Combining a Double Clustering Approach with Sentence Simplification to Produce Highly Informative Multi-document Summaries

Sara Botelho Silveira and António Branco
University of Lisbon
Edifício C6, Departamento de Informática
Faculdade de Ciências, Universidade de Lisboa
Campo Grande, 1749-016 Lisboa, Portugal
{sara.silveira, antonio.branco}@di.fc.ul.pt

Abstract

This paper presents a method for extractive multi-document summarization that explores a two-phase clustering approach that, combined with a sentence simplification procedure, aims to generate more useful summaries. First, sentences are clustered by similarity, and one sentence per cluster is selected, to reduce redundancy. Then, in order to group them according to topics, those sentences are clustered considering the collection of keywords. Finally, the summarization process includes a sentence simplification step, which aims not only to create simpler and more incisive sentences, but also to make room for the inclusion of further relevant content in the summary. Evaluation reveals that the approach pursued produces highly informative summaries, containing relevant data and no repeated information.

1 Introduction

Automatic text summarization is the process of creating a summary from one or more input texts through a computer program. It seeks to combine several goals: (1) the preservation of the idea expressed in the input texts; (2) the selection of the most relevant content; (3) the reduction of eventual redundancy; and (4) the organization of the final summary. While meeting these demands, it must ensure that the final summary complies with the desired compression rate. This is, thus, a complex task to be accomplished by a human, let alone a computer.

The most common automatic summarization approaches are typically enclosed in two categories: shallow or deep. Shallow approaches use statistical methods to perform summarization, while deep ones rely on formal and linguistic theories that aim to create the content that defines the summary. However, both shallow and deep techniques can be combined, resulting in a hybrid approach, which merges the best strategies from each methodology in order to improve the system results.

This paper presents a multi-document summarization system, SIMBA, that uses a hybrid approach. In our approach, the summary content is identified using statistical techniques, which compute sentence relevance based on text elements. Furthermore, a sentence simplification procedure, which relies on linguistic knowledge, is performed. It creates simplified sentences, containing as much information as needed, that will afterwards compose the summary.

The main goals of multi-document summarization are tackled through a double clustering approach, which includes a similarity clustering phase and a keyword clustering phase. Redundancy is addressed by clustering all the sentences based on a measure of similarity. Afterwards, the sentences are assembled by topics, using the keywords retrieved from the collection of texts. This approach impacts on the content of the summary. On the one hand, the similarity clustering ensures that this content is not repetitive. On the other hand, keyword clustering assures the selection of the most relevant content, the preservation of the idea of the input texts, and the organization of the final summary.

In order to produce highly informative summaries, the summarization process includes a simplification step, at the end of its processing pipeline. Text simplification aims at clarifying natural language texts, by simplifying its sentences structurally, into shorter and simpler ones, while preserving at the same time the meaning and information these sentences contain. Text simplification may also ease the comprehension of the text by humans. Three main reasons lead us to decide to apply simplification after summarization: (1) this way we ensure that every feature regarding the summarization procedure has been computed before the simplification process modifies the sentences; (2) the sim-

IEEE IRI 2012, August 8-10, 2012, Las Vegas, Nevada, USA

plification algorithm described, which takes into account the aforementioned features of the sentence, does not remove the identified structures without constraints, since we aim to ensure that no crucial content is deleted from the sentence; (3) during the compression process, it is possible to add more relevant content to the summary that was not being considered in the initial sentence candidates list. Our simplification procedure creates simpler and more incisive sentences that can improve the summary readability.

This paper is organized as follows: Section 2 provides an overview on both summarization and simplification approaches; Section 3 describes SIMBA; Section 4 reports the system evaluation; and, in Section 5, conclusions are drawn.

2 Related Work

Hybrid approaches combine statistical features with linguistic knowledge to optimize the generated summaries. Statistical features are typically mapped into measures of significance of the sentence, calculated over the texts to determine a score for each word.

In fact, the most used score measure has been *tf-idf* (Term Frequency × Inverse Document Frequency), where the salience of a term in a document is related to the number of documents in which the term occurs.

Vocabulary overlap measures are also used to define sentence scores. The Dice coefficient [7], the Jaccard index [9], and the Cosine similarity coefficient compute a similarity metric between pairs of sentences, determining a relation between them. For instance, in [15], along with other features, to define the sentence score, the overlap is computed between each sentence and the first sentence of the text, which, in news articles, is considered one of the most important sentences. Yet, in [21], the authors compute the overlap between each sentence and the document title, in order to reward the sentences that have a high degree of similarity with it.

Other works created new metrics to address specific multi-document challenges, as redundancy removal. The Maximal Marginal Relevance (MMR) [4] is a linear combination metric that relates query-relevance with information-novelty and strives to reduce redundancy while considering query relevance to select the appropriate passages to be part of the summary.

Once all the sentences in the texts have been scored, they are ordered based on those scores. The ones that will compose the final summary are then selected, seeking to fulfill the requested compression rate.

Concerning text simplification, automatic systems usually use linguistic knowledge to analyze the sentences. A representation of the sentence structure is created and afterwards simplification rules are applied over that structure. This way, it is possible to create shorter and simpler sentences, while preserving their meaning and the information they convey.

The very first works that addressed simplification ([5] and [10]) have identified several types of structures, which are afterwards removed based on rules that were induced using an annotated aligned corpus of complex and simplified texts. These structures include for instance passages delimited by punctuation marks, subordination and coordinating conjunctions, relative pronouns, and boundaries of clauses and phrases.

In [11], simplification is applied to a single-document summarizer. Operations, derived from an analysis of human written abstracts that remove inessential phrases from the extracted sentences, are performed. In a preprocessing phase of a multi-document summarization process [2], appositives and relative clauses are removed. Yet, in [6], a HMM sentence selection approach is combined with a simplification method that uses shallow parsing to detect lexical cues that trigger phrase eliminations.

In [16], simplification is used to improve content selection, that is, before extracting sentences to be summarized. Parentheticals are simplified by removing relative clauses and appositives to improve sentence clustering.

Sentence compression techniques were also applied to multi-document summarization, by using a parse-and-trim approach [22]. Grammatical constituents are iteratively removed from the sentence parse tree, using linguistic rules, to produce a headline. These rules come from a study which compared the relative prevalence of certain constructions in human-written summaries and lead sentences in stories. The rules include the replacement of temporal expressions, preposed adjuncts, determiners, conjunctions, modal verbs and the selection of specific phenomena in the parse tree.

More recently, a word graph method was used to create a single simplified sentence of a cluster of similar or related sentences [8]. Considering all the words in these related sentences, a directed word graph is built by linking word A to word B through an adjacency relation, in order to avoid redundancy.

As described above, unlike our approach, most of the works that combined summarization with simplification have performed simplification before summarization.

3 The SIMBA system

SIMBA is an extractive multi-document summarizer for the Portuguese language. It receives a collection of Portuguese texts, from any domain, and produces informative summaries, for a generic audience. The length of the summaries is determined by a compression rate value that is submitted by the user. Summarization is performed by executing three phases described in the following sections: annotation, content selection and summary generation.

3.1 Annotation

Firstly, the input texts are annotated, using a set of shallow processing tools for Portuguese, LX-Suite [3]. Sentence and paragraph boundaries are identified and words are tagged, with its corresponding part-of-speech and lemma (non-inflected form).

Henceforth, the collection of texts is handled as a set of sentences.

3.2 Content selection

This stage identifies relevant information in the collection of texts. In the first step, sentence scores are computed. The second step consists of a double-clustering approach (described in detail in [19]), that firstly clusters sentences by similarity to remove redundancy and, then, clusters sentences by keywords to identify the most significant data in the collection.

It is important to note here that the summarization process includes three types of scores: the main score, the extra score and the complete score. The main score reflects the sentence relevance in the overall collection of sentences. The extra score is used in the summarization process to reward or penalize the sentences, by adding or removing predefined score values.[1] The complete score is the sum of these two scores.

The main score is the one that defines sentence order in the clustering phases. The extra score is used to assign relevance to sentences during those clustering phases, in order for the decisions made during the summarization process to have impact on sentence relevance. For instance, a sentence that has been clustered by keywords must be rewarded since this means that it is significant to the idea conveyed by the collection of texts. These clustering phases are the core of the summarization algorithm, since they define which sentences proceed to the next phases. Depending on those decisions in the clustering phases, sentences are rewarded or penalized using the extra score. At the end of the summarization pipeline, sentences are ordered considering the complete score, which combines the two previous mentioned scores. This way, the extra score impacts on the order of the sentences that will compose the summary, contributing to a better selection of its sentences.

Computing sentence main score. Once the sentences and the words have been identified, *tf-idf* score is computed for each word, by considering its lemma. The sentence main score is, then, the average of the *tf-idf* scores of its words.

Clustering sentences by similarity. In order to identify redundant sentences, the next step aims at clustering sentences by their degree of similarity.

The similarity between two sentences comprises two dimensions, computed considering the word lemmas: the sentences subsequences and the word overlap.

The subsequences value is inspired in ROUGE-L and consists in the sum of the number of words in all the subsequences common to each sentence, divided by the total number of words of each sentence being considered, and by the total number of subsequences found between the two sentences. The overlap value is computed using the Jaccard index [9].

The similarity value is the average of both these values: the overlap and the subsequences value. It is then confronted with a predefined similarity threshold[2] –, initially set to 0.75, determining that sentences must have at least 75% of common words or subsequences to be considered as conveying the same information.

Afterwards, the sentences are actually clustered considering their similarity value. A cluster is composed by a collection of sentences, a similarity value, and a representative sentence. In our clustering context, the representative sentence is not the sentence which is the closest to all the sentences in the cluster, instead it is the sentence with the highest main score.

The algorithm starts with an empty set of clusters. All sentences in the collection of texts are considered. The first sentence of the collection creates the first cluster. Then, each sentence in the collection of sentences is compared with the sentences already clustered. For each cluster, the similarity value is computed between the current sentence being compared and all the sentences in the collection of sentences of each cluster. The similarity value considered is the highest between the current sentence and all the sentences in the collection of sentences of the current cluster. Then, if the similarity value is higher than the similarity threshold, the sentence will be added to this cluster.

When a sentence is added to a cluster, its representative is updated. If the score of the sentence being added is higher than the representative one, the newly added sentence becomes the representative of the cluster. Also, each representative is given an extra score value (0.1), which is subtracted from the sentences which are replaced as representatives.

Finally, if all the clusters have been considered, and the sentence was not added to any cluster, a new cluster with this sentence is created, meaning that this sentence does not repeat information previously considered.

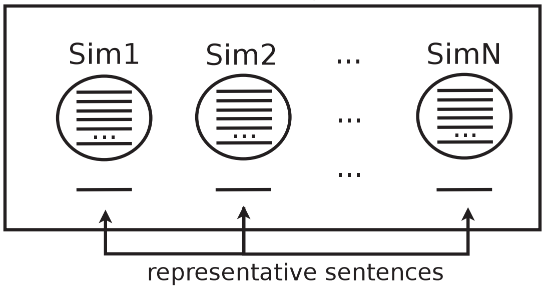
Once the procedure is finished, sentences with similar information are grouped in the same cluster and the one with

[1] The predefined value to be added to the extra scores is set to 0.1, both for the reward and for the penalty value. This value has been determined empirically, through a set of experiments.

[2] This threshold was determined empirically, using a set of experiments, since there is no reference for such a value for the Portuguese language.

the highest score is the representative sentence of all the sentences in the cluster. The similarity clustering process is depicted in Figure 1.

Figure 1. Similarity clustering



Redundant sentences are thus ignored, and a new collection of sentences is built by selecting only the representative of each similarity cluster. This collection is the input of the next phase of the summarization process.

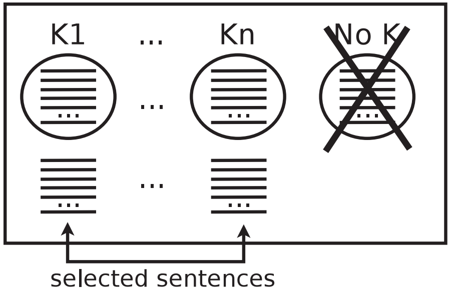
Clustering sentences by keywords. SIMBA produces a generic summary. Thus, the keywords that represent the global topic within the collection of texts are identified. The candidate keywords list contains common and proper names. It is built considering the lemma of the words, to ensure that the words in the collection are unique. Thereafter, the list is ordered considering the score of each word.

We define k, the number of keywords, as $k = \sqrt{\frac{N}{2}}$, where N is the total number of words in the collection of documents. The final list of keywords contains the first k words of the list of candidate keywords. Sentences are clustered based on these keywords.

In this phase, a cluster is identified by a keyword (the topic), and contains a representative sentence, and a collection of values (the sentences related to the keyword). The algorithm that clusters sentences by keywords is an adapted version of the K-means algorithm [13]. Firstly, each keyword k defines a cluster. Then, a sentence is added to the cluster that is represented by the keyword that occurs more often in the sentence. The cluster representative is recomputed if the current sentence has a higher score than the previous representative. Also, the current representative is rewarded with an extra score value and the previous representative is penalized by removing an extra score value. If the sentence does not contain any keywords, it is added to a specific set of sentences which do not have any keyword (”no-keyword” set). The set of keywords is recomputed if all the sentences have been considered, or if the ”no-keyword” set contains new sentences. This algorithm is repeated while the ”no-keyword” set changes between consecutive iterations.

Finally, an extra score is also assigned to the sentences in the clusters that represent the first set of keywords. These sentences are considered more significant than the others, since they address the main topics conveyed by the collection of texts. Still, sentences in the ”no-keyword” set are ignored, since they do not convey relevant information concerning the overall collection of texts, as shown in Figure 2.

Figure 2. Keyword clustering



Afterwards, only the sentences that have indeed been clustered by keywords are considered. The ones that have less than ten[3] words are penalized and an extra score value is subtracted from their extra score. The sentences that have more than ten words are assigned with an extra score value.

In addition, sentences are ordered based on their complete score. The complete score, defined in Equation 1, is used to rank all the sentences, defining the order in which the sentences are chosen to be part of the final summary.

$$completeScore_s = \frac{\sum_{t \in s} \textit{tf-idf}_t}{totalWords_s} + extraScore_s \quad (1)$$

3.3 Summary generation

This phase aims at creating the final summary delivered by SIMBA.

Sentences are simplified to reduce the original content, producing a summary composed by simpler and more informative sentences. In this approach, simplification is a form of compression. Each simplification step aims to compress the sentence by removing from it parts considered dispensable, that is, parts of the sentence that may add less relevant information to the general message of the sentence.

The first step of the algorithm is compression. The list ordered in the previous phase is used to create a new list containing only the sentences that add up to the maximum number of words allowed by the compression rate.

Afterwards, sentences are simplified. For each sentence, a parse tree is built using a constituency parser for Portuguese [17]. Subtrees that represent specific sentential

[3] As a rule of thumb, sentences with less than ten words are typically considered to have poor content.

structures are then identified in that parse tree. The simplification process removes these structures by replacing their subtrees by a null tree.

This process is executed in two steps: main phrase selection and phrase compression. The first step obtains the main phrase of the sentence, removing any additional information found. The original sentence is replaced by the simplified one, in which further simplification rules are applied. The phrase compression step deals with specific structures that contain explanatory information about the content already mentioned in the sentence. Five types of structures are targeted:

- Appositions – noun phrases that describe, detail or modify its antecedent (also a noun phrase);
- Adjectives;
- Adverbs or adverb phrases;
- Parentheticals – phrases that explain or qualify other information being expressed;
- Relative clauses – clauses that modify a noun phrase, which are introduced by a relative pronoun.

An example of a simplified sentence and its corresponding original sentence, from which an adverb phrase was removed, is shown below:

> ORIGINAL SENTENCE:
> *José Sócrates chegou* ***um pouco*** *atrasado ao debate.*
> José Sócrates arrived **a little late** to the debate.
>
> SIMPLIFIED SENTENCE:
> *José Sócrates chegou atrasado ao debate.*
> José Sócrates arrived late to the debate.

In this example, the adverb phrase does not add significant content to the sentence, thus it can be removed. Considering that in a set of sentences there are several phrases that can be removed, the rationale behind simplification is that it makes room for further information to enter the summary, creating a more informative text.

Thus, as the simplification process removes words from the already selected set of sentences, more sentences are added to this set in order to achieve the desired number of words again. These two steps, compression and simplification, are repeated until no more new sentences are added to the set of simplified sentences.

The simplification module [20] returns a collection of simplified sentences that define the final summary, ensuring that it is a concise text, focused in the most important information conveyed by the collection of texts.

4 Evaluation

Evaluation has been performed in four different ways. First, SIMBA was evaluated concerning the whole summarization process, by comparing its summaries with the ones generated using GISTSUMM [14], a summarizer built to deal with texts in Portuguese. It is based on the notion of gist, which is the most important passage of the text, conveyed by just one sentence, the one that best expresses the text's main topic. The system algorithm relies on this sentence to produce extracts. GISTSUMM is the only summarizer for Portuguese available on-line. Despite having been built to produce summaries from a single-document, it also performs multi-document summarization by means of an option in its interface that allows it to produce a summary from a collection of texts. GISTSUMM is then used as a baseline for the summarization process.

Secondly, the summary generation phase, which includes the simplification and compression steps, has also been evaluated. SIMBA summaries were compared to a simplification baseline. This simplification baseline is the very first version of our simplification module [18]. Once the candidate sentences have been ordered, they are simplified and then compressed to determine which ones will define the final summary. In this version of the simplification process, three types of structures are considered: relative clauses, apposition phrases and parenthetical phrases. These structures are removed from a sentence if the score of the simplified sentence, resulting from the removal of those structures, is higher than the one of the non-simplified sentence. Otherwise, the sentence is kept unmodified. Then, the set of sentences resulting from the simplification process is compressed. This is a very simple algorithm, different from the one detailed above, that can be considered as a baseline for simplification.

Thirdly, two versions of summaries built by SIMBA, simplified and non-simplified versions, were compared to infer the impact of simplification in the final summaries.

Finally, the simplification order was also evaluated. Summaries were generated using another version of SIMBA, in which simplification was performed before the double-clustering step, consisting in the inverse architecture of the approach suggested, to infer whether our claim that performing simplification after sentence selection produces indeed better summaries.

Since SIMBA was built specifically to deal with Portuguese texts, the *CSTNews* corpora [1], an annotated corpus composed of texts in Portuguese, was used. It contains 50 sets of news texts from several domains, for a total of 140 documents, 2,247 sentences, and 47,350 words. Each set contains, on average, 3 documents which address the same subject. The texts were retrieved from five Brazilian on-line newspapers. Also, each set of texts contains a manually built summary – the so-called ideal summary. There are 50 ideal summaries, containing an average of 137 words, resulting in an average compression rate of 85%.

Thus, all the generated summaries have a compression rate of 85%, meaning that the summary contains 15% of the words contained in the set of texts. Using the same compression rate of the ideal summaries allows a more accurate and fairer comparison between the ideal summaries and each of the automatically generated summaries.

After the summaries have been built, they were compared with the ideal summaries in the corpus using ROUGE [12]. In fact, a more precise metric of ROUGE was used, ROUGE-L (longest common subsequence), since it identifies the common subsequences between two sequences. As the simplification process introduces gaps in the extracted sentences, ROUGE-L is a fairer metric, once it does not require consecutive matches between the sentences.

Hence, four results will be presented. The first evaluates SIMBA summarization process over the summarization baseline (GISTSUMM). The second evaluates the simplification process considering the simplification baseline. The third evaluates the complete summarization process with and without simplification, to assess the actual gain that simplification brings to summarization. The fourth evaluates the arrangement of the double clustering approach along with the simplification module.

Summarization. Summaries for the corpora *CSTNews* were generated automatically using GISTSUMM and SIMBA. ROUGE-L metrics comparing these summaries with the ideal ones were computed. Results are shown in Table 1.

Table 1. Summarization evaluation

	GISTSUMM	SIMBA
Precision	0.4339	0.4833
Recall	0.3823	0.5372
F-measure	0.4012	0.5051

As shown in Table 1, SIMBA outranks the baseline when considering all the metrics. The f-measure value means that SIMBA summaries contain much of the information contained in the ideal summaries.

The difference between GISTSUMM and SIMBA f-measure values is of ten percentage points, which is a considerable difference. It means that SIMBA produces summaries containing more relevant information than GISTSUMM. Both SIMBA's precision and recall values also overcome the GISTUMM values. The recall value attained by SIMBA is very interesting. It means that the information contained in SIMBA summaries is very accurate, the summaries include the most important information conveyed by the ideal summaries, preserving, this way, the idea conveyed by the collection of texts. Yet, the shorter difference between the precision values can be justified by the simplification process. By removing information from the sentences, despite being less relevant, SIMBA summaries may have less in-sequence matches than would likely to be found. Still, SIMBA summaries have an higher precision value than the one by GISTSUMM, meaning that SIMBA summaries cover more topics mentioned in the input texts. Thus, the summaries produced by SIMBA are clearly better than the ones produced by GISTSUMM.

Simplification. Afterwards, SIMBA was compared with the simplification baseline. ROUGE-L metrics were also computed. Results are described in Table 2.

Table 2. Simplification evaluation

	Baseline	SIMBA
Precision	0.4960	0.4833
Recall	0.4387	0.5372
F-measure	0.4606	0.5051

The main difference between these summaries lies in the simplification algorithm, as the summarization process is the same. Thus, all the baseline values were expected to increase. However, the values being compared are still distant. SIMBA summaries have a better overall performance when compared to the summaries produced using the simplification baseline. All the values of the baseline are five percentage points lower than the ones achieved by SIMBA. This is a direct consequence of a more sophisticated algorithm performed by the current version of SIMBA. As the two systems have the same summarization procedure, the simplification process makes the difference by creating more informative summaries, containing more relevant information. SIMBA's results overcome the baseline, mainly through the usage of more sophisticated rules that remove content that is not crucial, making room for the addition of new relevant information that helps to refer more of the significant topics conveyed by the input texts. Considering the results expressed in Table 2, we can conclude that our algorithm can indeed help produce better summaries.

Simplification vs. Non-simplification. Then, two different summaries were built by SIMBA: simplified and non-simplified summaries. The simplified summaries have been built by performing the complete summarization process, including the simplification process. The non-simplified summaries have been built by performing the summarization process without the simplification module. The results that illustrate the differences between both SIMBA summaries are shown in Table 3.

Table 3. Simplification vs. Non-simplification evaluation

	Non-simplified	Simplified
Precision	0.4876	0.4833
Recall	0.5158	0.5372
F-measure	0.4955	0.5051

In what concerns the comparison between summaries produced by SIMBA either with or without simplification, the claim that simplification helps to improve summarization can be confirmed.

The recall values obtained by SIMBA are very encouraging. These values indicate that there is a higher number of words that are both in the automatic summaries and in the ideal summaries. Retrieving the most relevant information in a sentence by discarding the less significant data ensures that the summary indeed contains the most important information conveyed. This is a direct result of the simplification process, as when compared to non-simplified summaries, the simplified ones still have a better performance.

The precision values of the two types of summaries are closer than the ones concerning recall. Intuitively, the precision values should be similar or even decrease. In fact, this is what indeed occurs, as in comparison to the ideal summary, less in-sequence matches are found in simplified summaries due to the simplification process.

Still, the f-measure value, by combining both precision and recall, evidences that the simplified summaries include sentences that contain much of the information present in the ideal summaries, resulting in more informative texts.

Clustering and simplification. Finally, two types of summaries were created using two inverse approaches. In the ”before clustering” approach, simplification is performed before both clustering phases are performed. The ”after clustering” approach is the one suggested in this paper, that is, simplification is performed after both clustering phases have been performed, after the sentences have been ordered and while the selection of the sentences that will figure in the summaries is made. Table 4 shows the results for these two approaches.

Table 4. SIMBA evaluation when simplifying before and after the clustering phases

	Before Clustering	After Clustering
Precision	0.4902	0.4833
Recall	0.5035	0.5372
F-measure	0.4901	0.5051

Results shown in Table 4 demonstrate that simplifying sentences before clustering does not improve the final summaries. The main reason for this is that when a sentence is simplified, its main score changes. This impacts in the similarity clustering phase, since the representative sentence is the sentence with the highest score in the cluster. Those are the sentences that are kept in the summarization process, so that it is important that the best ones are selected. In addition, changing the sentence main score can also impact on the ordering phase. The ordered list of sentences can be different if the sentences have been previously simplified, since sentences with less words can have lowest scores, depending on the structures that have been simplified.

Moreover, simplification is a computationally expensive task. If this task is performed before clustering, linguistic information must be obtained for all the sentences submitted, increasing the system processing time.

These facts point out that simplification should not be performed before selecting and ordering the summary sentences. Other way, relevant data can be missed when comparing the sentences during the double-clustering procedure, and we can not ensure that the most significant sentences in the collection of sentences are indeed in the final summary.

Thus, by performing simplification at the end of the processing pipeline, we are able to ensure that all the information submitted is considered in the selection process, and that no relevant data is missed.

5 Concluding remarks

The results reported in this paper show that the quality of an automatic summary can be improved by (1) performing specific multi-document tasks – such as removing the redundant information, or considering all the texts in each set as a single information source; and (2) executing an algorithm that seeks to optimize the content selection, combined with a simplification process that removes less relevant content and makes room for more relevant information.

The multi-document summarizer presented relies on statistical features to perform summarization of a collection of texts in Portuguese, using a shallow yet accurate approach. A double clustering approach combined with a sentence simplification procedure has indeed proven to produce better summaries.

There are two points yet to consider. On the one hand, the order of the sentences in the final summary is an open issue, whose impact has to be assessed through a human evaluation. On the other hand, the simplification process can be improved. An algorithm that tries several combinations of removing the targeted structures in order to maximize the simplified sentence score is being tested.

Despite this, the final automatic evaluation shows very promising results. SIMBA's results overcome all baseline results and also its own results for non-simplified summaries. Both f-measure and recall values are very encouraging, since they reflect the high relevance of the sentences present in the summaries produced by SIMBA. In addition, the high recall obtained determines that the information in the summary is in fact relevant and that the simplification process impacts positively on the informativeness of the final summary. Thus, the results obtained point out that SIMBA summaries preserve the idea of the original collection of texts, and contain, at the same time, incisive and simple sentences, conveying the most significant information covered in the input texts.

References

[1] P. Aleixo and T. A. S. Pardo. CSTNews: Um córpus de textos jornalísticos anotados segundo a teoria discursiva multidocumento CST (cross-document structure theory). Technical report, Universidade de São Paulo, 2008.

[2] S. Blair-Goldensohn, D. Evans, V. Hatzivassiloglou, K. Mckeown, A. Nenkova, R. Passonneau, B. Schiffman, A. Schlaikjer, Advaith, A. Siddharthan, and S. Siegelman. Columbia university at duc 2004. In *Proceedings of the 2004 document understanding conference (DUC 2004)*, HLT/NAACL 2004, pages 23–30, Boston, Massachusetts, 2004.

[3] A. Branco and J. Silva. A suite of shallow processing tools for portuguese: Lx-suite. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'06)*, 2006.

[4] J. G. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Research and Development in Information Retrieval*, pages 335–336, 1998.

[5] R. Chandrasekar, C. Doran, and B. Srinivas. Motivations and methods for text simplification. In *In Proceedings of the Sixteenth International Conference on Computational Linguistics (COLING '96)*, pages 1041–1044, 1996.

[6] J. Conroy, J. Schlesinger, and J. Stewart. Classy query-based multidocument summarization. In *Proceedings of 2005 Document Understanding Conference*, Vancouver, BC, 2005.

[7] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[8] K. Filippova. Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 322–330, Stroudsburg, PA, USA, 2010. ACL.

[9] P. Jaccard. Nouvelles recherches sur la distribution florale. *Bulletin de la Sociète Vaudense des Sciences Naturelles*, 44:223–270, 1908.

[10] H. Jing. Sentence reduction for automatic text summarization. In *Proceedings of the sixth conference on Applied natural language processing*, pages 310–315, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[11] H. Jing and K. R. McKeown. Cut and paste based text summarization. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, NAACL 2000, pages 178–185, Stroudsburg, PA, USA, 2000. ACL.

[12] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. ACL.

[13] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[14] T. A. S. Pardo, L. H. M. Rino, and M. das Graças V. Nunes. Gistsumm: A summarization tool based on a new extractive method. In *PROPOR*, Lecture Notes in Computer Science, pages 210–218. Springer, 2003.

[15] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization*, NAACL-ANLP-AutoSum '00, pages 21–30. ACL, 2000.

[16] A. Siddharthan, A. Nenkova, and K. McKeown. Syntactic simplification for improving content selection in multi-document summarization. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 896, Morristown, NJ, USA, 2004. ACL.

[17] J. Silva, A. Branco, S. Castro, and R. Reis. Out-of-the-box robust parsing of Portuguese. In *Proceedings of the 9th Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR)*, pages 75–85, 2010.

[18] S. B. Silveira and A. Branco. Enhancing multi-document summaries with sentence simplification. In *ICAI 2012: International Conference on Artificial Intelligence*, Las Vegas, USA, July 2012.

[19] S. B. Silveira and A. Branco. Extracting multi-document summaries with a double clustering approach. In *Proceedings of the 17th International Conference on Applications of Natural Language Processing to Information Systems*, Groningen, The Netherlands, 2012. Springer.

[20] S. B. Silveira and A. Branco. Using a double clustering approach to build extractive multi-document summaries. In *TSD 2012: 15th International Conference on Text, Speech and Dialogue*, Brno, Czech Republic, September 2012.

[21] M. White, T. Korelsky, C. Cardie, V. Ng, D. Pierce, and K. Wagstaff. Multidocument summarization via information extraction. In *HLT '01: Proceedings of the first international conference on Human language technology research*, pages 1–7, 2001.

[22] D. Zajic, B. J. Dorr, J. Lin, and R. Schwartz. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Inf. Process. Manage.*, 43(6):1549–1570, 2007.