Abstractive Multi-Document Text Summarization Using a Genetic Algorithm

Verónica Neri Mendoza, Yulia Ledeneva(✉),
and René Arnulfo García-Hernández

Autonomous University of the State of Mexico,
Instituto Literario No. 100, 50000 Toluca, Mexico
veronica.nerimendoz@gmail.com, yledeneva@yahoo.com,
renearnulfo@hotmail.com

Abstract. Multi-Document Text Summarization (MDTS) consists of generating an abstract from a group of two or more number of documents that represent only the most important information of all documents. Generally, the objective is to obtain the main idea of several documents on the same topic. In this paper, we propose a new MDTS method based on a Genetic Algorithm (GA). The fitness function is calculated considering two text features: sentence position and coverage. We propose the binary coding representation, selection, crossover and mutation operators to improve the state-of-the-art results. We test the proposed method on DUC02 data set, specifically, on Abstractive Multi-Document Text Summarization (AMDST) task demonstrating the improvement over the state-of-art methods. Four different tasks for each of the 59 collection of documents (in total 567 documents) are tested. In addition, we test different configurations of the most used methodology to generate AMDST summaries. Moreover, different heuristics such as topline, baseline, baseline-random and lead baseline are calculated. The proposed method for AMDTS demonstrates the improvement over the state-of-art methods and heuristics.

Keywords: Multi-Document Text Summarization (MDTS) · Language-independent methods · MDTS methodology · Genetic algorithm · Heuristics

1 Introduction

The extensive use of Internet has caused the enormous growth in the usage of digital information. Currently, there are a great variety of users of online information services with a huge amount of unstructured digital information [9, 18]. The user accesses the information through queries, but the precision is always an issue due to the information overload. One way to resolve this issue is by generating summaries [6, 12].

The general process of summarization consists of rewriting the full text into a brief version [19]. Automatic Text Summarization (ATS) is a task of Natural Language Processing (NLP). ATS consists in selecting the most important units which could be paragraphs, sentences, part of sentences or keywords from a document or collection of documents using the state-of-the-arts methods or commercial systems.

J. A. Carrasco-Ochoa et al. (Eds.): MCPR 2019, LNCS 11524, pp. 422–432, 2019.
https://doi.org/10.1007/978-3-030-21077-9_39

ATS methods can be abstractive or extractive. In the abstractive text summarization, the summaries are composed from fusing and generating new text that describes the most important facts [10]. In the extractive text summarization, the sentences or other parts of a text are extracted and concatenated to compose a summary [14, 18].

Depending on the number of documents, summarization techniques can be classified in two tasks: Single-Document Text Summarization (SDTS) and MDTS. The main goal of the MDTS is to allow to the users to have an overview about the topics and important information that exists in collection of documents within relatively a short time [1, 3, 22]. The MDTS has gained interest since mid-1990s [4], starting with the development of evaluation programs such as Document Understanding Conferences (DUC) [23] and Text Analysis Conferences (TAC) [26].

In this paper, we consider the methodology for building the final summary that considers all sentences of all documents [2, 18, 21, 29]. In this paper, a new MDTS method based on a GA is proposed.

The organization of the paper is as follows: the Sect. 2 describes the proposed method. The Sect. 3 shows experimental configuration and results. Finally, in the Sect. 4 the conclusions are presented.

2 Proposed Method

2.1 Pre-processing

The proposed method consists of three steps. In the first step, the documents of the collection were chronologically ordered, then the original text is adapting to the entry of the format of the GA, where the original text is separated in sentences. Also, the text pre-processing is applied to the collection of documents. Firstly, the text was divided into words separated by commas, then some tags were placed in the text to be able to differentiate quantities, emails, among others, and finally the lexical analysis is carried out [8, 20].

2.2 Text Model

The goal of text modeling is to predict the probability of natural word sequences. It assigns high probability on word sequences that occur, and low probability on word sequences that never occur. The simplest and most successful form for text modeling is the n-gram model. n-gram is defined as a subsequence of consecutive elements in a given sequence [15, 20].

2.3 Genetic Algorithm

The basic configuration of GA is defined as follows [5]: the initial population is randomly generated, while the population of other generations are generated from some selection/reproduction procedure. The search process terminates when a termination criterion is met. Otherwise a new generation will be produced, and the search process continues. The termination criterion can be selected as a maximum number of generations, or the convergence of the genotypes of the individuals. Genetic operators are constructed according to the problem to be solved, so the crossover operator has been applied to the generation of summaries.

Encoding. The binary encoding is used for each individual, where each sentence of the document constitutes a gene. The values 1 and 0 determine if the sentence will appear or no in the final summary. The initial population is randomly generated [8, 20].

Selection Operator. *Roulette* selects individuals from a population according to their aptitude, and is intended to select stronger individuals (with greater value in the fitness function) [20].

Crossover Operator. This operator has been used in [8]. It was designed for ATS, where each individual represents a selection of sentences. The process of cross over is randomly select parents, only those with genes with a value of 1, and this value is assigned to the new individual. Genes with a value of 1 in both parents will be more likely to be chosen. To meet the condition of the summary, a gene is selected to be part of a new individual, the number of words is counted [20].

Mutation Operator. This operator performs the mutation according to a certain probability as described in [8, 20].

Stop Condition. The stop condition that was applied for the term of the GA is the maximum number of generations. For the execution of the GA, consideration must be given to the number of words that the summary must have. In this case, the lengths of 10, 50, 100 and 200 words were used.

The number of individuals and the number of generations are automatically calculated by the GA through the Eqs. 1 and 2 respectively. The number of individuals is determined by the number of sentences that the document contains by means of the following equation [20]:

$$Number\,individuals = Number\,Sentences * 2 \tag{1}$$

The number of generations is calculated trough the following equation:

$$Number\,Generations = 4 * 15 * \; Number\,Sentences \tag{2}$$

Fitness Function. The fitness function was used in the method [8, 20]. In this fitness function are evaluated two features, position sentences and coverage. The main idea is that if all the sentences (see the Eq. 3) had the same importance, it is could draw a line with the points that make up those coordinates as it is showed in Eq. 4.

$$\{X_1, X_2, X_3, \ldots X_n\} \tag{3}$$

$$\{(X_1, y), (X_2, y), (X_3, y), \ldots (X_n, y)\} \tag{4}$$

The idea for assigning more importance for the first sentences, would be consider the first sentence with the importance X_n, the second with importance $X_n - 1$.

Since the placement of the line indicates its importance, the midpoint of that line can be used to determine the slope of the line; thus, softening the importance of sentences. This would allow us to know how important a sentence is with respect to the following. For this can use the general equation of the slope of the line.

For a text with n sentences, if the sentence i is selected for the summary then its relevance is defined as $t(i-x)+x$, $where\, x = 1 + (n-1)/2$ and t is the slope to be discovered. With the objective to normalize the measurement of the position of the sentence $(Sentence\, Importance)$, the importance of the first k sentences is calculated, where k is the number of selected sentences. Then the formula to calculate the importance of the first sentences would be as follows:

$$Sentence\, importance = \frac{\sum_{|c_i|}^{n} = 1^{t(i-x)+x}}{\sum_{j=1}^{k} t(j-x)+1},\, x = 1 + \frac{(n-1)}{2} \tag{5}$$

However, it is not the only value by which the GA should be governed since it would try to obtain only the first sentences. It is also necessary to evaluate that the summary has different ideas, that is, it is not repetitive, but at the same time it has important words $(Precision - Recall)$. To measure both things the fitness function makes the summation of the frequencies of the n-grams that the summary weigh how significant are the n-grams obtained is the same but considering the original text, in this case only the most frequent n-grams according to the number of minimum words. This weighing are Precision and Recall. Precision defines as a sum of the frequencies of the n-grams consider the original text, expressed as follows:

$$\sum Original\, text\, frequency \tag{6}$$

Recall defines as a sum of the frequencies of the different n-grams of summary:

$$\sum Frequency\ Summary \tag{7}$$

Therefore, the formula for obtaining Precision-Recall is:

$$Presicion - Recall = \frac{\sum Original\, text\, frequency}{\sum Frequency\ Summary} \tag{8}$$

Finally, to obtain the value of the fitness function, the following formula is applied, which is multiplied by 1000.

$$FA = Presicion\_Recall * Sentence\ Importance * 1000 \tag{9}$$

3 Experimental Results

We test the proposed method based on the MDTS using the dataset provided in DUC [23]. We use ROUGE[1] to evaluate the proposed method, which is widely applied by DUC for performance evaluation [16]. It measures the performance of a summary by counting the unit overlaps between the candidate summary and two reference summaries.

Dataset. In order to empirically evaluate the summarization results, DUC02 dataset is used. This corpus contains 59 clusters of news texts documents, every cluster contains from 5 to 14 text documents. The summary lengths are 10, 50, 100, and 200 words [17].

Experiment Configuration. In each experiment, we followed the standard sequence of steps explained in the Sect. 2. The Table 1 presents the best obtained result of the proposed method with different slop value, for four different summary lengths, and selection operator *(Roulette*). The sentence selection considered parameter is *k-best +first* which consists in selecting the first sentences of the text, until the desired size of the summary is reached [13, 14].

In the Sects. 3.1, 3.2 and 3.3, we describe and compare the best obtained results of the proposed method to the state-of-the-art methods, commercial systems and heuristics.

3.1 Description of the State-of-the-Art Methods

R2N2 [6]**:** Recursive Neural Networks (R2N2) to rank sentences for MDTS were presented. This is a supervised method because analyzes the syntactic structure of an input sentence and produces a sentence parse tree. The R2N2 are used to automatically learn ranking features over parsing tree, and used features as POS, named entity, sentence depth, among other. Two sentence selection methods were used: **R2N2_G:** uses greedy algorithm *G* which selects the most salient sentences with a similarity threshold. **R2N2_ILP:** The sentence selection is based on Integer Linear Programming (ILP). ILP is intended to find the global optimum. In addition to the previous selection methods in [6] were used three support machine regression baselines: **UR:** Unigram regression with *G* selection. **SR:** Sentence regression with *G* selection. **U+SR:** ILP method assigns weights to the words of the sentences that are measured by related regressions.
LexRank [29]**:** It computes sentence importance based on the concept of centrality in a graph representation of sentences. In this model, a connectivity matrix based on intra sentence cosine similarity is used.
WF (Word Frequency) [18]**:** The sentences that contains the words with the most frequency from the source document (without stop-words) are considered for the final summary. **TE** (Textual Entailment) [18]: It consists of using textual implication in ATS that has been considered as a useful approach for obtaining a preliminary summary,

[1] ROUGE (Recall-Oriented Understudy for Gisting Evaluation) version 1.5.5.

Table 1. The best obtained results with several parameters of the proposed method.

Summary length	Value of slope	Selection operator	Elitism	Results		
				Recall	Precision	F-measure
10 words	−0.20	Roulette	3	16.312	15.465	15.840
50 words	−0.70	Roulette	3	28.383	28.170	28.268
100 words	−0.70	Roulette	3	35.377	35.063	35.214
200 words	−0.72	Roulette	3	42.728	42.332	42.526

where the sentences have not associated with any other sentence of document. **TE + WF** [18]: This method applies prior recognition of the textual entailment as a previous step to the words frequency in the summarization process.

GS, Knapsack, ILP [21]**:** This work considers tree inference global algorithms, the first is a greedy approximate method **GS**, the second is a dynamic programming approach based on solutions to the knapsack problem **Knapsack**, and the third is an exact algorithm that uses an Integer Linear Programming formulation of the problem **ILP**.

MFS (Maximal Frequent Sequences) [11]**:** This method analyses several options for language-independent features and corresponding term weighting feature based on units larger than one word.

3.2 Description of Heuristics

Topline [25]**:** It is a heuristic that allows to obtain the maximum value that any state-of-the-art method can achieve due to the lack of concordance between evaluators, since it selects sentences considering one or several gold-standard summaries.

Baseline-First [25]**:** It is a heuristic that take the first sentence in the 1st, 2nd, 3rd, etc. document collection in chronological sequence until you have the target summary size.

Baseline-Random [25]**:** It is the state-of-the-art heuristic that randomly selects sentences to present them as an extractive summary to the user.

Baseline-First-Document: It is a heuristic that take the first sentences in the 1st document of a document collection, until you have the target summary size. This heuristic is proposed in this work.

Lead Baseline [18]**:** It is a heuristic that take the first 10, 50, 100, and 200 words in the last document in the collection, where documents are assumed to be chronologically ordered.

3.3 Description of Systems

Svhoong Summarizer [12]**:** This system is available online. The text should be copied to the web page. The final summary is the text underlined in the same page.

Pertinence Summarize [12]**:** This system is available online. For each document, Pertinence automatically calculates percentages depending on the number of words in the document.

Tool4noobs Summarizer [12]**:** This system is available online and it uses three steps: extraction of text, identification of the key-words, and identification of sentences and generation of summary.

Copernic Summarizer [12]**:** The system was exclusively developed for the generation of automatic summaries. The version 2.1 was used on the Microsoft Windows operating system.

Microsoft Office Word Summarizer [12]**:** This system can be found in versions of Microsoft Office Word 2003 and Microsoft Office Word 2007.

3.4 Experimental Results (4 MDTS Tasks)

It is considered that any method can be worse than randomly choosing sentences (baseline-random), so the advance of baseline-random can be recalculated as 0%. The best possible performance is called topline and it is considered as 100% of advance. Using baseline-random and topline is possible to recalculate F-measure results in order to see the advance compared to the worst and the best results. In the Tables 2, 3, 4 and 5, the results of F-measure of ROUGE-1, ROUGE-2 and Advance are presented.

10 Words Task: The summary length of this task is 10 words. This task is rarely tested in the state-of-the-art, because of the difficulties related to find less words to describe the most important information. We calculate all heuristics: topline, baseline-first, baseline-random, baseline-first-document and lead-baseline (see Table 2) where we show the results of all state-of-art method and heuristics. The topline shows an enormous margin which exists between the best method and the best possible result to obtain. The difference is 62.09%. We hope that these experimental results serve as a reference for the future works.

Table 2. Comparison of the results to other methods for 10 words.

Type of Method	Method	ROUGE-1	ROUGE-2	Advance (%)
Unsupervised methods	Proposed	15.840	3.699	27.50%
Heuristics	Topline	37.636	16.732	100%
	Baseline-first	18.970	5.994	37.91%
	Baseline-first-document	18.970	5.994	37.91%
	Lead baseline	12.869	3.675	17.62%
	Baseline-random	7.570	1.226	0%

50 Words Task: The summary length of this task is 50 words. Several state-of-the-art unsupervised methods are 5 heuristics are presented in Table 3 (3 are calculated in the state-of-the-art: topline, baseline-first and baseline-random, and 2 heuristics are calculated in this paper: baseline-first-document and lead-baseline). In the Table 3, we see that the proposed method overcome the results of all state-of-art method and heuristics. As topline heuristic shows an extensive margin exists between the best method and the best possible result to obtain. The difference is 68.79%.

Table 3. Comparison of the results to other methods for 50 words.

Type of method	Method	ROUGE-1	ROUGE-2	Advance (%)
Unsupervised methods	Proposed	28.268	6.084	31.21%
	ILP [21]	28.100	5.800	30.42%
	Knapsack [21]	27.900	5.900	29.48%
	GS [21]	26.800	5.100	24.34%
Heuristics	Topline [25]	42.967	16.084	100%
	Baseline-first [25]	26.939	5.241	24.99%
	Baseline-first-document	25.286	4.331	17.25%
	Lead baseline	22.587	3.733	4.62%
	Baseline-random [25]	21.599	2.298	0%

100 Words Task: The summary length of this task is 100 words. The task of 100 words summary length is the most tested in the state-of-the-art. We can see several unsupervised and supervised methods, commercial tools, heuristic calculated in the state-of-the-art such as topline, baseline-first and baseline-random, and heuristics calculated in this paper such as baseline-first-document and lead baseline. In the Table 4, we see that the proposed method overcome the results of the state-of-art method, heuristics and commercial systems. However, only one supervised method from [2] overcome results of the proposed method, our conclusion is because the usage of language-dependent features. As topline result shows, an enormous margin exists between the best method and the best possible result to obtain. The difference is 55.47%. The summaries of commercial systems were provided by the authors and were evaluated.

Table 4. Comparison of the results to other methods for 100 words.

Type of method	Method	ROUGE-1	ROUGE-2	Advance (%)
Unsupervised methods	Proposed	35.214	8.145	31.41%
	LexRank [29]	35.090	7.510	30.82%
	Knapsack [21]	34.800	7.300	29.44%
	ILP [21]	34.600	7.200	28.48%
	GS [21]	33.000	6.800	20.84%
	MFS [11]	32.640	5.987	19.12%
	WF [18]	29.620	5.200	4.69%
Supervised methods	R2N2_ILP [2]	37.960	8.800	44.53%
	R2N2_G [2]	36.840	8.520	39.18%
	Ur + Sr [2]	35.130	8.020	31.01%
	Sr [2]	34.230	7.810	26.71%
	Ur [2]	34.160	7.660	26.38%
	TE + WF [18]	31.333	5.780	12.87%

(*continued*)

Table 4. (*continued*)

Type of method	Method	ROUGE-1	ROUGE-2	Advance (%)
Heuristics	Topline [25]	49.570	18.998	100%
	Baseline-first [25]	33.385	7.042	22.68%
	Baseline-first-document	30.170	5.326	7.32%
	Lead baseline	28.770	5.097	0.63%
	Baseline-random [25]	28.637	3.798	0%
Commercial systems	Coopernic [12]	32.489	5.692	18.40%
	Svhoong [12]	31.555	6.121	13.93%
	Word 2007 XP [12]	29.383	5.930	3.56%
	Word 2003 7 [12]	28.792	5.055	0.74%
	Word 2003 XP [12]	28.792	5.055	0.74%
	Word 2003 Vista [12]	28.792	5.055	0.74%
	Word 2007 Vista [12]	28.713	6.172	0.36%
	Word 2007 7 [12]	28.284	4.975	0.36%
	Pertinence [12]	27.377	3.495	−6.01%
	Tool4noobs [12]	26.688	3.143	−9.31%

200 Words Task: The summary length of this task is 200 words. Several unsupervised and supervised methods, and 4 heuristics calculated in the state-of-the-art such as topline, baseline-first, lead baseline and baseline-random and 1 heuristic calculated in this paper is baseline-first-document. In the Table 5, we see that the proposed method overcome the results of all the state-of-art methods and heuristics. The topline result shows an extensive margin exists between the best method and the best possible result to obtain. The difference is 67.37%. We logically understand that the MDST task needs the summary of more length, so our proposed method could find the most important ideas to compose the summary and obtain the best result in the task where the summary of bigger length is required.

Table 5. Comparison of the results to other methods for 200 words.

Type of method	Method	ROUGE-1	ROUGE-2	Advance (%)
Unsupervised methods	Proposed	42.617	11.67	32.63%
	ILP [21]	41.500	10.300	27.06%
	Knapsack [21]	41.200	10.000	25.57%
	GS [21]	40.100	9.500	20.08%
Supervised method	TE + WF [18]	37.762	8.004	8.42%
Heuristics	Topline [25]	56.120	23.682	100%
	Baseline-first [25]	41.118	10.362	25.16%
	Baseline-first-document	36.739	8.042	3.31%
	Baseline-random [25]	36.074	6.308	0%
	Lead-baseline [25]	34.716	6.497	−6.77%

4 Conclusions

In this paper, we proposed the method for AMDTS based on GA. The fitness function was calculated considered sentence position and coverage. We proposed the binary coding representation, selection, crossover and mutation operators. Four different tasks for each of the 59 collection of documents of DUC02 data set (specifically AMDST task) were tested. We tested different configurations of the most used methodology to generate AMDST summaries. Moreover, different heuristics such as topline, baseline, baseline-random and lead baseline were calculated. The proposed method for AMDTS demonstrates the improvement over the state-of-art methods and heuristics.

As future work we will use more language-independent features as redundancy reduction, sentence length and similarity with the title [28]. Also, we will consider other text models like sn-grams [27] and MFS [7], and other language [24].

References

1. Bakkar, H., et al.: Multi-document summarizer (2018)
2. Cao, Z., Wei, F., Dong, L., Li, S., Zhou, M.: Ranking with recursive neural networks and its application to multi-document summarization, vol. 7 (2015)
3. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: SIGIR 1998, pp. 335–336 ACM Press, New York (1998)
4. Das, D., Martins, A.F.T.: A survey on automatic text summarization (2007)
5. Du, K.L., Swamy, M.N.S.: Search and optimization by metaheuristics: techniques and algorithms inspired by nature (2016)
6. Ferreira, R., et al.: A multi-document summarization system based on statistics and linguistic treatment. Expert Syst. Appl. **41**(13), 5780–5787 (2014)
7. García-Hernández, R.A., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A.: A new algorithm for fast discovery of maximal sequential patterns in a document collection. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 514–523. Springer, Heidelberg (2006). https://doi.org/10.1007/11671299_53
8. García-Hernández, R.A., Ledeneva, Y.: Single extractive text summarization based on a genetic algorithm. In: Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Rodríguez, J.S., di Baja, G.S. (eds.) MCPR 2013. LNCS, vol. 7914, pp. 374–383. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38989-4_38
9. Kaushik, A., Naithani, S.: A comprehensive study of text mining approach (2016)
10. Kumar Bharti, S., et al.: Automatic keyword extraction for text summarization in multi-document e-newspapers articles (2017)
11. Ledeneva, Y., García-Hernández, R., Gelbukh, A.: Multi-document summarization using maximal frequent sequences, vol. 47, pp. 15–24 (2010). ISSN 1870-4069
12. Ledeneva, Y., et al.: Experimenting with maximal frequent sequences for multi-document summarization, vol. 45, pp. 233–244 (2010). ISSN 1870-4069
13. Ledeneva, Y., 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). https://doi.org/10.1007/978-3-540-78135-6_51

14. Ledeneva, Y.N., García-Hernández, R.A.: Generación automática de resúmenes - Retos, propuestas y experimentos (2017)
15. Ledeneva, Y.N., Gelbukh, A.: Automatic Language-Independent Detection of Multiword Descriptions for Text Summarization. Instituto Politécnico Nacional (2013)
16. Lin, C.-Y.: ROUGE: a package for automatic evaluation of summaries, vol. 34, no. 12, pp. 1213–1220 (2011)
17. Lin, H., Bilmes, J.: Multi-document summarization via budgeted maximization of submodular functions, 912–920 (2010)
18. Lloret, E., et al.: Incorporating textual entailment recognition in single-and multi-document summarization systems (2008)
19. Mani, I., Bloedorn, E.: Multi-document summarization by graph search and matching (1997)
20. Matías, M.G.A.: Generación Automática De Resúmenes Usando Algoritmos Genéticos. Universidad Autónoma del Estado de México (2013)
21. McDonald, R.: A study of global inference algorithms in multi-document summarization. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 557–564. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71496-5_51
22. Nayeem, M.T., Chali, Y.: Extract with order for coherent multi-document summarization (2017)
23. Over, P., Dang, H.: DUC in context. Inf. Process. Manag. **43**(6), 1506–1520 (2007)
24. Rojas-Simón, J., Ledeneva, Y., García-Hernández, R.A.: Calculating the upper bounds for portuguese automatic text summarization using genetic algorithm. In: Simari, G.R., Fermé, E., Gutiérrez Segura, F., Rodríguez Melquiades, J.A. (eds.) IBERAMIA 2018. LNCS (LNAI), vol. 11238, pp. 442–454. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03928-8_36
25. Rojas Simón, J., et al.: Calculating the upper bounds for multi-document summarization using genetic algorithms. Comput. Sist. **22**, 1 (2018)
26. Saggion, H., Poibeau, T.: Automatic text summarization: past, present and future. In: Poibeau, T., Saggion, H., Piskorski, J., Yangarber, R. (eds.) Multi-source, Multilingual Information Extraction and Summarization, pp. 3–21. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-28569-1_1
27. Sidorov, G.: N-gramas sintácticos no-continuos. Polibits **48**, 69–78 (2013)
28. Vázquez, E., et al.: Sentence features relevance for extractive text summarization using genetic algorithms. J. Intell. Fuzzy Syst. **35**(1), 353–365 (2018)
29. Wang, D., et al.: Multi-document summarization using sentence-based topic models. In: ACL and AFNLP, p. 297 (2010)