Bag-of-senses versus bag-of-words: comparing semantic and lexical approaches on sentence extraction

Jorge García Flores Laurent Gillard Olivier Ferret
Gaël de Chalendar

Institut de Radioprotection et de Sûreté Nucléaire
DS/DICST
31, avenue de la Division Leclerc
92260 Fontenay aux Roses, France
{jorge.garcia-flores}@irsn.fr

CEA, LIST, Laboratoire d'Ingénierie de la Connaissance Multimédia Multilingue
18 route du Panorama, BP6
F-92265 Fontenay aux Roses, France.
{laurent.gillard, olivier.ferret, gael.de-chalendar}@cea.fr

Abstract

Sentence extraction is a valuable technique for automatic summarization. This paper presents LIC2M's first participation in TAC evaluation campaign (update summarization task). We describe two main extractive approaches for summarization. The semantic strategy makes use of a bag-of-senses to calculate sense concentration on each sentence. In the lexical strategy, source text sentences are ranked according to their lexical similarity against a topic statement represented as a bag-of-words. Both approaches are compared and the evaluation results analyzed. An alternative version of the semantic strategy is proposed, where sense concentration ranking takes into account syntactic dependencies.

1 Introduction

Recognizing relevant sentences in a text is a fundamental task for both human and computer summarizers. Sentence extraction has been widely used since the early years of automatic summarization [Man01], in part because of the linguistic quality of an extract is considered higher than that of an automatically generated sentence. Nowadays, sentence extraction engines are critical components of more complex summarization approaches.

Extractive approaches suppose that meaning is highly concentrated on certain textual units. It would be then possible to produce a condensed version of the source text made from its most significant fragments. The main advantage of sentence extraction is a discursive one: a sentence is a minimally structured argumentation unit. Its bounds being typographical, a sentence is easier to manipulate than other textual elements. Furthermore, extracting sentences allows better control over the summary's compression level.

However, as Spärck Jones points out, the main drawback of extractive approaches is the lack of discursive cohesion [SJ07]. References in an extract (like anaphora or pronouns) might loose their referents during the extraction process. This is better observed in long document summarization, where the topical references can changed completely between two extracted sentences.

This paper presents LIC2M first experience in a summarization campaign. Our goal is to evaluate the performance of the sentence extraction component, which will be part of more complex applications, like a book summarization tool or a question answering system. Three sentence extraction systems were submitted for evaluation. *ceaList3* semantic approach makes use of a *bag-of-senses* to calculate sense concentration on each sentence.

ceaList2 is based on a lexical strategy, where source text sentences are ranked according to their lexical similarity against a topic statement represented as a bag-of-words. System *ceaList1* is an extended version of *ceaList3*, where sense concentration ranking takes into account syntactic dependencies. Systems *ceaList2* and *ceaList1* were submitted to TAC 2008 for manual evaluation and *ceaList3* was submitted as an additional run.

System *ceaList2* is derived from a method developed for the answer extraction component of a question answering system [GSB$^+$06]. It performs a similarity analysis between an input question and each sentence from the source text. Ranking depends on two parameters: density (defined as the amount of lexical coincidences between the question and the potential answer) and proximity (the closeness between coincidences). System *ceaList2* participated in DUC 2007 [BFB$^+$07] as a part of a multi-feature system but was never evaluated as a standalone system. Systems 1 and 3 were developed as part of the Morcas project [CBF$^+$05, GFdC08]. Morcas main objective is to develop an opinion oriented summarizer for long texts related to nuclear energy. Both systems *ceaList1* and *ceaList3* depend on the same semantic resource: a base of word senses built from the co-occurrence relations extracted from a large corpus. Each word from the source text is projected on this base to find the predominant senses of the text. Its sentences are then weighted and ranked according to their sense concentration.

2 Related work

Despite the fact that sentence extraction has been used since the early days of automatic summarization [Luh58, Man01], extractive summarization is still a very active research line [RJST04, GKMC99, KSH01], where the open questions are 1) how to weight and rank sentences and 2) how to combine the results of different features. Systems *ceaList3* and *ceaList1* follow the line of summarizers using semantic resources and frequency analysis as their main ranking feature. Frequency concerns not only to look for recurrent information, but to reduce redundancy [NV05, CG98]. Semantic resources used in automatic summarization go from WordNet [HyL98] to domain-specialized thesaurus [RHB07]. System *ceaList2* relies on an approach that was previously developed in [GSB$^+$06, BFB$^+$07] and takes place among the works based on lexical similarity.

3 Extracting word senses from lexical co-occurrences

The bag-of-senses approach that underlies the sentence extraction component of two of our three systems requires a reference repository of word senses. WordNet [Fel98] is of course an obvious solution to this problem but we didn't adopt it for three main reasons:

- first, WordNet-like resources exist for different languages but few are developed enough for being used in an effective way. For instance, the EuroWordNet for French, a language we are specifically interested in, is far from being as elaborate as the Princeton's WordNet;

- secondly, even if a WordNet-like network is available for a target language, we can't assume that it will cover accurately all possible topics and especially specialized domains such as the medical domain or the nuclear energy domain;

- finally, WordNet's senses are mainly defined through their relations with other WordNet's senses. As a consequence, WordNet doesn't offer much information about the context in which a sense occurs. It is why most of word sense disambiguation systems need annotated corpora for acquiring this kind of information. In our case, we don't perform word sense disambiguation strictly speaking but we tackle a closely related issue as we want to select the word senses that are the most representative of a piece of text.

More specifically, we have chosen to achieve this selection by relying on a repository of word senses that were automatically discovered from a corpus following the method described in [Fer04]. This method is based on the hypothesis that each sense of a word appears in specific contexts and as a consequence, discovers the senses of a word by clustering its co-occurrents.

3.1 From documents to co-occurrences

The discovering of word senses from a corpus starts in [Fer04] with the linguistic pre-processing of its documents. This pre-processing represents each of them as the sequence of its lemmatized plain words, that is, nouns, verbs and adjectives. After filtering low frequency words (frequency < 10), the co-occurrents of the remaining words are classically collected by recording the co-occurrences in a fixed-size window (19 plain words) moved over the pre-processed texts. The result of this first step is a large set of lexical co-occurrences that can be viewed as a lexical network in which the similarity between two words is given by the pointwise mutual information measure.

3.2 From co-occurrences to word senses

The building of the senses of a word starts by delimiting from the above network the subnetwork that is made of its co-occurrents and the co-occurrence relations between them. Moreover, theses relations are filtered by applying thresholds both on their frequency and their cohesion. The resulting graph is considered as a similarity graph and then clustered by relying on an adaptation of the Shared Nearest Neighbor (SNN) algorithm described in [ESK01]. This algorithm particularly fits our needs as it automatically determines the number of clusters – in our case the number of sense of a word – and does not take into account the elements that are not representative of the clusters it builds.

Algorithm 1 SNN algorithm

1. sparsification of the similarity graph
2. building of the SNN graph
3. computation of the distribution of strong links
4. search for topic seeds and filtering of noise
5. building of word senses
6. removal of insignificant word senses
7. extension of word senses

The SNN algorithm (see Algorithm 1) performs clustering by detecting high-density areas in a similarity graph. In our case, the similarity graph is directly built from the similarity matrix: each vertex represents a co-occurrent and an edge links two co-occurrents whose similarity is not null.

This first stage of the SNN algorithm starts by sparsifying the similarity graph, which is done by keeping only the links towards the k (k=10) most similar neighbors of each co-occurrent (step 1). Figure 1 shows the resulting graph for the word *mouse*. Then, the similarity graph is transposed into a shared nearest neighbor (SNN) graph (step 2). In this graph, the similarity between two co-occurrents is given by the number of direct neighbors they share in the similarity graph. Strong links in the SNN graph are finally detected by applying a fixed threshold to the distribution of shared neighbor numbers (step 3). A co-occurrent with a high number of strong links is taken as the seed of a word sense as it is representative of the set of co-occurrents that are linked to it. On the contrary, a co-occurrent with few strong links is supposed to be outlier (step 4).

The second stage of the SNN algorithm first builds word senses by associating to sense seeds the remaining co-occurrents that are the most similar to them provided that their number of shared neighbors is high enough (step 5). Moreover, seeds that are judged as too close to each other are also grouped during this step in accordance with the same criteria. The last two steps bring small improvements to the results of this clustering. First, when the number of co-occurrents gathered by a sense is too small, this sense is judged as insignificant and it is discarded (step 6). Its co-occurrents are added to the set of co-occurrents without sense after step 5. Finally, the remaining word senses are extended by associating to them the co-occurrents that are neither noise nor already part of a sense (step 7). As senses are defined at this point more precisely than at step 4, the integration of co-occurrents that are not strongly linked to a sense seed can be safely performed by relying on the average strength of their links in the SNN graph with the co-occurrents of the sense.

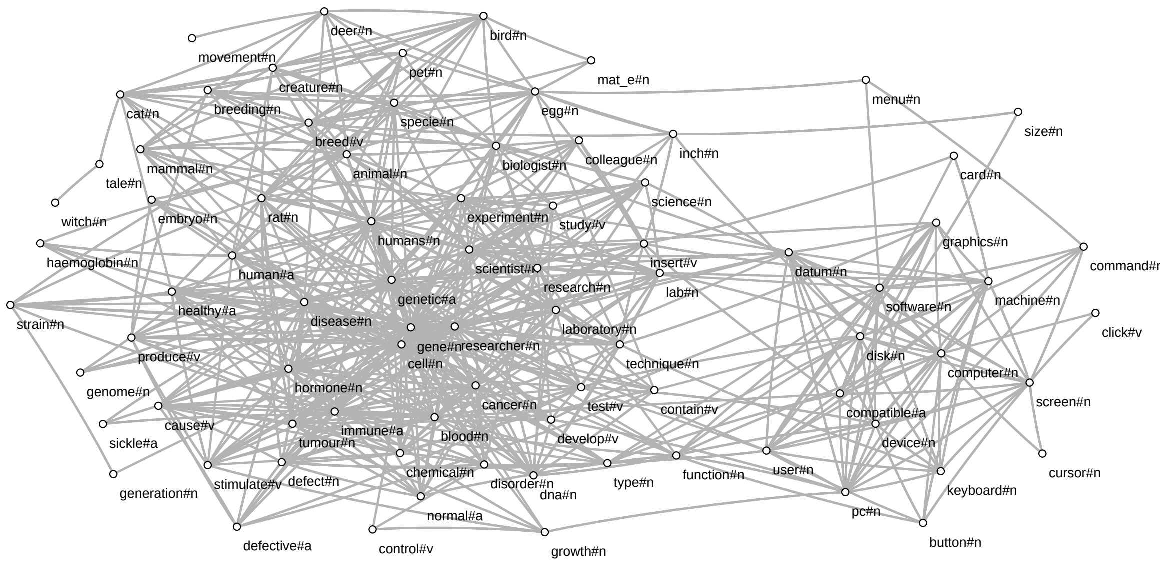

Figure 1: Similarity graph after its sparsification

3.3 Word senses for the TAC Summarization evaluation

For the TAC Summarization task, the corpus used for building our repository of word senses was made of the *Los Angeles Times part* of the TREC corpus, which represents two years of this newspaper. Table 1 gives some statistics about the vocabulary we have taken into account and about the resulting word senses. We can notice that the percentage of words without any word sense is quite high but could certainly be increased by adapting more dynamically thresholds for filtering co-occurrence relations.

	nouns	**adjectives**	**verbs**	**all**
#words	16,604	8,872	4,946	30,422
#words with senses	6,066 (37%)	2,084 (23%)	1,688 (34%)	9,838 (32%)
#senses by word	2.2	1.6	1.8	2.0

Table 1: Statistics about the results of the word sense discrimination procedure

Table 2 shows as an example the two senses distinguished by the method we have presented for the word "mouse" and illustrates the fact that this method is rather interesting for dealing with this kind of homonymy.

mouse-device	compatible, software, computer, machine, user, desktop, pc, graphics, keyboard, device
mouse-animal	laboratory, researcher, cell, gene, generic, human, hormone, research, scientist, rat

Table 2: Two senses of the word "mouse"

4 Semantic approach: bag-of-senses and nominal dependencies

Systems *ceaList3* and *ceaList1* are both avatars of Morcas summarizer [CBF$^+$05]. The idea that gave birth to Morcas was to rewrite the *tf*idf* hypothesis in semantic terms. It is well known that, when applied to the problem of automatic summarization, the *tf*idf* hypothesis states that relevance is a numerical function of word's frequencies [SJ07]. Instead of defining relevance in lexical terms, Morcas defines it as a function of word "senses". The underlying idea is that the frequency of a set of semantically related words is more significant than the frequency of isolated words. Therefore, we suppose that the recurrent occurrence of certain senses in a text is an indicator of a relevance and can be used as a weighting measure.

4.1 Bag-of-senses

The main semantic resource of Morcas is a base of word senses. Morcas reads from this base the definition of the senses of each plain word from the source text (nouns, verbs, adjectives). Each sense of the word is projected on the source text in order to weight its relevance. For instance, the word *party* has three senses: the first one is defined with political words (*republican party, democrat party*); the second bag contains "champagne" words (meaning *pleasure gathering*; the third sense is an specialization of the latter, referring more precisely to people participating to the party (*invited, hostess, caterer, party-goer*).

Each occurrence of a word that is part of the definition of a sense of the word *party* will increase the relevance of that sense. Morcas calculates the relevance of each sense in order to identify salient topics of the source text. Relevant senses give a higher weight to the sentences that refer to them (see section 4.3).

4.2 Morcas pipeline

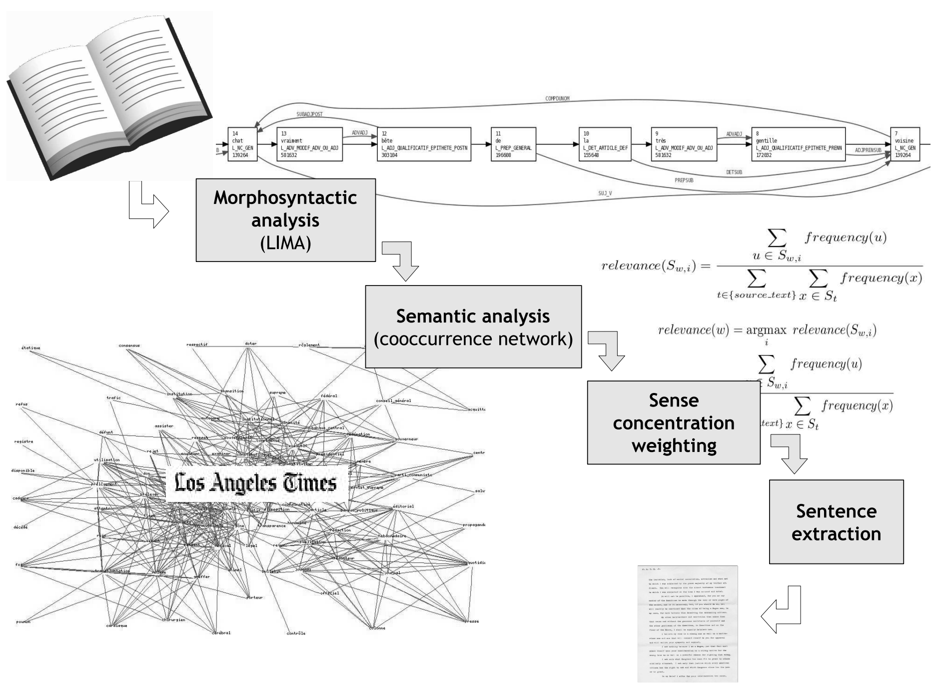

Figure 2: *ceaList3* pipeline

The first step in Morcas chain of treatment is a morpho-syntactic analysis of the source text (see Figure 2). This analysis is necessary because the base of senses stores lemmatized forms of words plus the grammatical category. Syntactic analysis is performed by LIMA (Lic2M Multilingual Analyzer) by means of a dependency grammar [BdC05]. LIMA makes also the segmentation of the source text in paragraphs and sentences.

The result of this first step is a morpho-syntactic graph which represents both the linear succession of phrases and the dependency tree. LIMA also annotates named entities, dates and acronyms. The second step in the pipeline

Dependency	Description	Example
COMPDUNOM	noun's complement	*capital of the country*
SUBADJPOST	adjective after a noun	*fuel efficient*
ADJPRENSUB	adjective before a noun	*political issue*
SUBSUBJUX	noun in juxtaposition	*mountain bike*
SUBSUBPOS	possessive nouns	*government's acceptance*
ATB_S	noun's attribute	*unprecedented crisis*

Table 3: Maximum load and nominal tension

is the semantic analysis. Every noun, verb, adjective and adverb in the source text is treated in order to get the definition of its senses from the base of senses. Instead of counting it individually, the occurrence of a word increases the count of all the bag-of-senses that define it. This analysis produces a twofold result: on the one hand, Morcas identifies predominant bags of senses; on the other hand it keeps a count of all the semantic bags that got at least one mention in the source text. The goal of the next steps is to identify those sentences concentrating relevant senses. *ceaList3* (Morcas original approach) relies only in the bag-of-senses criteria to calculate semantic concentration. Systems *ceaList1* fine-tunes the *ceaList3* calculation by including nominal dependencies (that is, those syntactic dependencies necessary to structure noun phrases).

4.3 System *ceaList3*

Given a word lemma w, $frequency(w)$ is the number of times that w appears in the source text. $\{S_{w,1}, S_{w,2}, ..., S_{w,n}\}$ represents the set of definitions of w's senses provided by the base of senses. For instance, the word $mouse$ has two senses: an animal one (*laboratory, genetics...*) and a technological one ($software, computer...$). *ceaList3* computes the frequency of each word defining a sense of w:

$$frequency(S_{w,i}) = \sum_{u \in S_{w,i}} frequency(u)$$

The relevance of a given sense $S_{w,i}$ is calculated by normalizing the frequency of each word of its definition against the frequency of all the words found in the source text:

$$relevance(S_{w,i}) = \frac{\sum\limits_{u \,\in\, S_{w,i}} frequency(u)}{\sum\limits_{t \in \{source\_text\}} \sum\limits_{x \,\in\, S_t} frequency(x)}$$

Therefore, the relevance of w is the relevance of its predominant sense:

$$relevance(w) = \underset{i}{\mathrm{argmax}}\ relevance(S_{w,i})$$

and the sentence relevance is the sum of the relevance of each word, normalized against the sentence size:

$$relevance(sentence) = \frac{\sum\limits_{w \in sentence} relevance(w)}{|\{w \in sentence\}|}$$

4.4 System *ceaList1*

Morcas summaries are intended to support information research on documents from the nuclear energy domain. A first evaluation in french language [GFdC08] showed that Morcas got low scores summarizing scientific documents (articles, PhD thesis). *ceaList1* modifications were motivated by the need of fine-tuning the bag-of-senses approach to improve scientific discourse summarization. Our nominal dependencies hypothesis is a consequence of this need. We suppose that those sentences with the highest amount of nominal dependencies are indicators of conceptual complexity, so we can considered them as more relevant. Table 3 shows the LIMA syntactic dependencies that we use as a tuning factor for relevance assessment.

Nominal dependencies hypothesis is motivated by the fact that the syntactic structure of scientific language is characterized by a high use of structured concepts. To tackle *ceaList3* preference for short sentences, we use the number of nominal dependencies as a tuning factor for the relevance assessment formula.

$$relevance(sentence) = \frac{\sum_{w \in sentence} relevance(w)}{|w \in sentence|} * nominal\_deps(sentence)$$

5 Lexical similarity approach: compactness, a score using density and proximity

5.1 System *ceaList2*

System *ceaList2* relies on an extraction score method (called *compactness*) that was developed for a previous work in Question-Answering (QA) [GSB+06]. This score was used to select factual answers to factual question according to semantic features: on the one hand the expected answer type derived from a question; on the other hand the corresponding named entity found inside a passage. The underlying idea of the *compactness* score for QA is that the best candidate answer is closely surrounded by relevant words found in the question. Any word that is not included in the question can disturb the relationship between a candidate answer and its responsiveness to a question. Thus, *compactness* can be seen as a lexical similarity measure, which depends on density and proximity. *ceaList2* approach includes different steps:

- A topic is seen as a bag-of-words, lemmatized and stop-listed. Remaining words (or lemmas) compose a $TSet$.
- Each sentence of a document is considered as an answering fragment. Unlike the QA task, no semantic answer type is associated to the sentences, therefore we consider each item inside the $TSet$ as a kind of answer candidate, and compute its *compactness* score. The best *compactness* score gives us a centered window of the most interesting phrases of the sentence. This score is extrapolated to the full sentence score. This extrapolation is made in sake of linguistic coherence, but *ceaList2* clearly lacks some component to remove other uninteresting phrases from this interesting window (moreover, from an extraction point of view, the system should have kept only content inside this window).
- Sentences are chosen from their *compactness* score until they reach the n-words limit ($n = 100$).

For each $x_i \in TSet$, inside each $Sentence$, *compactness* score is computed as follow:

$$compactness(x_i) = \frac{\sum_{\substack{y \in TSet \\ y \neq x_i}} p_{y_m,x_i}}{|TSet|}$$

with y_m being the occurrence of y in the current $Sentence$ which maximizes:

$$p_{y_m,x_i} = \frac{|W|}{2R+1}$$

and where:

$$R = distance(y_m, x_i)$$
$$W = \{z | z \in TSet \cap Sentence, \text{distance}(z, x_i) \leq R\}$$

The *compactness* score is computed by searching for the one with the best contribution y_m rather than the nearest occurrence of y.

6 The update summarization task

Adaptations of these systems were needed in order to participate in TAC 2008 update summarization task. Neither of our systems were intended for multi-document summarization; so we had to treat multi-document input as one long document with multiple chapters.

Neither *ceaList3* nor *ceaList1* were able to process an input topic statement; so the sense concentration algorithm was modified as well. Instead of projecting the whole source text onto the base of senses, we projected only the topic statement, getting a limited set of relevant senses. In other words, instead of calculating sense concentration of all the source text topics, we limit the analysis to senses issued from the input topic statement.

SET A (58 systems)						
	ceaList1		**ceaList2**		**best**	**best**
Metric	**score**	**rank**	**score**	**rank**	**system**	**human**
Overall responsiveness	2.13	(43)	2.35	(30)	2.79	4.79
Linguistic quality	2.38	(35)	2.19	(42)	3.00	4.92
Pyramid	24.2%	(39)	26.3%	(33)	35.9%	84.1%

Table 4: *ceaList* results on Set A (manual evaluation)

SET B (58 systems)						
	ceaList1		**ceaList2**		**best**	**best**
Metric	**score**	**rank**	**score**	**rank**	**system**	**human**
Overall responsiveness	2.21	(19)	1.96	(37)	2.60	4.88
Linguistic quality	2.46	(25)	2.29	(33)	3.21	4.96
Pyramid	21.8%	(28)	20.8%	(32)	33.6%	76.1%

Table 5: *ceaList* results on Set B (manual evaluation)

New information research is not a priority for the further developments of these components (which, as it was mentioned above, will be part of a long document summarizer and a question answering system). Because of time and human resources limitations, no particular development was made to look for new information other than to treat documents documents from set A in chronological order (from older to newer) and documents from set B in inverse chronological order (from newer to older). No particular modifications were made to *ceaList2* for the update task, where sets A and set B were summarized in an independent way.

SET A (72 systems)							
	ceaList1		**ceaList2**		**ceaList3**		**best**
Metric	**score**	**rank**	**score**	**rank**	**score**	**rank**	**system**
Overall responsiveness	0.106	(53)	0.114	(44)	0.095	(66)	0.143
Linguistic quality	0.070	(51)	0.078	(43)	0.056	(68)	0.111
Pyramid	0.043	(43)	0.043	(42)	0.032	(63)	0.064

Table 6: *ceaList* results on Set A (automatic evaluation)

SET B (72 systems)							
	ceaList1		ceaList2		ceaList3		best
Metric	**score**	**rank**	**score**	**rank**	**score**	**rank**	**system**
Overall responsiveness	0.108	(42)	0.109	(40)	0.109	(56)	0.137
Linguistic quality	0.069	(38)	0.069	(37)	0.056	(58)	0.101
Pyramid	0.047	(35)	0.046	(36)	0.037	(53)	0.076

Table 7: *ceaList* results on Set B (automatic evaluation)

7 Conclusion and further work

Manual evaluation suggests that the bag-of-words approach (*ceaList2*) fits better for the generic summarization task (set A) while the bag-of-senses approach (*ceaList3*) gets higher scores on the update summarization task. Summaries produced using the bag-of-words approach got higher readability scores on both generic and update summarization, probably as a consequence of the nominal dependencies feature, which tends to extract longer, syntactically richer sentences.

Our best rank was 19 for Overall Responsiveness in update summarization (bag-of-senses), with no significant difference at the 95% confidence level with the best system. However, the same run was ranked 43 for Overall Responsiveness for set A. This gap might be caused by the fact that input texts from set A where treated in chronological order while those from set B where treated in inverse chronological order, that is from the most recent to the oldest documents.

Our systems were handicapped by the somehow brutal way in which a mono-document summarization strategy was modified to treat a multi-document input. The lack of any redundancy processing was penalizing as well. The bag-of-words approach *ceaList2* considers the TAC'08 summarization task from a certainly reductive perspective, as it only performs sentence selections by using compactness scores rather than a real summarization task, but could be useful if integrated to a more complex approach combining different features.

Automatic evaluation confirms our observations: bag-of-words for generic summarization and bag-of-senses for more complex tasks. Results from Basic Elements (BE) also confirms this observation but in both ROUGE evaluations, *ceaList2* gets much better scores than *ceaList1*, which points out the fact that, being based on lexical comparisons, *ceaList2* extracts have a higher word-by-word similarity with the topical statement.

Comparing two different versions of the bag-of-senses approach (*ceaList1* and *ceaList3*) it is clear that nominal dependencies are a valuable feature: *ceaList3* (without nominal dependencies) gets lower scores in every evaluation and for ROUGE 2 (set B), this difference is significance at the 95% confidence level.

TAC 2008 results showed us that the *ceaList2* and *ceaList1* were complementary approaches: one could be used for relevant semantic fragments extraction; the other one both for similarity and redundancy analysis. Further work will include the fusion of these criteria into a first-stage sentence extractor that will be connected to reformulation and syntactic compression modules. Other bases of senses are being developed for specialized domains (in particular, for the nuclear energy field).

References

[BdC05] Romaric Besançon and Gaël de Chalendar. L'analyseur syntaxique LIMA dans la campagne d'évaluation EASY. In *Actes de la 12ème conférence sur le Traitement Automatique des Langues, TALN 2005*, 2005.

[BFB+07] Florian Boudin, Benoît Favre, Frédéric Béchet, Marc El-Bèze, Laurent Gillard, and Juan-Manuel Torres-Moreno. The LIA-Thales summarization system at DUC-2007. In *DUC-NIST Proceedings Document Understanding Conference (DUC-2007)*, 2007.

[CBF$^+$05] Gaël de Chalendar, Romaric Besançon, Olivier Ferret, Gregory Grefenstette, and Olivier Mesnard. Crosslingual summarization with thematic extraction, syntactic sentence simplification, and bilingual generation. In *Workshop on Crossing Barriers in Text Summarization Research, 5th International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*, 2005.

[CG98] Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR'98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, 1998.

[ESK01] Levent Ertöz, Michael Steinbach, and Vipin Kuma. Finding topics in collections of documents: A shared nearest neighbor approach. In *Text Mine'01, Workshop of the 1st SIAM International Conference on Data Mining*, 2001.

[Fel98] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[Fer04] Olivier Ferret. Discovering word senses from a network of lexical cooccurrences. In *20th International Conference on Computational Linguistics (COLING 2004)*, pages 1326–1332, 2004.

[GFdC08] Jorge Garcia-Flores and Gaël de Chalendar. Syntactico-semantic analysis: a hybrid sentence extraction strategy for automatic summarization. In *MICAI 2008: Advances in Artificial Intelligenc*. Springer, Lecture Notes in Artificial Intelligence, 2008.

[GKMC99] Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Research and Development in Information Retrieval*, pages 121–128, 1999.

[GSB$^+$06] L. Gillard., L. Sitbon, L. Blaudez, P. Bellot, and M. El-Bèze. Relevance measures for question answering, the LIA at QA@CLEF-2006. In Carol Peters, Paul Clough, Fredric C. Gey, Jussi Karlgren, Bernardo Magnini, Douglas W. Oard, Maarten de Rijke, and Maximilian Stempfhuber, editors, *CLEF*, volume Lecture Notes in Computer Science of *4730*, pages 440–449. Springer, 2006.

[HyL98] Eduard Hovy and Chin yew Lin. Automated text summarization and the summarist system. In *Proceedings of the TIPSTER Text Program*, 1998.

[KSH01] Wessel Kraaij, Martijn Spitters, and Martine Van Der Heijden. Combining a mixture language model and naive bayes for multi-document summarisation. In *Proceedings of the DUC 2001 workshop*, pages 95–103, 2001.

[Luh58] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[Man01] Inderjeet Mani. *Automatic Summarization.* John Benjamins Publishing Company, 2001.

[NV05] Ani Nenkova and Lucy Vanderwende. The impact of frequency on summarization. Technical Report MSR-TR-2005-101, Microsoft Research, 2005.

[RHB07] Lawrence H. Reeve, Hyoil Han, and Ari D. Brooks. The use of domain-specific concepts in biomedical text summarization. *Information Processing and Management*, 43(6):1765–1776, 2007.

[RJST04] Dragomir R. Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938, 2004.

[SJ07] Karen Spärck Jones. Automatic summarising: The state of the art. *Information Processing and Management*, 43(6):1449–1481, 2007.