Towards an automatic characterization of criteria

Benjamin Duthil[1], François Trousset[1], Mathieu Roche[2], Gérard Dray[1], Michel Plantié[1], Jacky Montmain[1], and Pascal Poncelet[2]

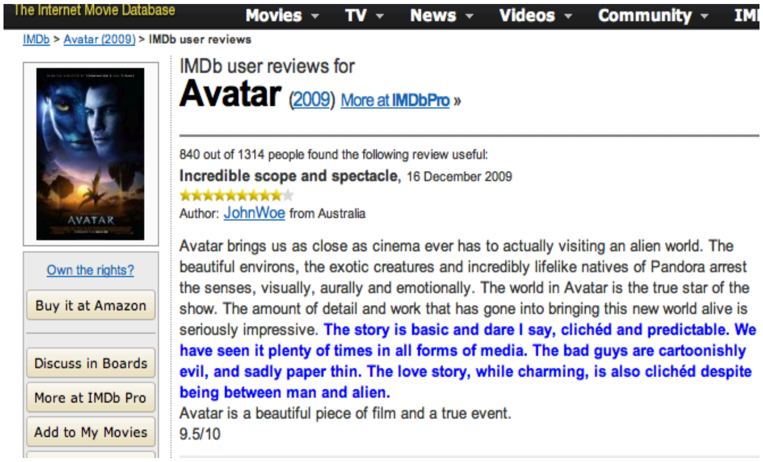
[1] EMA-LGI2P, Parc Scientifique Georges Besse, 30035 Nîmes Cedex, France
name.surname@mines-ales.fr
[2] LIRMM CNRS 5506, 161 Rue Ada, 34392 Montpellier, France
name.surname@lirmm.fr

Abstract. The number of documents is growing exponentially with the rapid expansion of the Web. The new challenge for Internet users is now to rapidly find appropriate data to their requests. Thus information retrieval, automatic classification and detection of opinions appear as major issues in our information society. Many efficient tools have already been proposed to Internet users to ease their search over the web and support them in their choices. Nowadays, users would like genuine decision tools that would efficiently support them when focusing on relevant information according to specific criteria in their area of interest. In this paper, we propose a new approach for automatic characterization of such criteria. We bring out that this approach is able to automatically build a relevant lexicon for each criterion. We then show how this lexicon can be useful for documents classification or segmentation tasks. Experiments have been carried out with real datasets and show the efficiency of our proposal.

Keywords: Criteria characterization, Mutual Information, Classification, Segmentation

1 Introduction

With the development of web technologies, always increasing amounts of documents are available. Efficient tools are designed to help extracting relevant information. Information and Communication Technologies are thus a kernel factor in developing our modes of organisation, if not our societies. Everybody has already visited recommendation sites to consult opinions of other people before choosing a movie or a e-business website. Automatically classifying and indexing documents are computerized tasks that contribute to the development of our information society. For example, lots of tools are already available to extract opinions on movies, (e.g. http://www.premiere.fr/Cinema/Critique-Film) and support cinemagoers to select their movie theatre. Nevertheless, when considering Figure 1 that describes a cinemagoer's opinion with regard to movie Avatar, we may notice that the cinemagoer's opinion regarding criterion scenario is rather negative (although the overall assessment is 9.5/10).



IMDb user reviews for

Avatar (2009) More at IMDbPro »

840 out of 1314 people found the following review useful:

Incredible scope and spectacle, 16 December 2009

★★★★★★★★★☆

Author: JohnWoe from Australia

Avatar brings us as close as cinema ever has to actually visiting an alien world. The beautiful environs, the exotic creatures and incredibly lifelike natives of Pandora arrest the senses, visually, aurally and emotionally. The world in Avatar is the true star of the show. The amount of detail and work that has gone into bringing this new world alive is seriously impressive. **The story is basic and dare I say, clichéd and predictable. We have seen it plenty of times in all forms of media. The bad guys are cartoonishly evil, and sadly paper thin. The love story, while charming, is also clichéd despite being between man and alien.**
Avatar is a beautiful piece of film and a true event.
9.5/10

Fig. 1. an opinion example

Our goal in this paper is to automatically identify all parts in a document that are related to a same center of interest, i.e. a specific criterion in the area of interest of an Internet user. Such an approach allows providing cinemagoers, for example, with more accurate and relevant critics: indeed, when a user is just interested in the casting relevancy of a movie, then he does not expect the search engine to return him critics providing general opinions or opinions related to other criteria such as scenario, soundtrack, etc. Criteria characterization is traditionally performed using supervised classification algorithms (e.g. Mindserver Categorization, Thunderstone, ...). A training set of texts, critics, etc. is first annotated. Then, these methods learn useful features from the classification stage. However, in the Web context, these approaches cannot be easily implemented since it is not realistic to build training sets representative of any criteria in documents (e.g. blogs, forums, tweets, newspapers, ...) that may be found on the Web. This task becomes even harder if misspelling of words is to be taken into account. Similarly, if we consider a criterion like actor, the spelling of names is quite tricky. For example, the main actor in ”Avatar”, Sam Worthington may be spelled Wortington or Wortingthon... Furthermore, the same idea can be expressed in very different ways depending on the type of documents. For example, let us consider criterion scenario: we may find ”scary scenarios ought to make you hit the panic button” in a blog or a forum, while it would rather be ”scary scenarios make yourself get out of the fear” in an official newspaper. Besides these difficulties, the huge amount of data makes manual annotation very thorny if not impossible.

In this paper, we present a new automatic approach named *Synopsis* which tags items of texts according to predefined criteria. First, *Synopsis* builds a lexicon containing words that are characteristic of a criterion and words that are not characteristic of this criterion from a set of documents merely downloaded using a web search engine (google for example). The way this lexicon is built is of great influence in documents classification and segmentation activities.

The paper is organized as follows. Section 2 first describes the main principles of Synopsis approach. Then, a detailed description of Synopsis is provided step by step. Section 3 presents the different experiments we have carried out. A state of the art is presented in section 4 to facilitate the understanding of results and section 5 finally presents some concluding remarks and future work.

2 The Synopsis approach

In this section, an overview of *Synopsis* process is first presented. The detail of each step (document retrieving, words extraction, ...) required by the approach is then provided.

2.1 General presentation

All along this paper, "movie" is our application domain and we focus on two criteria: *actor* and *scenario*. The general architecture of the approach is described in figure 2.

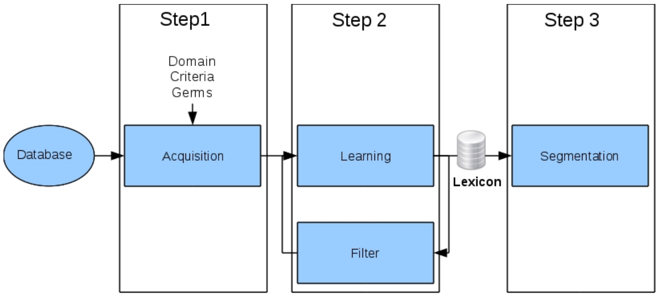


Fig. 2. General architecture

• Step 1 consists in defining the domain and the criteria. For each criterion, the user just needs to give a set of words called *germs* to specify what this criterion intends to be. For example, the germs for our two criteria may be :

scenario → *adaptation, narrative, original screenplay, scriptwriter, story, synopsis*
actor → *acting , actor, casting, character, interpretation, role, star*

A corpus of documents is built for each germ of a specific criterion (relatively to a domain) by querying a search engine for documents containing at least one occurrence of this germ (*c.f.* Example 1). The final goal is to identify words which are correlated to one germ. At the end of this step, the corpus of documents named *class* of the criterion is the union of all the corpora related to criterion's germs. Similarly, a second corpus is built for this criterion: this time, the search engine gathers documents containing none of the criterion's germs. Intuitively, this corpus named *anti-class* intends to give a better characterization of the criterion. Indeed, a general term of the domain (for example, film) should not be identified as a specific term of criterion *actor* or *scenario* because it is not discriminating. Since film should appear in *anti-class* corpora of criteria, it should be eliminated. Furthermore, a pre-processing is performed over the entire corpus to remove text formatting tags and meaningless parts... Texts are then lemmatized using a morpho-syntactic tool (*c.f.* Example 2).

Example 1 *To illustrate our purpose in the domain "*movie*", while using Google as a search engine, the foollowing requests will be used to buids the corpus of criterion "*actor*":*

`"+movie +cast"` *for the set of documents associated to germ "*cast*"*

`"+movie -cast -acting -actor -character -interpretation -role -star"`
to build the anti-class

The symbols + (resp. -) indicate that these words must be (resp. not be) present in the returned document.

Example 2 *After preprocessing, the text "*The role of this actor*" is rewritten : the$^{[DT]}$ role$^{[NN]}$ of$^{[IN]}$ this$^{[DT]}$ actor$^{[NN]}$.*

• Step 2 intends to identify the word representative (resp. non- representative) of the criterion from the lemmatized texts from Step 1. This is achieved by studying the frequency of words strongly correlated with germs of the criterion. The method especially focuses on words which are close to the seeds. The following assumption is thus made: the more frequently a word appears near a germ, the more likely it is to characterize the criterion. These words are identified by using a text window centered on germs (*c.f.* Section 2.2). The size of the window is set to an a priori given number of common nouns (grammatically speaking), e.g. the windows exactly contain two common nouns. Processing documents in both corpora (class and anti-class) provides a set of words close to germs with their frequency in the class or words with their frequency in the anti-class. Four kinds of words can then be defined:

1. Very frequent words in the *class* and poorly frequent in the *anti-class*;
2. Very frequent words in *anti-class* and poorly frequent in the *class*;
3. Very frequent words in both corpora ;
4. Poorly frequent words in both corpora.

In the first two cases, information from frequencies is sufficient to take a decision upon the word's membership of the criterion. A score that characterizes the degree of membership of the criterion is computed from the word's frequency. Case 3 illustrates words which are not discriminating because they belong to both classes and are therefore to be eliminated. In the last case, the corpora of documents related to the word cannot be used to decide whether the word belongs or not to the criterion. In that latter case, another additional stage shall be performed to get new documents related to poorly frequent words. However, because of the large number of words generally obtained at this stage, a first filtering phase is performed. This one is made by applying a web measure named *Acrodef* [10], which considers the number of results returned by the search engine. It is based on the following assumption: the more documents found on the web by the search engine contain a word close to a germ, the more

characteristic of the criterion the word. The remaining words, named *candidate word*, are then processed one by one (*c.f.* Section 2.4). A set of documents is downloaded for each of these words and a processing is performed to obtain a frequency commensurable with the ones obtained for frequent words (*c.f.* Section 2.4). These frequencies are then processed to give a score to each word. The value of this score provides information on the proximity of the word with regard to the criterion. The higher is the score, the more representative of the criterion (the class) is the word. The lower is the score, the more representative of the anti-class is the word. Once computed, the scores are stored in the lexicon of the criterion.
• Step 3 consists in using the lexicon provided in Step 2 for classification, indexation or segmentation relatively to the criterion.

2.2 Characterization criteria

Acquisition. As explained in section 2.1, the first step consists in acquiring and pre-processing Web documents. Acquisition is done by using a search engine (see Example 1). For each germ g of a criterion C the system retrieves about 300 different documents containing words: germ g and domain D (e.g. actor and movie). The resulting set of documents for all germs of criterion C defines the criterion's *class*. Similarly, the system seeks about 300 documents[3] containing none of the germs of the criteria C . This set of documents defines the anti-class of criterion C. It is designed to limit the expansion of the class. Thus, the class of a criterion C is composed of about $n * 300$ documents (where n is the number of germs) and its anti-class is of about 300 documents. All the documents in the class and anti-class are then analyzed to remove HTML tags, advertising, ... and processed using a morpho-syntactic analyzer for lemmatization purposes (See Example 2).

Learning step. Our learning process is based on the following assumption: Words that characterize a criterion are often associated with this criterion. In our case, a criterion is primarily defined by germs. Thus, we are looking for words that are strongly correlated with germs, i.e. words that frequently appear very close to germs. To identify those words we define windows centered on germs in each document t. A window is formally defined as follows:

$$F(g, sz, t) = \{m \in t/d^t_{NC}(g, m) \leq sz\} \qquad (1)$$

where g is the germ, sz represents the given size of the window, and $d^t_{NC}(g, m)$ is the distance between m and g: It counts the number of grammatical *common nouns* words between m and g. We focus on the common nouns
as they are known to be meaningful words in texts [6].

[3] In section "Experiments", we study the influence of the number of documents on the quality results.

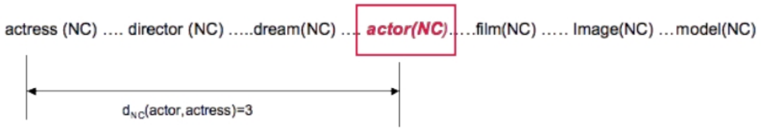

actress (NC) .... director (NC) .....dream(NC) ... actor(NC).....film(NC) ..... Image(NC) ...model(NC)
d_NC(actor,actress)=3

Fig. 3. An example for a window of size 3

Example 3 *Figure 3 shows a sample window of size 3 centered on the germ* actor*: there are 3 common names on its left (*actress*,* director*,* dream*) and 3 common names on its right (*film*,* image*,* model*).*

When a word m appears in a window $F(g, sz, t)$, we have then to take into account its relative position (explanation below) with regard to germ g at the center of the window. This is done by introducing the notion of influence: $I(m, g, sz, t)$ (for a window size sz in text t) :

$$I(m, g, sz, t) = \left[\begin{array}{l} 0 \textbf{ if } m \notin F(g, sz, t) \\ h(d^t_*(m, g)) \textbf{ if } m \in F(g, sz, t) \end{array}\right. \tag{2}$$

where $d^t_*(m, g)$ is the distance between words m and g regardless of the grammatical nature of words. The h function is used to balance the weight associated to a word according to the distance that separates it from the germ. There are, of course, many ways to define this function. In our experiments, we first consider the constant function, which affects the same weight to any word in the window regardless of the distance (number of words) that separates it from the germ. We also consider a notion of semi-Gaussian to smooth the effect of words nearby the edges of the window. This is done by normalizing the size of the window for distance $d^t_*(m, g)$ in order to get an interval centered on g with radius 1. Then, we introduce a Gaussian distribution centered on g. Let us note l and r respectively the words that are the left and right edges of the window. Thus, h is defined by:

$$h = \begin{cases} gauss\left(\frac{d^t_*(g,M)}{d^t_*(g,l)}, \mu, \sigma\right) & \text{for a word to the left of } g. \\ gauss\left(\frac{d^t_*(g,M)}{d^t_*(g,r)}, \mu, \sigma\right) & \text{for a word to the right of } g. \end{cases}$$

$$gauss(x, \mu, \sigma) = \exp - \frac{(x-\mu)^2}{2\sigma^2} \tag{3}$$

2.4 Resolve candidate words

Example 4 *Considering previous example in Fig. 3, words* dream *and* film *get higher weight than other words when using a semi-Gaussian for h whereas a constant function $h = 1$ would have provided the same weight for all words.*

Representativeness. For each word M previously established, we will now compute its representativeness which is a couple of values $(X, \overline{X})$. Where X is the representativeness component regarding to the *class* and $\overline{X}$ is representativeness component relatively to the *anti-class*. Let

$\mathcal{O}(M,T)$ be the set of occurrences of the word M in a text T. Let S be the set of germs for the studied criterion. Then the components of the representativeness are computed as follows:

$$X(M,sz) = \sum_{g \in \mathcal{S}} \sum_{t \in \mathcal{T}(g)} \sum_{g \in \mathcal{O}(g,t)} \sum_{m \in \mathcal{O}(M,t)} I(m,g,sz,t) \qquad (4)$$

$X(M,sz)$ is thus the cumulative impact of all germs g of the criterion on word M in all the texts of the *class*.

$$\overline{X}(M,sz) = \sum_{t \in anti-class} \sum_{g \in \mathcal{O}(D,t)} \sum_{m \in \mathcal{O}(M,t)} I(m,g,sz,t) \qquad (5)$$

$\overline{X}(M,sz)$ is thus the cumulative impact of germ of the domain D on the word M in all documents of the *anti-class*. As the size of the *anti-class* is quite different from the one of the *class*, the values X and $\overline{X}$ are normalized according to the number of germs in the criterion and to the number of documents in both corpora. Both components of representativeness of a word are isomorphic to a frequency respectively in the *class* for X and in the *anti-class* for $\overline{X}$. They are used as such in the following.

	X	$\overline{X}$
film	1080	460
actress	170	0
theater	0	370
poster	700	700
Matt Vaughn	ϵ	ϵ
Sam Wothington	ϵ	ϵ
story	100	120

Fig. 4. Example of representativeness computed for a subset of words for criterion *actor*. ϵ is a small positive quantity

Example 5 *Figure 4 provides a sample of representativeness degrees computed for criterion* actor*. This sample highlights the fact that words are distributed in different categories as explained in Section 2.1. We can notice that the word* poster *is as frequent in the* class *as in the* anti-class*. Thus it gives no information that could be used to discriminate* class *from* anti-class *and therefore it is removed. Word* film *is much more frequent in the* class *than in the* anti-class*. Thus it is characteristic of the criterion* actor*. Considering the words* Matt Vaughn *and* Sam Wothington[4]*, as they have very low representativeness in both* class *and* anti-class*, we cannot deduce anything for those words because we got little information on them. But unlike word poster which can be safely removed because we know that it is not informative at all, eliminating those words is not safe because we actually know nothing about them. To get a safe conclusion with regard to those words, additional information is required for them. This kind of words are named candidate words.*

Figure 5 illustrates the different kinds of words. Words that are common in both corpuses and thus are not discriminating are in the green region

[4] For simplicity, we consider that Matt Vaughn and Sam Wothington are single words.

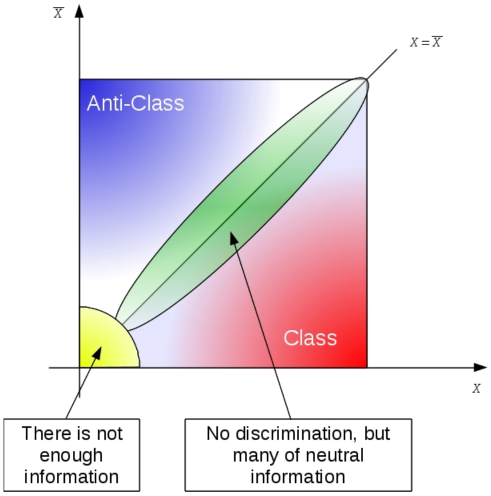

$\overline{X}$
$X = \overline{X}$
Anti-Class
Class
X
There is not enough information
No discrimination, but many of neutral information

Fig. 5. Representation of different kind of words

(ellipse along the bisector). The red (below the bisector) and the blue (above the bisector) regions correspond to words characteristics respectively characteristics of *class* and *anti-class*: they are very common in one of them but not in the other one. The yellow region (around the origin) contains words for which no decision can be made. It is easy to decide to retain or eliminate words having a high frequency either in the *class* or *anti-class* or both (green, red and blue regions). But for those having low frequencies (yellow region), so called *candidate words*, it is essential to get additional information to be able to make a safe decision. However, the amount of words of this kind may be very important. As the complete processing of those words may consume a lot of time, in order to decrease the complexity we use a quick filter processing to remove the words which are too far from the criterion (*c.f.* Section 2.2). This processing significantly reduces the number of downloaded documents.

Validation of candidate words. To determine whether a candidate word helps or not in criteria discrimination, we apply $AcroDef_{IM3}$ measure described in [10] whose the objective is evaluate the correlation of two words in a given context. It relies on the following assumption: A candidate word close to seed words must appear in a large number of documents of the domain. This measure is based on the Mutual Information to the cube and takes into account the context notion [3] by computing correlation between two words m1 and m2. . Let U be a context (defined by a set of words), $AcroDef_{IM3}$ is defined as follows:

$$AcroDef_{IM3}(m_1, m_2) = log_2 \left(\frac{nb\left((m_1,\ m_2)\ and\ U\right)^3}{nb\left(m_1\ and\ U\right) \times nb\left(m_2\ and\ U\right)} \right) \quad (6)$$

Where $nb(l\ and\ U)$ with l a set of words means the number of documents containing all words in l and U with the constraint that all words in l are close to each other. For our purpose m_1 and m_2 are respectively the considered germ word and the candidate word. We use a search engine on the Internet to carry out this task.

Example 6 *By considering the candidate word Sam Wothington and the germ actor in domain $U = (movie)$ the equation becomes:*

$$AcroDef_{IM3}(SamWothington, actor) =$$
$$log_2\left(\frac{(nb((''SamWothington,actor'')\ and\ ''movie'')+nb((''actor,SamWothington'')\ and\ ''movie''))^3}{nb(SamWothington\ and\ movie)\times nb(actor\ and\ movie)}\right)$$

According to the values returned by *Acrodef*, we only retain the words whose $AcroDef_{IM3}$ value is above a threshold. This threshold is obtained experimentally and currently fixed to -25.

	X	$\overline{X}$
Matt Vaughn	*deleted by AcroDef*	
Sam Worthington	300	10

Fig. 6. Words selection after filtering by *AcroDef* and enrichment of the corpus for *actor* criteria.

All words kept after the $AcroDef_{IM3}$ filtering step are then processed to get a new value of X and $\overline{X}$ for each of them (C.f. figure 6). To do that, for each candidate word, a set of documents is downloaded from the Web (*c.f.* Section 2.4). Each of them must contain both the candidate word and one of the germs of the criteria.

Discrimination By using the value of representativeness X and $\overline{X}$ computed as previously explained, we can now define a score for each word as follows:

$$Sc(M, sz) = \frac{(X(M, sz) - \overline{X}(M, sz))^3}{(X(M, sz) + \overline{X}(M, sz))^2} \qquad (7)$$

The cubic power of the numerator allows signed discrimination: the words unrepresentative of the criterion (frequently found in the *anti-class* but not in the *class*) receive a negative score, and the ones representative of the criterion (frequently found in *class* but not in the in the *anti-class*) have positive scores. The square Power for the denominator is used to normalize the score.

	Score
film	1040
actress	450
theater	-460
Sam Wothington	2640

Fig. 7. Example of common lexicon for criterion *actor*.

The scored obtained for each words M are stored in the lexicon associated to the criterion. This is illustrated in Figure 7 for criterion *actor* in domain *movie* after processing the *candidate words* (c.f. Section 2.2). We can see that the score of *Sam Wothington* is now very high while *theater* is low. Thus *Sam Wothington* is representative of the criterion *actor* in the domain *movie* whereas *theater* is no more considered since stands for a general term. This results correspond to the expected behavior of our assessment system. The lexicon may now be used in Step 3 for classification, segmentation or indexation.

2.3 Lexicon usage

In this section we illustrate how to use the lexicon in a text segmentation context. We focus on identifying parts of a document related to the criterion on study. For a document t, a sliding window is introduced (*c.f.* Section 2.2) (it is successively centered on each occurrence of a common name in text t). Its size is denoted sz. From the lexicon, a score is computed for each window f as follows:

$$Score(f) = \sum_{M \in f} Sc(M, sz) \tag{8}$$

For a given text t, a window is related to the criterion, when its score is greater than a threshold value. For obvious length constraints, then automatic choice of the threshold value can not be developed in this paper. The idea consists in analyzing the number of words reputed to be related to the criterion as a function of the threshold value. The changes of the number of selected words due to threshold variations are very slow except for some remarkable value (Figure 8).They correspond to distinct granularity degrees of text analysis according to user's point of view.The threshold value of our algorithm is automatically selected among them.

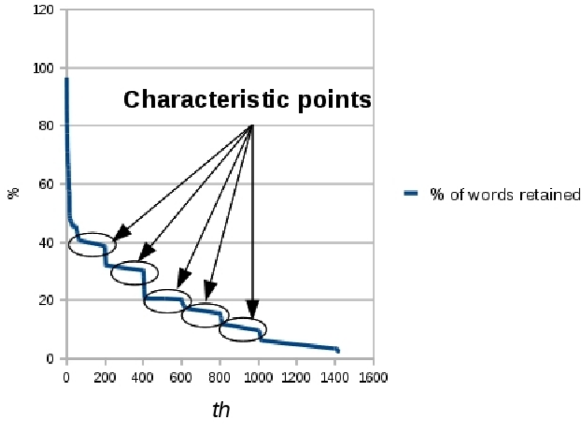


Fig. 8. Percentage of words retained based on the threshold

2.4 Resolve candidate words

As candidate words are both infrequent in the domain and in the criterion it is much harder to obtain representative samples of texts including this word. Then, X and $\overline{X}$ for a candidate word are irrelevant. We have developed a method to collect texts containing the candidate word and we have introduced a web measure to finally get X and $\overline{X}$ values for the candidate word that are normalized relatively to values of the other words in the lexicon.

3 Experiments

In order to analyze *Synopsis* performances, several experiments have been carried out with *Google* as search engine and *TreeTagger* [11] as lemma and morphosyntaxic analyzer. The field of experiment is the one described all along this paper: movie is the domain, actor and scenario are the criteria. Classification and Segmentation tests were performed using a test corpus containing about 700 sentences that have been labeled by experts for each criterion. This corpus mainly contains film criticisms originated from different sources: blogs, journalism reviews...

3.1 System Performance

The following experiments are performed in a classification context. They highlight interest of enriching the lexicon with candidate words. The *test corpus* is the one described above. Validation tests are based upon conventional indicators: recall, precision and F-measure. They are defined as follows:

- $recall = \frac{relevantwords \cap retriviedwords}{relevantwords}$
- $precision = \frac{relevantwords \cap retriviedwords}{retriviedwords}$
- $FScore = 2 * \frac{recall * precision}{recall + precision}$

***Synopsis* Performance**				
	***actor* criteria**		***scenario* criteria**	
	Without CW^* resolution	With CW	Without CW	With CW
FScore	0.39	**0.89**	0.49	**0.77**
precision	0.25	**0.87**	0.59	**0.67**
recall	0.94	**0.90**	0.41	**0.88**

Table 1. System performance in classification for criterion *actor* and *scenario*. CW^* *: Candidate Word*

Table 1 highlights interest of enriching the lexicon with candidate words. Note that if Learning is only based on frequent word (”Without candidate word” column) lots of details are lost. Thus the F-measure gets very low value (*i.e.* 0.39 for *actor* and 0.49 for *scenario*). As soon as candidate words are taken into account (”With candidate word” column) the F-measure rapidly increases (0.89 for *actor* and 0.77 for *scenario*).

3.2 Learning phase analysis

Determining the number of documents required for lexical Learning (prior to enrichment of the corpus) To evaluate the minimum number of documents required to reach a stability in the Learning process, we study the evolution of the three indicators (*precision*, *recall* and *F-measure*) as functions of the number of documents. Results are shown in Figures 9, 10 and 11.

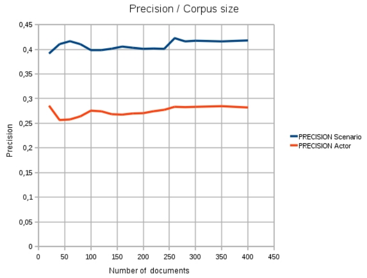


Fig. 9. Precision measure as a function of the documents in the corpus

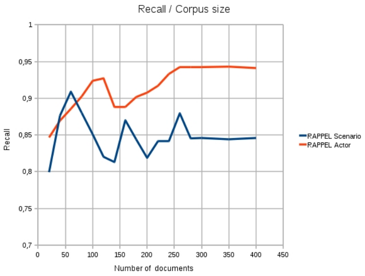


Fig. 10. Recall measure as a function of the documents in the corpus

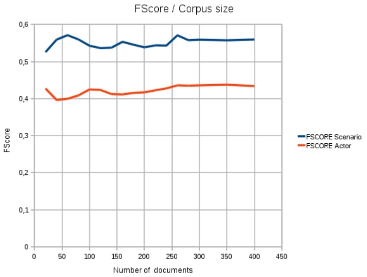


Fig. 11. FScore measure depending on the number of documents in the corpus

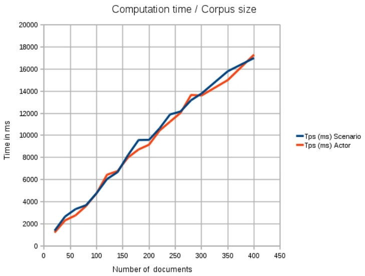


Fig. 12. Computation time depending on the number of files in the corpus

In these experiments, stability in the Learning process is achieved when about 280 documents have been analyzed as shown in Figures 9, 11) and 10. Figure 12 shows that the computation time only increases in a linear way with the number of documents analyzed during the Learning process.

Analysis of the lexicon on the criteria *actor.* The lexicon contains a set of words with their signed score. The objective is to study the repartition of kind of word in the lexicon for three steps: before $AcroDef$ filtering, after $AcroDef$ filtering and after adding candidate words in the lexicon. This repartition is presented in Table 2. $Acrodef$ filter suppress 6000 candidate words and only 500 are still to be computed. After computing them, candidate words have been assigned a score and then are distributed in the three other categories. At the end of the process no more candidate word remains.

Lexicon constitution (for *actor* criteria)			
	Before Filtering	After $AcroDef$ filter	After CW^* computation
Words with a positive non-zero score.	760	760	1000
Words with a score of zero	350	350	410
Candidate words	6500	500	0
Words with a negative score	350	350	550

Table 2. Number of words in the lexicon before and after filtering.
CW : Candidate Word

3.3 Comparison with two standard tools in the context of segmentation: *C99* and *TextTiling*

This section is devoted to the comparison of our approach *Synopsis* with other ones usually used for segmentation tasks: *C99* and *TextTiling*. As these approach do not associate labels with the segment they identify, we are compelled to associate labels to the segments to make our comparison. The affectation of labels (*actor* or *non actor* in this experiment) to each segment is carried out in such a way that *C99* and *TextTiling* obtain the maximal score for any of the three assessment rates: (*precision*, *recall* or *F-measure*). These maximal compared values are then compared to *Synopsis* results in Table 3.

Performance							
		for actor			**for scenario**		
Maximize		C99	TextTiling	Synopsis	C99	TextTiling	Synopsis
$FScore_{max}$	FScore	0.39	0.43	**0.89**	0.36	0.38	**0.77**
$Precision_{max}$	Precision	0.25	0.29	**0.87**	0.25	0.28	**0.67**
$Rappel_{max}$	Recall	0.80	0.81	**0.90**	0.65	0.65	**0.88**

Table 3. Comparison of the three segmenters for criteria *actor*

Synopsis clearly provides much better performances. F-measure and accuracy are always higher than those that could ever be obtained with *C99* or *TextTiling*. However this result is to be moderated since both usual segmenters do not use any initial Learning corpus.

4 RelatedWork

The approach described in this paper is able to identify fragments of texts in relation with a given topic of interest or domain. Thus, our work may appear rather close to segmentation of texts and thematic extraction approaches. Segmentation is the task that identifies topic changes in a text. A segment or extract is then supposed to have a strong internal semantic while being without semantic links with its adjacent extracts. Synopsis not only detects topic changes in the text but also builds the subset of extracts that are related to a same topic, i.e the text is not only segmented but also indexed by the criteria defined over the domain. It results in the identification of the thematic structure of the text [8]. As many studies, our approach relies on statistical methods. For example, TextTiling studies the distribution of terms according to criteria [5]. Analyzing the distribution of words is a widely spread technique in segmentation process [9]. Other methods, such as C99 approach, are based on the computation of similarities between sentences in order to detect thematic ruptures [1]. Note that segmentation approaches have, in our point of view, a major weakness: they cannot identify the topic an extract deal with. As a consequence, segmentation approaches cannot identify topic repetition in a document. Techniques issued from text summarization may in turn identify parts of a document that are related with the dominant topic of the document [2]. Other methods aim to identify excerpts in relation with the title of the document [7]. Identifying segments of text related to a set of criteria in a given domain, i.e. identifying the thematic structure of a document with Synopsis is yet another challenge. Most of automatic summarization techniques are based upon supervised training methods and thus require a high degree of human intervention to create a training corpus. Our non-supervised framework is a major asset of Synopsis. The only required intervention of human beings consists in providing a small set of germ words for each criterion of interest in the domain.

5 Conclusion

In this paper, we have proposed a new approach *Synopsis* to identify the thematic structure of a text. It consists in identifying segments of texts related to criteria defined upon a domain. *Synopsis* is to be considered as a non-supervised approach since the only human intervention consists in providing a subset of germ words for each criterion. We have discussed the interest of the anti-class concept. We have demonstrated that partitioning words into words related to a criterion and words absent from this criterion provide safer classification results. In order to eliminate as quickly as possible noisy words, we have shown that mutual influence measures like *Acrodef* could help in order to minimize the number of words that require a more detailed analysis. Finally, experiments have highlighted that *Synopsis* performances are relevant both in classification and segmentation. Prospects related to this work are numerous. First, we want to extend the approach in order that *Synopsis* could incrementally learn new words. This step is a major challenge when studying a given domain. Indeed, let us consider the case of proper nouns. As earlier discussed, classification is significantly improved when proper nouns are included into the lexicon. Let us consider again the case of word Sam

Worthington. An analysis of results from *Google Trends* shows that before *"Avatar"* there was almost no documentation about Sam Worthington. After *"Avatar"* successful show, more and more documents were related to its main actor. As a consequence, adding this information in the lexicon necessarily improves quality of results but the difficulty is due to the dynamical aspects of the corpus of proper nouns related to a domain. Secondly, we wish to extend our approach by extracting opinions expressed in excerpts of specific criteria (that is the reason why subtopic of a domain are named criteria in Synopsis). In previous work [4], we have demonstrated that the way opinions are expressed depend on the domain: opinions detection thus appears as an obvious further development of *Synopsis*.

References

1. Choi, F.Y.Y.: Advances in domain independent linear text segmentation. In proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics 23, 26–33 (2000)
2. Chuang, W.T., Yang, J.: Extracting sentence segments for text summarization: A machine learning approach. Proceedings of the 23 th ACM SIGIR pp. 152–159 (2000)
3. Daille, B.: Approche mixte pour l'extraction automatique de terminologie : statistique lexicale et filltres linguistiques. Thse de Doctorat, Universit Paris VII, France. (1994)
4. Harb, A., Plantie, M., Dray, G., Roche, M., Trousset, F., Poncelet, P.: Web opinion mining: how to extract opinions from blogs? International Conference on Soft Computing as Transdisciplinary Science and Technology (2008)
5. Hearst, M.A.: Texttiling: segmenting text into multi-paragraph subtopic passages. ACM 23, 33–64 (March 1997)
6. Kleiber, G.: Noms propres et noms communs : un problème de dénomination. Meta pp. 567–589 (1996)
7. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval pp. 68–73 (1995)
8. McDonald, D., Hsinchun, C.: Using sentence-selection heuristics to rank text segments in txtractor. JCDL'02 pp. 28–35 (2002)
9. Reynar, J.C.: Topic segmentation: Algorithms and applications. PhD thesis (2000)
10. Roche, M., Prince, V.: Acrodef : A quality measure for discriminating expansions of ambiguous acronyms. CONTEXT pp. 411–427 (2007)
11. Schmid, H.: Treetagger. In TC project at the institute for Computational Linguistics of the University of Stuttgart (1994)