An Assessment of Word Sequence Models for Extractive Text Summarization*

René Arnulfo García-Hernández, Yulia Ledeneva,
Alexander Gelbukh and Citlalih Gutiérrez

Autonomous University of the State of Mexico, Santiago Tianguistenco
Center for Computing Research, National Polytechnic Institute, Mexico
Toluca Institute of Technology, Mexico
renearnulfo@hotmail.com, yledeneva@yahoo.com, www.Gelbukh.com
Paper received on 04/08/08, accepted on 06/09/08.

Abstract. The main problem for generating an extractive text summary is to detect the most relevant information in the source document. For such purpose, recently some approaches have successfully employed the word sequence information from the self-text for detecting the candidate text fragments for composing the summary. In this paper, we employ the so-called *n-grams* and *maximal frequent word sequences* as features in a vector space model in order to determine the advantages and disadvantages for extractive text summarization.

1 Introduction

In the last two decades, we have experienced an exponential increase in the electronic text information available for being queried. The best example of the hugest and ever-increased collection of documents most frequently queried is Internet, with millions of web documents. According to a study by the Netcraft magazine [1] in June of 2007 Internet had 122 million of web sites, and according to Jaso Curtis [2] in February of 2007 Google had indexed around 25,000 million of web pages. Nowadays, it is common to use Google for retrieving a list web pages, but the user has to decide if a document is interesting only with the extracted text where the words of the request query appears. Therefore, it is necessary to download and read each document until the user finds satisfactory information. It was unnecessary and time-consuming routine. Thus, it is indispensable to develop automatic methods for detecting the most relevant content from a source document in order to show it as a summary. In addition, there are a number of scenarios where automatic construction of such summaries is useful. Other examples include automatic construction of summaries of news articles or email messages for sending them to mobile devices as SMS; summarization of information for government officials, businesspersons, researches, etc., and summarization of web pages to be shown on the screen of a mobile device, among many others. These examples show that it is desirable that text summarization approaches work more in language and dominion independent way.

* Work done under partial support of PROMEP, CONACyT, SNI, SIP, PIFI.

Advances in Intelligent and Information Technologies
Research in Computing Science 38, 2008, pp 253-26?

Automatic Text Summarization (ATS) is an active research area that deals with single- and multi-document summarization tasks. In single-document summarization, the summary of only one document is built, while in multi-document summarization the summary of a whole collection of documents (such as all today's news or all search results for a query) is built. While we believe that our ideas apply to either case, in this work we have experimented only with single-document summaries.

Summarization methods can be classified into abstractive and extractive summarization [3]. An abstractive summary is an arbitrary text that describes the contexts of the source document. Abstractive summarization process consists of "understanding" the original text and "re-telling" it in fewer words. Namely, an abstractive summarization method uses linguistic methods to examine and interpret the text and then to find new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original document. While this may seem the best way to construct a summary (and this is how human beings do it), in real-life setting immaturity of the corresponding linguistic technology for text analysis and generation currently renders such methods practically infeasible.

An extractive summary, in contrast, is composed with a selection of sentences (or phrases, paragraphs, etc.) from the original text, usually presented to the user in the same order—i.e., a copy of the source text with most sentences omitted. An extractive summarization method only decides, for each sentence, whether or not it will be included in the summary. The resulting summary reads rather awkward; however, simplicity of the underlying statistical techniques makes extractive summarization an attractive, robust, language-independent alternative to more "intelligent" abstractive methods. In this paper, we consider extractive summarization.

A typical extractive summarization method consists in several steps, at each of them different options can be chosen. We will assume that the units of selection are sentences (these could be, say, phrases or paragraphs). Thus, final goal of the extractive summarization process is sentence selection.

The main problem for generating an extractive automatic text summary is to detect the most relevant information in the source document. Although, some approaches claim being domain and language independent, they use some degree of language knowledge like lexical information [4], key-phrases [5] or golden samples for supervised learning approaches [6, 7, 8]. Furthermore, training on a specific domain tends to customize the extractions process to that domain, so the resulting classifier is not necessarily portable. In our opinion, these works present a high dominion and language dependence degree.

Recently, Villatoro *et al.* [6], Ledeneva *et al.* [9, 10] and García *et al.* [11] have successfully employed the word sequences from the self-text for detecting the candidate text fragments for composing the summary. Villatoro *et al.* [6] and García *et al.* [11] have extracted all the sequences of n words (*n-grams*) from the self-text as features of his model. Ledeneva *et al.* [9, 10] has proposed to extract all the frequent grams from the self-text, but they only considers those that are not contained (as subsequence) in other frequent grams (maximal frequent word sequences). In comparison with n-grams, the Maximal Frequent Sequences (*MFS*) are attractive for extractive text summarization since it is not necessary to define the gram size (n), it means, the length of each MFS is determined by the self-text. Moreover, the set of

all extracted MFSs is a compact representation all frequent word sequences, reducing in this way the dimensionality in a vector space model.

In this work, we evaluate the n-grams and maximal frequent sequences as domain- and language- independent models for automatic text summarization. For such purpose, we use the framework of Garcia *et al.* [11] approach where the summarization is made by sentence extraction using an unsupervised learning algorithm.

The paper is organized as follows. Section 2 summarizes the state of the art of text summarization methods. Section 3 describes the general scheme of the proposed approach. Section 4 presents the experimental settings followed for the experimentation. Section 5 concludes the paper.

2 Related Work

Ledeneva *et al.* [9] suggest a typical automatic extractive summarization approach composed by term selection, term weighting, sentence weighting and sentence selection steps. One of the ways to select the appropriate sentences is to assign some numerical measure of usefulness of a sentence for the summary and then select the best ones; the process of assigning these usefulness weights is called *sentence weighting*. One of the ways to estimate the usefulness of a sentence is to sum up usefulness weights of individual terms of which the sentence consists; the process of estimating the individual terms is called *term weighting*. For this, one should decide what the terms are: for example, they can be words; deciding what objects will count as terms is the task of *term selection*. Different extractive summarization methods can be characterized by how they perform these tasks.

Ideally, a text summarization system should "understand" (analyze) the text and express its main contents by generating the text of the summary. For example, Cristea *et al.* [12] perform sentence weighting according to their proximity to the central idea of the text, which is determined by analysis of the discourse structure.

However, the techniques that try to analyze the structure of the text involve too sophisticated and expensive linguistic processing. In contrast, most of the methods discussed in the literature nowadays represent the text and its sentences as a bag of simple features, using statistical processing without any attempts to "understand" the text.

Supervised learning methods consider sentence selection as a classification task: they train a classifier using a collection of documents supplied with existing summaries. As features of a sentence such methods can consider text units (in such a case we can speak of term selection) or other, non-lexical characteristics. Villatoro-Tello *et al.* [6] use as terms *n*-grams found in the text.

On the contrary, the *n*-grams in the work of Garcia *et al.* [11] are used as features of a sentence in an unsupervised learning method. This method tries to find the different ideas (sentences) presented in the text by clustering similar sentences. Then, from each cluster is selected the most representative sentence for composing the summary.

However, the majority of current methods are purely heuristic: they do not use any learning but directly state the procedure used for term selection, term weighting,

and/or sentence weighting (given that sentence selection in most cases consists in selecting the best-weighted sentences).

A very old and very simple sentence weighting heuristic does not involve any terms at all: it assigns highest weight to the first sentences of the text. Texts of some genres—such as news reports or scientific papers—are specifically designed for this heuristic: e.g., any scientific paper contains a ready summary at the beginning. This gives a baseline [13] that proves to be very hard to beat on such texts. It is worth noting that in Document Understanding Conference (DUC) competitions [13] only five systems performed above this baseline, which does not demerit the other systems because this baseline is genre-specific. Though the method proposed in this paper very slightly outperforms this baseline, such a comparison is unfair.

From the works devoted to term-based methods, most concentrate on term weighting. Xu *et al.* [14] derives relevance of a term from an ontology constructed with formal concept analysis. Song *et al.* [4] basically weight a word basing on the number of lexical connections, such as semantic associations expressed in a thesaurus, that the word has with its neighboring words; along with this, more frequent words are weighted higher. Mihalcea [15] presents a similar idea in the form of a neat, clear graph-based formalism: the words that have closer relationships with a greater number of "important" words become more important themselves, the importance being determined in a recursive way similar to the PageRank algorithm used by Google to weight web pages.

The latter idea can be applied directly to sentence weighting without term weighting: a sentence is important if it is related to many important sentences, where relatedness can be understood as, say, overlap of the lexical contents of the sentences [15].

Recently, a novel approach quite different from other methods was presented by Ledeneva *et al.* [9]. In this work, the sentences are weighted by using the terms derived from the maximal frequent word sequences. Then, the best sentence is combined with the baseline sentences for composing the summary. This approach is ranked, according to ROGUE evaluation system, in third place.

The methods presented in [15, 16, 17] and [9] are those that currently give the best results and with which we compare our suggested method.

While in the experiments reported in the papers discussed above were based on words as terms, this is not the only possible option. Liu *et al.* [18] uses pairs of syntactically connected words (basic elements) as atomic features (terms). Such pairs (which can be thought as arcs in the syntactic dependency tree of the sentence) have been shown to be more precise semantic units than words [19, 20]. However, while we believe that trying text units larger than a word is a good idea, extracting the basic elements from the text requires dependency syntactic parsing, which is language-dependent. Simpler statistical methods, as the use of *n*-grams as terms in [6], may prove to be more robust and language-independent.

3 Proposed Methodology

Commonly, an extractive summarization approach achieves the term selection, term weighting, sentence weighting and sentence selection steps. However, the strategy of sentence selection step is reduced to simply to take the weightiest sentences. Although, this strategy could work well for the first ranked sentence, the strategy could convey that similar sentences to the first one tend to be ranked after the first one; producing redundant sentences for the summary. This problem affected negatively the recall measure of the work of Ledeneva [9]. For avoiding this problem, we employ the unsupervised learning algorithm of Garcia *et al.* [11] for automatically detecting the groups of similar sentences from which is selected the most representative sentence; reducing in this way the redundancy in the summary. In this section, we describe the general steps of the proposed approach.

3.1 Term Selection

An n-gram is a sequence of n words. We say that an n-gram occurs in a text if these words appear in the text in the same order immediately one after another. For example, a 5-gram (n-gram of length 5) *words appear in the text* occurs once in the previous sentence, while *appear immediately after another* does not (these words do not appear on adjusting positions), neither does *the text appear in* (order is different).

The definition of n-gram depends on what one considers words. For example, one can consider capitalized (*Mr. Bush*) and non-capitalized (*a bush*) words as the same word or as different words; one can consider words with the same morphological stem (*live*, *lived*, *living*), the same root (*educate*, *education*), or the same meaning (*facilitate*, *assist*) as the same word; one can omit the stop-words (*the*, *in*) when counting word positions, etc. Say, one can consider that in our example sentence above there occur the n-grams *we say* (capitalization ignored), *word appear* (plural ignored), *appear text* (*in the* ignored). This can affect counting the n-grams: if one considers *occur* and *appear* as equivalent and ignores the stop-words, then in our example sentence the bigram *appear text* occurs twice.

A maximal frequent word sequences is a gram (the size is not restricted) that it is frequent in text, but it is not contained (as subsequence) in other frequent gram. In this case, for considering a gram as frequent it is necessary to establish a frequency threshold.

3.2 Term Weighting

Boolean Weighting (BOOL): It is the easiest way to weight a term. It models the presence or absence of a term in the document, defined as:

$$w_i(t_j) = \begin{cases} 1, & \text{if the term } t_j \text{ appears in document } i \\ 0 & \text{other case} \end{cases}$$

Term Frequency (TF) was proposed in [21]. This weighting takes into account that a term that occurs in a document can better reflect the contents of document than a term that occurs less frequent. Therefore, the weighting TF assigns a greater relevance to terms with greater frequency and consists in evaluating the number of times the term appears in the document.

$w_i(t_j) = f_{ij}$, where f_{ij} is the frequency of the term j in document i.

Inverse Document Frequency (IDF) was proposed by Salton and Buckley [22] for improving information retrieval systems (IR). The problem of TF weighting in IR is that, when a term appears in almost all the documents in the collection; this term is useless for discriminating relevant documents. For example, the stop-word *and* could have a high TF, but it is useless for discriminating the relevant documents since tends to appear in most of the documents. IDF is defined as:

$w_i(t_j) = \log\left(\frac{N}{n_j}\right)$, where N is the number of documents in the collection and n_j is the number of documents where the term j appears.

TF-IDF. The problem of IDF weighting in IR is that it is not possible distinguish between two documents with the same vocabulary (list of different words), even thought if the term is more frequent in a document. TF-IDF weighting gives more relevance to the terms that are less frequent in the collection but more frequent into the document.

$$w_i(t_j) = f_{ij} \times \log\left(\frac{N}{n_j}\right)$$

Note that in this paper we propose to use these term weights for single document summarization. Therefore, for applying these term weights we can consider the document as a collection of sentences instead of a collection of documents.

3.3 Sentence Selection Using an Unsupervised Learning Algorithm

An unsupervised learning algorithm form groups of objects in order to achieve, in the one hand, the greatest possible similarity between objects of a group, in other hand, the greatest possible dissimilarity between objects of different groups.

In this step, we use the approach of Garcia *et al.* [11] where they proposed an unsupervised algorithm for discovering the groups of sentences with similar meaning. Then, the approach can select the most representative sentence from each group

in order to compose the summary. In particular, he proposed to use the well-known K-means algorithm, which assumes that the number of clusters is previously known. Sometimes this characteristic is a disadvantage in K-means, however in this case is an advantage because lets to control the number of groups to create; which lets, at the same time, estimate the number of words in the final summary. For example, if the average of words per sentence is 20 and a user desires a 100-word summary then K-means must create 5 clusters, obviously this is only an estimation of the number of words in the final summary. K-means represents each sentence in a vector space model. So, each document is represented as a vector of features, where the features correspond to the different terms in the document, in this case n-grams.

K-means is based on centroids, which are points in the vector space model calculated as the mean of the objects in a group. K-means iteration consists in to assign each object to the closest centroid and then the new centroids are recaculated again. The algorithm finishes when the centroids do not change. In the beginning, the K-means algorithm need seeds as the initial centroids for each group. Thus, the successful of K-means depends on selecting good initial seeds. In Garcia *et al.* approach [11], first sentences are considered as initial seeds, since it is known that the Baseline sentences are good candidate sentences for composing the summary. In our case, the initial seeds are calculated in random way in order of being more independent from the baseline heuristic.

For measuring the similarity between two sentences the Euclidean distance is used, defined as:

$\text{Distance}\quad (X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$, where X and Y are sentences expressed as vectors with n features.

4 Experimental results

We have conducted several experiments to compare the n-gram and MFS models.

Algorithm: In each experiment, we followed the standard sequence of steps:

- *Preprocessing*: First, eliminate stop-words, and then apply stemming;
- *Term selection*: decide which size of n-grams as features are to be used to describe the sentences. The frequency threshold used for MFS model was 2.
- *Term weighting*: decide how the importance of each feature is to be calculated, it can be BOOL, TF, IDF or TFIDF;
- *Sentence clustering*: decide the initial seeds for the k-means algorithm, in this case Baseline sentences;
- *Sentence selection*: after K-means finishes, select the closest sentence to each centroid for composing the summary;
- The specific settings for each step varied between the experiments and are explained below for each experiment.

Test data set. We used the standard DUC 2002 collection provided [13]. In particular, we used the data set of 567 news articles of different length and with different topics. Each document in the DUC collection is supplied with a set of human-generated summaries provided by two different experts.[3] While each expert was asked to generate summaries of different length, we used only the 100-word variants.

Evaluation procedure. We used the ROUGE evaluation toolkit [23] which was found to highly correlate with human judgments [24]. It compares the summaries generated by the program with the human-generated (gold standard) summaries. For comparison, it uses *n*-gram statistics. Our evaluation was done using *n*-gram (1, 1) setting of ROUGE, which was found to have the highest correlation with human judgments, namely, at a confidence level of 95%. ROUGE calculates Precision, Recall, and F-measure values. We consider F-measure because it represents a balance (not an average) of recall and precision results.

Table 1 shows the results obtained with ROUGE for different word sequence models and different term weights. In particular, tables 1, 2 and 3 shows Recall, Precision and F-measure results, respectively. As Recall, Precision and F-measure tables show, the **best recall result** is obtained with 3-grams and IDF; the **second place** is obtained by MFS and BOOL; and the third place is obtained by 1-gram and BOOL. The ***worst result*** was obtained with 4-grams and TFIDF.

Table 1. Recall results for different word sequence models for DUC 2002 collection.

Model	BOOL	TF	IDF	TFIDF
1-gram	**0.44567**	0.44205	0.44132	0.44196
2-gram	0.44005	0.43990	0.44092	0.44128
3-gram	0.44241	0.43681	**0.44833**	0.44289
4-gram	0.43691	0.43663	0.44198	***0.43303***
5-gram	0.43794	0.44047	0.44511	0.43763
MFS	**0.44698**	0.44123	0.44127	0.44226

Table 2. Precision results for different word sequence models for DUC 2002 collection.

Model	BOOL	TF	IDF	TFIDF
1-gram	**0.44207**	0.43895	0.43793	0.43823
2-gram	0.43646	0.43684	0.43789	0.43801
3-gram	0.43891	0.43382	**0.44482**	0.43992
4-gram	0.43355	0.43383	0.43881	***0.43005***
5-gram	0.43467	0.43761	0.44201	0.43470
MFS	**0.44371**	0.43859	0.43801	0.43912

It is possible to observe in F-measure results that the quality of the results is not increased when the grams size do it. Hence, it is not clear which gram size must be chosen, for this reason, we have calculated F-measure average for each model and for each weighting. Although the MFS model is in second place for F-measure re-

[3] While the experts were supposed to provide extractive summaries, we observed that the summaries provided in the collection were not strictly extractive: the experts considerably changed the sentences as compared with the original text.

sults, it is interesting to observe that the MFS model obtains the *best F-measure average* Moreover, the *best weighting average* is got by IDF.

Table 3. F-measure results for different word sequence models for DUC 2002 collection.

Model	BOOL	TF	IDF	TFIDF	Average
1-gram	**0.44374**	0.44037	0.43949	0.43996	0.44089
2-gram	0.43814	0.43824	0.43927	0.43953	0.43879
3-gram	0.44054	0.43519	**0.44640**	0.44127	0.44086
4-gram	0.43511	0.43510	0.44027	***0.43142***	0.43547
5-gram	0.43617	0.43892	0.44340	0.43604	0.43864
MFS	**0.44520**	0.43979	0.43953	0.44056	*0.44127*
Average	0.43982	0.43793	*0.44140*	0.43813	

5 Conclusions and future work

In this work, we compare two successful word sequence models, the *n*-gram and MFS, using an unsupervised learning algorithm for automatic extractive text summarization. In particular, the K-means algorithm was used for creating groups of similar sentences. After that, for each group of sentences, the most representative sentence was selected for composing the summary. In this comparison, the 3-gram model obtained the best F-measure results, followed by MFS and then by 1-gram. However, the MFS model obtained the best F-measure average. It shows that the MFS model obtains, in general, competitive F-measure results. In the future, it is necessary to find *a priori* way of determining the best gram size for text summarization—what is not clear how to do. Another option is to combine the MFS and *n*-grams in a super model.

References

1. NetCraft. Web Server Survey, England. June 2007. http://news.netcraft.com/
2. Jason Curtis. "Marketing the Most of Google", Electronic Resources Librarian, United Kindom, 2007.
3. Lin, C.Y. and Hovy, E. Automated Text Summarization in SUMMARIST. In Proc. of ACL Workshop on Intelligent, Scalable Text Summarization, Madrid, Spain, 1997.
4. Song, Y., *et al.* A Term Weighting Method based on Lexical Chain for Automatic Summarization. CICLing 2004, LNCS, vol. 3878, Springer-Verlag 2004.
5. HaCohen-Kerner, Y., Zuriel, G., Asaf, M. Automatic Extraction and Learning of Keyphrases from Scientific Articles. CICLing 2005, LNCS, vol. 3878, pp. 645–657, Springer-Verlag 2005.
6. Villatoro-Tello, E., Villaseñor-Pineda, L., Montes-y-Gómez, M. Using Word Sequences for Text Summarization. TSD, LNAI Springer 2006.
7. Chuang T.W., Yang J. Text Summarization by Sentence Segment Extraction Using Machine Learning Algorithms. Proc. of the ACL-04 Workshop. Barcelona, España, 2004.

8. Neto L. Freitas A. A.,and Kaestner C. A. A. Automatic Text Summarization using a Machine learning Approach. Proceedings of the ACL-04 Workshop. Barcelona, España, 2004.
9. Ledeneva Yulia, Gelbukh Alexander, René Arnulfo García-Hernández. Terms Derived from Frequent Sequences for Extractive Text Summarization. CICLing'2008. LNCS vol. 4919 Springer-Verlag, 2008. pp 593-604.
10. Ledeneva Yulia, Gelbukh Alexander, García H. René. Keeping Maximal Frequent Sequences Facilitates Extractive Summarization, CORE 2008, Research in Computing Science Vol. 34, México, 2008.
11. René Arnulfo García-Hernández, Romyna Montiel, Yulia Ledeneva, Eréndira Rendón, Alexander Gelbukh, Rafael Cruz. Text Summarization by Sentence Extraction Using Unsupervised Learning, 7th Mexican International Conference on Artificial Intelligence (MICAI08), Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 5317. pp.133-143.
12. Cristea D., *et al.* Summarization through Discourse Structure. CICLing 2005, LNCS, vol. 3878, Springer-Verlag 2005.
13. DUC. Document Understanding Conference 2002; www-nlpir.nist.gov/projects/duc.
14. Xu, W., Li, W., *et al.* Deriving Event Relevance from the Ontology Constructed with Formal Concept Analysis. CICLing 2006, LNCS, vol. 3878, Springer-Verlag 2006.
15. Mihalcea, R. Random Walks on Text Structures. CICLing 2006, LNCS, vol. 3878, pp. 249–262, Springer-Verlag 2006.
16. Mihalcea, R., Tarau, P. TextRank: Bringing Order into Texts, in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Barcelona, Spain, 2004.
17. Hassan, S., Mihalcea, R., Banea, C. Random-Walk Term Weighting for Improved Text Classification, Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2007), Irvine, CA, 2007.
18. Liu, D., He, Y., Ji, D., Hua, J. Multi-Document Summarization Based on BE-Vector Clustering. CICLing 2006, LNCS, vol. 3878, Springer-Verlag 2006.
19. Bolshakov, I.A. Getting One's First Million…Collocations. CICLing-2004, Seoul, Korea, LNCS 2945, p. 229-242, Springer-Verlag 2004.
20. Grigori Sidorov, Alexander Gelbukh. Automatic Detection of Semantically Primitive Words Using Their Reachability in an Explanatory Dictionary. NLPKE 2001 at Proc. International IEEE SMC-2001 Conference: Systems, Man, And Cybernetics, pp. 1683–1687.
21. Luhn H.P A Statical Approach to Mechanical Encoding and Searching of Literary Information. IBM Journal of Research and Development, pp. 309-317, 1975.
22. Salton G., Buckley, C. Term-Weighting Approaches in Automatic Text Retrieval. Information Processing & Management, Vol 24. pp. 513–523, 1988.
23. Lin C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of Workshop on Text Summarization of ACL, Spain, 2004.
24. Lin C.Y., Hovy E. Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In Proceedings of HLT-NAACL, Canada, 2003.