

# Generating Update Summaries : Using an Unsupervised Clustering Algorithm to Cluster Sentences

Aurélien Bossard

Laboratoire d'Informatique de Paris-Nord (UMR 7030, CNRS et U. Paris 13)  
99, av. J.-B. Clément  
93430 Villetaneuse

**Abstract.** In this article, we present a summarization system dedicated to update summarization. We first present the method on which this system is based, CBSEAS, and its adaptation to the update summarization task. Generating update summaries is a far more complicated task than generating "standard" summaries, and needs a specific evaluation. We describe TAC 2009 "Update Task", which we used in order to evaluate our system. This international evaluation campaign allowed us to confront our system to others automatic summarization players. Finally, we show and discuss the interesting results obtained by our system.

**Keywords:** automatic summarization, update summarization, sentence clustering, sentence extraction

## 1 Introduction

During the past decade, automatic summarization, supported by evaluation campaigns and a large research community, has shown fast and deep improvements. Indeed, the research in this area is led by strong industrial needs: fast processing despite ever increasing amount of data. As the field of automatic summarization widens, the applications become more varied. We can quote e-mail streams, blogs, scientific articles or newswire articles as application fields, and opinion-oriented, update and differential summaries as different kinds of summaries.

And update summary is based on the assumption that the user who needs the summary has already read some documents. The update summary has to summarize what is new to the user in the documents to be summed up. This automatic summarization task is strongly supported by industrial partners, as it meets real needs.

In this paper, we present our research in automatic update summarization: we first developed an automatic standard summarization system, and adapted it to update summarization. This system is based on sentence clustering, which efficiently improves the informational diversity in the summaries.

Automatic generation of update summaries has been proposed as a task of DUC<sup>1</sup> 2007, TAC<sup>2</sup> 2008 and 2009 evaluation campaigns. This task provided the evaluation of two different kinds of summary: standard request-guided summary and update request-guided summary, and allows us to estimate the quality of both of our summarization system and its adaptation to automatic generation of update summaries.

This article is based on a generic multi-document summarization system, CBSEAS[1], which differs from other similar systems by trying to use redundancy in order to produce better summaries. Redundancy is indeed crucial for multi-document summarization: we put forward the hypothesis that the most repeated pieces of information are the most important. Also, detecting the sentences that do not convey the same piece of information can help to create summaries with a better informational diversity. Apart from the standard summarization problem, the other main issue in this article is the update management: how can we distinguish already known from new information?

This article shows how the different aspects of these two issues are processed. We also present the reassessment of our system during TAC 2008 and 2009 evaluation campaigns. The evaluation of automatic summaries is still an open research area. Several evaluation metrics are used, about which we discuss as they differ in the method and in the aspects of the summaries they try to evaluate.

The structure of this article is the following one: we first give a quick overview of the state of the art. We then describe our generic summarization system, CBSEAS, and the adaptations made to manage the updating problem. Finally, we describe the evaluation and give the details of the results obtained on TAC 2008 and 2009 Update task.

## 2 State of the Art

In this section, we present an overview of existing methods for automatic summarization and update management. These fields have been widely explored, so we limit this overview to the main work and the work which inspired our system.

### 2.1 Overview of automatic multi-document summarization

Automatic summarization is being studied since the beginning of data processing. Automatic summarization process based on advanced linguistic theories has soon proven to be too ambitious: indeed, it requires paraphrases recognition which involves the need of complex linguistic resources, and text generation processes which are still in an exploratory stage. Recently, some researches such as those of Marcu [22] tried to analyse the rhetorical structure prior to the sentence selection process, but this method is still theoretically limited to specific area.

---

<sup>1</sup> Document Understanding Conference: <http://www-nlpir.nist.gov/projects/duc/index.html>

<sup>2</sup> Text Analysis Conference: <http://www.nist.gov/tac>

Therefore, right from the 1950's [20], research in automatic summarization has focused on the excerpt of important sentences – creation of extracts – rather on the generation of abstracts. The extracted sentences have to constitute a coherent text which is faithful to the ideas/information expressed in the original documents. Sentence extraction basically consists on scoring each sentences from the documents to be summarized, and to extract those which get the highest scores in the summary. The number of sentences or words in the summary can be set in advance, but can also be dynamically set using a compression ratio – for example 10% of the original documents.

Edmundson [10] defined textual clues which can be used to determine the importance of a sentence. In particular, he set a list of cue words, such as "hardly" or "impossible", and used term frequency, sentence position and the number of words occurring in the title. These clues are still used by recent systems, like the one of Kupiec [17].

Other systems focus on term frequency. Luhn [20] led the way of frequency-based sentence extraction systems. He proposed to build a list of important terms. The importance of a term depends on whether or not its frequency belongs to a predefined range. The more a sentence presents words belonging to this list, the more important it is. Radev [24] took advantage of the advances in text statistics by integrating the tf.idf metric to Luhn's method. The list of important terms, that Radev calls "centroid", is composed of the  $n$  terms with the highest tf.idf. The sentences are ranked according to their similarity to the centroid. The clue-based and term frequency-based methods are efficient when selecting the sentences which reflect the global content of the documents to be summed up. Such a sentence is called "central". However, these methods are not designed to generate good summaries according to informational diversity. Now, informational diversity is almost as important as centrality when evaluating a summary. Indeed, a summary should contain all the important pieces of information which should not be repeated.

In order to deal with diversity, the MMR [4] method – for Maximum Margin Relevance – selects iteratively the sentence which maximizes the score function shown in (1). This score function works with a user query, which means that the generated summaries are guided towards the informational need of a user. The MMR score function takes into account diversity by subtracting the similarity of the evaluated sentence to already selected sentences from its centrality score. This method has been widely used and adapted to different summarization tasks [14, 5, 25, 28].

$$MMR = \operatorname{argmax}_{P_i \in D \setminus S} \left[ \lambda \operatorname{sim}_1(P_i, Q) - (1 - \lambda) \operatorname{argmax}_{P_j \in S} \operatorname{sim}_2(P_i, P_j) \right] \quad (1)$$

where  $Q$  is the user query,  $D$  the sentences to summarize,  $S$  the already selected sentences, and  $\lambda$  the novelty factor.

Redundancy in multi-document summarization is a good clue of the importance of a piece of information. MMR does take redundancy into account, but only in order to filter sentences and not as an extraction criterion. Radev took advantage of the recent advances in social networks analysis to use information redundancy as the main criterion to judge sentence importance for automatic summarization. He builds a graph of the documents to be summarized, in which the nodes are the sentences, and the edges the sentences similarities. He then uses the *prestige* notion as in the social networks area in order to extract the most important sentences. The nodes with the highest *prestige* are those which are strongly linked to other high *prestige* nodes.

All the methods we presented in this overview consider the global content of the documents to summarize when evaluating the sentences' centrality. Yet we consider the documents not as a whole, but as multiple clusters of sentences grouped according to their informational content. In each of these clusters, central sentences emerge, which are those that we want to extract.

## 2.2 Overview of update summarization systems

DUC 2007 and TAC 2008 "Update task" revealed that generating update summaries is a far more complex task than generating "standard" summaries [7]. We here present different strategies aiming to manage the update summarization.

Some authors, such as Galanis and Malakasiotis [12] remove from the update documents all the sentences of which similarity to a sentence of the initial documents is beyond a predefined empirical threshold. Others chose to work on a global similarity between the two sets of documents [15]. The empirical sentence similarity threshold constitutes a bias that the authors have chosen to move to a more global view on the documents. Sentences are iteratively removed from the update cluster until the similarity between the update cluster and the initial one falls under a threshold.

The method presented in [3] selects the sentences for the update summary using the MMR method described above. The weight of dissimilarity to the already selected sentences in the scoring function has been increased in order to ensure that the extracted sentences do not carry information that the user has already read.

Another method, exposed in [27], aims to evaluate the novelty of a word. The novelty factor ( $fn$ ) of a word in a document published at a date  $t$  depends on its number of occurrences in the previous documents and its number of occurrences in the later documents, as shown in (2).

$$fn(w) = \frac{|nd_t|}{|pd_t| + |D|} \quad (2)$$

$$\begin{aligned} nd_t &= d : w \in d \wedge t_d \leq t \\ pd_t &= d : w \in d \wedge t_d > t \\ D &= d : t_d \leq t \end{aligned}$$

The novelty factor is used to measure sentence novelty. Though this method has proven to be efficient, be it during TAC 2008 or during TAC 2009. However, we want to evaluate a novel method based on the similarity between sentences, which doesn't need to empirically set a similarity threshold.

### 3 CBSEAS, a Generic Approach for Automatic Multi-Document Summarization

We want to specifically manage the multi-document aspect by considering redundancy as the main issue of multi-document summarization. Indeed, we consider the documents to summarize as made up by groups of sentences carrying the same information. In each of these clusters, one sentence can be considered as central. Extracting this sentence, and not another one, in every cluster can lead to summaries in which the risk of redundancy is minimized. The summaries generated with this method may carry a good informational diversity.

Our approach implements this method. The first step is to cluster sentences, and then select one sentence per cluster.

#### 3.1 Sentence Clustering

We here describe the first part of our algorithm: sentence clustering. We wanted a flexible clustering algorithm in which we could easily adapt the clustering criterion. *Fast global k-means* seemed to be appropriated for that purpose. It takes a similarity or dissimilarity matrix as input. The model created after the sentence clustering cannot be used only to extract sentences, but also for post-processing such as sentence ordering, which will be the subject of future publications.

**Pre-processing** The input documents undergo preprocessings before they can be processed by CBSEAS. We here present the different preprocessings used by our system.

*POS Tagging* The documents are morphosyntactically analysed and the textual units tagged by *tree-tagger*<sup>3</sup> [26]. It enables the differentiation of morphosyntactic types during the sentence similarity computation, which should not be only considered as a simple similarity computation between sets of elements. In fact, natural language processing involves syntactic and semantic knowledge.

*Sentence Segmentation* Some authors choose to work on small textual structures rather on sentences. So they work on the extraction of groups of syntactically related words, dividing sentences in clauses [22]. Extracting clauses rather than sentences leads to the problem of clauses identification – though automatic syntactic analysis has recently greatly improved – and clauses independence. Other authors choose to extract entire paragraphs in order to improve the linguistic

---

<sup>3</sup> *tree-tagger* webpage: <http://www.ims.uni-stuttgart.de/projekte/complex/TreeTagger/>

$$sim(s_1, s_2) = \frac{\sum_{mt} weight(mt) \times fsim(s_1, s_2)}{\frac{fsim(s_1, s_2) + gsim(s_1, s_2)}{\sum_{mt} weight(mt)}} \quad (3)$$

$$fsim(s_1, s_2) = \sum_{n_1 \in s_1} \sum_{n_2 \in s_2} tsim(n_1, n_2) \times \frac{tfidf(n_1) + tfidf(n_2)}{2} \quad (4)$$

$$gsim(s_1, s_2) = card((n_1 \in s_1, n_2 \in s_2) \mid tsim(n_1, n_2) < \delta) \quad (5)$$

where  $mt$  are the morphological types,  $s_1$  and  $s_2$  the sentences,  $tsim$  the similarity between two terms using WordNet and the JCN similarity measure [16] and  $\delta$  a similarity threshold.

---

coherence of the summaries. However, it increases the risk of extracting non pertinent sentences, as whole paragraphs are extracted.

We made the choice to extract whole sentences in order to avoid generating ungrammatical summaries and extracting irrelevant sentences.

*Named Entity Tagging* Named entity tagging enables the refinement of the sentence similarity computation. Indeed, the tagging enables to take into account a complex lexical group such as "President George W. Bush" as a unique lexical entity. Such a group should in fact be identified as a single named entity. So will it be considered as a single term, not as four distinct terms. Named entities are tagged using GATE architecture and ANNIE system [6].

**Sentence Similarity Computation** We put forward the hypothesis that sentence similarity must take into account the type of documents which CBSEAS has to summarize and the kind of summary asked by the user. For example, the characteristics that will determine if two sentences are similar will differ whereas it is for opinion summarization or for market analysis summarization. In the first case, adjectives, adverbs and sentiment verbs are discriminant; in the second case, it will be currencies, amounts, and company names.

We want to take this fact into account by using a parameterizable similarity measure, which can easily be adapted to the different tasks to which a summarization system can be confronted. We also want to take into account linguistic relationships between terms – *e.g.* synonymy, hyperonymy. For that purpose, we used the WordNet database [11] and the JCN similarity measure [16] which is based on the distance between terms in the WordNet taxonomy. Eq 3, 4, and 5 present this measure.

*Clustering Algorithm* Once the similarity matrix has been computed, CBSEAS automatically clusters the sentences, grouping together semantically close ones. This step is achieved using *fast global k-means* algorithm [18], an incremental variant of the well known *k-means* clustering algorithm [21]. *Fast global k-means*

avoids the problematical choice of the  $k$  initial cluster centers. The incrementality of *fast global k-means* makes it also interesting for the purpose of generating update summaries.

*Fast global k-means* first creates one cluster that contains all the elements. At each step of the algorithm, a news cluster is creating, whose center is the farthest element to its cluster center. Each element is then placed in the cluster of which it is the closest to the center. The clusters centers are then reinitialized. The algorithm stops when it has created the number of clusters asked by the user.

Figure 1 presents a cluster generated by CBSEAS. Words shared by at least a half of this cluster's sentences are boldfaced in order to identify the reason of the clustering.

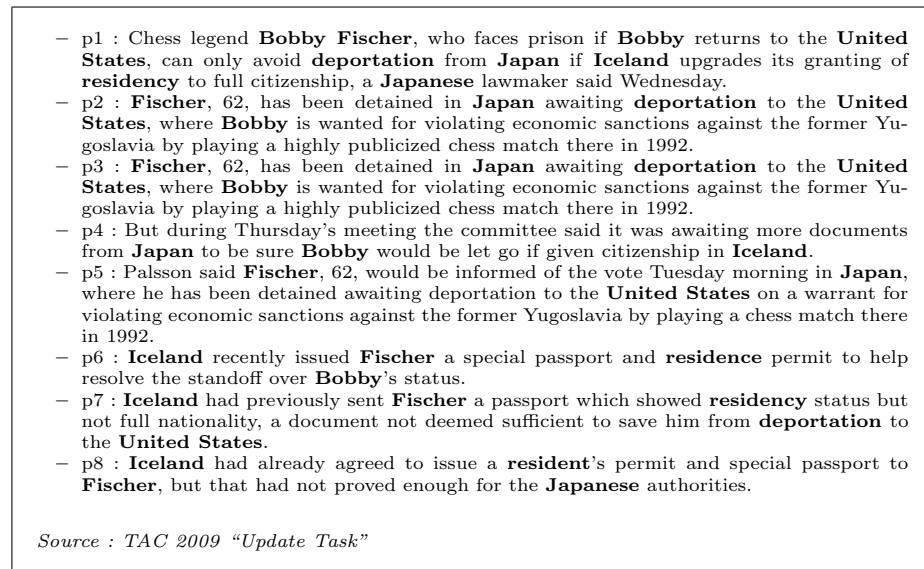


Fig. 1. Example of a cluster generated by CBSEAS

### 3.2 Sentence Selection

After the sentences clustering step, CBSEAS extracts one sentence per cluster. Let us remind the reader that the sentences thus extracted should minimize the information redundancy, and so provide a summary with a good informational diversity. The extraction criterion is as important as the clustering. In fact, the way the sentences are extracted does influence the centrality of the summary.

We here present the three criteria used by CBSEAS to extract sentences: local centrality, global centrality, sentence length, and the way they are encoded. The

final score of a sentence is computed with a weighted sum of these three scores. The weights are set using a genetic algorithm which is detailed in [2].

**Local Centrality** Local centrality is the relevance of a sentence to its cluster’s content. We want the extracted sentences to reflect the most the information of their cluster. The idea behind the local centrality is the following one: the overall sentences of a cluster  $C$  express a set of pieces of information which we name  $I$ . The most central sentence in  $C$  is the one which contains the most important pieces of information of  $I$ . We work under the assumption that information redundancy is correlated to information importance. The sentence which maximizes the sum of similarities to the other sentences,  $s_{max}$ , is the most central sentence. It receives 1 as local centrality score. The other sentences receive a local centrality score equal to their similarity to  $s_{max}$ . Figure 2 illustrates this measure.

**Global Centrality** The major problem of extracting sentences exclusively upon local centrality measure is that it does not take into account the global content nor a user query (if one). In order to generate precise summaries which meet the informational need of a user, we must add a global centrality score to the local centrality score. For that purpose, we identify two cases:

- the user has a query, so the summary must be connected to that query;
- the user does not have a query, so the summary must be relevant to the overall content of the documents.

In the first case, we use the similarity to the request as global centrality score. This sentence to query similarity is computed same as the similarity between sentences, defined in Sect. 3.1. In the second case, we use the *centroid* score as defined in [24].

**Sentence Length** The length of the summaries is often limited to a certain amount of words. For this reason, we have chosen to give a length score to every sentence, in order to penalize too short or too long sentences. This score function is defined in (6).

$$score_{length} = \frac{1}{e^{(|length(sentence) - length_{required}|)}} \quad (6)$$

## 4 Generating Update Summaries

With the development of news feed websites, the update detection and summarization has become an important research problematic. Indeed, users tracking a topic do not want to have to read every newly published article, but only the information they have not already perused. So update summarization answers

List of the atomic pieces of information (AI) found in the Fig. 1 sentences:

AIs	Weight
Fischer faces deportation	5
Fischer chess player	4
Iceland issued Fischer a resident permit	4
Fischer violated economic sanctions against Yugoslavia	3
Iceland issued Fischer a passport	3
Fischer is 62	3
Fischer has been detained in Japan	3
Fischer faces prison	1
Fischer could avoid deportation	1
Iceland special passport can not avoid him deportation	1
Iceland wants to be sure Fischer would be let go if given citizenship	1

s2, s3 and s5 are the sentences carrying most of the AIs:

Sentences	s1	s2	s3	s4	s5	s6	s7	s8
Sum of AI weights	15	21	21	1	21	7	8	7

Sentences similarities:

	s1	s2	s3	s4	s5	s6	s7	s8	sum
s1	1.0	.171	.171	.156	.150	.1	.133	.133	2.014
s2	.171	1.0	1.0	.091	.821	.031	.094	.064	3.272
s3	.171	1.0	1.0	.091	.821	.031	.094	.064	3.272
s4	.156	.091	.091	1.0	.083	.075	.069	.077	1.642
s5	.15	.821	.821	.083	1.0	.027	.108	.055	3.065
s6	.1	.031	.031	.075	.027	1.0	.167	.315	1.746
s7	.133	.094	.094	.069	.108	.167	1.0	.115	1.78
s8	.133	.064	.053	.077	.055	.315	.115	1.0	1.812

Local centrality scores as defined in Sect. 3.2:

Sentence	s1	s2	s3	s4	s5	s6	s7	s8
Score	.171	1.0	1.0	.091	.821	.031	.094	.064

Fig. 2. Illustration of the local centrality concept

to industrial needs in content access. Moreover, as standard summarization systems now achieve a good quality in terms of informational content, research can now concentrate on more complex tasks, such as those recently proposed by the DUC and TAC evaluation campaigns: opinion summarization, update summarization, or topic-based summarization. In this section, we present our method for managing update summarization.

#### 4.1 Intuitions

CBSEAS – Clustering-Based Sentence Extractor for Automatic Summarization – clusters together semantically close sentences. In other terms, it creates different clusters for semantically distant sentences. Our clustering method can also be used to differentiate sentences carrying new pieces of information from sentences

carrying already known pieces of information, and so for managing update. In fact, sentences carrying old pieces of information are semantically close from the sentences that a user has already read.

The main weakness of such a method resides in its lack of advanced dedicated linguistic processing. For example, the sentences “After hemming and hawing and bobbing and weaving, the board of directors of Fannie Mae finally jettisoned Franklin D. Raines, the mortgage finance giant’s former chief executive, and Timothy Howard, its former chief financial officer.” extracted from the AQUAINT-2 corpus<sup>4</sup> is easily manually identifiable because of the use of “finally”. However, if using linguistic tags can be helpful for detecting update sentences, it also limits the method to a unique language, whereas we want to later apply our method to other languages.

What is more, CBSEAS has proven to be efficient at grouping together semantically close sentences and differentiate semantically far ones. In fact, CBSEAS ranked itself at the third place for avoiding redundancy on TAC 2008 “Opinion Summarization Task” [1]. This is another reason for using our clustering method to differentiate update sentences from non-update ones.

## 4.2 Update Algorithm

Before trying to identify update sentences, we need to modelize the pieces of information that the user requesting the update summary has already read. We can then confront the new documents to this model in order to determine if sentences from these documents carry new pieces of information. So the first step of our algorithm is to cluster the sentences from the documents already read by the user – which we call  $D_I$  – into  $k_I$  groups, as in Sect. 3.1 for the generation of a standard summary.

The model thus computed –  $M_I$  – is then used for the second step of our algorithm, which consists in determining if a sentence from the new documents –  $D_U$  – is to be grouped with the sentences from  $D_I$ , or to create a new cluster which will only contain update sentences. *Fast global k-means* algorithm, slightly modified, can be used to confront elements to a previously established model in order to determine if these elements can be an integral part of the model. We here describe the part of our algorithm dedicated to update.

First, our algorithm computes the similarities between sentences from  $D_U$  with the clusters centers of  $M_I$  and between all the sentences from  $D_U$ . Then it adds the new sentences to  $M_I$ , and iterates *fast global k-means* from the  $k_I$  iteration with the following constraints:

- The sentences from  $D_I$  can not be moved to another cluster; this is done to preserve the  $M_I$  model which encodes the old pieces of information. It also avoids to disturb the semantic range of the new clusters that bear novelty.

---

<sup>4</sup> The AQUAINT-2 collection is a subset of the LDC English Gigaword Third Edition composed of news articles from different press agencies

- The  $M_I$  clusters centers can not be recomputed; as the semantic range of a cluster directly depends on its center, this prevents the semantic range of  $M_I$  clusters from being changed by the integration of new elements from  $D_U$ .

The main problem of this algorithm, which is detailed in Fig. 3 relies in the choice of  $k_I$  – the number of  $M_I$  clusters – and that of  $k_U$  – the number of update clusters.

We empirically decided to set  $k_U$  to the desired number of sentences for the update summary, and  $k_I$  to  $\frac{|S_I|}{|S_U|} \times k_U$ , where  $S_I$  and  $S_U$  are respectively the sentences from  $D_I$  and  $D_U$ . For our participation to TAC 2009 “Update Task”, we have chosen to let a genetic algorithm decide the values of  $k_I$  and  $k_U$ . This algorithm, trained on TAC 2008 data, is presented in [2]. Neither of these solutions are ideal, as they suppose the existence of at least  $k_U$  new pieces of information carried by  $k_U$  different sentences. Other solutions could be conceived, such as determining if adding update clusters improves or deteriorates the quality of the clustering. A clustering quality index such as Davies Bouldin’s [9] could be used for that purpose.

Once the update clusters have been populated, the update summary is generated by extracting one sentence per update cluster, as in Sect. 3.2.

## 5 Evaluation: Participation to TAC “Update Task”

We evaluated our work under the “Update Task” of the TAC 2009 evaluation campaign conducted by the NIST<sup>5</sup>. We here present in details the task, the different metrics used for evaluation, and the results obtained by our update summarization system.

### 5.1 Detailed Description of the Task

The “Update Task” of TAC 2009 evaluation campaign requires to produce two different kinds of summaries: standard summaries and update summaries, both query-based.

The task consists of 44 topics which comport a short title, a query and two sets of documents: initial and update documents. The systems must generate two summaries for each of the document sets: a standard summary which synthesizes the initial documents content, and an update summary which synthesizes the information contained in the update documents, taking into account that the user requesting the summary has already read the initial documents. The summaries length is limited to 100 words, whatever the length of the original documents.

Each document set is constituted of 10 documents extracted from the AQUAINT-2 corpus. These documents are news articles written in english and coming from different sources: AFP, NYT, APW, LTW and Xinhua press agencies.

The queries are complex and are written in english. For example, the query of Topic D0902 is: “Describe the debate over use of emergency contraceptives, also

<sup>5</sup> NIST: National Institute of Standards and Technology

```

//MI Clustering
for all p in PI
do
  cluster(p) ← C1
for end
for i in 1 kI
do
  for n in 1 i
  do
    center(Cn) ← argmaxpj ∈ Cn ( ∑pm ∈ Cn sim(pj, pm))
  for end
  for all p in PI
  do
    cluster(p) ← argmaxCm, 1 < m < u (sim(center(Cm), p))
  for end
  if i < kI alors
    cluster(argminp ∈ DI (sim(p, center(cluster(p)))) ← Ci+1
  end if
for end
//Update detection
for all p in PU
do
  cluster(p) ← argmaxCi, 1 < i < kI (sim(center(Ci), p))
for end
for i in kI kI + kU
do
  for n in kI + 1 i
  do
    center(Cn) ← argmaxpj ∈ Cn ( ∑pm ∈ Cn sim(pj, pm))
  for end
  for all p dans PU
  do
    cluster(p) ← argmaxCm, 1 < m < i (sim(center(Cm), p))
  for end
  if i < kU alors
    cluster(argminpm ∈ DI (sim(pm, center(cluster(p)))) ← Ci+1
  end if
for end

```

**Fig. 3.** Update detection algorithm

---

called the morning-after pill, and whether or not it should be available without a prescription.”. Figure 4 presents a topic from TAC 2008 “Update Task”. The 2008 “Update Task” was unchanged in 2009.

## 5.2 The metrics used for evaluation

The NIST used three different methods for evaluating the participants runs. The first method is the widely used ROUGE<sup>6</sup> package [19]. The ROUGE metrics are based on n-gram co-occurrences between the automatic summary and reference summaries established by experts. Their main advantage resides in their complete automation. However, the evaluation of summaries cannot be limited to

<sup>6</sup> ROUGE: Recall-Oriented Understudy for Gisting Evaluation

<b>Topic D0848 : Airbus A380</b>		
Describe developments in the production and launch of the Airbus A380		
Initial documents		
16/01/2005	AFP	The Airbus A380 : from drawing board to runway-ready in a decade
16/01/2005	AFP	A380 'superjumbo' will be profitable from 2008 : Airbus chief
16/01/2005	APW	Airbus prepares to unveil 1380 "superjumbo", world's biggest passenger jet
17/01/2005	LTW	Can Airports Accomodate the Giant Airbus A380 ?
19/01/2005	AFP	After fanfare, Airbus A380 now must prove it can fly
25/01/2005	AFP	Airbus mulls boosting A380 production capacity
10/04/2005	AFP	While US government moans, airports ready for Airbus giant
27/04/2005	AFP	Paris airport neighbors complain about noise from giant Airbus A380
27/04/2005	NYT	Giant Airbus 380 makes maiden flight
04/05/2005	AFP	Airbus A380 takes off on second test flight
Update documents		
01/06/2005	AFP	Airbus announces delay in delivering new superjumbo A380
03/06/2005	AFP	German wing of Airbus denies superjumbo A380 parts delay
05/10/2005	AFP	US aviation officials to study A380 turbulence
15/10/2005	AFP	Airbus says it cannot meet demand for A380 superjumbo
18/10/2005	APW	Second Airbus A380 makes maiden flight
13/11/2005	APW	Airbus executive says company will pay millions in compensation for late A380 deliveries
17/02/2006	APW	Airbus sees no delay to A380 after wing ruptured during test
22/02/2006	AFP	Airbus confident of A380 certification
26/03/2006	APW	33 people injured in evacuation frill for A380 super-jumbo
29/03/2006	APW	Airbus A380 superjumbo passes emergency evacuation test

**Fig. 4.** Example of a topic from the TAC 2009 "Update Task".

n-grams of n-gram sequences comparison. So the NIST have chosen to used more precise evaluation methods which are not entirely automatic.

The second method used by the NIST is the Pyramid method, described in [23]. The authors define the notion of SCUs – Summarization Content Units – which are pieces of information that appear in the summaries. The Pyramid method first requires human judges to extract a list of SCUs from the reference summaries. The SCUs are then ranked according to their number of occurrences in the reference summaries, and can be seen as forming a Pyramid where the most important pieces of information are at the top and the least important at the base. The SCUs are also extracted from the evaluated summaries, and compared to the pyramid in order to obtain the Pyramid score.

The Pyramid score takes into account the linguistic quality of the summaries: if a sentence is ungrammatical, it doesn't carry any SCU. However, this evaluation method does not take into account the coherence between sentences nor the global structure of a summary. That is the reason why the NIST intro-

duced completely manual evaluation measures. They are described in [8]. The manual measures evaluate both *overall responsiveness* and *readability*. *Overall responsiveness* reflects the degree to which a summary is responding to the informational need expressed in the topic statement, considering its informational content as well as linguistic quality. The readability score reflects the fluency and structure of a summary, independantly of content. It is based on grammaticality, non-redundancy, referential clarity, focus, structure and coherence. *Overall responsiveness* and *readability* were evaluated according to a five-point scale:

- **5**: very good
- **4**: good
- **3**: barely acceptable
- **2**: poor
- **1**: very poor.

It would have been interesting to have a view on the different aspects on which the readability measure was based, as it was the case for TAC 2008 “Opinion Summarization Task” evaluation. It would have allowed us to better understand the cause of our system’s results.

### 5.3 Baselines

The NIST provided three baselines for the “Udpate Task” of TAC 2009. The first one (*Baseline 1*) consists in extracting the first sentences of the latest document until the limit of 100 words is reached. This baseline provides a lower bound of what can be achieved with an automatic summarizer.

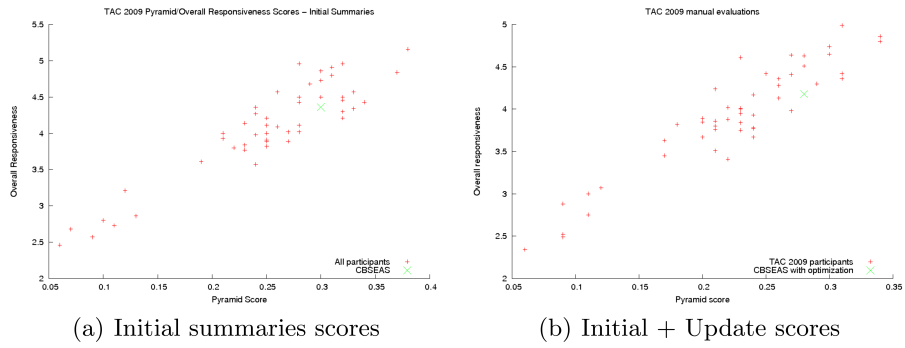
The second baseline (*Baseline 2*) is built by randomly ordering the sentences of a reference summary. It gives an overview of the sentence ordering impact on linguistic quality and overall responsiveness.

The third baseline (*Baseline 3*) is made up of entire sentences manually extracted from the documents to summarize. The extraction method is detailed in [13]. The idea behind this baseline is to provide an upper bound of what can be achieved with a purely extractive summarizer, in both terms of content and linguistic quality.

### 5.4 Results and Discussion

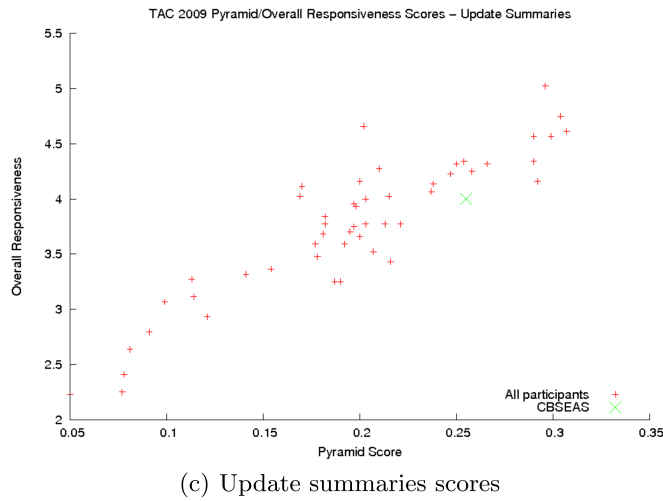
In this section, we present the results of our update summarization system, compared to the other participants’.

Figure 5 shows the pyramid evaluation and the overall responsiveness score for all the participants. Our system performs well, ranking mong the first third of participants for initial summaries, and among the 10 best for update summaries. The “overall responsiveness” score is not aas good. This is due to the poor linguistic quality of the summaries generated by our system. In fact, CBSEAS does not apply any post-processing, such as anaphora resolution or sentence ordering, which could improve the coherence of the summaries.



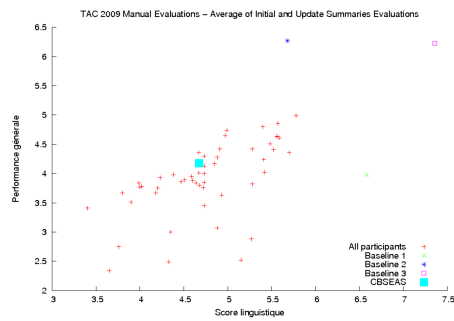
(a) Initial summaries scores

(b) Initial + Update scores



(c) Update summaries scores

**Fig. 5.** TAC 2009 “Update Task” results : Pyramid and overall responsiveness scores



**Fig. 6.** TAC 2009 “Update Task”: Overall responsiveness and linguistic scores of the three baselines and all participants

**Mean scores of initial and update summaries**

	ROUGE-2	ROUGE-SU4	Pyr.	Ling.	Ov. resp.
CBSEAS rank	9/53	10/53	11/53	31/53	18/53
CBSEAS score	0.0919	0.1305	0.28	4.67	4.18
Meilleur score	0.0273	0.0583	0.06	3.40	2.34
Lowest score	0.1127	0.1452	0.34	5.78	4.99
Mean score	0.0786	0.1168	0.226	4.751	3.922

**Initial summaries**

	ROUGE-2	ROUGE-SU4	Pyr.	Ling.	Ov. resp.
CBSEAS rank	8/53	8/53	15/53	35/53	19/53
CBSEAS score	0.1027	0.1338	0.3	4.91	4.3
Lowest score	0.0282	0.0591	0.06	3.43	2.46
Best score	0.1216	0.1510	0.38	5.93	5.16
Mean score	0.0853	0.1214	0.252	4.762	4.075

**Update summaries**

	ROUGE-2	ROUGE-SU4	Pyr.	Ling.	Ov. resp.
CBSEAS rank	8/53	15/53	10/53	24/53	22/53
CBSEAS score	0.0811	0.1223	0.26	4.75	3.98
Lowest score	0.0264	0.0576	0.05	3.36	2.23
Best score	0.1039	0.1395	0.31	5.89	5.02
Mean score	0.0719	0.1122	0.198	4.742	3.769

**Fig. 7.** Detailed numeric results of TAC 2009 “Update Task”

Figure 8 presents the two summaries generated by CBSEAS for the D0911 toopic. One can see that the last sentence is cut. This is due to the 100 words limit set by the NIST. CBSEAS does not automatically remove the entire sentence which exceeds this limit. This also negatively affects the linguistic score.

Figure 7 presents the different scores obtained by CBSEAS, and its position in relation to the other systems. One can see that the linguistic quality is the real weakness of our system. However, this display brings to light the efficiency of our update management strategy. CBSEAS loses “only” 13% of its pyramid score when managing the update summarization, compared to the initial summaries score, while the overall participants show an average 21.5% decrease. Five systems on the fifteen which overtake CBSEAS on initial summaries Pyramid scores undertake CBSEAS on update summaries pyramid scores. Generally speaking, update summarization is a difficult task, and one can notice that systems perform better on initial summaries than on update ones.

One could argue that the proposed evaluation is not complete: the initial and update summaries are evaluated independantly. The update summary evaluation could have been pushed further, evaluating the presence of SCUs in the automatic summaries that can be found in the initial documents. The redundancy between the update summaries and the already known content is not evaluated as is.

One interesting result is the linguistic score of the *Baseline 2*, presented in Fig. 6: with a score of 5.68, it is overtaken by two automatic summarizers. The best systems equal the *Baseline 3* – which consists of manually extracted sentences – in selecting the most important pieces of information (Pyramid score). However, these systems remain far away from this baseline (*cf* Fig. 6) in units of linguistic quality and overall responsiveness. These facts prove the impact of sentence ordering on linguistic quality, and so on the user satisfaction towards a summary.

The TAC 2009 evaluation campaign has shown that our system, although competitive for generating summaries, still needs to improve their linguistic quality. The summaries informational content is good, as it performs 83.3% of *Baseline 3* – which provides an upper bound of what can be achieved with an extractive summarizer – in units of Pyramid score. The update management is also satisfactory, as CBSEAS performs better than the vast majority of participants.

<b>D0911 Bobby Fischer : initial summary</b>
Describe efforts to secure asylum in Iceland for chess legend Bobby Fischer.
Chess legend Bobby Fischer was on Monday granted citizenship by the parliament of Iceland, a move which could allow him to avoid deportation from Japan to the United States where he is wanted for violating sanctions against the former Yugoslavia.
Chess legend Bobby Fischer, who faces prison if he returns to the United States, can only avoid deportation from Japan if Iceland upgrades its granting of residency to full citizenship.
Iceland's parliament voted Monday to grant citizenship to fugitive U.S. chess star Bobby Fischer.
Lawmakers in Iceland are likely to grant citizenship to mercurial chess genius Bobby Fischer, a
Pyramid score: 0.622      Linguistic score: 6
<b>D0911 Bobby Fischer : update summary</b>
Describe efforts to secure asylum in Iceland for chess legend Bobby Fischer.
Iceland said Wednesday it hoped to give detained chess legend Bobby Fischer a passport before the weekend after granting him citizenship in a move that could allow him to avoid a US prison term.
An Icelandic supporter of Bobby Fischer said Tuesday he had paid a registration fee that would allow the American chess legend to settle in Iceland.
Chess legend Bobby Fischer could leave his Japanese detention cell by the weekend, his supporters said Tuesday, a day after Iceland's parliament voted to grant him citizenship.
Japan said Tuesday it may let detained chess legend Bobby Fischer leave for Iceland,
Pyramid score: 0.345      Linguistic score: 6

**Fig. 8.** Example of a pair of summaries generated by CBSEAS.

## 6 Conclusion

In this article, we presented CBSEAS, a generic summarization system, and a novel algorithm designed to manage update summarization. Our system obtained competitive results on TAC 2009 “Update Task”. The compared results of initial and update summaries show that our update management strategy is efficient. However, it could be pushed further by filtering the sentences from the update documents, using such a method as the one described in [27], which is based on words novelty factor. The results also put forward the quality of CBSEAS’ sentence selection method. However, our system lacks in linguistic post-processing. This results in user satisfaction barely better than the average of all TAC 2009 participants. If sentence ordering has been spotted as an important element for automatic summaries coherence and comprehension, the impact of other post-processings such as sentence compression or anaphora resolution should be evaluated in future works.

## References

1. Bossard, A., Génereux, M., Poibeau, T.: Description of the lipn systems at tac2008: Summarizing information and opinions. In: Notebook papers and results of TAC 2008. Gaithersburg, Maryland, USA (2008)
2. Bossard, A., Rodrigues, C.: Combining a multi-document update summarization system – cbseas – with a genetic algorithm. Smart Innovation, Systems and Technologies, Springer (2011)
3. Boudin, F., Torres-Moreno, Juan-Manuel, E.B.M.: A scalable MMR approach to sentence scoring for multi-document update summarization. In: Proceedings of the 2008 COLING Conference. pp. 21–24. Manchester, UK (2008)
4. Carbonell, J., Goldstein, J.: 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. pp. 335–336. ACM, New York, NY, USA (1998)
5. Chowdary, C.R., Kumar, P.S.: Esum: An efficient system for query-specific multi-document summarization. In: Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval. pp. 724–728. ECIR ’09, Springer-Verlag, Berlin, Heidelberg (2009)
6. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics. Philadelphia, PA, USA (2002)
7. Dang, H.T., Owczarzak, K.: Overview of the TAC 2008 update summarization task. In: Notebook papers and results of TAC 2008. pp. 10–23. Gaithersburg, Maryland, USA (2008)
8. Dang, H.T., Owczarzak, K.: Overview of the TAC 2009 update summarization task. In: Notebook papers and results of TAC 2009. Gaithersburg, Maryland, USA (2009)
9. Davies, D.L., Bouldin, D.W.: A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1(2), 224–227 (April 1979)

10. Edmundson, H.P., Wyllys, R.E.: Automatic abstracting and indexing—survey and recommendations. *Commun. ACM* 4(5), 226–234 (1961)
11. Fellbaum, C.: *WordNet: An Electronic Lexical Database* (1998)
12. Galanis, D., Malakasiotis, P.: Aueb at tac 2008. In: *Notebook papers and results of TAC 2008*. Gaithersburg, Maryland, USA (2008)
13. Genest, P.É., Lapalme, G., Yousfi-Monod, M.: Hextac: the creation of a manual extractive run. In: *Notebook papers and results of TAC 2009*. Gaithersburg, Maryland, USA (novembre 2009)
14. Goldstein, J., Mittal, V., Carbonell, J., Kantrowitz, M.: Multi-document summarization by sentence extraction. In: *NAACL-ANLP 2000 Workshop on Automatic Summarization - Volume 4*. pp. 40–48. Association for Computational Linguistics, Morristown, NJ, USA (2000)
15. He, T., Chen, J., Gui, Z., Li, F.: Ccnu at tac 2008: Proceeding on using semantic method for automated summarization yield. In: *Notebook papers and results of TAC 2008*. Gaithersburg, Maryland, USA (2008)
16. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: *International Conference Research on Computational Linguistics (ROCLING X)* (September 1997)
17. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. pp. 68–73. ACM, New York, NY, USA (1995)
18. Likas, A., Vlassis, N., Verbeek, J.: The global k-means clustering algorithm. *Pattern Recognition* 36, 451–461 (2001)
19. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*. Barcelona, Spain (2004)
20. Luhn, H.: The automatic creation of literature abstracts. *IBM Journal* 2(2), 159–165 (1958)
21. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: *Le Cam, L.M., Neyman, J. (eds.) Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. vol. 1, Statistics. University of California Press (1967)
22. Marcu, D.: *Improving summarization through rhetorical parsing tuning* (1998)
23. Nenkova, A., Passonneau, R.J., McKeown, K.: The pyramid method: Incorporating human content selection variation in summarization evaluation. *TSLP* 4(2) (2007)
24. Radev, D., Winkel, A., Topper, M.: Multi document centroid-based text summarization. In: *Proceedings of the ACL 2002 Demo Session*. Philadelphia, PA, USA (2002)
25. Ribeiro, R., de Matos, D.M.: Extractive summarization of broadcast news: comparing strategies for european portuguese. In: *Proceedings of the 10th international conference on Text, speech and dialogue*. pp. 115–122. TSD'07, Springer-Verlag, Berlin, Heidelberg (2007)
26. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK (1994)
27. Varma, V., Bysani, P., Bharat, K.R.V., Kovelamudi, S., GSK, S., Kumar, K., Maganti, N.: Iit hyderabad at tac 2009. In: *Notebook papers and results of TAC 2009*. Gaithersburg, Maryland, USA (2009)

28. Wang, B., Liu, B., Sun, C., Wang, X., Li, B.: Adaptive maximum marginal relevance based multi-email summarization. In: Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence. pp. 417–424. AICI '09, Springer-Verlag, Berlin, Heidelberg (2009)