ARTEX *is AnotheR TEXt summarizer*

Juan-Manuel TORRES-MORENO[1,2]

[1] Laboratoire Informatique d'Avignon,
BP 91228 84911, Avignon, Cedex 09, France
juan-manuel.torres@univ-avignon.fr
[2] École Polytechnique de Montréal,
CP. 6128 succursale Centre-ville, Montréal, Québec, Canada

Abstract

This paper describes ARTEX, another algorithm for Automatic Text Summarization. In order to rank sentences, a simple inner product is calculated between each sentence, a document vector (text topic) and a lexical vector (vocabulary used by a sentence). Summaries are then generated by assembling the highest ranked sentences. No ruled-based linguistic post-processing is necessary in order to obtain summaries. Tests over several datasets (coming from Document Understanding Conferences (DUC), Text Analysis Conference (TAC), evaluation campaigns, etc.) in French, English and Spanish have shown that ARTEX summarizer achieves interesting results.

Keywords: Automatic Text Summarization, Space Vector Model, Text extraction, Ultra-stemming

1 Introduction

Automatic Text Summarization (ATS) is the process to automatically generate a compressed version of a source document [15]. Query-oriented summaries focus on a user's request, and extract the information related to the specified topic given explicitly in the form of a query [2]. Generic mono-document summarization tries to cover as much as possible the information content. Multi-document summarization is a oriented task to create a summary from a heterogeneous set of documents on a focused topic. Over the past years, extensive experiments on query-oriented multi-document summarization have been carried out. Extractive Summarization produces summaries choosing a subset of representative sentences from original documents. Sentences are ordered and then assembled according to their relevance to generate the final summary [10].

This article introduces a new method of summarization based in sentences extraction on Vector Space Model (VSM). We score each sentence by calculating their inner product with a pseudo-sentence vector and a pseudo-word vector. Results show that ARTEX not only preserves the content of the summaries generated using this new representation, but often, surprisingly the performance can be improved. ARTEX could be an interesting and simple algorithm using the extractive summarization paradigm. Our tests on trilingual corpora (English, Spanish and French) evaluated by the FRESA algorithm (without human references) confirm the good performance of ARTEX.

In this paper, related work is given in Section 2. Section 3 presents the new algorithm of Automatic Text Summarization. Experiments are presented in Section 4, followed by Results in Section 5 and Conclusions in Section 6.

2 Related works

Research in Automatic Text Summarization was introduced by H.P. Luhn in 1958 [9]. In the strategy proposed by Luhn, the sentences are scored for their component word values as determined by tf*idf-like weights. Scored sentences are then ranked and selected from the top until some summary length threshold is reached. Finally, the summary is generated by assembling the selected sentences in the original source order. Although fairly simple, this extractive methodology is still used in current approaches. Later on, [3] extended this work by adding simple heuristic features such as the position of sentences in the text or some key phrases indicate the importance of the sentences. As the range of possible features for source characterization widened, choosing appropriate features, feature weights and feature combinations have become a central issue.

A natural way to tackle this problem is to consider sentence extraction as a classification task. To this end, several machine learning approaches that uses document-summary pairs have been proposed [6, 12], An hybrid method mixing statistical and linguistics algorithms is presented in [1]. [10] and [15] propose a good state-of-art of Automatic Text Summarization tasks and algorithms.

2.1 Document Pre-processing

The first step to represent documents in a suitable space is the pre-processing. As we use extractive summarization, documents have to be chunked into cohesive textual segments that will be assembled to produce the summary. Pre-processing is very important because the selection of segments is based on words or bigrams of words. The choice was made to split documents into full sentences, in this way obtaining textual segments that are likely to be grammatically corrects. Afterwards, sentences pass through several basic normalization steps in order to reduce computational complexity.

The process is composed by the following steps:

1. **Sentence splitting**. Simple rule-based method is used for sentence splitting. Documents are chunked at the period, exclamation and question mark.

2. **Sentence filtering**. Words lowercased and cleared up from sloppy punctuation. Words with less than 2 occurrences ($f < 2$) are eliminated (*Hapax legomenon* presents once in a document). Words that do not carry meaning such as functional or very common words are removed. Small stop-lists (depending of language) are used in this step.

3. **Word normalization**. Remaining words are replaced by their canonical form using lemmatization, stemming, ultra-stemming or none of them (raw text). Four methods of normalization were applied after filtering:

   - Lemmatization by simples dictionaries of morphological families. These dictionaries have 1.32M, 208K and 316K words-entries in Spanish, English and French, respectively.
   - Porter's Stemming, available at Snowball (web site `http://snowball.tartarus.org/texts/stemmersoverview.html`) for English, Spanish, French among other languages.
   - Ultra-stemming. This normalization seems be very efficient and it produces a compact matrix representation [16]. Ultra-stemming consider only the n first letters of

each word. For example, in the case of ultra-stemming (first letter, FIX$_1$), inflected verbs like "sing", "song", "sings", "singing"... or proper names "smith", "snowboard", "sex",... are replaced by the letter "**s**".

4. **Text Vectorization**. Documents are vectorized in a matrix $S_{[P\times N]}$ of P sentences and N columns. Each element $s_{i,j}$ represents the occurrences of an object j (a letter in the case of ultra-stemming, a word in the case of lemmatization or a stem for stemming), $j = 1, 2, ..., N$ in the sentence i, $i = 1, 2, ..., P$.

3 *AnotheR TEXt summarizer* (ARTEX)

ARTEX[1] is a simple extractive algorithm for Automatic Text Summarization. The main idea is the next one: First, we represent the text in a suitable space model (VSM). Then, we construct an average document vector that represents the average (the "global topic") of all sentences vectors. At the same time, we obtain the "lexical weight" for each sentence, i.e. the number of words in the sentence. After that, it is calculated the angle between the average document and each sentence; narrow angles indicate that the sentences near of the "global topic" should be important and therefore extracted. See on the figure 1 the VSM of words: P vector sentences and the average "global topic" are represented in a N dimensional space of words.

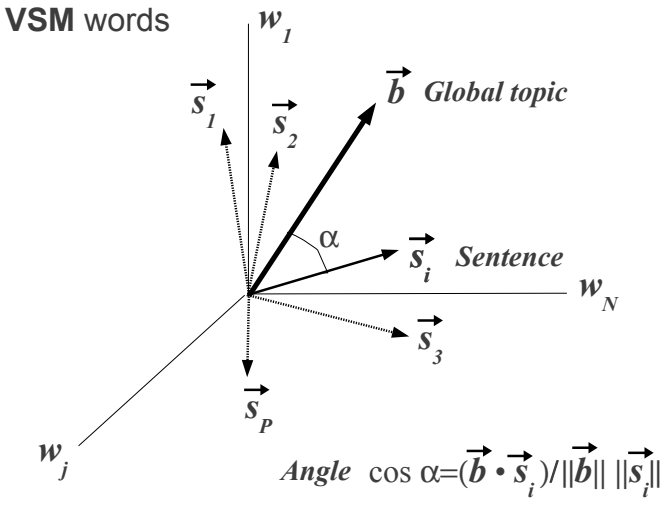

Figure 1: The "global topic" in a Vector Space Model of N words.

Next, a score for each sentence is calculated using their proximity with the "global topic" and their "lexical weight". In the figure 2, the "lexical weight" is represented in a VSM of P sentences.

Finally, the summary is generated concatenating the sentences with the highest scores following their order in the original document.

[1]In French, ARTEX *est un Autre Résumeur TEXtuel.*

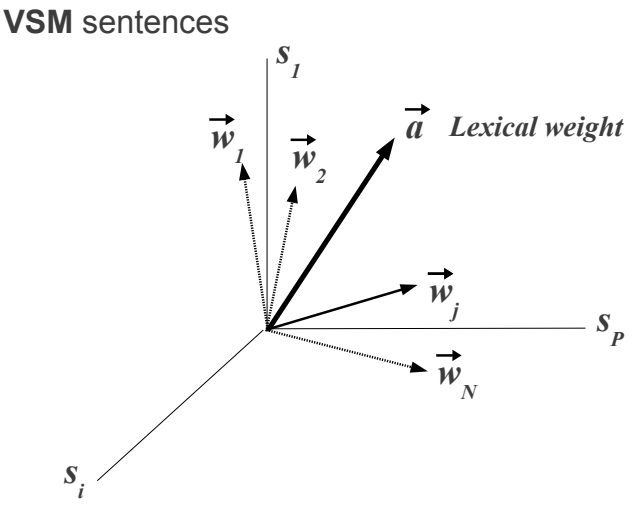

Figure 2: The "lexical weight" in a Vector Space Model of P sentences.

3.1 Algorithm

Formally, ARTEX algorithm computes the score of each sentence by calculating the inner product between a sentence vector, an *average pseudo-sentence vector* (the "global topic") and an *average pseudo-word vector* (the "lexical weight").

Once a pre-processing (word normalization and filtering of stop words) is completed, it is created a matrix $S_{[P\times N]}$, using the Vector Space Model, that contains N words (or letters) and P sentences.

Let $s_i = (s_1, s_2, ..., s_N)$ be a vector of the sentence i, $i = 1, 2, ..., P$. We defined $\vec{a}$ the *average pseudo-word vector*, as the average number of occurrences of N words used in the sentence i:

$$(1) \quad a_i = \frac{1}{N}\sum_j s_{i,j}$$

and $\vec{b}$ the *average pseudo-sentence vector* as the average number of occurrences of each word j used trough the P sentences:

$$(2) \quad b_j = \frac{1}{P}\sum_i s_{i,j}$$

The score or weight of each sentence s_i is calculated as follows:

$$(3) \quad \text{score}(s_i) = \left(\vec{s}\times\vec{b}\right) \times \vec{a} = \frac{1}{NP}\left(\sum_j s_{i,j}\times b_j\right)\times a_i \,;\; i = 1, 2, ..., P \,;\; j = 1, 1, ..., N$$

The score(•) computed by equation 3 must be normalized between the interval [0,1]. The calculation of $\vec{s}\times\vec{b}$ indicates the proximity between the sentence $\vec{s}$ and the *average pseudo-sentence* $\vec{b}$. The product $(\vec{s}\times\vec{b})\times\vec{a}$ weigh this proximity using the *average pseudo-word* a_i.

If a sentence s_i is near of $\vec{b}$ and their corresponding element a_i has a high value, s_i will have, therefore, a high score. Moreover, a sentence i far of main topic (i.e. $\vec{s_i} \times \vec{b}$ is near 0) or a less informative sentence i (i.e. a_i are near 0) will have a low score.

In computational terms, it is not really necessary to divide the scalar product by the constant $\frac{1}{NP}$, because the angle $\alpha = \arccos \vec{b}.\vec{s}/|\vec{b}||\vec{s}|$ between $\vec{b}$ and $\vec{s}$ is the same if we use $\vec{b} = \vec{b'} = \sum_i s_{i,j}$. The element a_i is only a scale factor that does not modify α.

In fact, if the matrix $S_{[P\times N]}$ is approximated to a binary matrix[2] $S'_{[P\times N]}$, where each element $s'_{i,j} = \{0, 1\}$ has a probability of $p = \frac{1}{2}$, we can normalize vectors $\vec{a}$, $\vec{b}$ and matrix S, as follows:

$$|\vec{a}| = \sum_i^P \sqrt{{s'_{i,j}}^2} = \sum_i^P \sqrt{(\{0,1\}^P)^2} = N\sqrt{P} \tag{4}$$

$$|\vec{b}| = \sum_j^N \sqrt{{s'_{i,j}}^2} = \sum_j^N \sqrt{(\{0,1\}^N)^2} = \sqrt{N}P \tag{5}$$

$$|\vec{s_i}| = \sum_j^N \sqrt{{s'_{i,j}}^2} = \sum_j^N \sqrt{\{0,1\}^2} = N \tag{6}$$

Vectors then will be represented in hyper-spheres of N or P dimensions, and the normalized score' in this space would be:

$$\begin{aligned} \text{score'}(s_i) &= \left(\frac{\vec{s}}{|\vec{s}|} \times \frac{\vec{b}}{|\vec{b}|}\right) \times \frac{\vec{a}}{|\vec{a}|} = \frac{1}{N\sqrt{N}PN\sqrt{P}} \left(\sum_j s_{i,j} \times b_j\right) \times a_i \\ &= \frac{1}{\sqrt{N^5P^3}} \left(\sum_j s_{i,j} \times b_j\right) \times a_i \,;\, i = 1, 2, ..., P \,;\, j = 1, 2, ..., N \end{aligned} \tag{7}$$

However, the term $1/\sqrt{N^5P^3}$ is a constant value (i.e. a simple scale factor), and then the score($\bullet$) calculated using the equation 3) and the score'($\bullet$) using the equation 7, are both equivalent.

4 Experiments

ARTEX algorithm described in the previous section has been implemented and evaluated in corpora in several languages.

We have conducted our experimentation with the following languages, summarization tasks, summarizers and data sets: 1) Generic multi-document-summarization in English with the corpus DUC'04; 2) Generic single-document summarization in Spanish with the corpus *Medicina Clínica* and 3) Generic single document summarization in French with the corpus PISTES.

We have applied the summarization algorithms and finally, the results have been evaluated using FRESA while processing times for each summarizer have been measured and compared.

The following subsections present formally the details of the summarizers, corpora and evaluations studied in different experiments.

[2]This is a reasonable approximation in this context, because $S_{[P\times N]}$ is a sparsed matrix with many term occurrences equal to one or zero.

4.1 Other Summarizers

To compare the performances, two other summarization systems were used in our experiments: CORTEX and ENERTEX. To be in the same conditions, these two systems have used exactly the same textual representation based on Vector Space Model, described in Section 2.1.

- CORTEX is a single-document summarization system using several metrics and an optimal decision algorithm [4, 14, 15, 18].
- ENERTEX is a summarization system based in Textual Energy concept [5]: text is represented as a spin system where spins $\uparrow$ represents words that their occurrences are $f > 1$ (spins $\downarrow$ if the word is not present).

4.2 Summarization Corpora Description

To study the impact of our summarizer, we used corpora in three languages: English, Spanish and French. The corpora are heterogeneous, and different tasks are representatives of Automatic Text Summarization: generic multi-document summary and mono-document guided by a subject.

- Corpus in English. Piloted by NIST in Document Understanding Conference[3] (DUC) the Task 2 of DUC'04[4], aims to produce a short summary of a cluster of related documents. We studied generic multi-document-summarization in English using data from DUC'04. This corpus with 300K words (17 780 types) is compound of 50 clusters, 10 documents each.
- Corpus in Spanish. Generic single-document summarization using a corpus from the scientific journal *Medicina Clínica*[5], which is composed of 50 medical articles in Spanish, each one with its corresponding author abstract. This corpus contains 125K words (9 657 types).
- Corpus in French. We have studied generic single-document summarization using the Canadian French Sociological Articles corpus, generated from the journal *Perspectives interdisciplinaires sur le travail et la santé* (PISTES)[6]. It contains 50 sociological articles in French, each one with its corresponding author abstract. This corpus contains near 400K words (18 887 types).

4.3 Summaries Content Evaluation

DUC conferences have introduced the ROUGE content evaluation [7], wich measures the overlap of n-grams between a candidate summary and reference summaries written by humans. However, to write the human summaries necessaries for ROUGE is a very expensive task.

Recently metrics without references have been defined and experimented at DUC and Text Analysis Conferences (TAC)[7] workshops.

FRESA content evaluation [13, 17] is similar to ROUGE evaluation, but human reference summaries are not necessary. FRESA calculates the divergence of probabilities between the

[3] http://duc.nist.gov
[4] http://www-nlpir.nist.gov/projects/duc/guidelines/2004.html
[5] http://www.elsevier.es/revistas/ctl_servlet?_f=7032&revistaid=2
[6] http://www.pistes.uqam.ca/
[7] www.nist.gov/tac

candidate summary and the document source. Among these metrics, Kullback-Leibler (KL) and Jensen-Shannon (JS) divergences have been widely used by [8, 17] to evaluate the informativeness of summaries.

In this article, we use FRESA, based in KL divergence with Dirichlet smoothing, like in the 2010 and 2011 INEX edition [11], to evaluate the informative content of summaries by comparing their n-gram distributions with those from source documents.

FRESA only considered absolute log-diff between the terms occurrences of the source and the summary. Let T be the set of terms in the source. For every $t \in T$, we denote by C_t^T its occurrences in the source and C_t^S its occurrences in the summary.

The FRESA package computed the divergence between the document source and the summaries as follows:

$$\mathcal{D}(T||S) = \sum_{t \in T} \left| \log\left(\frac{C_t^T}{|T|} + 1\right) - \log\left(\frac{C_t^S}{|S|} + 1\right) \right| \quad (8)$$

To evaluate the information content (the "quality") of the generated summaries, after removing stop-words, several automatic measures were computed: FRESA_1 (Unigrams of single stems), FRESA_2 (Bigrams of pairs of consecutive stems), FRESA_{SU4} (Bigrams with 2-gaps also made of pairs of consecutive stems) and finally, $\langle \text{FRESA} \rangle$, i.e. the average of all FRESA values.

The FRESA values (scores) are normalized between 0 and 1. High FRESA values mean less divergence regarding the source document summary, reflecting a greater amount of information content. All summaries produced by the systems were evaluated automatically using FRESA package.

5 Results

In this section we present the results for each corpus with different summarizers and the several normalization strategies used. Based on these results, firstly, we have verified that ultra-stemming improves the performance of summarizers. Secondly, we show that ARTEX is a system that has a similar performances –in terms of information content and processing times– to other state-of-art summarizers.

5.1 Content evaluation

- **English corpus.** Figure 3 shows the performance of the three summarizers using FIX_1, stemming and lemmatization. Results show that ultra-stemming improves the score of the three automatic summarizer systems. ARTEX and CORTEX expose a similar performances in information content.

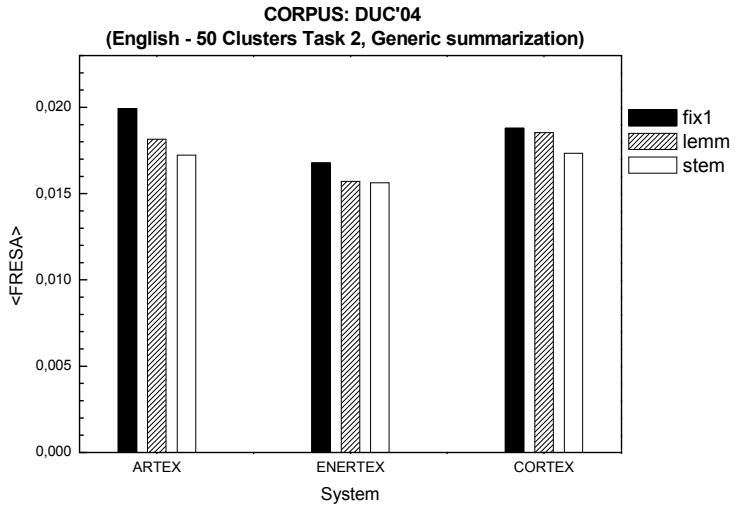

Figure 3: Histogram plot of content evaluation for corpus DUC'04 Task 2, with ⟨FRESA⟩ measures, for each summarizer and each normalization.

- **Spanish corpus.** Spanish is a language with a greater variability than English. Results in figure 4 shown that ARTEX summarizer outperforms CORTEX and ENERTEX if stemming or lemmatization are used as normalization.

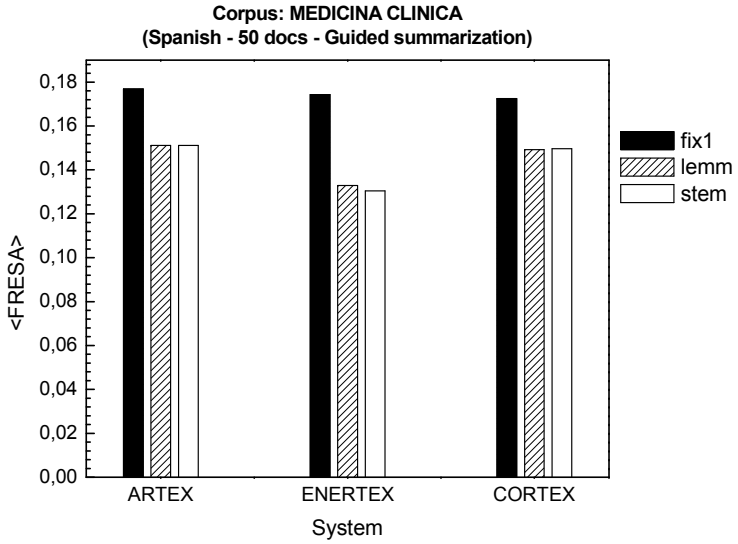

Figure 4: Histogram plot of content evaluation for Spanish corpus *Medicina Clínica* with ⟨FRESA⟩ scores for each summarizer.

- **French corpus.** French is a language with a large variability too. Figure 5 shows the score ⟨FRESA⟩ on the French corpus *Pistes*. Results show a similar behavior: Ultra-stemming improves the score of the three automatic summarization systems used. In particular, the efficacy of ARTEX is less sensible to word normalization than others summarizers.

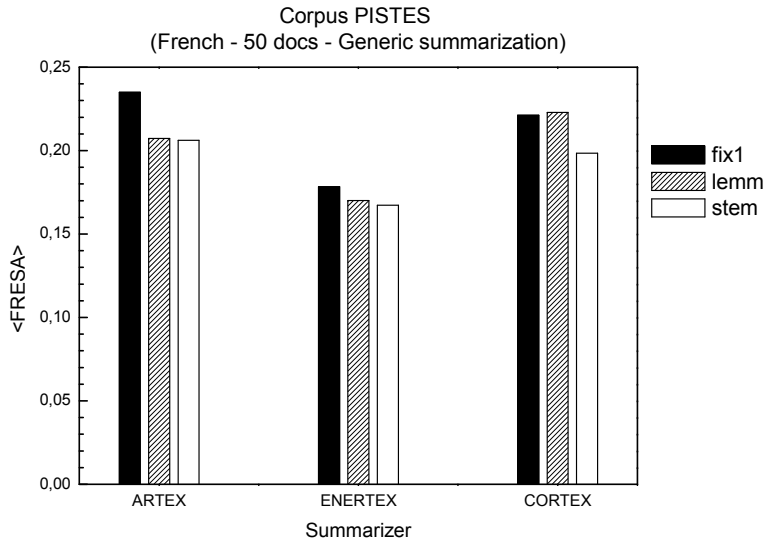

Figure 5: Histogram plot of content evaluation for French corpus PISTES with ⟨FRESA⟩ scores for each summarizer.

5.2 Processing Times Evaluation

Table 1 shows processing times for each corpus, following the normalization method for CORTEX, ARTEX and ENERTEX summarizers[8]. Processing times of ultra-stemming FIX_1 are shorter compared to all others methods. By example, CORTEX is a very fast summarizer with $O(\log \rho^2)$ (where $\rho = P \times N$), and processing times for stemming and FIX_1 are close. In other hand, ENERTEX summarizer has a complexity of $O(\rho^2)$, then it needs more time to process the same corpus. Performances of ARTEX algorithm remain close to CORTEX.

	Summarizer Average Time (all corpora)		
Normalization	CORTEX	ARTEX	ENERTEX
Lemmatization	1.60'	2.50'	10.42'
Stemming	0.54'	1.29'	9.47'
FIX_1	**0.32'**	*0.40'*	4.25'

Table 1: Statistics of processing times (in minutes) of three summarizers over three corpora.

[8]All times are measured in a 7.8 GB of RAM computer, Core i7-2640M CPU @ 2.80GHz × 4 processor, running under 32 bits GNU/Linux (Ubuntu Version 12.04).

6 Conclusions

In this article we have introduced and tested a simple method for Automatic Text Summarization. Artex is a fast and very simple algorithm based in VSM model and the extractive paradigm. The method uses a matrix representation to calculate a normalized score for each sentence, using the inner product of pseudo-(sentences|words) vectors. The algorithm retains the salient information of each sentence of document. An important aspect of our approach is that it does not requires linguistic knowledge or resources which makes it a simple and efficient summarizer method to tackle the issue of Automatic Text Summarization.

Summaries generated by Artex system are pertinents. The results obtained on corpora in English, Spanish and French show that Artex can achieve good results for content quality. Tests with other corpora (DUC and TAC evaluation campaigns, INEX, etc.) in mono-and multi-document guided by a subject, using content evaluation with (ROUGE evaluations) or without reference summaries still in progress.

References

[1] Iria da Cunha, Silvia Fernández, Patricia Velázquez-Morales, Jorge Vivaldi, Eric SanJuan, and Juan Manuel Torres-Moreno. A new hybrid summarizer based on vector space model, statistical physics and linguistics. In *Proceedings of the 6th Mexican International Conference on Advances in Artificial Intelligence (MICAI'07)*, pages 872–882, Aguascalientes, Mexico, 2007. Springer-Verlag.

[2] Harold Daumé III. *Practical structured learning techniques for natural language processing.* PhD thesis, Los Angeles, CA, 2006.

[3] H. P. Edmundson. New Methods in Automatic Extraction. *Journal of the Association for Computing Machinery*, 16(2):264–285, 1969.

[4] B. Favre, F. Béchet, P. Bellot, F. Boudin, M. El-Bèze, L. Gillard, G. Lapalme, and J-M. Torres-Moreno. The LIA-Thales summarization system at DUC-2006. In *Proceedings of the Document Understanding Conference (DUC'06)*, Brooklyn, New York, United States, 2006. http://duc.nist.gov.

[5] Silvia Fernández, Eric SanJuan, and Juan-Manuel Torres-Moreno. Textual Energy of Associative Memories: performants applications of Enertex algorithm in text summarization and topic segmentation. In *Proceedings of the Mexican International Conference on Artificial Intelligence (MICAI'07)*, pages 861–871, Aguascalientes, Mexico, 2007. Springer-Verlag.

[6] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th Conference ACM Special Interest Group on Information Retrieval (SIGIR'95)*, pages 68–73, Seattle, WA, United States, 1995. ACM Press, New York.

[7] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In Marie-Francine Moens and Stan Szpakowicz, editors, *Proceedings of the Workshop Text Summarization Branches Out (ACL'04)*, pages 74–81, Barcelone, Spain, july 2004. ACL.

[8] Annie Louis and Ani Nenkova. Automatic Summary Evaluation without Human Models. In *First Text Analysis Conference (TAC'08)*, Gaithersburg, MD, United States, 17-19 November 2008.

[9] H.P. Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[10] I. Mani and M. Mayburi. *Advances in Automatic Text Summarization.* MIT Press, Cambridge, 1999.

[11] Eric SanJuan, Patrice Bellot, Véronique Moriceau, and Xavier Tannier. Overview of the INEX 2010 Question Answering Track (QA@INEX). In Shlomo Geva, Jaap Kamps, Ralf Schenkel, and Andrew Trotman, editors, *Comparative Evaluation of Focused Retrieval*, volume 6932 of *Lecture Notes in Computer Science*, pages 269–281. Springer Berlin / Heidelberg, 2011.

[12] Simone Teufel and Marc Moens. Sentence extraction as a classification task. In I. Mani and M. Maybury, editors, *Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain, 11 July 1997.

[13] J.-M. Torres-Moreno, Horacio Saggion, I. da Cunha, P. Velazquez-Morales, and E. SanJuan. Evaluation automatique de résumés avec et sans réferences. In *Proceedings de la conference Traitement Automatique des Langagues Naturelles (TALN'10)*, Montréal, QC, Canada, 19-23 July 2010. ATALA.

[14] J.-M. Torres-Moreno, P.-L. St-Onge, M. Gagnon, M. El-Bèze, and P. Bellot. Automatic Summarization System coupled with a Question-Answering System (QAAS). *CoRR*, abs/0905.2990, 2009.

[15] Juan-Manuel Torres-Moreno. *Résumé automatique de documents: une approche statistique.* Hermès-Lavoisier, Paris, 2011.

[16] Juan-Manuel Torres-Moreno. Beyond Stemming and Lemmatization: Ultra-stemming to Improve Automatic Text Summarization. *CoRR*, arXiv:1209.3126 [cs.IR], 2012.

[17] Juan-Manuel Torres-Moreno, Horacio Saggion, Iria da Cunha, and Eric SanJuan. Summary Evaluation With and Without References. *Polibits: Research journal on Computer science and computer engineering with applications*, 42:13–19, 2010.

[18] Juan-Manuel Torres-Moreno, Patricia Velázquez-Morales, and Jean-Guy Meunier. Cortex : un algorithme pour la condensation automatique des textes. In *Proceedings of the Conference de l'Association pour la Recherche Cognitive*, volume 2, pages 365–366, Lyon, France, 2001.