GistSumm: A Summarization Tool Based on a New Extractive Method[1]

Thiago Alexandre Salgueiro Pardo
Lucia Helena Machado Rino
Maria das Graças Volpe Nunes

Núcleo Interinstitucional de Lingüística Computacional (NILC)
CP 668 – ICMC-USP, 13560-970 São Carlos, SP, Brazil
http://www.nilc.icmc.usp.br
{thiago@nilc.icmc.usp.br; lucia@dc.ufscar.br; gracan@icmc.usp.br}

Abstract. This paper presents a new extractive approach to automatic summarization based on the gist of the source text. The gist-based system, called GistSumm (GIST SUMMarizer), uses the gist as a guideline to identify and select text segments to include in the final extract. Automatically produced extracts have been evaluated under the light of gist preservation and textuality.

1 Introduction

This paper presents a method for text summarization based on the gist of a source text that differs from the related ones in Computational Linguistics. Gist, here, is understood as the main idea intended by the writer or grasped by the reader. Using simple statistical measures, gist is identified as the most important passage of the source text, conveyed by just one sentence. It serves, then, as the guideline to identify and select other sentences to compose the final extract. Those are added to the extract provided that they satisfy summarization requirements, namely, gist preservation, textuality, relevance, and compression constraints. The novelty of our gist-based method, embedded in the so-called GistSumm (GIST SUMMarizer) system, consists of both the way gist is identified and used to produce the extract.

It has long been common sense that gist should guide information selection to produce a summary or extract [1] [2] [3] [4] [5]. Deep-based approaches, e.g., those based on the rhetorical structuring of source texts, address nuclearity to select relevant information [4] [6]. Often, they provide a hierarchy of rhetorical relations, as does O'Donnel's approach [6], or determine the importance of text spans according to their position in a rhetorical structure, as do Marcu's [4] and Ono et al.'s [2] approaches. Those works aim at guiding the selection of discourse segments that are related to the most salient nucleus and organizing them to produce the final summary structure. Since a variety of discourse segments can be chosen, distinct rhetorical structures can be produced, which result in distinct summaries after surface realization. In doing so, the source rhetorical structure is pruned, but its discourse backbone is kept unaltered, implying that nuclearity referring to the main idea is preserved in the final summary. In a similar deep approach based on the Rhetorical Structure Theory [7], we also address text summarization by using a central proposition of a text [3] [8]. However, we do not build the rhetorical structure of the source text and, thus, the central proposition is embedded in its information structure. By focusing on the central proposition, our system builds a summary structure from the scratch, through the combination of intentions [9] and those propositions pinpointed by relations appearing in the informa-

[1] This work was supported by FAPESP and CNPq.

tion structure. Similarly to the other deep-based work, our system can also produce various summaries for the same source text, for distinct strategies may be corresponding to varied choices of information units or discourse relations. Additionally, it also guarantees that the central proposition is preserved in every summary.

Although GistSumm also determines the gist of a source text, its approach is purely statistical, avoiding the inherent complexity of those deep-based ones. There are other surface-based approaches that address gist preservation in both mono- and multi-document summarization. Hovy and Lin's SUMMARIST [10], for example, is able to produce both extracts and abstracts of a source text by carrying out three main tasks: topic identification, interpretation (which fuses the identified topics) and summary generation. Topic identification, specially, corresponds to selecting a set of sentences that best expresses the gist through a combination of techniques (keywords, cue phrases, sentences location in text, etc.). To summarize a collection of texts, Radev et al. [11] consider word counting to score each sentence from the source texts and those highest-scored, above a specified threshold, are considered to be the most representative ones from the collection. Together, they finally compose the extract.

Gist identification in GistSumm is similar to that of SUMMARIST, although it focuses on only one sentence. Complementary sentences to produce the extract are those that are above a threshold, like in Radev et al.'s approach. GistSumm novelty relies on the extra constraint that additional sentences correlate to gist. Correlation is computed through lexical cohesion [12] [13], which is carried out by computing words co-occurrence. GistSumm is described in Section 2, followed by its evaluation (Section 3). Remarks on our proposal are presented in Section 4.

2 GistSumm Description

By making GistSumm gist-based, we assume it emulates human summarization in that, when a person summarizes a text, s/he first tries to identify the gist and, then, adds information drawn from the text to complement it [5]. Considering that the amount of complementary information to appear in the extract depends on how long the extract is intended to be[2], extraction is based, thus, on two parameters: the gist, which triggers the process, and the intended compression rate of the corresponding extracts.

Gist can be determined through either the Keywords [14] or the Text Mining [15] method. Its determination itself is very simple: a) based on the former method, gist is calculated on the basis of a list of keywords of the source text, considering a threshold of word significance; b) based on the latter, it is the result of the measurement of the representativeness of intra- and inter-paragraphs sentences. In both cases just one sentence is assigned to gist: in the former, it is the sentence that corresponds to the most significant distribution of keywords; in the latter, it is that whose frequency distinguishes it as the most representative of the source text, similarly to the way a topic or a search phrase is derived [16].

2.1 GistSumm Premises

In GistSumm, the following premises hold: 1) every text is built around a main topic, or idea; 2) it is possible to identify in a text one sentence that best expresses its main topic, i.e., the gist sentence. Based on these, we adopt the following hypotheses:

I. Through simple statistics, we can determine either the gist sentence or a quite satisfactory approximation of it;

[2] It also depends on the intended level of detail, but this is not measurable in GistSumm.

II. By means of the gist sentence, it is possible to build coherent extracts, which will convey the gist sentence itself and those extra sentences that may complement it, and, thus, make extracts more informative.

The keypoint in GistSumm is, thus, to identify those sentences that better correlate to gist. This is explained below, along with the description of its processes.

2.2 GistSumm Processes

GistSumm comprises three processes, namely, segmentation, sentence ranking, and extract production. Segmentation addresses sentences as minimal units. After delimiting them, sentence ranking proceeds to gist determination through the selected ranking method (either Keywords or Text Mining). Hereafter, we will refer to GistKey to signal the use of the Keywords method by GistSumm; otherwise, as GistTFISF, after the measure TF-ISF (Term Frequency – Inverse Sentence Frequency) [15]. Besides indicating the gist sentence, sentence ranking also classifies the other sentences to identify those that will appear in the extract. In this stage, for a more accurate calculation GistSumm makes use of the following sub-processes: stopwords removal, stemming and case folding, as suggested in [17]. Extract production finally identifies sentences to include in the final extract that satisfy (1) gist correlation, (2) relevance and (3) compression rate constraints. In what follows, sentence ranking and extract production are detailed and exemplified for the sample text shown in Figure 1, whose sentence segments are numbered. This has been extracted from a corpus of scientific texts in Computer Science.

> [English is the dominant language in the writing and publishing of scientific research in the form of scientific articles.]$_1$ [However, many non-natives users of English suffer the interference of their mother tongues when writing scientific papers in English.]$_2$ [These users face problems concerning rules of grammar and style, and/or feel unable to generate standard expressions and clauses, and the longer linguistic compositions which are conventional in this genre.]$_3$ [In order to ease these users' problems, we developed a learning environment for scientific writing named AMADEUS (Amiable Article Development for User Support).]$_4$ [AMADEUS consists of several interrelated tools - reference, support, critic and tutoring tools - and provides the context in which this dissertation is inserted.]$_5$ [The main goal of this research is to implement AMADEUS as an agent-based architecture with collaborative agents communicating with a special agent embodying a dynamic user model.]$_6$ [In order to do that we introduce the concept of adaptivity in computer systems and describe several user model shells.]$_7$ [We also provide details about intelligent agents which were used to implement the user model for the AMADEUS environment.]$_8$

Figure 1 – Sample text

2.3 Sentence Ranking

The scoring of the sentences is carried out in two steps: preprocessing and ranking itself. The former corresponds to vectoring the sentences of the source text [16] and, then, for each vector, removing stopwords and stemming (following [18]), and case folding the remaining ones. Finally, words frequency is computed through either the Keywords or the TF-ISF method to rank each sentence. For any of the employed methods, the sentence with the highest score is assumed to be the gist sentence. When two or more sentences in the source text have overlapping scores, the last one to be processed is chosen. This decision is due to the verification that, in a corpus of scientific texts, namely, the Theses Corpus [19], the gist sentence usually appears near the end of the text. This has also been corroborated by Aretoulaki [20].

Table 1 shows the sentence scores for the sample text shown in Figure 1. As it can be seen, the Keywords method signals sentence 4 as the gist sentence, but the TF-ISF one pinpoints sentence 3 (the highest-scored ones). By thoroughly reading the sample text and comparing it with such choices, we confirm that the former method identifies more clearly the gist sentence, since it mirrors the main idea more properly than sentence 3. Indeed, if we discourse-analyze the sample text based on the RST Theory [7], for example, sentence 3 still refers to background information. So, it could not be the gist sentence. Alternatively, the main topic refers to a tool to help non-native English speakers. So, sentence 4 suffices well such a role[3], being pretty satisfactory as the gist of the text. Corroborating this, we can also acknowledge that the remaining sentences 5-8 just add further details on the tool itself. Having determined the gist sentence, GistSumm can now proceed in selecting complementary sentences to build the corresponding extract.

Table 1 – Sentences scores

Sentence	Keywords	TF-ISF
1	24	0,465
2	22	0,628
3	23	**0,671**
4	**42**	0,598
5	22	0,643
6	37	0,663
7	17	0,571
8	25	0,575

2.4 Extract Production

To build the extract, GistSumm executes the following steps:

1) It averages the sentence scores, determining their threshold;
2) Besides the gist sentence, GistSumm selects others that both
   a. Contain at least one word whose stem also corresponds to some word in the gist sentence (i.e., it assures that lexical cohesion be observed);
   b. Have scores above the threshold (i.e., it guarantees that only relevant sentences to gist will be chosen).

The above steps are also constrained to satisfying the compression rate. If this is too strict that only the gist sentence satisfies it, the extract will be mono-sentential and as informative as its gist sentence (step 2 will thus be excluded). So, there is a compromise between informativeness and compression that will even delineate if gist can be complemented or not. Clearly, step (2a) is responsible for the distinctive idea underlying GistSumm when compared with other extractive approaches: it is this step that is intended to prove Hypothesis (II). This is addressed in Section 3 along with the proof of Hypothesis (I).

Figures 2 and 3 show extracts of the sample text for a 60% compression rate, respectively referring to GistKey and GistTFISF methods. It is possible to notice that the first extract conveys the gist, while the second one does not. Besides not resolving anaphors, GistSumm also does not

[3] Considering that sentence 4, i.e., the gist sentence, will be part of the final extract, it is evident that post-processing is mandatory to resolve occurrences such as the introduced dangling segments. We stress that extractive methods usually do not address such an issue.

address other types of non-cohesive devices resulting from the extractive process (e.g., the contrastive starting sentence in Extract 2), like most of the existing simple extractive methods.

3 Evaluating GistSumm Performance

Our evaluation of GistSumm has been carried out aiming at two distinctive goals: 1) to see how effective the proposed ranking methods are in identifying the gist sentence of a source text, i.e., to certify Hypothesis (I) is pertinent; 2) to assess GistSumm performance as such, by means of focusing on the quality of the generated extracts, i.e., to certify Hypothesis (II) is attainable. The evaluation has been taken mostly after three proposals, namely, those in [21], [22] and [5].

> English is the dominant language in the writing and publishing of scientific research in the form of scientific articles. In order to ease these users' problems, we developed a learning environment for scientific writing named AMADEUS (Amiable Article Development for User Support). The main goal of this research is to implement AMADEUS as an agent-based architecture with collaborative agents communicating with a special agent embodying a dynamic user model. We also provide details about intelligent agents which were used to implement the user model for the AMADEUS environment.

Figure 2 – GistKey-based Extract 1

> However, many non-natives users of English suffer the interference of their mother tongues when writing scientific papers in English. These users face problems concerning rules of grammar and style, and/or feel unable to generate standard expressions and clauses, and the longer linguistic compositions which are conventional in this genre.

Figure 3 – GistTFISF-based Extract 2

3.1 Experiment 1: Identifying the Gist Sentence

In this experiment, we used the already mentioned Theses Corpus [19], composed of 10 scientific texts on Computer Science (c.a. 530 words). To assess Hypothesis (I), 10 human judges, computational linguists and native Brazilian Portuguese speakers, were asked to identify their corresponding gist sentences. We, thus, used them as *gist gold standards* (GGSs)[4], to compare with the ones generated by GistSumm. Table 2 synthesizes the figures for GistKey and GistTFISF methods when using a 60% compression rate. When GGSs do not fully match the automatic gist ones, we also verify how close they are.

Table 2 – GistSumm effectiveness in determining gist sentences

	GGS identified		Proximity of GGSs to the calculated gist sentences		
Methods	Yes	No	None	Vague	Close
GistKey	20%				
		80%	30%	0	50%
GistTFISF	20%				
		80%	70%	10%	0

[4] After (Teufel and Moens, 1999).

As it can be seen, GistKey significantly outperforms GistTFISF in identifying the gist sentence. In 20% of the cases, the GGSs were completely identified and in 50%, they got a very close score. In both cases, the gist sentences were selected by GistSumm to be in the final extracts. Oppositely, when calculating the gist sentence through GistTFISF, only 20% GGSs were found to correspond to the automatic corresponding ones and only 10% got vaguely close to those. However vague they were graded, they were also selected by GistSumm to be in the final extracts. Considering both methods, when there was no proximity at all, the gist sentences were excluded from the final extracts. This shows the following: a) the GistKey method can signal gist sentences that suggest a good proximity to the idea humans get from the source texts; b) GistTFISF may be discarded as indicator of gist sentences, at least for the current test corpus.

This very same experiment was also used to compare extracts generated by Microsoft Office AutoSummarize with the ones generated by GistSumm. Since we cannot get gist sentences from the AutoSummarize, we could not verify if they matched the GGSs. So, we customized it to produce only one sentence-sized extract and, then, verified if those were corresponding to our GGSs. AutoSummarize failed in 100% of the cases. Considering a 60% compression rate, AutoSummarize included the GGSs in the extracts in 60% of the cases, which were outperformed by GistKey (70%) when using the same data. Although GistSumm performed well in this investigation, such results are still inconclusive, due to the size and significance of the test corpus.

3.2 Experiment 2: Evaluating the Extracts Overall Quality

A different test corpus has been used in this experiment, composed of 20 newspaper texts in English, from the WSJ financial section (c.a. 410 words). For each text, 2 extracts were automatically generated considering also the 60% compression rate. We had 12 human judges reading the source texts and scoring their extracts based on two decision points [22]: gist preservation and textuality (see Table 3). Textuality, here, is the property of a text being both cohesive and coherent [3]. Graphic 1 synthesizes the average scores, indicating again that GistKey outperforms GistTFISF when automatic extracts are compared with their corresponding source texts.

Table 3 – Possible scores for quality

Gist	Textuality	Score
Preserved	Assured	9
Preserved	Partially assured	8
Preserved	Not assured	7
Partially preserved	Assured	6
Partially preserved	Partially assured	5
Partially preserved	Not assured	4
Not preserved	Assured	3
Not preserved	Partially assured	2
Not preserved	Not assured	1

Graphic 1 – Average of extracts scores

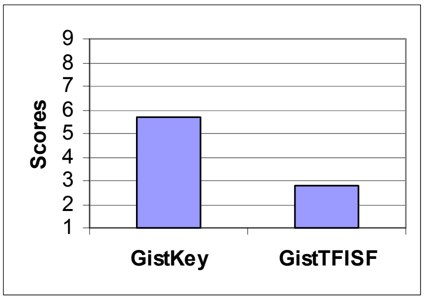

Table 4 shows the distribution of the extracts according to the means, i.e., the scores average [22]. Although only 55% of the ones generated through GistKey are above the means, this still outperforms GistTFISF. Table 5 shows the figures for gist and textuality, as suggested by [5]. When using GistKey, 90% of the extracts have been judged to totally or partially preserve the gist. Textuality was also highly graded: 85% of them were totally or partially coherent and cohesive. Again, GistKey significantly outperformed GistTFISF.

Table 4 – Means distribution

Means Methods	Above	In It
GistKey	55%	14%
GistTFISF	39%	10%

Table 5 – Extracts quality

Points	Gist Preservation		Textuality	
Methods	Total	Partial	Total	Partial
GistKey	50%	40%	50%	35%
GistTFISF	10%	20%	5%	25%

4 Final Remarks

GistSumm has been devised to produce extracts from texts of any domain, genre, and natural language, provided that the corresponding NL resources (i.e., stopwords repository and stemmer) are assembled into it. So far, we have explored it for Brazilian Portuguese and English. We have chosen two simple statistical methods in order to determine a gist sentence, which is the backbone of text extraction, for two reasons: they are easy to implement and they do not demand complex and sophisticated linguistic resources. So, they seemed appealing for us to verify and compare their effectiveness in determining the gist sentence. The experiments described here made evident that the correct determination of the gist sentence usually influences the quality of the related extracts. Moreover, they show that gist conveys better the content of the corresponding source text when it is computed through GistKey, instead of GistTFISF. Two conclusions can be withdrawn from this: the gist identification method based on the keywords distribution performs better than that based on the inverse distribution of sentences in the source text, for the test corpus adopted so far. However, further investigations should explore more deeply such a difference, for other text genres and domains and for more significant corpora.

Although many authors have stressed the need to convey the main idea and to warrant the textuality of the results in automatic summarization [1] [23] [24], GistSumm is novel because of both the way gist is determined and used as a guide for extraction (through lexical cohesion): by observing the words co-occurrence, the extracts are more likely to be coherent; by including gist, they are more likely to convey well the main idea of their source texts.

A quite considerable limitation of our proposal refers to gist being corresponding to just one sentence, since very often it is embedded in the thread of discourse and, thus, it may be diffuse in the text [8] [3]. To overcome it, we should extend GistSumm to signal multiple segments, instead of just one sentence. This certainly will compromise compression rates. In order to avoid it, thresholds or other means to correlate sentences to gist must be explored (e.g., [23]).

References

1. Sparck Jones, K.: Discourse Modelling for Automatic Summarising. Technical Report No. 290. University of Cambridge (1993).
2. Ono, K., Sumita, K., Miike, S.: Abstract Generation based on Rhetorical Structure Extraction. In the *Proceedings of the 15th International Conference on Computational Linguistics* – COLING'94, Vol. 1, pp. 344-348. Kyoto, Japan (1994).
3. Rino, L.H.M.: *Modelagem de Discurso para o Tratamento da Concisão e Preservação da Idéia Central na Geração de Textos*. Tese de Doutorado. IFSC-USP. São Carlos – SP (1996).
4. Marcu, D.: *The Theory and Practice of Discourse Parsing and Summarization*. Cambridge, MA. The MIT Press (2000).
5. Mani, I.: *Automatic Summarization*. John Benjamins Publishing Co., Amsterdam (2001).

6. O'Donnell, M.: Variable-Length On-Line Document Generation. In the *Proceedings of the 6th European Workshop on Natural Language Generation*, Gerhard-Mercator University, Duisburg, Germany (1997).
7. Mann, W.C. and Thompson, S.A.: *Rhetorical Structure Theory: A Theory of Text Organization*. Technical Report ISI/RS-87-190 (1987).
8. Pardo, T.A.S. and Rino, L.H.M.: DMSumm: Review and Assessment. In E. Ranchhod and N. J. Mamede (eds.), *Advances in Natural Language Processing*, pp. 263-273 (Lecture Notes in Artificial Intelligence 2389). Springer-Verlag, Germany (2002).
9. Grosz, B. and Sidner, C.: Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, Vol. 12, No. 3 (1986).
10. Hovy, E.H. and Lin, C.Y.: Automated Text Summarization in SUMMARIST. In M. Maybury and I. Mani (eds), *Advances in Automatic Text Summarization*. Cambridge, MIT Press (1998).
11. Radev, D.R., Jing, H., Budzikowska, M.: Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In the *Proceedings of the ANLP/NAACL Workshop on Summarization*. Seattle, WA (2000).
12. Morris, J. and Hirst, G.: Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. *Computational Linguistics*, Vol. 17, No. 1 (1991).
13. Hoey, M.: *Patterns of Lexis in Text*. Oxford University Press (1991).
14. Luhn, H. P.: The automatic creation of literature abstracts. *IBM Journal of Research and Development*, Vol. 2, pp. 159-165 (1958).
15. Larocca Neto, J., Santos, A.D., Kaestner, A.A., Freitas, A.A.: Generating Text Summaries through the Relative Importance of Topics. In M.C. Monard and J.S. Sichman (eds.), *Lecture Notes in Artificial Intelligence*, No. 1952, pp. 300-309. Springer-Verlag (2000).
16. Salton, G.: *Automatic Text Processing. The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley (1989).
17. Witten, I.H., Moffat, A., Bell, T.C.: Managing Gigabytes. *Van Nostrand Reinhold*. New York (1994).
18. Porter, M.F.: An algorithm for suffix stripping. *Program*, Vol. 14, No. 3 (1980).
19. Feltrim, V.D., Nunes, M.G.V., Aluísio, S.M.: *Um corpus de textos científicos em Português para a análise da Estrutura Esquemática*. Série de Relatórios do NILC. NILC-TR-01-4 (2001).
20. Aretoulaki, M.: *COSY-MATS: A Hybrid Connectionist-Symbolic Approach To The Pragmatic Analysis of Texts For Their Automatic Summarisation*. PhD. Thesis. University of Manchester (1996).
21. Sparck Jones, K. and Galliers, J. R.: Evaluating Natural Language Processing Systems. *Lecture Notes in Artificial Intelligence*, Vol. 1083 (1996).
22. White, J.S., Doyon, J.B., Talbott, S.W.: Task Tolerance of MT Output in Integrated Text Processes. In *ANLP/NAACL 2000: Embedded Machine Translation Systems*, pp. 9-16. Seattle, WA (2000).
23. Barzilay, R. and Elhadad, M.: Using lexical chains for text summarization. In the *Proceedings of the ACL-97/EACL97 Workshop on Intelligent Scalable Text Summarization*. Madrid, Spain (1998).
24. Teufel, S. and Moens, M.: Argumentative classification of extracted sentences as a first step towards flexible abstracting. In I. Mani and M. Maybury (eds.), *Advances in automatic text summarization*, MIT Press (1999).