	**0.202**	**0.436**	**1**	0.574

Table 3. Wikipedia articles with a length between 1000 and 2000 characters

Title of the page	Sangmelima	Yaou Aissatou	Kousseri	Mayo rey	Peter Mafany Musonge	Lalla Malika Issoufou	Bafoussam	Luc Ayang	Mbo people (cameroon)	Chantal Biya
LA	1178	1233	1386	1490	1509	1613	1790	1890	1915	1951
LSS	441	354	293	371	373	446	185	364	453	434
LRS	281	1230	459	1323	148	118	1516	193	1320	100
TACOS	1	0.287	0.638	0.280	1	1	0.122	1	0.343	1
ROUGE-1	0.73	**0.689**	0.576	**0.785**	**1**	**1**	**0.52**	0.842	**1**	**1**
ROUGE-2	0.68	**0.607**	0.56	**0.703**	**1**	**1**	**0.458**	0.666	**1**	**1**
ROUGE-SU4	0.692	**0.607**	0.535	**0.697**	**1**	**1**	**0.425**	0.704	**1**	**1**

Table 4. Wikipedia articles with length over 2000 characters

Title of the page	Nadia Buari	Moussa Faki	Foumban	Chipset	African Union	History	Chemistry	Oxygen	Sudan	Singapour
LA	2161	2639	4618	5255	36502	38640	42192	43838	56928	67880
LSS	399	357	357	428	237	346	407	295	270	337
LRS	176	566	559	428	654	2204	2784	2321	2628	2790
TACOS	1	0.630	0.638	1	0.362	0.156	0.146	0.127	0.102	0.120
ROUGE-1	0.631	**1**	0.12	0.208	0.137	0.035	**0.185**	0.125	**0.655**	0.066
ROUGE-2	0.555	**1**	0	0.043	0.035	0	0	0	**0.571**	0
ROUGE-SU4	0.53	**1**	0.014	0.046	0.031	0.061	0.061	0.039	**0.55**	0.018

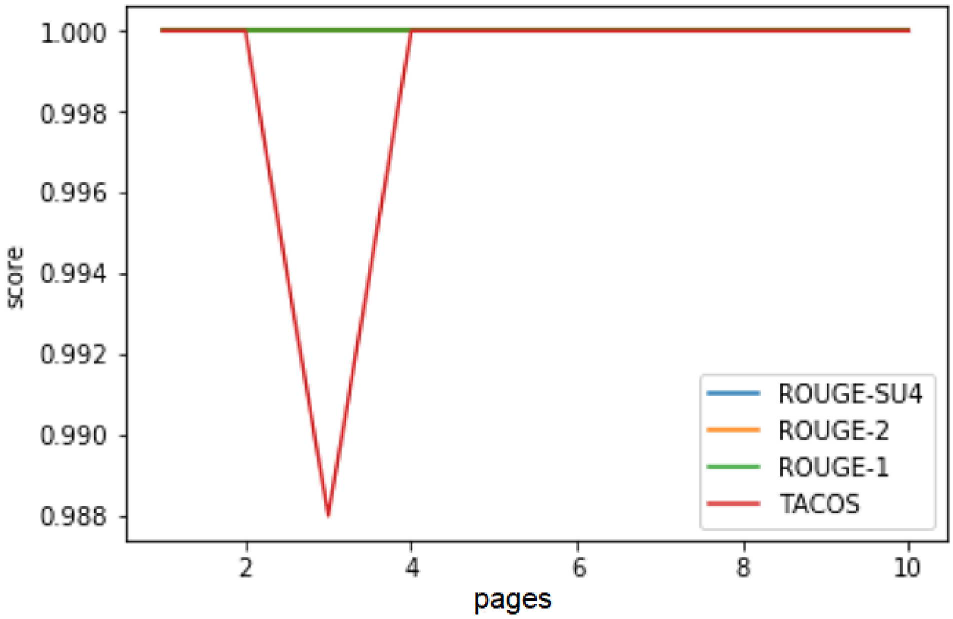


Figure 3. Approach of summary generation

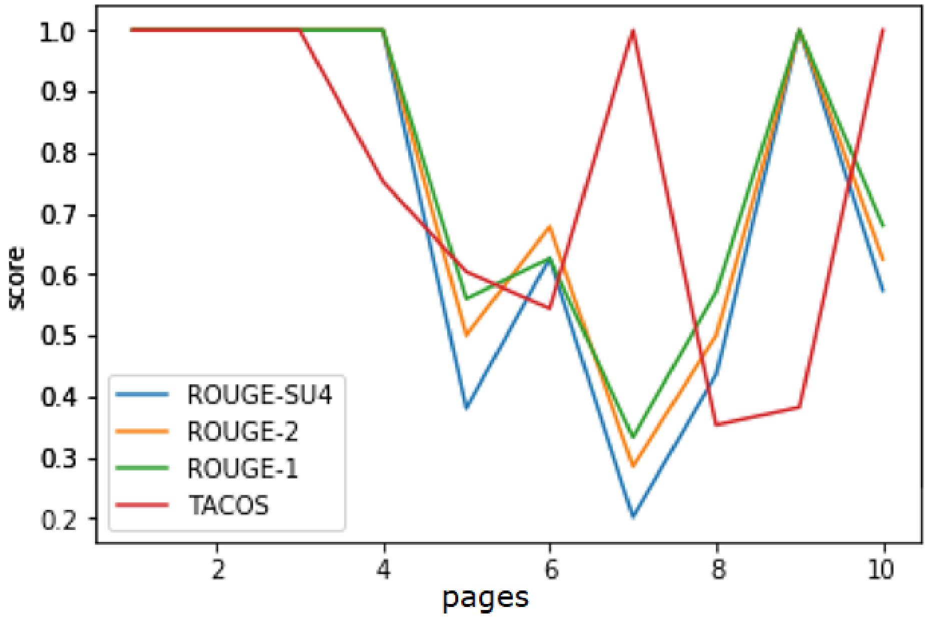


Figure 4. Approach of summary generation

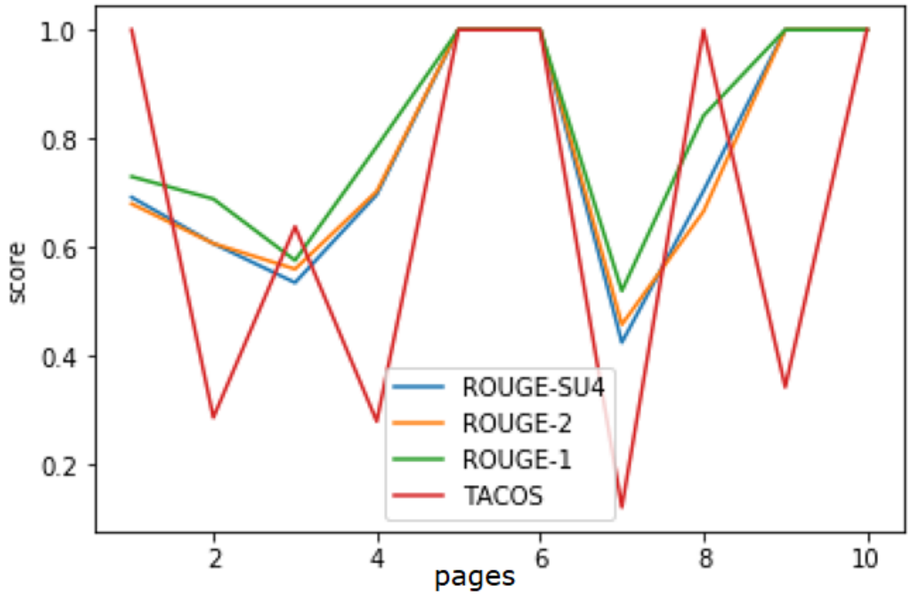


Figure 5. Approach of summary generation

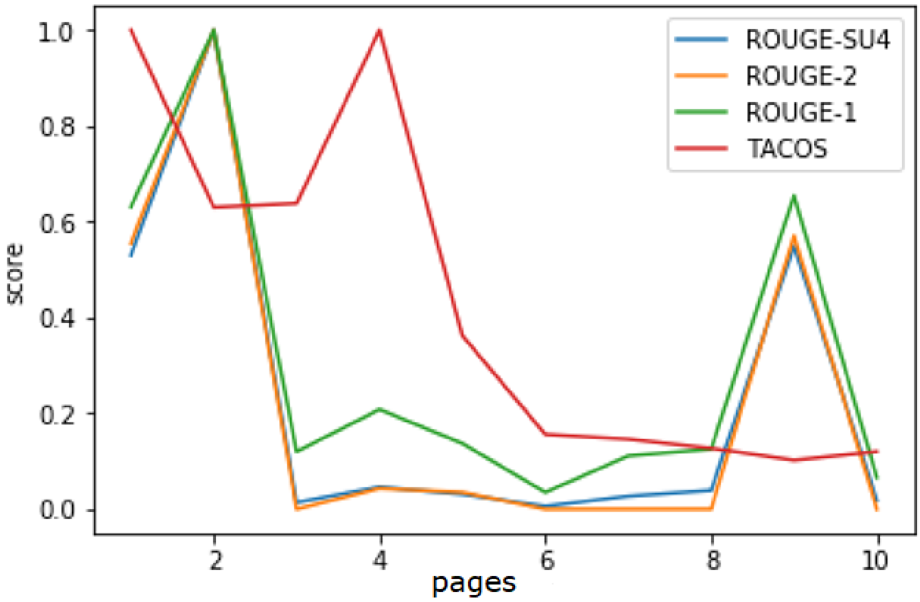


Figure 6. Approach of summary generation

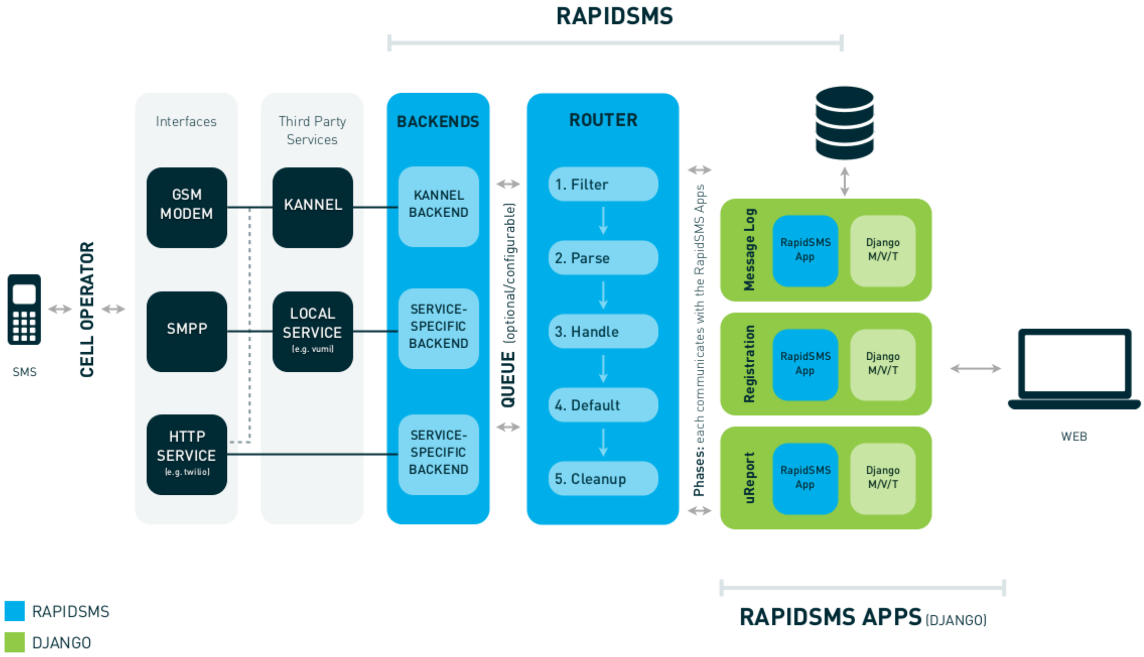
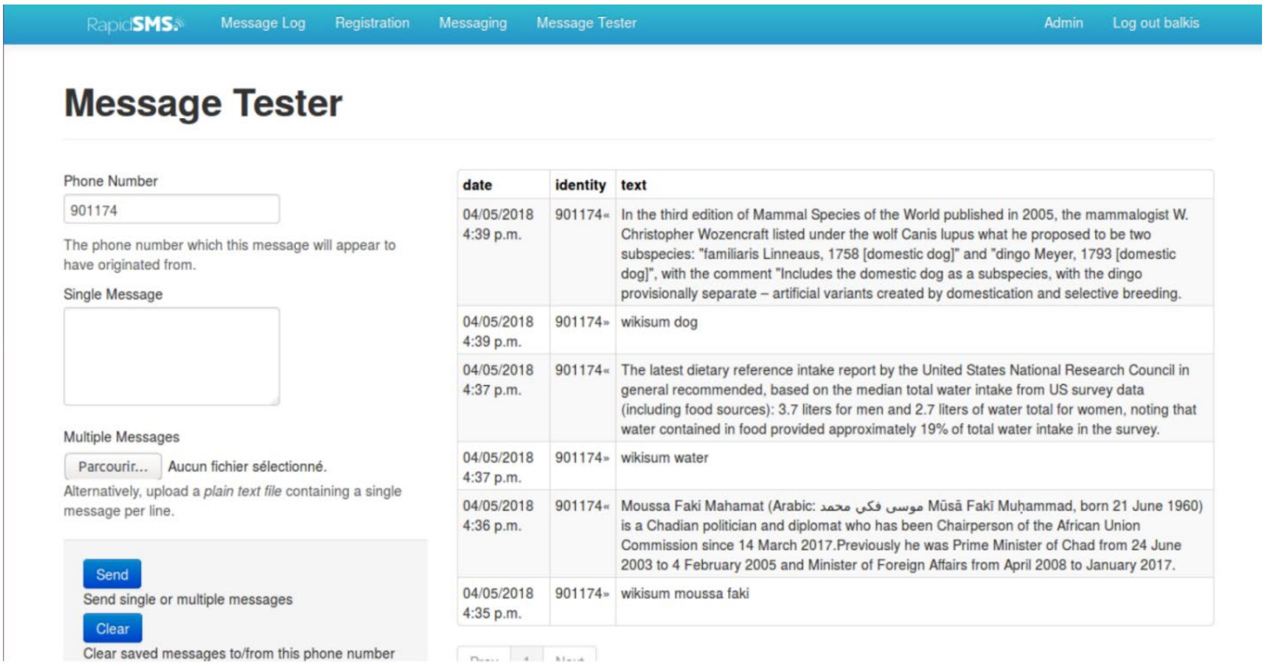


Figure 7. RapidSMS architecture[§]



Message Tester

Phone Number

901174

The phone number which this message will appear to have originated from.

Single Message

Multiple Messages

Parcourir... Aucun fichier sélectionné.

Alternatively, upload a *plain text file* containing a single message per line.

Send

Send single or multiple messages

Clear

Clear saved messages to/from this phone number

date	identity	text
04/05/2018 4:39 p.m.	901174«	In the third edition of Mammal Species of the World published in 2005, the mammalogist W. Christopher Wozencraft listed under the wolf Canis lupus what he proposed to be two subspecies: "familiaris Linneaus, 1758 [domestic dog]" and "dingo Meyer, 1793 [domestic dog]", with the comment "Includes the domestic dog as a subspecies, with the dingo provisionally separate – artificial variants created by domestication and selective breeding.
04/05/2018 4:39 p.m.	901174»	wikisum dog
04/05/2018 4:37 p.m.	901174«	The latest dietary reference intake report by the United States National Research Council in general recommended, based on the median total water intake from US survey data (including food sources): 3.7 liters for men and 2.7 liters of water total for women, noting that water contained in food provided approximately 19% of total water intake in the survey.
04/05/2018 4:37 p.m.	901174»	wikisum water
04/05/2018 4:36 p.m.	901174«	Moussa Faki Mahamat (Arabic: موسى فكي محمد Mūsā Fakī Muḥammad, born 21 June 1960) is a Chadian politician and diplomat who has been Chairperson of the African Union Commission since 14 March 2017.Previously he was Prime Minister of Chad from 24 June 2003 to 4 February 2005 and Minister of Foreign Affairs from April 2008 to January 2017.
04/05/2018 4:35 p.m.	901174»	wikisum moussa faki

Figure 8. Test results using RapidSMS

[§] https://rapidsms.readthedocs.io/en/develop/topics/architecture.html

5. Testing

5.1. RapidSMS Framework

RapidSMS is an open-source framework for application development using Short Message Service (SMS). It is the continuously increasing penetration rate of GSM technologies on the planet that seems to have motivated its development. Through its web interface, users can log in and access the system to view data as they arrive. They can also send SMS. RapidSMS is written in Python and integrates with Django, a Web development platform also written in Python. Still, according to the UNICEF website, RapidSMS is designed to work on small hardware configurations and requires at least a GSM modem and a SIM card. The complete architecture of RAPIDSMS is provided in Figure

5.2. Results of testing

To get a summary of a Wikipedia page, the user sends an SMS to a short number with the following syntax: ”wikisum Key_word”. Figure 8 presents some requests with their corresponding summaries.

Request: “*wikisum dog*”

Corresponding summary: “*In the third edition of Mammal Species of the World published in 2005, the mammologist W. Christopher Wozencraft listed under the wolf Canis lupus what he proposed to be two subspecies: "familiaris Linneaus, 1758 [domestic dog]" and "dingo Moyer. 1793 [domestic dog]", with the comment "Includes the domestic dog as a subspecies. with the dingo provisionally separate - artificial variants created by domestication and selective breeding.*” 439 characters including spaces.

Request: “*wikisum water*”

Corresponding summary: “*The latest dietary reference intake report by the United States National. Research Council in general recommended, based on the median total water intake from US survey data (Including food sources): 3.7 liters for men and 2.7 liters of water total for women, noting that water contained in food provided approximately 19% of total water intake in the survey.*” 359 characters including spaces.

6. Conclusions and perspectives

The main objective of this work was to summarize Wikipedia pages into a maximum of three SMS (455 characters). To do this, we proposed an approach combining two approaches namely: LSA and TextRank. Generated summaries have been evaluated using ROUGE metrics. Since those metrics have been developed for summarization approaches using words as units, their results cannot directly be interpreted for character-based summarization approaches. A new metric called Threshold of Acceptability for Character Oriented Summaries (TACOS) has therefore been introduced to have a relative appreciation of the quality of summaries. The complete system has been tested using RapidSMS, allowing a user to send a request and to receive the corresponding summary of the Wikipedia page on a mobile phone.

Although the fact that TACOS provides a threshold for the appreciation of the quality of the summary, it will be of interest to better elaborate this metric. An approach could be to find a relationship between characters in both system summary and reference summary. In addition, to better guide the summarization process, the system can also make use of a profiling process based on the user’s previous requests.

References

[1] Ebongue JLFK. Rethinking Network Connectivity in Rural Communities in Cameroon. ArXiv Prepr. ArXiv150504449, Lilongwe, Malaw: 2015.

[2] Fendji JLEK, Nlong JM. Rural wireless mesh network: A design methodology. ArXiv Prepr ArXiv150408213 2015.

[3] Mani I, Maybury MT. Advances in automatic text summarization (Vol. 293). Camb MA 1999.

[4] Liu PJ, Saleh M, Pot E, Goodrich B, Sepassi R, Kaiser L, et al. Generating wikipedia by summarizing long sequences. ArXiv Prepr ArXiv180110198 2018.

[5] Sizov G. Extraction-based automatic summarization: Theoretical and empirical investigation of summarization techniques. Master’s Thesis. Institutt for datateknikk og informasjonsvitenskap, 2010.

[6] Moratanch N, Chitrakala S. A survey on extractive text summarization. 2017 Int. Conf. Comput. Commun. Signal Process. ICCCSP, IEEE; 2017, p. 1–6.

[7] Moratanch N, Chitrakala S. A survey on abstractive text summarization. 2016 Int. Conf. Circuit Power Comput. Technol. ICCPCT, IEEE; 2016, p. 1–7.

[8] Chklovski T, Pantel P. Verbocean: Mining the web for fine-grained semantic verb relations. Proc. 2004 Conf. Empir. Methods Nat. Lang. Process., 2004, p. 33–40.

[9] Miller GA. WordNet: An electronic lexical database. MIT press; 1998.

[10] Gambhir M, Gupta V. Recent automatic text summarization techniques: a survey. Artif Intell Rev 2017;47:1–66.

[11] Allahyari M, Pouriyeh S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, et al. Text summarization techniques: a brief survey. ArXiv Prepr ArXiv170702268 2017.

[12] Hatipoglu A, Omurca Sİ. A Turkish Wikipedia Text Summarization System for Mobile Devices. IJ Inf Technol Comput Sci 2016;1:1–10.

[13] AJMERA S. AUTOMATIC TEXT SUMMARIZATION. PhD Thesis. INDIAN INSTITUTE OF TECHNOLOGY MANDI, 2015.

[14] Hingu D, Shah D, Udmale SS. Automatic text summarization of wikipedia articles. 2015 Int. Conf. Commun. Inf. Comput. Technol. ICCICT, IEEE; 2015, p. 1–4.

[15] Ramanathan K, Sankarasubramaniam Y, Mathur N, Gupta A. Document summarization using Wikipedia. Proc. First Int. Conf. Intell. Hum. Comput. Interact., Springer; 2009, p. 254–260.

[16] Gabrilovich E, Markovitch S. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. AAAI, vol. 6, 2006, p. 1301–1306.

[17] Dumais ST, Furnas GW, Landauer TK, Deerwester S, Harshman R. Using latent semantic analysis to improve access to textual information. Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 1988, p. 281–285.

[18] Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. J Am Soc Inf Sci 1990;41:391–407.

[19] Gong Y, Liu X. Generic text summarization using relevance measure and latent semantic analysis. Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., 2001, p. 19–25.

[20] Steinberger J, Jezek K. Using latent semantic analysis in text summarization and summary evaluation. Proc ISIM 2004;4:93–100.

[21] Mihalcea R, Tarau P. Textrank: Bringing order into text. Proc. 2004 Conf. Empir. Methods Nat. Lang. Process., 2004, p. 404–411.

[22] Page L, Brin S, Motwani R, Winograd T. The pagerank citation ranking: Bringing order to the web. Stanford InfoLab; 1999.

[23] Sarkar D. Text Analytics with Python 2016.

[24] Lin C-Y, Och FJ. Looking for a few good metrics: ROUGE and its evaluation. Ntcir Workshop, 2004.

[25] Gelbukh A. Computational linguistics and intelligent text processing. Springer; 2011.

Appendix 1: LSA Algorithm

1. Decompose the document $\boldsymbol{D}$ into individual sentences and use these sentences to form the candidate sentence set $\boldsymbol{S}$, and set $\boldsymbol{k} = \mathbf{1}$.
2. Construct the terms by sentences matrix $\boldsymbol{A}$ for the document $\boldsymbol{D}$.
3. Perform the SVD on $\boldsymbol{A}$ to obtain the singular value matrix $\sum$, and the right singular vector matrix $\boldsymbol{V^T}$. In the singular vector space, each sentence $\boldsymbol{j}$ is represented by the column vector $\boldsymbol{\phi_j} = [\boldsymbol{v_{j1}}, \boldsymbol{v_{j2}}, \dots, \boldsymbol{v_{jr}}]^{\boldsymbol{T}}$ of $\boldsymbol{V^T}$.
4. Select the $\boldsymbol{k^{th}}$ right singular vector from matrix $\boldsymbol{V^T}$.
5. Select the sentence which has the largest index value with the $\boldsymbol{k^{th}}$ right singular vector and include it in the summary.
6. If $\boldsymbol{k}$ reaches the predefined number, terminate the operation; otherwise, increment $\boldsymbol{k}$ by one, and go to step 4.

Appendix 2: TextRank Algorithm

1. Tokenize and extract sentences from the document to be summarized.
2. Decide on the number of sentences $\boldsymbol{k}$ that will part of the final summary.
3. Build document term feature matrix using weights like $\boldsymbol{TF} \times \boldsymbol{IDF}$ or Bag of Words.
4. Compute a document similarity matrix by multiplying the matrix with its transpose.
5. Use these documents as the vertices and the similarities between each pair of documents as the weight or score coefficient mentioned earlier and feed them to the PageRank algorithm.
6. Get the score for each sentence.
7. Rank the sentences based on score and return the top $\boldsymbol{k}$ sentences