Single Extractive Text Summarization Based on a Genetic Algorithm

René Arnulfo García-Hernández and Yulia Ledeneva

Autonomous University of the State of Mexico,
Santiago Tianguistenco, México
renearnulfo@hotmail.com, yledeneva@yahoo.com

Abstract. Extractive text summarization consists in selecting the most important units (normally sentences) from the original text, but it must be done as closer as humans do. Several interesting automatic approaches are proposed for this task, but some of them are focused on getting a better result rather than giving some assumptions about what humans use when producing a summary. In this research, not only the competitive results are obtained but also some assumptions are given about what humans tried to represent in a summary. To reach this objective a genetic algorithm is proposed with special emphasis on the fitness function which permits to contribute with some conclusions.

1 Introduction

According to Lee [1], the amount of information in Internet continues growing, but much of this information is redundant. Therefore, we need new technologies to efficiently process information. The automatic generation of document summaries is a key technology to overcome this obstacle. Given this, it is essential to develop automated methods that extract the most relevant information from a text, researched by Automatic Text Summarization (ATS) area [2], [3], [4], [5]. 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 both cases, in this work we have experimented only with single-document summaries.

Summarization methods can be classified into abstractive and extractive summarization. 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.

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2013, LNCS 7914, pp. 374–383, 2013.

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 single extractive summarization.

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 [5], key-phrases or a golden sample for supervised learning approaches [6] [7]. Furthermore, training on a specific domain tends to customize the extraction process to that domain, so the resulting classifier is not necessarily portable. For that reason, these works present a high dominion and language dependence degree.

A typical extractive summarization method [8] [9] consists in 5 steps: preprocessing, term selection, term weighting, sentence weighting and sentence selection; 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.

Usually, in the preprocessing step the document is analyzed for removing words without meaning (stop words) and for getting a canonical representation of each word by applying a stemming algorithm in order to find relations between significant words. Moreover, some methods use more complex resources such as *Part-of-speech tagging*, lemmatization (instead of stemming), key words, key phrases, etc.

Most of the language-independent methods employing the n-gram as the unit in term selection step which is composed by all the sequences of n words of the document. Recently, the Maximal Frequent Sequence (MFS) model has been proposed as text model [8] [10] [11] [12] which tried to select only the important terms according to the frequency without the need of determine n. A MFS text model can be defined in terms of grams as all the frequent grams (of any size) that are not subsequence of other frequent grams. For considering that a gram is frequent it must be repeated at least a threshold times in the text, when the threshold it not specified it is assumed that is taken the lowest possible, *i.e.* two.

In third step is given an importance to the selected terms, for example the presence or absence of a term can be used as Boolean weighting, but in this weighting it is not possible to know which term is more important. An alternative is to use the frequency of the term as TF weighting, but a very frequent term is not always important since could be a stop word or a term that it is repeated in most of the sentences; therefore it is important for the entire document and not for a single sentence. This problem can be solving if the inverse document frequency is used as IDF weighting, in this case the frequency of a term is divided by the number of sentences where the term is presented; it means a frequent term is more important if it appears in a single sentence instead of all the sentences.

Normally, for composing the summary the sentences are selected according to its relevance in the sentence selection step. This way of sentence selection tends to produce redundant summaries. In this research, special attention is given to sentence selection step since this process must consider all the relevant information getting in previous steps. In this paper, a genetic algorithm is proposed for optimizing the sentences selection step based on the frequency of the words (1-grams).

2 Related Works

In different ways, several approaches have employed a genetic algorithm for the ATS task based on attribute selection [13] [14]. However, these kinds of approaches have in common that represent each sentence as a set of attributes extracted from the original text. The following features are gotten using only static and structural information from the original text, without linguistic knowledge.

Similarity to title [13] [14] [15] is a measure that arise sentences that have common words with the title. This is determined by counting the number of matches between the words in a sentence and the words in the title [13] or it is calculated as the cosine similarity [14].

Similarity to keywords [14] is an analogous measure to similarity to title.

Sentence length feature [13] [14] [15] gives more preference for longer sentences, under the idea that short sentences could bring, for example: datelines, numbers or author names. This measure is normalized to the longest sentence in the document.

Term weight feature [13] is based on the frequency of the terms presented in the sentence. The score of a sentence can be calculated as the sum of the weights of the terms in the sentence. A term will be more important if it appears frequently into the document but simultaneously it does not appear in others sentences.

Sentences position feature [13, 14, 16] relies in the baseline heuristic [17] that establishes the first sentences of a text can be considered as a good summary. Document collections created specifically for ATS systems has proved that it is a hard line to overcome. Normally, this feature assigns the inverse order number as the importance for the sentence, for example, if there are 10 sentences in the document, the first sentence has a score of 10/10, and the second one has a score of 9/10 and so on.

Sentence similarity feature [13] measures the similarity that has a sentence against the rest of sentences in order to avoid getting untypical sentences. Therefore, a sentence with high score is more probable to appear in the summary. One option to get the similarity between two sentences is to use the cosine similarity measure.

Numerical feature [13] is based on the idea that in the sentences where numerical data appears are more relevant. For measure this feature is calculated as the ratio of the number of the numerical data in the sentences over the sentence length.

It is possible to extract other dependent-linguistic features based on *Proper Noun*, [13] [14] *Thematic Word* [13], *Anaphors* [14], *Discourse Markers* [14].

Some approaches that using a genetic algorithms for the ATS task [13] [14] are based on attribute sentence selection in a supervised classification scheme [13], thus, for these approaches is needed to account with a previously set of golden summaries for training. Other approaches [15] use the GA in an unsupervised classification scheme, where the fitness function is formulated with some of the above features for evaluating the summarized sentences.

3 Proposed Genetic Algorithm

Genetic Algorithm (GA) is the most traditional evolutionary technique that has proved to be an alternative solution for an optimization problem. In the first step, the GA proposes a population of random solutions (*initial population step*) that are evaluated according to the objective function to optimize (*fitness function step*). In this sense, a solution for one problem is not absolute, it means, there is set of possible solutions where some are better than others. Considering mostly the best solutions (*parents selection step*), the GA proposes a new population mixing (*crossover step*) some parts from a canonical codification (*Chromosome encoding step*) of these good solutions in order to get better solutions (*evolution principle*). Eventually, the way of mixing some parts from the canonical codification could produce repeated solutions. Therefore, the GA applies a small variation (*mutation step*) to the canonical codification in the new population in order to explore new solutions. The new population is evaluated again and the process is repeated until a satisfactory solution is reached or until some arbitrary stop-criteria is reached (*stop condition*).

3.1 Proposed Genetic Operators

Preprocessing. Before the original text could be used for the GA, it is needed to adapt the entry of the original text to the format of the GA. In this step, the original text is separated in sentences. Also, the text is preprocessed with the well-known Porter Stemmer [18] in order to find related words. Since the proposed method is based on the frequency of the words as a measure of its relevance (section 4.3), this does not take into account the frequency of stop words because it is higher than meaningful words.

Chromosome Encoding. GA must encode each solution (chromosome) using a canonical way. One of the most used encodes for a chromosome is the binary representation. For the ATS problem we propose to represent the genes of a chromosome (C) with a vector of length n of binary values (C_n), where the C_i gene corresponds with the i-th sentence in the original text. If C_i gene has a value of 1 ($C_i = 1$) means that the i-th sentence is included in the summary, otherwise not.

Initial Population. After the chromosome encoding is setup, it is possible to create the first generation considering some parameters. Each gene can take a binary random value ($C_{i=1\ldots n} = Random[0,1]$). However, if a sentence is selected to appear in the summary ($C_i = 1$) then the number of words of the i-th sentence are summed to the number of words in the summary. The number of words in the summary must contain at least the number of words specified by the user (m). To guarantee that each sentence could be selected for the summary, there are created n number of chromosomes in the initial population and in each one a different gene is arbitrary set to 1.

$$Population = \left\{C_i^j \middle| j = 1 \ldots n, i = 1 \ldots n, C_{i=j}^j = 1, C_{i\neq j}^j = Random[0,1]\right\}$$

Fitness Function. One of the key steps of a genetic algorithm is the Fitness Function which in this case it is based on the idea of f-measure that it is a harmonic balance of recall and precision measures. Usually in information retrieval, precision is defined as

the number of correctly recovered units divided by the number of recovered units; and recall is defined as the number of correctly recovered units divided by the number of correctly units. In this way, precision measures the fraction of retrieved units that are relevant, while recall measures the fraction of relevant instances that are retrieved. However, for generating a summary (S), the maximum-words threshold (m) of a summary is considered. Consequently, the number of recovery units always is limited by the maximum-word threshold. Therefore the golden summary must have, for one side, the most relevant words of the original text (T) and, for the other side, must have expressivity, it means, it must not be redundant.

The relevance of a word w is represented by the appearing frequency of the word in the original text ($frequency(w,T)$), and the expressivity is represented if only are considered the different words that the summary can have ($\{\text{word} \in S\}$). In this sense, the best summary would contain the most frequent words with respect to the original text and each word must be different. In order to have a normalized measure the sum of the frequencies of the different words in the summary is divided by the sum of the frequencies of the most frequent words with respect to the original text:

$$\beta = \frac{\sum_{\text{p}=\{\text{word} \in S\}}^{m} frequency(p,T)}{\sum_{\text{q}=\{\text{word} \in T\}}^{m} frequency(q,T)}$$

Sentence position feature is a heuristic that has proved that the first sentences from the original text are good candidates of being part for the summary. Normally, the inverse position order of the sentence it is used as a measure of its relevance. The problem of measuring this feature in this way is that, for example, with a 30-sentence text, the first sentence will be 30 times more important that the last one. It makes almost impossible that the last sentence could appear in the summary. In contrast to [13] [14], we propose to make this difference softer using the linear equation with slope t, if t is -1 we will measure the sentence position as in [13] [14], and if t is 0, it will give the same relevance to each sentence. For a text with n sentences, if the sentence i was selected for the summary (it is, the chromosome$|C_i| = 1$) then its relevance is defined as: $t(i - x) + x$, where $x = 1 + (n-1)/2$ and t is the slope for discovering. In order to normalize the sentence position measure (δ), it is calculated the relevance of the first k sentences, where k is the number of selected sentences.

$$\delta = \frac{\sum_{|C_i|=1}^{n} t(i-x)+x}{\sum_{j=1}^{k} t(j-x)+x}, \quad x = 1 + \frac{(n-1)}{2}$$

Therefore the fitness function is: $fitness = \beta \times \delta$

Parent Selection. In this point, each chromosome must have associated a fitness value that will let to mostly select the best chromosomes. The evolution principle establishes that normally if two good solutions are crossing it could produce better solutions; nevertheless, in some cases the solution could be worse. In this step, we employ the classical roulette selection that gives more probability of being selected as

a parent, to the chromosomes that have a greater fitness value. In this way, the worst chromosome has the possibility of being selected, although it was slight probable.

Crossover. Classical crossover operators as *n-point crossover* does not work properly because the new child chromosome could represent a summary with more or less words than the user specified. Therefore, to create the new chromosome we propose to choose from both parents the genes randomly, but consider only those with value 1. In this way, if the C_i gene has a value of 1 in both parents, it has more probability of being selected for the child chromosome. Each time a gene in the child chromosome is selected the minimum number of words for the summary is reviewed.

Mutation. According to the evolution scheme, the mutation slightly happen in the nature with a low probability of 0.1 percentages, however is one of the fundamental mechanisms to preserve the evolution. The classical operator *inverse mutation operator* inverts the binary value of a randomly selected gene. In our proposed scheme, this operator will produce summaries with more or less words than the user specified. In this step we propose to apply the invert operator twice to the child chromosome, but the first time only the genes with value 1 are considering for invert the value; in the second time only the genes with value 0 are considering for invert the value. After that, the number of words in the summary is review it, if the numbers of words do not have the number of words specified by the user, another gene with value 0 is inverted, this process continues until the number of words specified by the user is satisfied.

4 Experimentation

We used the standard DUC 2002 document collection provided [19]. 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[1]. 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 [20] which was found to highly correlate with human judgments [21]. 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 lets to know the f-measure that is a balance (not an average) of recall and precision results.

Table 1 shows the ROUGE evaluation from our approach with the whole DUC-2002 collection; varying the slope from -0.25 to -0.75. In figure 1, it is possible to observe that our approach has the best f-measure when the slope is -0.625.

[1] 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.

Table 1. Results of our proposed approach varying slope from -0.25 to -0.75

Slope	Recall	Precision	F-measure
-0.25	0.4737	0.4686	0.4710
-0.375	0.4736	0.4736	0.4760
-0.5	0.480	0.4747	0.4772
-0.55	0.4829	0.4775	0.4801
-0.625	**0.4855**	**0.4802**	**0.4827**
-0.7	0.4843	0.4789	0.4815
-0.75	0.4790	0.4735	0.4761

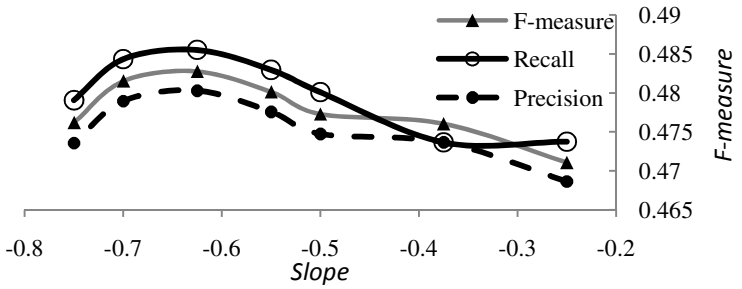

Fig. 1. Behavior of the performance when the slope is varying

5 Comparison with Related Works

In table 2, our proposed approach is compared to others approaches that have used the same DUC-2002 document collection for text summarization.

- **Baseline** (random) [3]: This is a heuristic in which the summaries are built from a set of sentences selected in random way. This simple strategy has the purpose of determine how significant the results can be achieved.
- **TextRank** [17]**:** The approach is a ranking algorithm based on graphs. A graph is built to represent the text, so that the nodes are words (or other text entities) interconnected by vertices with meaningful relationships. For the task of extracting sentences, the goal is to qualify whole sentences and sort highest to lowest rating. Therefore, a vertex is added to the graph for each sentence in the text. To establish connections (cycles) between sentences, define a relationship of similarity, where the relationship between two sentences can be seen as a process of "recommendation": a sentence that points to some concept in the text gives the reader a "recommendation" to refer to other sentences in the text that point to the same concepts and therefore a link can be established between any two sentences that share a common content.
- **Maximal Frequent Sequences (MFSs)** [3] [8] [9]. Ledeneva *et al.* [3] [8] [9] experimentally shows that the words which are parts of bigrams (2-word sequences) which are repeated more than once in the text are good terms to describe the content of that text, so also called the maximal frequent sequences

(sequences of words that are repeated a number of times and also are not contained in other frequent sequences). This work also shows that the frequency of the term as ranking of terms gives good results (while only count the occurrences of a term in repeated bigrams).

- **Baseline** (first): This heuristic selects the first sentences of the document until the desired size of the summary is reached [9]. Besides of being a simple heuristic, only four DUC-2002 systems (S1,S2,S3,S4) could outperform the baseline results (showed in table 2).
- **K-means:** The k-means algorithm creates clusters of similar objects. In [3] the k-means is used for creating clusters of sentences from the original text that allow identifying the main ideas; after that, from each cluster the most representative sentence is selected for the summary.
- **Topline [6].** In this work, a GA was used to calculate the best summaries that it is possible to find according with the ROUGE evaluation.

The comparison of the best F-measure results of our proposed approach with the above state-of-the-art approaches is presented in table 2. Since, any method can be worse than choosing random sentences (baseline: random) the significance of f-measure is recalculated as 0%. In opposite way, since any method can outperform the Topline is considered as 100%. Using baseline and topline is possible to recalculate the f-measure results in order to see how significant the results are (see table 2).

Table 2. Results of f-measure with other methods

System	F-measure	Significance
Baseline: random	0.3881	0%
TextRank:	0.4432	26.50%
MFS's (k-best)	0.4529	31.16%
Baseline: first	0.4599	34.53%
GA	0.4662	37.56%
S1	0.4683	38.57%
S2	0.4703	39.53%
TextRank	0.4708	39.77%
S3	0.4715	40.11%
MFS's (1best+first)	0.4739	41.26%
K-means	0.4757	42.13%
MFS's-EM-5	0.4774	42.95%
S4	0.4814	44.87%
Proposed GA	**0.4827**	**45.50%**
Topline [6]	0.596	100%

6 Conclusions

We have proposed a genetic algorithm for automatic single extractive text summarization task. Specifically, we proposed the preprocessing, chromosome

encoding, initial population, fitness function, parent selection, crossover and mutation step. Our genetic algorithm allow to consider the number of words that a user desire. All the parameters that the GA could need are calculated automatically considering the structure of the original text (in fact, it was applied to 567 documents of the DUC collection). In this sense, from the original text was possible determine the number of chromosomes in the population and the number of maximum iterations.

In contrast to the state-of-the-art works related to GA, the proposed GA is not based in a database that was built from features whose were extracted from sentences. Instead, our GA evaluates how good the summary is with respect to the original text, without the necessity of having a collection for training a classifier. In these sense, fitness function tell us more what a summary must contain instead of what process humans follows for building a summary.

Furthermore, we found that if there were a linear relevance with respect to sentence position in the original text, it is of 0.625 considering two consecutive sentences. This parameter was calculated for the DUC-2002 collection. As a future work, other collections will be tested with this parameter.

There are different terms that can be chosen as words, n-grams or MFS; we use words that are easier for extracting for the original text. There are other features that can be extracted from the sentences as similarity to title, sentence length, etc.; the proposed approach uses only the frequency of words and the sentence position. Also, it is important to note that our purposed approach works independently from linguistic resources. We think that this research is relevant since employing basic language-independent information from the original text, it was possible to outperform the others approaches that use the same collection.

References

[1] Lee, J.-H., Park, S., Ahn, C.-M., Kim, D.: Automatic Generic Document Summarization Based on Non-negative Matrix Factorization. Information Processing and Management 45, 20–34 (2009)

[2] Luhn, H.P.: The automatic creation of Literature abstracts. IBM Journal of Research and Development (1958)

[3] Garcia-Hernandez, R.A., Montiel, R., Ledeneva, Y., Rendon, E., Gelbukh, A., Cruz, R.: Text Summarization by Sentence Extraction Using Unsupervised Learning. In: Orejas, F., Ehrig, H., Jantke, K.P., Reichel, H. (eds.) Abstract Data Types 1990. LNCS (LNAI), vol. 534, pp. 133–143. Springer, Heidelberg (1991)

[4] Edmondson, H.P.: New Methods in Automatic Extraction. Journal of the Association for Computing Machinery (1969)

[5] Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: SIGIR 1995 (1995)

[6] Villatoro-Tello, E., Villaseñor-Pineda, L., Montes-y-Gómez, M.: Using Word Sequences for Text Summarization. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2006. LNCS (LNAI), vol. 4188, pp. 293–300. Springer, Heidelberg (2006)

[7] Chuang, T., Yang, J.: Text Summarization by Sentence Segment Extraction Using Machine Learning Algorithms. In: Proc. of the ACL 2004 Workshop, Barcelona, España (2004)

[8] Ledeneva, Y.: PhD. Thesis: Automatic Language-Independent Detection of Multiword Descriptions for Text Summarization. National Polytechnic Institute, Mexico (2009)
[9] Ledeneva, Y.N., Gelbukh, A., García-Hernández, R.A.: Terms Derived from Frequent Sequences for Extractive Text Summarization. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 593–604. Springer, Heidelberg (2008)
[10] Garcia-Hernandez, R.A., Martinez-Trinidad, J.F., Carrasco, A.: Finding maximal sequential patterns in text document collections and single documents. Informatica. International Journal of Computing and Informatics (34), 93–101 (2010)
[11] Ledeneva, Y., Garcia-Hernandez, R., Gelbukh, A.: Multi-document summarization using Maximal Frequent Sequences. Research in Computer Science 47, 15–24 (2010)
[12] Garcia-Hernandez, R., Ledeneva, Y., Gelbukh, A., Citlalih, G.: An Assessment of Word Sequence Models for Extractive Text Summarization. Research in Computing Science (38), 253–262 (2008)
[13] Suanmali, L., Salim, N., Salem Binwahlan, M.: Genetic Algorithm based Sentence Extraction for Text Summarization. International Journal of Innovative Computing 1(1) (2011)
[14] Silla, C.N., Pappa, G.L., Freitas, A.A., Kaestner, C.A.A.: Automatic text summarization with genetic algorithm-based attribute selection. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) IBERAMIA 2004. LNCS (LNAI), vol. 3315, pp. 305–314. Springer, Heidelberg (2004)
[15] Qazvinian, V., Sharif, L., Halavati, R.: Summarising text with a genetic algorithm-based sentence extraction. Int. J. Knowledge Management Studies 2(4), 426–444 (2008)
[16] Cruz, C.M., Urrea, A.M.: Extractive Summarization Based on Word Information and Sentence Position. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 653–656. Springer, Heidelberg (2005)
[17] Rada, M., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2004 (2004)
[18] van Rijsbergen, C.J., Robertson, S.E., Porter, M.F.: New models in probabilistic information retrieval. En línea (1980)
http://tartarus.org/~martin/PorterStemmer/index.html
(Último acceso: Enero 28, 2013)
[19] Document Understanding Conferences. En línea (Julio 16, 2002),
http://www-nlpir.nist.gov/projects/duc/index.html2
[20] Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of Workshop on Text Summarization of ACL (2004)
[21] Lin, C., Hovy, E.: Automatic Evaluation of Summaries Using N-gram Co-Occurrence. In: Proceedings of HLT-NAACL, Canada, (2003)
[22] Ledeneva, Y., Hernández, R.G., Soto, R.M., Reyes, R.C., Gelbukh, A.: EM Clustering Algorithm for Automatic Text Summarization. In: Batyrshin, I., Sidorov, G. (eds.) MICAI 2011, Part I. LNCS, vol. 7094, pp. 305–315. Springer, Heidelberg (2011)