Using a Double Clustering Approach to Build Extractive Multi-document Summaries

Sara Botelho Silveira and António Branco

University of Lisbon, Portugal
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. 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 that represent the topics in the set of texts. Evaluation reveals that the approach pursued produces highly informative summaries, containing many relevant data and no repeated information.

1 Introduction

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

This paper presents a multi-document summarization system, SIMBA, that receives a collection of texts and returns an extract 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 grouping the sentences based on a similarity measure. Afterwards, the sentences are assembled by topics, using the keywords retrieved from the collection of texts. Furthermore, to support the compression process, this system includes a sentence simplification module, which aims to produce simpler and more incisive sentences, allowing more relevant content to enter the summary. Finally, an automatic evaluation of SIMBA is presented.

Previous works addressed multi-document summarization in different ways. MEAD [1] is a multi-document system that summarizes clusters of news articles automatically grouped by a topic detection system. It uses information from the centroids of the clusters to select the sentences that are most likely to be relevant to the cluster topic. By identifying similarities and important differences across sets of documents, Newsblaster [2] builds summaries from on-line news sources. It performs summarization through different modules that use different strategies depending on the type of

P. Sojka et al. (Eds.): TSD 2012, LNCS 7499, pp. 298–305, 2012.

the documents in the input set. In order to produce extracts from a set of documents, NeATS [3] selects relevant portions about a topic and presents them in coherent order, using several metrics: term frequency, sentence position, stigma words and maximum marginal relevance. Concerning the Portuguese language, GistSumm [4] was the first single-document summarizer built. It is based on the notion of gist, which is the most important passage of the text, conveyed by just one sentence that best expresses the text's main topic. The system algorithm relies on this sentence to produce extracts. GistSumm is available on-line and though it has been built to produce summaries from a single-document, it also performs multi-document summarization by means of an option in its interface. In their work, [5] treat multi-document summarization as a classification problem, by combining features, as sentence position and sentence size, with sophisticated linguistic features, given by the CST model, such as semantic relations between sentences from different texts.

Henceforth, this paper is organized as follows: Section 2 describes our system, Section 3 reports system evaluation, and in Section 4 some conclusions are drawn.

2 The SIMBA System

SIMBA is an extractive multi-document summarizer for the Portuguese language, that receives a collection of texts, from any domain, and produces informative summaries, for a generic audience. Summarization is performed by means of two main phases executed in sequence: clustering by similarity and clustering by keywords. The length of the summaries is determined by a compression rate value that is submitted by the user.

2.1 Methodology

The procedure starts by processing automatically the documents submitted to be summarized. A set of shallow processing tools for Portuguese, LX-Suite [6], is used to annotate the texts. Sentence and paragraph boundaries are identified and words are tagged, with its corresponding POS and lemmata. Also, a parse tree representing each sentence syntactic structure is built, using LX-Parser [7]. Henceforth, the collection of texts is handled as a set of sentences.

Afterwards, sentence scores are computed. The scoring procedure includes two scores, the main score and the extra score, that are combined to define the sentence final score – named complete score. The main score reflects the sentence relevance in the overall collection of sentences, and is the sum of the `tf-idf` score (computed considering the word lemma) of each word of the sentence, smoothed by the number of words in the sentence. The extra score is used during the clustering phases 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 next stages of processing aim to identify relevant information in the collection of texts in two steps: similarity clustering and keyword clustering.

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

Similarity clustering. In order to identify sentences conveying the same information, they are clustered considering their degree of similarity.

The similarity between two sentences (Equation 3) comprises two dimensions, computed considering the word lemmas: the sentences subsequences (Equation 1) and the word overlap (Equation 2).

$$subsequences(s_1, s_2) = \frac{\sum_i \left(\frac{\text{subsequence}_i}{\text{totalWords}_{s_1}} + \frac{\text{subsequence}_i}{\text{totalWords}_{s_2}} \right)}{totalSubsequences} \tag{1}$$

$$overlap(s_1, s_2) = \frac{\sum \text{commonWords}(s_1, s_2)}{\text{totalWords}_{s_1} + \text{totalWords}_{s_2} - \sum \text{commonWords}(s_1, s_2)} \tag{2}$$

$$similarity(s_1, s_2) = \frac{\text{subsequences}(s_1, s_2) + \text{overlap}(s_1, s_2)}{2} \tag{3}$$

$\text{overlap}(s_1, s_2)$ – number of overlapping words between the two sentences.
$\text{subsequence}(s_1, s_2)$ – number of overlapping words in the subsequences between the two sentences.
$\text{commonWords}(s_1, s_2)$ – common words between the two sentences.
totalWords_{s_i} – total words in the sentence i.
subsequence_i – number of words of the subsequence i.
totalSubsequences – number of subsequences between the two sentences.

The subsequence value is inspired in ROUGE-L and consists of the sum of the number of words in all the subsequences common to each sentence, smoothed by the total number of words of each sentence being considered, and divided by the total number of subsequences found between the two sentences. The overlap value is computed using the Jaccard index [8].

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

Two examples are discussed below. Taking into account this threshold, the sentences in the following example are considered to be similar.

Sentence#1:	**Sentence#2:**
A casa que os Maias vieram habitar em Lisboa, no outono de 1875, era conhecida pela casa do Ramalhete.	A casa que os Maias vieram habitar, no outono de 1875, era conhecida pela casa do Ramalhete.
The house in Lisbon to which the Maias moved in the autumn of 1875, was known as the Casa do Ramalhete.	*The house to which the Maias moved in the autumn of 1875, was known as the Casa do Ramalhete.*

Overlap	Subsequences	Similarity Value
0.89	0.95	0.92

These sentences share most of the words, but there is a leap (*"em Lisboa"*) between Sentence#1 and Sentence#2. Both the overlap and the subsequences values are high,

[2] This threshold was determined empirically, using a set of experiments.

so the similarity value is also high (0.92), and thus the sentences are considered to be similar.

The two sentences in the following example are not similar, despite having many words in common.

Sentence#1:	**Sentence#2:**
A casa que os Maias vieram habitar em Lisboa, no outono de 1875, era conhecida pela casa do Ramalhete.	A casa que os Maias vieram habitar em Lisboa, no outono de 1875, era conhecida na vizinhança da Rua de S. Francisco de Paula, pela casa do Ramalhete ou simplesmente o Ramalhete.
The house in Lisbon to which the Maias moved in the autumn of 1875, was known as the Casa do Ramalhete.	*The house in Lisbon to which the Maias moved in the autumn of 1875, was known in Rua S. Francisco de Paula, as the Casa do Ramalhete or, more simply, as Ramalhete.*

Overlap	Subsequences	Similarity Value
0.59	0.79	0.69

Despite Sentence#1 is being contained in Sentence#2, both sentences are considered not to be similar, since their similarity value is below the threshold.

Afterwards, sentences are clustered considering their similarity value. As the primary goal of this phase is to determine the sentences that represent each cluster, a simple algorithm, that seeks to optimize sytem execution, is used.

A cluster contains a collection of sentences, a similarity value, and a centroid (the highest scored sentence of the collection). 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 the 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. If the similarity value is higher than the similarity threshold, the sentence is added to that cluster.

When a sentence is added to a cluster, its centroid is updated. If the score of this sentence is higher than the centroid one, the newly added sentence becomes the centroid of the cluster, and is rewarded with an extra score. Likewise, an extra score value is subtracted from the sentences which are replaced as centroids.

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 redundant information are grouped in the same cluster and the one with the highest score (the centroid) represents them. So, this phase returns a collection of sentences built by selecting only the centroid of each similarity cluster. The sentences in the collection of sentences, by being redundant, are discarded.

Keyword clustering. Our system produces a generic summary, so it is not focused on a specific matter. Thus, the keywords that represent the global topic within the collection of texts are identified. A list with the candidate keywords is constructed containing

words that are common and proper names, since these words identify ideas or themes. Words are compared considering their lemmas, 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 that final list of keywords.

In this phase, a cluster is identified by a keyword, and contains a centroid (a sentence), and a collection of sentences (related to the keyword). The algorithm that clusters sentences by keywords is an adapted version of the K-means algorithm [9], and follows the steps described below:

1. Choose the number of clusters, k, defined by the number of keywords;
2. Create the initial clusters, represented by each keyword obtained;
3. Consider each sentence:
   (a) Compute the occurrences of each keyword in the sentence;
   (b) Assign the sentence to the cluster whose keyword occurs more often;
4. Recompute the cluster centroid. If the current sentence has more occurrences of the keyword than the previous centroid sentence had, the newly added sentence becomes the cluster centroid;
5. 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);
6. Recompute the set of keywords if:
   (a) All the sentences have been considered;
   (b) The "no-keyword" set contains new sentences.
7. Repeat previous steps (2 – 6) while the "no-keyword" set of sentences remains different in consecutive iterations.

As in the similarity algorithm, when the centroid is changed, the extra scores of the current centroid and of the previous centroid are updated.

In addition, an extra score is also assigned to the sentences in the clusters that represent the initial 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.

The next step of the summarization procedure orders sentences based on their complete score, defining the order of the sentences to be included in the summary. Afterwards, this set of sentences is compressed in order to select the ones composing the summary. Compression is applied in two ways. First, the compression rate given by the user is applied to the collection of sentences. When the total number of words of the sentences already added to the summary reaches or surpasses the maximum compression, no more sentences are selected. Afterwards, a sentence simplification procedure [10] removes, from a sentence, syntactic structures whose removal is less detrimental to the comprehension of the text. Simplification is performed by identifying and removing appositions, parentheticals, and relative clauses from the sentence parse tree. This process seeks to make room for more relevant data to be included in the summary, aiming to produce a more informative text. 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 sentences that defines the summary. Finally, the summary is delivered to the user in the form of a text file.

3 Evaluation

In order to perform evaluation, *CSTNews* [11], an annotated corpus 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 newspapers. Also, each set of texts contains a manually built summary – the ideal summary. There are 50 ideal summaries, containing an average of 137 words, resulting in an average compression rate of 85%.

In order to understand the impact of this approach, Table 1 details the sentences considered before and after the clustering phases have been executed.

Table 1. Sentences involved in the clustering phases

Clustering by similarity:			**Clustering by keywords:**		
Before	After	Difference	Before	After	Difference
2,247	2,115	132	2,115	1,599	516

The similarity clustering is the first step of the summarization process, so that it takes all the sentences in the corpus (2,247). After executing this step, 132 sentences have been considered redundant. This corresponds to 5% of the sentences in the corpora. Thus, these sentences are not considered in the next steps of the summarization procedure. If these sentences have been selected to be part of the final summaries, those would contain many superfluous data that would impact negatively on their informativity.

The double clustering approach executes both phases in sequence. Thus, after the set of sentences has been filtered in the similarity clustering, the remainder of the sentences (2,115) are clustered by keywords. Considering all the document sets, there were 516 sentences that do not mention the main topics of the texts. This corresponds to 24% of the sentences considered in this phase. Therefore, only 76% of the sentences considered in the keywords clustering phase are indeed relevant to the topic. This way, after executing the double clustering procedure, a total of 648 sentences were discarded, either by being redundant or irrelevant.

Concerning the evaluation itself, we compared the summaries generated automatically by SIMBA with summaries produced by GISTSUMM. The compression rate used was the one of the ideal summaries (85%), meaning that the summary contains 15% of the words contained in the set of texts.

Afterwards, the summaries were compared with the ideal summaries 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, this is considered a fairer metric.

Table 2 details precision, recall and F-measure metrics for both summarizers.

Table 2. Multi-document evaluation metrics

	GISTSUMM	SIMBA
Precision	0.43616	0.48534
Recall	0.38469	0.54014
F-measure	0.40398	0.50752

The SIMBA process has an overall better performance than the baseline.

The recall values obtained by SIMBA are very encouraging. These values indicate that there is a high density of words that are both in SIMBA summaries and in the ideal summaries. Retrieving the most relevant information in a sentence by discarding the less relevant data ensures that the summary indeed contains the most important information conveyed.

The precision values of the two systems are closer than the ones concerning recall. Intuitively, the precision values should be similar or even decrease, since, in comparison to the ideal summary, less in-sequence matches are likely to be found in SIMBA summaries due to the simplification process. Still, SIMBA has a higher precision value than GISTSUMM, meaning that its summaries cover more significant topics than the ones produced by GISTSUMM. Both the precision and recall values attained by SIMBA are a direct result of the combination of both clustering phases, along with the simplification process.

Consequently, when computing the F-measure value, by combining both precision and recall, the claim that SIMBA produces better summaries than GISTSUMM can be confirmed.

4 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 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 and allows the addition of more relevant information.

The multi-document summarizer presented relies on statistical features to perform summarization of a collection of texts in Portuguese. Despite the core algorithm being language-independent, this system uses language-specific tools that aim to improve not only the content selection, but also the general quality of a summary produced from a collection of texts written in Portuguese.

The final evaluation demonstrates promising results. 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.

This approach impacts on the content of the final summary in two ways. On the one hand, similarity clustering ensures that the content is not repetitive. On the other hand, keyword clustering insures the selection of relevant content, and the preservation of the idea of the input texts. Thus, the combination of these two clustering phases allows the creation of highly informative summaries.

References

1. Radev, D.R., Jing, H., Budzikowska, M.: 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 2000, pp. 21–30. ACL (2000)
2. McKeown, K.R., Hatzivassiloglou, V., Barzilay, R., Schiffman, B., Evans, D., Teufel, S.: Columbia multi-document summarization: Approach and evaluation. In: Proceedings of the Document Understanding Conference, DUC 2001 (2001)
3. Lin, C.Y., Hovy, E.: From single to multi-document summarization: A prototype system and its evaluation. In: Proceedings of the ACL, pp. 457–464. MIT Press (2002)
4. Pardo, T.A.S., Rino, L.H.M., das Graças Volpe Nunes, M.: GistSumm: A Summarization Tool Based on a New Extractive Method. In: Mamede, N.J., Baptista, J., Trancoso, I., Nunes, M.d.G.V. (eds.) PROPOR 2003. LNCS, vol. 2721, pp. 210–218. Springer, Heidelberg (2003)
5. Jorge, M.L.C., Agostini, V., Pardo, T.A.S.: Multi-document summarization using complex and rich features, pp. 1–12 (July 2011)
6. Branco, A., Silva, J.: 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 2006 (2006)
7. Silva, J., Branco, A., Castro, S., Reis, R.: Out-of-the-box robust parsing of Portuguese. In: PROPOR 2010: Proceedings of the 9th Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada, pp. 75–85 (2010)
8. Jaccard, P.: Nouvelles recherches sur la distribution florale. Bulletin de la Sociète Vaudense des Sciences Naturelles 44, 223–270 (1908)
9. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press (1967)
10. Silveira, S.B., Branco, A.: Enhancing multi-document summaries with sentence simplification. In: ICAI 2012: International Conference on Artificial Intelligence (2012)
11. Aleixo, P., Pardo, T.A.S.: CSTNews: Um córpus de textos jornalísticos anotados segundo a teoria discursiva multidocumento CST (cross-document structure theory). Technical report, Universidade de São Paulo (2008)
12. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out: Proceedings of the ACL 2004 Workshop, Barcelona, Spain, pp. 74–81 (July 2004)