The Performance of BLMSumm: Distinct Languages with Antagonistic Domains and Varied Compressions

Marcelo Arantes de Oliveira[1] Marcus Vinicius C. Guelpeli[2]
1Departamento de Engenharia da Computação

Centro Universitário de Barra Mansa (UBM) –Volta Redonda, RJ – Brasil

2Departamento de Computação - DECOM

Universidade Federal dos Vales do Jequitinhonha e Mucuri – (UFVJM) –Diamantina, MG – Brasil

marceloarantes19@gmail.com[1], marcus.guelpeli@ufvjm.edu.br[2]

Abstract—**The article describes the BLMSumm summarizer, whose aim is to use various local search methods and Metaheuristics to create summaries. Summarization is treated as an issue of optimization and an attribute identification method based on bipartite graphs is presented. BLMSumm is enhanced with a heuristic for attribute selection, the purpose of which is to make it language independent and then compare it to both professional algorithms and those in the literature. The results obtained from the experiments and evaluated using the Rouge tool are promising.**

I. Introduction

DUE to the excess of information currently circulating in the media, text summarization is fundamental to the process of knowledge acquisition. Way before the advent of the digital age, human beings have had the need to synthesize, abstract and, in short, summarize the information they receive. It's an integral part of attaining knowledge. Faced with the volume of information and the ease with which information can be accessed in the digital media, such as the Internet, text summarization has become crucial to the process of knowledge formation. Absorbing so much knowledge, from so many different sources, is an almost superhuman task. It is in this scenario that text summarization arises.

Text summarization generates a version of the original text that maintains the main features of the author's idea. The process of summarizing a text aims to retain the original text's most significant information in a few sentences, thereby making the reader's life easier, since he or she can absorb the text's main idea by reading only a few short lines. The main objective is to reduce the time spent by the reader in this task, which, in turn, increases the chances that the knowledge will be absorbed and, automatically, results in more available time for the reader to spend by increasing his knowledge on the subject, reading new texts and exploring new sources of information.

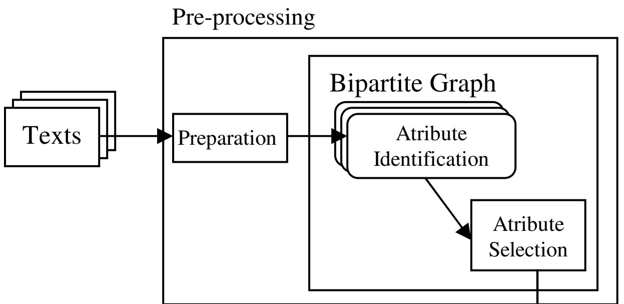
One of the greatest problems in this field is how to generate a summary that is quite concise (highly compressed) yet that does not lose the informativity of the original text [1].

Along with this problem, another issue arises, which is the evaluation of the AS. Evaluating the AS depends fundamentally on human evaluation, and among these evaluators (specialists), due to the high level of complexity of the process, there is no consensus when it comes to analyzing the results obtained in the AS. The aims of this work are twofold: produce extractive summaries that retain informativity and evaluate the performance of the BLMSumm summarizer with various languages, domains and compression rates, as compared to other summarizers described in the literature [2].

This work is organized as follows. Section 2 presents the model used in the application of the iterative improvement methods in summarization. Section 3 describes the methodology, the corpus that was used, the evaluation method and the statistical analyses. In Section 4, the experiments and the results obtained are described. In Section 5 we present the conclusion and suggest future works.

II. The BLMSumm Model

The BLMSumm Models [16] allows for the use of heuristics and metaheuristics in solving the problem of generating automatic summaries. Figure 1 presents the model, the details of which are described in this section. The BLMSumm model is made up of three phases: Pre-processing, Processing and Post-processing.



Oliveira M. A. is with the Universidade Federal de Itajubá (Unifei); Itajubá,MG-Brasil; e-mail:marceloarantes19@ gmail.com.

Marcus V. C. Guelpeli is with the Instituto de Computação – Universidade Federal Fluminense(UFF); Niterói, RJ – Brasil; e-mail: mguelpeli@ic.uff.br.

Fig. 1. BLMSumm Model.

In BLMSumm, pre-processing is executed in two stages, preparation and generation of a bipartite graph, whose purpose is the identifying and then select the attributes, as described in section 2.1.2. The pre-processing phase delivers to the processing module a list of sentences that have been duly classified and "valorized" based on a bipartite graph. The task of the processing module is to choose the sentences that will be a part of the summary, delivering to the next phase a list of sentences to comprise the summary. The post-processing phase in turn generates a summarized text file based on the sentences received from the processing module.

A. *Pre-processing*

1) Preparation: the only preparation technique used in this work was case folding, which consists in transforming all the letters in the document into lowercase (or uppercase). In this stage, a bipartite graph is generated to aid in the process of identifying and selecting the attributes that will be used in the next stage.

2) Bipartite Graph: a graph, G=(V, E), consists of a non-empty set of vertices (V) and a set of edges (E), in which each edge is a set made up of two vertices of V. A bipartite graph is when there is a partitioning of V into two subsets, V_1 and V_2 , so that each and every edge possesses one end in V_1 and the other in V_2 [3]. In the bipartite graph generated, the set of vertices $V_1 \in V$ is given by the set of sentences in the text, whereas the set of vertices $V_2 \in V$ is given by the set of words in the text. An edge $E_x \in E$ is drawn on the graph to determine that a word is part of a sentence. Consider the text in Figure 2.

1. Luxemburgo rebate especulações.
2. Em momento algum existiu veto ao Adriano.
3. O problema de Adriano é na questão filosófica.
4. Luxemburgo afirma que momento não cabe a discussão sobre Adriano: "Não sei o por que dessa discussão no momento."
5. Luxemburgo afirma que Adriano ainda pode ser contratado.

Fig. 2. Fictional text based on the news story in globoesporte.globo.com on 18/03/2011

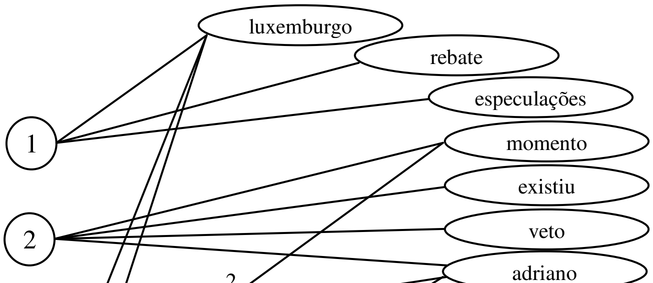


Fig. 3. Bipartite graph representing the text: V1 to the left and V2 to the right.

The text presented in Figure 2 generates the bipartite graph shown in Figure 3, where V_1 = {1, 2, 3, 4, 5} and V_2 = {Luxemburgo, rebate, especulações, momento, existiu, veto, Adriano, problema, questão, filosófica, afirma, cabe, discussão, sei, porque, pode, ser, contratado}, and the edges are: E = {(1, Luxemburgo), (1, rebate), (1, especulações), (4, Luxemburgo), (2, momento), (2, existiu), ...}.

If a word occurs more than once in a sentence, the frequency of the occurrence is added to the edge that represents it, as in the following edges: {(4, discussão), (4, momento)}.

3) Attribute Identifier: The Attribute Identifier of the BLMSumm model goes over the bipartite graph determining the value of each valid word and/or sentence based on a previously established frequency calculation. In this particular work, two types of word frequency calculation were employed:

a. Each sentence will be given the value of the sum of the frequencies of the words in the sentence.

b. Each sentence will be given the value of the sum of the weight of the words in the sentence. The words in the first sentence receive a weight ten times their frequency and the rest of the words have a weight that coincides with their frequency in the text.

One can also use other frequency calculations that take into consideration the following: the location of the word, the location of the sentences, the relation between the frequencies and the location of the sentences and the words in the text, the Relative Frequency, the Inverse Frequency Term per Sentence (TF-ISF) [4], Ranking by term frequency (RTF), Ranking by sentence frequency (RFS).

4) Attribute Selection: This phase is characterized by the choice of a subset based on the set of attributes in the text

[5]. This subset keeps its original position improving the comprehension of the model that is generated. In a data subset that possess n attributes, there are 2n -1 possibilities of obtaining a subset [6]. It is to be expected that in the attribute selection cuts are made in accordance to a given criterion [7].

Luhn proposed a technique to find relevant attributes, assuming that the most significant attributes to discern the content of the document are on an imaginary peak located in between two cut points. However, some degree of arbitrariness is involved when it comes to determining the cut points, as well as the imaginary curve. These are established by trial and error [8].

In this case, the cuts were established in order to deal with inherent aspects of the language in which the text was written. The idea is that, with the cuts, the following problems can be either resolved or minimized:

a. Using cuts of stop words, which makes the superficial approach employed language independent.

b. Using words that are not very significant for the text in choosing sentences that will become a part of the generated summary.

To make the selection, the attributes are ordered by frequency in a decreasing fashion. The first cut, called Cut 1, is made, which corresponds to 10% of the attributes with the highest frequency, as shown in Figure 4. The purpose of Cut 1 is to exclude the largest amount of stop words that can possibly be removed. After this, there is a selection of 30% of the attributes. Hence, only 60% of the attributes remain for Cut 2, which will discard words that are of little significance for the generation of the summary, as shown in Figure 4. Each sentence is given a value that is the sum of the values of the valid attributes that belong to the sentence.

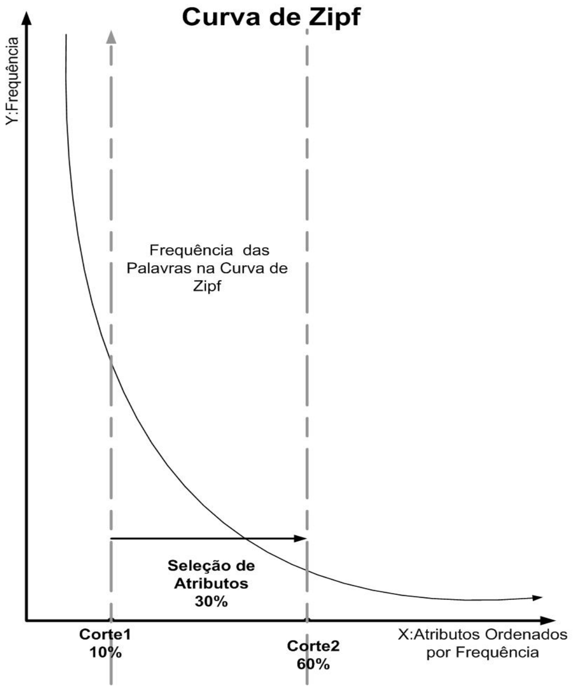


Fig. 4. Zipf's Curve and the cuts used in Attribute Selection in BLMSumm

B. *Processing*

1) Summarization as an issue of optimization: Considering that each sentence in the summary receives a value that determines its “importance” in regards to the text, choosing the best summary could be understood as the process that chooses the most “important” set of sentences in the text, without allowing the number of words in the set to surpass a certain compression rate. In this case, superficial summarization can be seen as an issue of optimization:

$$\text{Maximize} \sum_{i=1}^{n} h_i b_i \qquad (1)$$

$$\text{Subject to} \sum_{i=1}^{n} p_i b_i \leq C \qquad (2)$$

Where:

- n – is the total number of sentences in a text.
- i – is the i-eth sentence of a text.
- p_i – is the amount of words in the i-eth sentence of a text.
- h_i – is the value of the i-eth sentence of a text, given by the evaluation function.
- b_i – is a binary value that determines whether the i-eth sentence of a text is (b_i = 1) or is not (b_i = 0) part of a summary.
- C – is the maximum amount of words that are allowed to be in a summary, which is given as function of the compression rate.

Hence, each state is a summary and can be viewed as a list of sentences, a vector of binaries and/or a whole number.

2) Local Search Methods and Metaheuristics: Local search algorithms are built as a means of exploring the search

space. In them, an initial state is generated by a given method (either constructive or random) and improvements are made at each iteration until a stop condition is reached. Local search methods tend to become stuck in optimum locations, thereby not generating an optimum global solution to a given problem. Metaheuristics are developed in order to get away from these optimum locations and converge towards an optimum global solution [9].

a. Greedy Algorithms: A greedy algorithm will always make the choice that appears to be the best one at the moment; that is, it will make the optimum choice considering local conditions in the hope that this choice leads to an optimal solution for the global situation [10]. The greedy algorithm chooses the best solution locally, hence, at each iteration, the algorithm selects the sentence with the highest value and, if the sum of the words in the current summary added to the amount of words in the sentence does not exceed the compression rate, then it is added to the summary.

b. Simulated Annealing: The "locality" employed in algorithms such as the Greedy algorithm may lead to excellent solutions locally, that is, to local maximums that often times may be far from the optimal solution for a given problem.

   Simulated Annealing is a maximization/ minimization technique that mimics the process used in metallurgy of heating and cooling materials in order to reduce defects. At each iteration, simulated annealing generates a possible solution and if it is in fact a better solution than the one previously stored, it will replace the former solution and become the current option. There is a probability that a worse solution than the current one is chosen as a replacement; hence, simulated annealing attempts to escape from local maximums [9].

   Each and every summary is represented by a binary number. The valid bits determine which sentences will be included in the summary. At each iteration, a whole number is randomly selected and converted into a binary, which is called the next summary. This number is then associated to the solution it represents and the amount of words is verified (if there are more words than the compression rate allows, the solution is discarded), as is the score of the summary (taking into account the evaluation function). If the next summary receives a higher score than the current summary, then it replaces the current summary. If not, then there is a probability that the next summary could become the current summary, though this probability decreases with time.

III. Methodology

A. *Corpus*

In this work, the corpuses are primarily divided into two languages, Portuguese and English.

The Portuguese language texts encompass the journalism and the medical domains, with a total of 200 original texts, 100 in each domain. Texts in the medical domain were extracted from the Scielo which is specialized in scientific articles spanning a number of areas in the health sciences. The journalism texts were taken from the TeMário 2004 corpus, made up of articles from the online newspaper Folha de Sao Paulo and encompassing the following 5 sections: Special, International, World, Opinion and Politics.

Texts in the English language were also from the journalism and the medical domains, totalling 200 original texts. The journalism texts were extracted from the news agency Reuters and the medical texts came from the Scielo website.

The summaries were obtained using each of the summarization algorithms chosen for the experiment, as specified and defined in item 3.2. For the compression the following percentages were used: 50%, 70%, 80% and 90%.

B. *Summarization Algorithms*

For this experiment, we used Local Search methods and Metaheuristics, implemented as variations of the BLMSumm model that were proposed in item 2.1.3 and that work in the English and Portuguese languages.

In order to compare the performance of these methods, specific summarization algorithms for the Portuguese language and for the English language were used. A random function was developed that can also generate summaries in both language. All of these summarization algorithms – the one that was developed, the ones in the literature and the professional ones – are detailed below.

- Supor [11] – the sentences selected for the extract are the ones that include the most frequent words in the source-text, as these supposedly represent the most important concepts in the text. The choice of each sentence is made after the sentence has been classified according to its representativity in the text. For this, a score is attributed to each sentence based on the sum of the frequencies of the words as they appear in the text as a whole. After the scoring has been made, a threshold is determined based on statistical measures and, then, the sentences with the most frequently encountered words are selected.

- Gist_Average_Keyword [4] – the sentences are scored using one of two simple statistical methods: the keywords or the average keywords method, in which the difference is that the latter has a normalization function in relation to the size of the sentences (measured by the number of words). Afterwards, the sentences are ranked according to their scores and the sentence with the highest score is chosen as the gist sentence – that is, the sentence that best represents the main idea of text. The selected sentences meet the following criteria: (a) they contain at least one stem in common with the gist sentence selected in the previous step and (b) they have a score that is higher than the threshold, which is calculated by taking the average score of the sentences. Criteria (a) looks to select sentences that complement the text's main idea, whilste (b) aims to select only the relevant sentences, as per the compression percentage determined by the user.

- Gist_Intrasentença [4] – is employed in all the sentences for the exclusion of stop words.

- Copernic– is a professional summarizer that can be used for texts in the English language. From the information available online and after contacting the suppliers, we were unable to obtain specific details regarding the algorithm it uses.

- Intellexer Summarizer Pro - is also a professional summarizer that can be used for texts in the English language. From the information available online and after contacting the suppliers, we were unable to obtain specific details regarding the algorithm it uses.

- SweSum [12] - this is the summarizer from the literature. SweSum summarization engine was, originally intended for Swedish but applied to other languages since then . For each language, the system uses a lexicon to map the inflected forms of the words in the content to their respective roots. This is used to identify the theme, based on the hypothesis that the sentences that contain words with high frequency.

- AleatorioM is a function that randomly chooses the sentences that will be part of the summary. In short, at each iteration, AleatorioM chooses a whole number whose binary correspondent has a total of bits that is equal to the number of sentences in the text. Each bit represents a sentence. A sentence is chosen to be part of the summary if its corresponding bit is valid. This procedure is repeated many times until one finds a summary whose compression rate is in accordance to the previously determined rate.

C. *Evaluation of the Generated Summaries*

One of the most commonly used method of evaluating automatic summaries is the ROUGE measure, which stands for *Recall Oriented Understudy for Gisting Evaluation* [13]. ROUGE is a summary evaluation package that was created to allow for a direct comparison between an AS and a corresponding human-made summary using n-grams as a metric. With this tool, one is able to analyze the closeness in quality of the AS in relation to the reference human-made summary. Generally speaking, ROUGE calculates the level of informativity of the extracts. According to the definitions offered by Lin e Hovy (2003), the calculation employed by ROUGE is based on the common sets of words in sequence (n-grams) found between the human-made reference summaries and the automatic summaries. The higher the number of common words between the summaries, the higher the score given to the AS. To validate the results, we employed Friedaman's ANOVA statistical tests and Kendall's coefficient of concordance, methods used in international conferences for AS evaluation such as TAC (Text Analysis Conference), the most relevant in the field of AS.

IV. Results Analysis

The variations of BLMSumm are *Guloso* (greedy), *GusoloCorte* (greedy) and *SimAnn* (Simulated Annealing), the numbers (1 and 2) shows which frequency calculation were employed (section 2.1.2.1). As seen in Figure 5, the higher the compression rates in the summaries, the lower the performance of the BLMSumm variations (Guloso1, Guloso2, GulosoCorte1, GulosoCorte2, SimAnn1 and SimAnn2) and the lower the AleatorioM function. In this domain, in none of the compression rates, did we observe an improvement in the summarizers in the literature. The algorithms that employ any type of cut tend to show poorer performance as the compression rates increase. We did not see a significant increase during the increase in compression rates of the summarizers in the literature and it can be noted that the best performance in all compressions were the Greedy algorithms with no cuts.

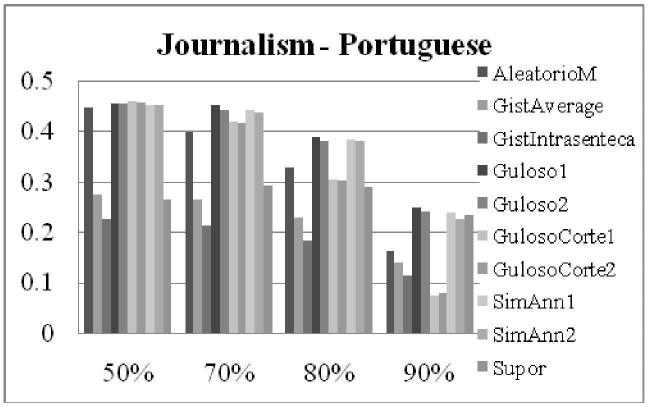


Fig. 5. Graphs showing the performance of the Gist_Average, Gist_Intrasenteça, Supor, AleatórioM and BLMSumm summarizers, with 50%, 70%, 80% and 90% compression, in the journalism domain in the Portuguese language. The results are the accumulated averages of the F-Measure obtained through ROUGE using a resample of 100.

The compressions 50%, 70% and 80% the variations of BLMSumm and the AleatórioM function have heightened performances, although at a compression rate of 90%, the algorithms based on randomness (simulated annealing and AlgoritmoM) show a steep decline in performance, while the Greedy algorithms seem to improve their performance. The algorithms in the literature improve their performances as the compression rate increases. The Gist_Average obtained the best performances in all compression rates, Figure 6.

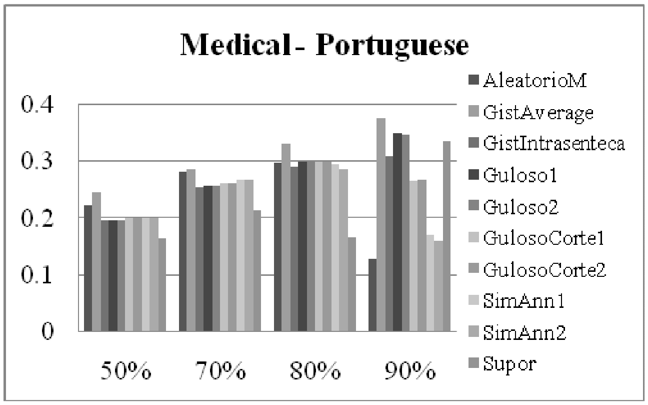


Fig. 6. Graphs showing the performance of the Gist_Average, Gist_Intrasenteça, Supor, AleatórioM and BLMSumm summarizers, with 50%, 70%, 80% and 90% compression, in the medical domain in the Portuguese language. The results are the accumulated averages of the F-Measure obtained through ROUGE using a *resample* of 100.

In figure 7, one can tell that the both the algorithm in the literature and the professional algorithms have better performances as the compression increases, in contrast to the algorithms with cuts, whose performance declines. The Copernic algorithm obtained the highest increase in performance throughout all compressions. In 70%, 80% and 90% compression, the Greedy algorithms performed consistently worse than the rest.

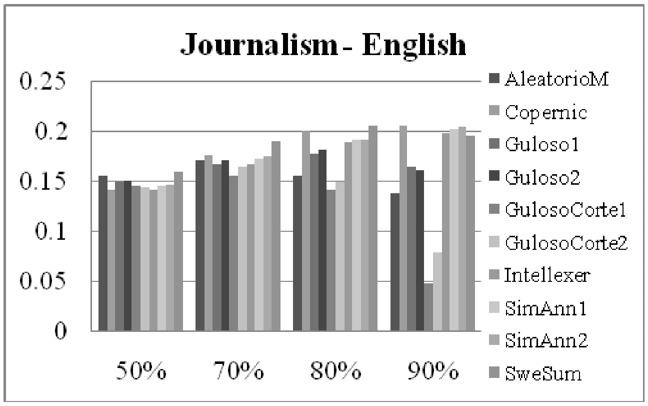


Fig. 7. Graphs showing the performance of the Copernic, Intellexer, SweSum, AleatórioM and BLMSumm summarizers, with 50%, 70%, 80% and 90% compression, in the journalism domain in the English language. The results are the accumulated averages of the F-Measure obtained through ROUGE using a resample of 100.

By analyzing figure 8, it is clear that again the algorithm in the literature as well as the professional algorithms perform better as the compression rates increase. The Simulated Annealing algorithm increases in performance as the compression varies from 50% to 80%, but there is a steep decline at 90% compression. As is the case for the professional algorithms and the one in the literature, the Greedy algorithm with no cuts performs better as the compression increases.

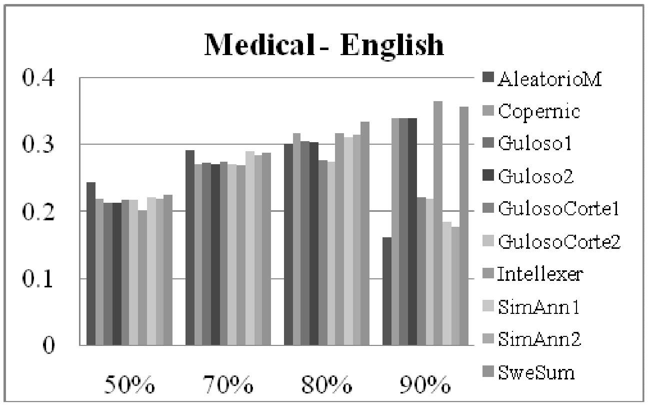


Fig. 8. Graphs showing the performance of the Copernic, Intellexer, SweSum, AleatórioM and BLMSumm summarizers, with 50%, 70%, 80% and 90% compression, in the medical domain in the English language. The results are the accumulated averages of the F-Measure obtained through ROUGE using a resample of 100.

V. Conclusion

In the journalism domain in the Portuguese language, the BLMSumm summarizer performed normally, that is, as the compression increased, its performance declined. This is natural and expected, since the decrease in word volume leads to a loss of informativity in the summaries. In this case, one can clearly see a greater decline in the F-Measure values for the journalism domain as the compression increases, which suggests that this domain is more susceptible to informativity losses.

In the medical domain the F-Measure results are better, since the BLMSumm performs very well in domains in which there is a greater frequency of uncommon words or neologisms. This observation does not apply to the domain with a poorer lexicon, where common words that are used very frequently lose informativity as the compression increases, such as is the case in journalism. It is worth noting that the best results occurred in the English language, due either to characteristics of the language itself or the good quality of the English language summarizers.

The results obtained using the BLMSumm summarizer are promising because although they did not indicate the best comparative performances in the various experiments that were conducted, some variation approximates those that obtained the best performances. It must be taken into account that the algorithms in the literature and the professional algorithms are language dependent, whereas the BLMSumm summarizer does not have this limitation.

VI. Future Works

Use the bipartite graph to generate many diverse simulations with different frequency calculations; vary the local search methods; implement population search methods that vary in domains and languages and test them in different compressions.

References

[1] C. H. Delgado, C. E.Vianna, and M. V. C. Guelpeli, Comparando sumários de referência humano com extratos ideais no processo de avaliação de sumários extrativos. IADIS Ibero-Americana WWW/Internet, Algarve, Portugal, p. 293, 2010.

[2] M. V. Guelpeli, F. C. Bernardini, and A. C. B. Garcia, Todas as Palavras da Sentença como Métrica para um Sumarizador Automático, Tecnologia da Informação e da Linguagem Humana - TIL. WebMedia. Vila Velha, 2008.

[3] A. Drozdek, *Data Structures and Algorithms in C++, 2nd edition*. Pacific Grove, CA - USA: Brooks / Cole, 2001.

[4] T. A. S. Pardo, Estrutura textual e multiplicidade de tópicos na sumarização automática: o caso do sistema GistiSumm, São Carlos, Brasil, 2006.

[5] C. Y. Liu, L. Yu, "Toward Integrating Feature Selection Algorithms for Classification and Clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491-502, 2005.

[6] B. M. Nogueira, "Avaliação de métodos não-supervisionados de," USP. São Carlos - Brasil, pp. 104, 2009.

[7] M. Dash, and H. Liu, "Feature Selection for Classification," *Inteligence Data Análise,* no. 1, pp. 131-136.

[8] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research an Development*, 2, 1958.

[9] W. Raynor, *The International Dictionary of Artificial Intelligence*. New York, NY - USA: Amacon, 1999.

[10] T. H. Cormen, and C. E. Leiserson, *Algoritmos - Teoria e Prática*. Segunda Edição. Rio de Janeiro, RJ: Campus, 2002.

[11] M. MÓDOLO, SuPor: um Ambiente para a Exploração de Métodos Extrativos para a Sumarização Automática de Textos em Português. Dissertação de Mestrado - UFSCar, 2003.

[12] H. Dalianis, *SweSum - A Text Summarizer for Swedish*. IPLab-174, NADA, KTH. [S.l.]. October 2000.

[13] T. A. S. Pardo, and L. H. M. Rino, *TeMário: Um Corpus para Sumarização Automática de Textos*. São Carlos, Brasil. 2004.

[14] E. Hovy, and C. Lin, "Automated text summarization in summarist," in *Intelligent Scalable Text Summarization Workshop*, Madrid, Spain, 1997, pp. 39-46.

[15] C. Lin, and E. H. Hovy, "Automatic evaluation of summaries using N-gram," in *Proccedings of the Language Technology Conference*, Edomonto, Canada, 2003.

[16] M. A. Oliveira, and M. V. Guelpeli, "BLMSumm Métodos de Busca Local e Metaheurísticas na Sumarização de Textos," *VIII Encontro Nacional de Inteligência Artificial - ENIA, 2011*, Natal, Brasil, 2011.