Combining statistical data analysis techniques to extract topical keyword classes from corpora

Mathias Rossignol

Pascale Sébillot

Irisa, Campus de Beaulieu, 35042 Rennes Cedex, France

(mrossign|sebillot)@irisa.fr

Corresponding author : Pascale Sébillot (sebillot@irisa.fr)

tel: 33 2 99 84 73 17 — fax: 33 2 99 84 71 71

Abstract: *We present an unsupervised method for the generation from a textual corpus of sets of keywords, that is, words whose occurrences in a text are strongly connected with the presence of a given topic. Each of these classes is associated with one of the main topics of the corpus, and can be used to detect the presence of that topic in any of its paragraphs, by a simple keyword co-occurrence criterion. The classes are extracted from the textual data in a fully automatic way, without requiring any* a priori *linguistic knowledge or making any assumptions about the topics to search for. The algorithms we have developed allow us to yield satisfactory and directly usable results despite the amount of noise inherent in textual data. That goal is reached thanks to a combination of several data analysis techniques. On a corpus of archives from the French monthly newspaper* Le Monde Diplomatique*, we obtain 40 classes of about 30 words each that accurately characterize precise topics, and allow us to detect their occurrences with a precision and recall of 85 % and 65 % respectively.*

Keywords: Topic detection, topic characterization, statistical data analysis, unsupervised classification.

1 Introduction

The term "Natural Language Processing" (NLP) covers a wide range of studies, from speech recognition to automatic text generation, some of which are concerned with the extraction of meaningful information from textual data. Researches carried out in that area can mostly be split into two categories, depending on whether they make use of existing linguistic data, such as electronic dictionaries, to extract information from text, or "create" linguistic knowledge by extracting it from textual data—the purpose of the latter being mostly to provide the information needed by the former. The work we present in this paper belongs to the second category, that of systems exploiting large collections of texts, or *corpora*, and processing them using various numerical or symbolic machine-learning techniques in order to extract linguistic knowledge from them. Statistical data analysis methods, by making it possible to extract the "average behavior" of a word from examples of the way it is used in a text and to bring together words sharing similar properties, constitute a very efficient approach for that purpose. However, there exists a need to adapt those techniques to the complexity inherent in textual data. This paper can be seen as a case study of such an adaptation to a specific task: it details the algorithms we have developed to exploit the results of a classical hierarchical clustering and automatically obtain from raw textual data a set of complete and consistent classes of topical keywords, directly exploitable for topic detection and characterization. Those algorithms are of general interest outside the field of NLP, for the issues they address are commonly encountered when applying statistical analysis techniques to complex data.

Our purpose is to characterize and detect the topics dealt with in a large textual corpus (about 10 million words) in order to split it into topical sub-corpora gathering excerpts from the initial text that deal with a given topic. Hence, the two main problems we have to solve are 1- finding out what topics are addressed in a corpus and providing a human-understandable characterization of them, and 2- locating the appearances of each of these topics in a text, so as to select the text excerpts

from which to build each sub-corpus.

Topic characterization and detection has many applications in NLP, such as document retrieval (finding a text "dealing with...") or the extraction of information from text (knowing what topic a segment of the text addresses being an important clue to direct the search). In our case, the topical sub-corpora we aim at generating are to be used as the basis of a subsequent treatment building lexical resources from textual data. We will not detail here the principles of this further development, but simply present "as is" the main requirements it imposes on our system:

- in order to fully reflect the type of language associated with each topic, sub-corpora must be made up of "self sufficient" text excerpts, that is, excerpts that are fully understandable if considered as stand alone texts. This is not the case of isolated sentences or parts of sentences, which most of the time lack a context in which they take their full meaning. We therefore choose to use full paragraphs as the atomic text unit, but accept that a paragraph may feature references to several topics;

- even though we wish to be able to extract the largest possible sub-corpora, we have to take into account the fact that subsequent treatments should be able to rely on the obtained result. Therefore, emphasis is put on precision rather than recall, in other words, the topical consistency of the extracted sub-corpora has precedence over their completeness;

- to achieve an optimal adaptation to the processed corpus regardless of its domain of main interest, we shall not rely on a predefined list of topics to look for, but let them "emerge" from the textual data itself—this also presents the interest of giving our system the ability to put to light some semantic information about the contents of the studied corpus;

- most importantly, in order to be seamlessly integrated into the wider-scale treatment chain we intend to build for the acquisition of lexical resources, the whole process of topic detection and characterization must be fully automatic, requiring neither human intervention nor external data.

In order to meet these requirements, we have chosen to characterize and detect topics using classes of topical keywords, that is, words whose presence in a text segment is strongly connected with the occurrence of a given topic. These classes are acquired from textual data in a completely unsupervised way thanks to the three stage system that we have developed:

1. several sets of imperfect classes are obtained from relatively small excerpts of the studied corpus thanks to a classical hierarchical clustering technique,
2. these various sets of classes are then compared and confronted to produce a "consensual" and reliable classification made of small but noise-free classes,
3. the classes thus obtained are used as the basis of a simple supervised learning method in order to obtain larger classes performing a more complete coverage of the characterized topics.

We present in section 2 several researches carried out in the field of topic detection and characterization; this lets us show how the specifities of our work (notably, the total automatization and the absence of resort to external data) call for a different approach. We then detail and justify our choice of using classes of topical keywords and exploiting statistical data analysis techniques to build them. In section 3, we examine more closely the properties of textual data with respect to our purpose, thus exposing the difficulties that have to be overcome for this study, and show how the three stage structure around which our system is built lets us address those issues and generate from a journalistic corpus 40 keyword classes of about 30 words each that efficiently point out the main topics a text corpus deals with. The rest of this paper details the operation of the algorithms we have developped, and finally proposes an evaluation of the performance of the system.

2 Review of existing work, and choices

The field of topic detection and characterization has seen numerous achievements over the last ten years, considering the question from various points of view. It

is often combined with the task of topical segmentation, as in [12], where the authors make use of linguistic "clues" to detect areas where a topic shift occurs in the text. As has been said, we choose not to consider that issue and predefine the text segmentation at paragraph boundaries (which is actually a most simple kind of linguistic clue) in order to guarantee the consistency of the extracted text segments. Moreover, the choice of linguistic clues is language-dependent and involves the intervention of an expert, which we wish to avoid.

Litman *et al.* do not focus on the problem of topic characterization, that is often better addressed by systems based on machine-learning techniques. Some of the most efficient of these, such as [3] or [2], build language models (n-gram grammars, cache memories [8]...) characterizing the kind of text corresponding to each topic. But these models are learned from pre-classified texts, thus requiring an important human intervention to prepare the training data. Moreover, such methods use an *a priori* list of topics to look for, whereas our goal is to make them emerge from the sole textual data.

Another family of topic detection tools works on lexical similarity between text segments, that is, considers that two text excerpts deal with the same subject if they use the same words. The work presented in [5], for example, performs a sequential scan of a whole corpus and computes for each word the lexical similarity between the lists of 10 words preceding and following that target. To make up for the small amount of data available for each computation (only 20 words), that measure makes use of additional collocation data collected on another (large) corpus. Local minima of the lexical similarity are considered to indicate a topic change in the text, which is segmented along these frontiers. The words that played the most prominent role in maintaining the lexical similarity at a high value between two such splitting points are used to build characterizing "topical signatures". Although this algorithm operates without human intervention, it does require additional data, which has to be found and selected to fit the type of the processed corpus. Moreover, this system performs, like [12], a topical segmentation of the studied text, which leads to the problem of "self-sufficiency" of the extracted texts excerpts that we

have already mentioned, and which we solve by predefining our text segments as whole paragraphs of text.

Some earlier works deal with the issue of topic detection for whole paragraphs, the best-known one being Hearst's *TextTiling* [6]. The purpose of this system is to perform a topical segmentation of a text into groups of consecutive paragraphs; every 20 words in the text, it computes a lexical similarity between the lists of 100 words before and after the target, and when a local minimum of this value is met, it splits the text at the closest paragraph boundary. Topical signatures are computed similarly to [5], but in a way that does not ensure that two non-contiguous paragraphs dealing with a same topic will be recognized as such, that is, share the same signature. Salton *et al.*, in [17], extend TextTiling to better cope with that possibility, and create topical links between non-consecutive paragraphs. This work could be a partial answer to our problematic if it was designed to deal with large corpora, which is not the case: like *TextTiling*, its purpose is to extract the topical structure of a scientific or newspaper article, and it is not meant to scale up.

The idea behind our system is to compensate our constraints (no human intervention, no external data) by the fact that, having predefined text segments, we are no longer limited to performing a sequential scan of the corpus, like [5], and can treat it as a whole—considering, for example, the matrix gathering the distribution of words over its paragraphs (*i.e.* the number of occurrences of each word in each paragraph). This lets us take advantage of the analytical power of more sophisticated statistical methods in order to detect peculiar linguistic constructs or regularities that can be related with the occurrences of specific topics. Thus, our work may be compared with research carried out in the field of data visualization, such as [14] or [9]. However, contrary to such studies, that mostly concentrate on the optimization of a single statistical data analysis method to produce easily human-interpretable results, we focus on the design of algorithms combining several data analysis techniques to yield "well-finished", directly computer-usable results.

With contrast to most recent works in the field of topic detection, we choose

to use an elementary means to characterize and detect topics: each of them is represented by a set of *keywords*, that is, words whose presence in a text segment significantly increases the probability for this topic to appear there. For example, *hook*, *rod*, *fish*, *catch* are all keywords for the topic "fishing". Although none of the mentioned words, considered alone, is sufficient to detect the topic, a combination of several of them clearly indicates what is dealt with in the text segment where they appear. Hence the detection criterion we employ: if at least two keywords of a same class appear in the same paragraph, then that paragraph deals with the topic "of the class" (from now on, we use that expression to refer to the topic that underlies, or "justifies", the grouping of the words contained in a class). Our goal is to automatically build from the textual data such keyword classes corresponding to all the main topics of the corpus. Rudimentary a tool as they may seem, such classes present many advantages: they have theoretical roots in many linguistic theories[1], but are also intuitive (keywords form the basis of the use of Internet search engines) and, being simple lists of words, can be easily interpreted by a human user, thus fulfilling our aim of topic characterization. Moreover, their nature of sets of objects makes them a "natural target" for data analysis techniques.

It comes from our definition of keyword classes that words belonging to the same class, being typical of a certain topic, will appear together with a relatively high frequency in the paragraphs dealing with it. This observation lies at the foundation of our system, and leads us to make use of an unsupervised statistical classification method to bring together words having a similar distribution over the paragraphs of our corpus. However, specificities of our data make it impossible to obtain satisfactory results by using this technique alone; it is therefore necessary to develop an "algorithmic environment" that fully exploits the potential of this method, more particularly by combining it with other, different approaches to the same problem.

The upcoming section examines some of the fundamental issues that arise from

[1]For our purpose of lexical information extraction, we use the formalism of Interpretative Semantics [15], in which keywords can be interpreted as a weakened form of semantic isotopies.

our problematics; it is shown that the three stage structure we have adopted for our system, which we present in greater detail, is both necessary and theoretically sufficient to overcome them.

3 Specific difficulties and proposed solution

From a data analyst's point of view, one of the most striking features of textual data is the amount of noise it contains. Several reasons for this poor definition can be put into light by a closer examination of the objects we are dealing with, that is, words:

- At the lowest level of abstraction, we can call words the space-separated character strings found in the raw textual data—this meets the definition of *tokens* in the field of formal languages. They depend directly on the production of the studied corpus, which involves human typing or OCR, two possible sources of errors, although marginal. The corpus we have used for this work is built from 14 years of archives from the French monthly newspaper *Le Monde Diplomatique*; an estimated 1 % of its tokens features a typographical error.

- Tokens are not suitable objects to work with in NLP, for they do not correspond to logical entities: several distinct tokens can be various shapes of a single word (for example, both *went* and *goes* are instances of the linguistic entity "to go"), and a given token can correspond to several objects (like *green*, that can be a noun or an adjective). Many tools exist to automatically find out the syntactic type ("part of speech (POS) tagging") and non-inflected form, or lemma ("lemmatization") of the words appearing in a text, thus making it possible to work on *lexical items* instead of simple tokens, but they are not totally reliable. We have prepared our corpus using the MULTEXT tagger and lemmatizer [1]; the error rate on our corpus is of about 5 % of mis-tagged or mis-lemmatized word occurrences.

- Lexical items roughly correspond to common dictionary entries, which is, compared to using tokens, a great step toward linguistic "correctness". But

our definition of keywords assumes that words have non-ambiguous meanings, which is generally not true: some words are clearly polysemous (like "mouse"), and most words exhibit slight variations in meaning depending on the context in which they are used. Therefore, we should ideally work on *semantic items* instead of lexical ones. Tools to perform the task of word-sense disambiguation have been developed [7], but all of them feature two characteristics that make them unsuitable for our needs: they assign word occurrences to predefined meanings, which means exploiting external data, and perform a much finer-grained semantic analysis of the studied text than what we intend to do. It is therefore not relevant to make use of them, and our system has to take into account the fact that it processes ambiguous objects.

Another difficulty we have to cope with is the issue of object selection: statistical classification methods need a certain amount of information about an object to be able to characterize it, which implies that they will only be able to study the most frequent words in the corpus. But many of those are of no interest for topic detection (e.g. "size", "Monday", *etc.*), and reciprocally, many uncommon words can show a great accuracy for that task. This raises two additional and complementary issues: the first is to select from the most frequent words in the text the ones that actually carry a strong topical information; the other to extend the raw results of the statistical analysis to include less frequent but topically relevant words into the final classes.

Finally, but also most immediately, the size of our data is a problem to be considered: experiments we have carried on show that, in order to obtain a reasonably complete coverage of the topics dealt with in the text, it is necessary to consider at least the 1,000 most frequent nouns and adjectives it contains. If we exploit data collected on the whole corpus and make use of the aforementioned matrix of distribution of words over paragraphs, these 1,000 objects will be characterized by 100,000 variables (the number of paragraphs in our corpus), which leads us to limits of tractability in terms of memory usage, processing time, and numerical precision.

We propose a solution that both overcomes this size problem, and provides a means to filter out the noise in our results and select the most relevant words for topic detection. It is widely inspired from classical random sampling techniques like Monte-Carlo or bootstrapping [4]. The idea behind random sampling is to iteratively analyze randomly selected excerpts from the considered data and "integrate" the various obtained results into a "consensual", reliable end-product. Following this example, we work on random samples of 10,000 paragraphs extracted from the initial corpus, and build using each of them a first classification on its 1,000 most frequent nouns and adjectives. These first classifications are inspired from a work presented in [13], which makes use of the CHAVL clustering method (*Classification Hiérarchique par Analyse de la Vraisemblance du Lien*, i.e. "hierarchical classification by linkage likelihood analysis") [10] to build a classification tree from the contingency table crossing words and paragraphs (that is, the matrix indicating for each word in which paragraphs it appears, and how many times). The use of a hierarchical classification method is imposed by the fact that having no previous knowledge of the number of topics in our corpus, and hence of desired classes, prevents us to employ directly class-yielding techniques, such as the classical k-means, which require such an assumption. But it leaves us with the problem of "intelligently" extracting classes from a classification tree. We present in section 4 the method we have devised for that task; its purpose is to perform a partial questioning of the aggregations proposed by the hierarchical classification technique while collecting the most interesting (according to a quality measure we define) classes found in that operation, thus yielding a first set of classes usable by the subsequent stage.

As has been said, we apply this first treatment to multiple random excerpts from our corpus, and then build a new set of classes by comparing the obtained results. Traditionally, random sampling is mostly used to evaluate the average value and standard deviation of a single statistical measure, and we have developed a method to extend this principle to the generation of a "consensual classification": the idea is to group words that were repetitively observed in the same class in

many classifications. This process, described in section 5, enables us to eradicate classification noise, which is essentially non-repetitive. It also removes non-topical words from the classification, since those words have no reason to exhibit a specific link with others, and assigns polysemous words to the class corresponding to their most frequent meaning in the corpus.

That second stage of our system enables us to generate a set of consistent but relatively small keyword classes (applied to our corpus, it yields about 40 classes of 4 to 10 words) corresponding to clearly identifiable topics. It solves many of the problems we have mentioned earlier: noise filtering, word-sense ambiguities, and object selection. But we still need, in order to obtain reasonably complete keyword classes, to extend the collection of words they contain to less frequent but equally meaningful words. This is done by using these first classes as a basis for a simple supervised learning method adding to each class the words appearing "abnormally often" in the paragraphs where it detects its topic. This more robust and less computationally expensive technique makes it possible to take into account relatively rare words and work on their distributions in the whole corpus at once. That third and last treatment, presented in section 6, extends our some 40 classes to an average of 30 keywords each without any loss of consistency, thus yielding complete and relevant keyword classes covering most of the topics addressed in the corpus.

Those results are exposed in section 7 and evaluated from two points of view: that of their expressivity for a human reader, and that of their efficiency for topic detection. As a conclusion (section 8), we briefly outline some considered directions of development for that system.

We present in the next section the first stage of our system, that exploits the results of a hierarchical classification of words based on their distributions over the paragraphs of an excerpt from the corpus in order to generate a set of classes constituting a suitable basis for the treatments presented in the rest of the paper.

4 First classification on one data excerpt

This first computation step is organized around the CHAVL clustering algorithm, which we use to build a word classification tree from the contingency table crossing word lemmas with paragraphs. The similarity measure between words we use for this task is integrated in the standard implementation of CHAVL; we have presented it in annex A, at the end of the paper, for enhanced readability. Algorithmically speaking, CHAVL follows the classical hierarchical clustering scheme of incremental aggregations. As has been said, the choice of this method is mostly justified by the fact that, knowing nothing of the number or sizes of the keyword classes we wish to obtain, we have to rely on a hierarchical classification technique. The issues met when applying such a method have proved to be quite independent of what hierarchical clustering technique (or even similarity measure, provided a certain degree of sophistication) is actually used, which is why we do not discuss more in depth the choice of CHAVL.

Although working on a smaller number of words would permit a better functioning of the methods employed during this stage of computation (including CHAVL), as shown in section 4.4, we rely on the noise-filtering ability of the next stage of our system and select for this first analysis the 1,000 most frequent nouns and adjectives in the excerpt, in order to achieve a fairly satisfactory coverage of the topics it deals with. The choice to retain only nouns and adjectives for topic characterization is a quite generally approved one in this domain, backed up for example by works in terminology, such as [18]. Concerning the size of the excerpt, 10,000 paragraphs has proved experimentally to constitute a balanced choice: choosing a smaller excerpt notably decreases the quality of the classification, whereas larger ones do not bring significant enough improvements to the quality of the obtained classification to justify the increase in computational demand.

Figure 1 shows a reduced version of the kind of classification tree produced by CHAVL[2]. Each node of the tree corresponds to a class gathering all the leaves of

[2]As for the results and examples presented in the rest of this paper, words displayed on the figure are translations of results actually obtained in French.

the descending subtree. As can be seen, classes of interest—like {*cinema*, *movie*, *scene*} or {*unemployment*, *job*, *unemployed*}—can appear at various levels in the classification tree. The traditional method of choosing one level in the tree (such as the dotted line on the figure) and cutting its "branches" at this level to obtain classes is therefore not suitable in our case. It can also be noted that some proposed classes are "meaningless" with regard to our research and the use we intend to make of them (e.g. {*capitalist*, *life*}), and that some of the mergings proposed by CHAVL would benefit from a slight reorganization: for example, moving *town* from its merging with {*cinema*, *movie*, *scene*} to the class {*city*, *lodgment*} would let us generate the other interesting class {*cinema*, *movie*, *scene*, *author*, *work*}.

That examination of the imperfections found in a typical (although scaled down) classification tree shows that the goal of a tree-reading algorithm should be not only to spot the "most interesting" classes in the tree, but also to question the mergings it performs, possibly bringing small modifications to its structure in order to let more relevant classes appear. This unusual exploitation of the classification tree is made necessary by the fact that we make use of a statistical method (and similarity measure) aiming at highlighting a statistical property on the objects it considers, whereas our goal is to reach some knowledge about a linguistic property of these objects—for which statistics can only give an estimate. To correct that approximation, we adopt a new point of view to define a quality measure that numerically reflects the "interest" of a potential keyword class. That new measure is essentially statistical as well, but the exploitation we make of it lets the two complementary approaches compensate each other's aberrations. The quality function we propose makes use of another classification built on paragraphs considering the words they contain: this is an application of the principle of lexical similarity presented in the introduction, and is a simple way to build classes of paragraphs dealing with a same topic. In order to evaluate the quality of a potential keyword class, we then compare the set of paragraphs in which it detects the appearance of its topic with the computed paragraph classes. Before we explain how this measure is precisely defined, we first present how the classification of paragraphs is performed.

Since the following sections alternatively refer to operations on words and paragraphs, we introduce a new notation to avoid any ambiguity: words prefixed with "p-" refer to operations on paragraphs (e.g. "p-classification": a classification of paragraphs), whereas the prefix "k-" is used to denote operations on potential keywords ("k-partition": a partition of the set of potential keywords).

4.1 Paragraph classification

Our purpose for this p-classification is not to build "perfect" p-classes, but only an intermediate result good enough to be used as a "stepping stone" for future treatments. In particular, we do not aim at creating p-classes bringing together *all* the paragraphs dealing with a given topic, but small, consistent classes whose members all address the same topic—that topic being possibly present in other classes as well. Our strategy is this time to not use any of the sophisticated similarity measures predefined in CHAVL, but to rely on the amount of available data to obtain fairly good results: the similarity we define to compare the vocabulary employed in two paragraphs is voluntarily simple, so as not to impose any constraint on the selection of words used to compute it, and remain computationally reasonable despite the large amount of data processed.

4.1.1 Definition of a similarity measure between paragraphs

Computing the lexical similarity between two paragraphs consists in evaluating how close the vocabularies they employ are—in its most elementary form, this idea leads us to counting the number of words the paragraphs have in common. That principle lies at the foundation of the measure we define. It is refined by a normalization taking into account the sizes of the compared paragraphs (in number of words), and, as an approximation of the kind of normalization a more sophisticated measure would feature (centering and reduction of the values, "normalized contributions"...), the importance of a shared word in the computation of the similarity measure is inversely proportional to its number of occurrences in the randomly selected subcorpus. The similarity measure between two paragraphs A and B is thus defined

as:

$$s(A,B) = \frac{1}{\min(n_a, n_b)} \sum_i \frac{\min(a_i, b_i)}{n_i} \quad (1)$$

where i browses the set of all words considered for the computation, $A = (a_i)$ and $B = (b_i)$ are the vectors gathering the number of occurrences of each of those words in each paragraph, n_i is the number of occurrences of word i in the 10,000 paragraphs of our random excerpt, and n_a and n_b are the number of words in paragraphs A and B respectively.

The simplicity of this formula makes it possible to take into account all nouns and adjectives appearing at least twice in the studied excerpt, that is about 9,000 words. To build a classification tree of paragraphs, we employ the same implementation of CHAVL we used to cluster words, this time with a similarity matrix computed according to the above formula. The tree we obtain is quite well-balanced, which is the sign of a smooth operation of the clustering algorithm, with a similarity measure well adapted to the collection of objects we process. That good balance of the tree enables us to extract reasonably good classes from it using a simple technique.

4.1.2 Generation of a collection of p-classes

Contrary to the k-classification, we cannot aim at building complete p-classes, for doing so would lead us once again to the problem of performing an intelligent extraction of classes from a tree. P-classes are therefore generated by a simple procedure selecting in the p-classification tree a level where the average size of a p-class is 12[3], and yielding all classes obtained by cutting the branches of the tree at that level. The p-classes deviating too much from the original target size of 12 paragraphs (less than half, or more than twice that size) are removed from that partition, and we obtain a collection of 600 p-classes.

Due to the size of the processed data and complexity of the studied objects (knowing what a paragraph deals with requires to actually read it), we cannot pro-

[3]This value was chosen as a good compromise between class consistency, our first goal, and generalization value.

pose a thorough evaluation of the generated classes. A few random checks have shown that they often bring together paragraphs sharing, in addition to a fairly similar "lexical background", a few uncommon words very specific to a given issue (e.g. "intifada" or "spacecraft"). Hence, although the exploited similarity measure would probably not yield satisfactory results if used to generate larger topical classes, it proves to be well adapted to our goal of producing small topically consistent p-classes.

From this point, we have at our disposal an "arrangement" of paragraphs in topical groups. We now show how, by confronting this low-level, fine-grained classification with the topic detection performed by a potential k-class, we can compute a numerical evaluation of the quality of that k-class.

4.2 Quality measure for potential keyword classes

The quality measure we now define is based on a simple idea born from our initial definition of keywords and of their roles: if two keywords of the same set appear in a given paragraph, then that paragraph deals with the topic underlying the set of keywords (we say in what follows that a k-class "recognizes" a paragraph). Hence, the set of all k-classes we intend to extract from the k-classification tree performs a thematic clustering of the set of studied paragraphs. Since this is also true of the collection of p-classes we have defined in section 4.1, those two classifications should coincide as much as possible (that correspondence being essentially limited by the fact that a paragraph may be recognized by several k-classes, but can only belong to one p-class).

If we consider the ideal case where all the paragraphs of a p-class actually address the same topic and each k-class recognizes *all* the paragraphs dealing with a given topic and *only* them, we can state the following rule: if a k-class recognizes one of the paragraphs of a p-class, since those paragraphs all address the same topic, it will recognize *all* of them. In that ideal case, the proportion of paragraphs of a p-class recognized by a given k-class can therefore only be zero or one. The quality measure we now define is an attempt to mathematically define a value evaluating

how close a k-class is to that ideal configuration.

4.2.1 Mathematical expression

Let $\mathcal{K}$ be a keyword class and $\mathcal{P}_1, \ldots, \mathcal{P}_p$ all the paragraph classes defined by the p-partition from section 4.1 (where $p = 600$). For any paragraph P, we write $\mathrm{rec}(\mathcal{K}, P)$ to express the fact that $\mathcal{K}$ recognizes P. With $\mathcal{K}$, we associate the vector $\overrightarrow{K} \in \mathbb{R}^p$, defined by:

$$\overrightarrow{K} = (k_1, \ldots, k_p), \text{ with } \forall i \in [1, p], k_i = \frac{\mathrm{Card}\{P \in \mathcal{P}_i \mid \mathrm{rec}(\mathcal{K}, P)\}}{\mathrm{Card}(\mathcal{P}_i)} \tag{2}$$

Each element k_i of $\overrightarrow{K}$ is the proportion of paragraphs of $\mathcal{P}_i$ recognized by $\mathcal{K}$. Since the numbering of the p-classes is totally arbitrary, we can without losing any information define $\overrightarrow{K}' \in \mathbb{R}^p$, $\overrightarrow{K}' = \left(k'_1, \ldots, k'_p\right)$, a vector containing the same values as $\overrightarrow{K}$ but sorted in descending order. We derive our quality measure from the global "profile" of that vector.

Figure 2 shows (in a simplified way) various possibilities for that profile. The first case is quite close to the ideal repartition we are aiming for (where all values are either 0 or 1): a clear separation can be observed between the first four p-classes, of which a relatively important proportion of paragraphs is recognized by the k-class, and the others, mostly "ignored" by it. In the second case, differences still exist between the "recognition rates" of the various p-classes, but they do not clearly fall into two categories. (3), finally, is the furthest from the ideal configuration, a situation where the keyword class does not express a "preference" for any p-class.

In order to detect distributions presenting, like (1) on fig. 2, a strong dichotomy between two sets of paragraph classes, a first idea would be to use a simple measure of standard deviation on the values of $\overrightarrow{K}'$. The drawback of this is that standard deviation reaches its maximum when *half* of the values are 1 and the others 0, whereas the proportion of p-classes that should be "well-recognized" by a given keyword class cannot be known in advance. To avoid that problem, we define a new vector $\overrightarrow{K}'' \in \mathbb{R}^{p-1}$ by:

$$\overrightarrow{K}'' = \left(k''_1, \ldots, k''_{p-1}\right), \text{ with } \forall i \in [1, p-1], k''_i = k'_i - k'_{i+1} \tag{3}$$

$\overrightarrow{K}''$ contains the sequence of all differences between consecutive values of $\overrightarrow{K}'$, so to speak its "derivative" (for each example of $\overrightarrow{K}'$ given on fig. 2, $\overrightarrow{K}''$ is given on the bottom line). The sum of all k''_i, being equal to $k'_1 - k'_p$, cannot be greater than 1, and the standard deviation of the values of $\overrightarrow{K}''$ (written $\sigma_{K''}$) can only reflect the "brutality" of the transition between high and low values of $\overrightarrow{K}'$. $\sigma_{K''}$ thus enables us to make a distinction between profiles (1) and (2) on fig. 2, but is nil for both (2) and (3). To distinguish those two last profiles, it is necessary to combine $\sigma_{K''}$ with the extent of $\overrightarrow{K}'$ (that is, $k'_1 - k'_p$). This combination is made using a simple $(1 + a)(1 + b) - 1$ formula, which lets us balance evenly the contributions of $\sigma_{K''}$ and $\left(k'_1 - k'_p\right)$ to the global quality measure q:

$$q\left(\mathcal{K}\right) = (1 + \sigma_{K''})(1 + (k'_1 - k'_p)) - 1 \tag{4}$$

The q function quite faithfully reflects the correspondence between the thematic classification of paragraphs implied by $\mathcal{K}$ and the p-partition obtained in section 4.1. We now give a general description of the algorithm developed to confront that evaluation of the relative quality of a k-class to the constraint imposed by the k-classification tree.

4.3 Extraction of keyword classes from the word classification tree

The algorithmic use of q aims at furnishing a means to enhance the reading and exploitation of Chavl's initial keyword classification tree in two directions: q is used to 1- automatically point out the relevant keyword classes in the tree independently of the level at which they appear, 2- ignore some of the mergings proposed by Chavl in the tree, and even modify them. The algorithm starts from the leaves of the tree (that correspond to elementary classes of one element), and goes up toward the root checking, for each node met on that way, if the merging suggested by that node between the classes corresponding to its daughters implies a progression of q. If that is the case, the algorithm performs the merging and keeps going up the tree with the new class resulting from it. Else, the algorithm keeps going up the tree,

Algorithm 1 Recursive expression of the tree-reading algorithm.

function extractPartition - **parameter**: a node N of the k-classification tree

if N is a leaf, **result** : $\{\{value\}\}$

classes = $\left\{\bigcup_{i=1..nbDaughters} \{C_i \mid C_i \in (\text{extractPartition}\,(daughter_i) \cup \varnothing)\}\right\}$

partition = $\varnothing$

repeat

C = **select** $c \in$ classes such that $q(c) = \max_{d \in \text{classes}}(q(d))$

partition = partition $\oplus$ C

classes = classes $\setminus \{c \in$ classes $\mid c \cap C \neq \varnothing\}$

until classes = $\{\varnothing\}$

result : partition

but *without* performing the suggested merging: after (b) in fig. 1, we no longer have a keyword class but a set of two classes, {{*cinema*, *movie*, *scene*}, {*town*}}.

When processing subsequent nodes, we now try all possible mergings between classes from those sets of keyword classes to find the most interesting ones in terms of the evolution of q. For example, the algorithm would yield for node (d) on figure 1 {{*cinema*, *movie*, *scene*, *author*, *work*}, {*town*}} + {*lodgment*, *city*}
→ {{*cinema*, *movie*, *scene*, *author*, *work*}, {*town*, *lodgment*, *city*}}

Algorithm 1 gives a more formal, recursive description of that procedure.

A heuristic allowing the early extraction from the algorithm of k-classes that have gone up several nodes without being merged with any other class is used to reduce the complexity of that procedure and lets it run in a reasonable time (about 5 minutes on a personal computer running at 1.5 GHz). Naturally, the classes thus removed from the merging process are then added to the final partition.

By applying that algorithm to the example tree, we finally obtain in (e), the root of the tree, the following partition of the set of all potential keywords: {{*cinema*, *movie*, *scene*, *author*, *work*}, {*town*, *lodgment*, *city*}, {*capitalism*, *life*}, {*employment*, *job*, *unemployment*}}.

4.4 First look at the intermediate results

We have presented in [16] a first system for topic detection and characterization that consisted only of this first set of treatments, followed by a simple filtering of classes based on their sizes. Since no further processing was to be applied to the produced classes (as is the case here in the following stages of our system), we have limited during that previous work the number of studied words to 400 nouns in order to limit the amount of noise in the results. We present here a quick overview of the results obtained by using this system alone, so as to better highlight the interest of the treatments presented in the following sections.

That first system yielded 35 classes, out of which 25 were "intuitively" acceptable as keyword classes. These classes gathered between 5 and 10 words, as the following example result, characterizing the "education / research" topic: {*office*, *center*, *teaching*, *institution*, *research*, *university*, *school*}.

That proposed k-class is a good illustration of the issue we introduced in section 3 as the "object selection problem": many more words come to mind to characterize the topic of education and research than the few ones brought together here, which is due to the limited size of the collection of objects used for the clustering. This is particularly critical here because of our strong constraint on the number of studied words (400 nouns only). Another aspect of the same problem (the selection of relevant words) is the presence of words such as "office" or "center" in the presented k-class: these words are linked with the characterized topic, but are not specific to it. Ideally, they should be withdrawn from any final result, for they have too general a meaning to be accepted as reliable keywords.

The second stage of our system, presented in the next section, retains only the words that show stable associations with others when applying this first system to various random excerpts from the corpus. That way, the second problem of filtering of "useless" words is solved. Moreover, by making the whole system more resistant to noise, it lets us increase the number of processed objects to 1,000 words, thus widening the available vocabulary and providing a partial answer to the first problem as well. The third stage, presented in section 6, extends the obtained

classes by adjoining less frequent words to them, which finally solves that issue of insufficient vocabulary.

5 Merging classifications obtained by random sampling

The problem we have to address now is to build, from a collection of classifications obtained by n executions of the treatment described in the previous section[4], a single classification such that the groupings of words it performs correspond to the majority associations these words exhibit in the n studied classifications. Ideally, each word should appear in the same class as the word with which it appears the most frequently in these n results. But this mechanism is not sufficient to guarantee the generation of k-classes large enough to be relevant for topic detection and characterization. To achieve that goal, we make use of the formalism of graph theory.

All obtained partitions are synthesized as a valued graph $G = (X, \Gamma, v)$, where X is the set of all studied words, Γ gathers all pairs of words that appear together in a same class at least once in the n considered partitions, and $v : \Gamma \rightarrow \mathbb{N}^*$ associates with each such pair the number of times the words are grouped together. A part of the graph thus built is shown on figure 3, where three groups of keywords can be observed around the words "university", "researcher", and "hospital".

The extraction consists in finding strongly connected parts of the whole graph, and is performed by algorithm 2. It proceeds by iteratively selecting the edge of greatest weight in G (from now on referred to as (a, b), of weight V) and automatically define a threshold under which all edges are ignored. Once that thresholding is performed, the vertices contained by the connected part of the graph that feature a and b make up a k-class, which is removed from the graph before repeating that procedure.

The difficulty lies in the definition of a relevant threshold that generates consis-

[4]Experimentally, we observe that results tend to stabilize for $n = 40$.

Algorithm 2 Extraction of strongly connected parts in the graph synthesizing the results of multiple word classifications.

function extractClassesFromGraph - **parameter**: a graph $G = (X, \Gamma, v)$

classes $= \varnothing$

repeat

select $(a, b) \in \Gamma$ such that $V = v(a, b) = \max \{v(m, n) \mid (m, n) \in \Gamma\}$

with each $V' \leq V$, associate the set $\mathcal{S}_{V'}$ of all vertices n such that:

(1) $\forall (n, n') \in \Gamma$, if $v(n, n') \geq V'$, then $n' \in \mathcal{S}_{V'}$

(2) $\exists\, m_0, ..., m_i \in X$ such that

$m_0 \in \{a, b\}$ and $m_i = n$

$\forall\, j < i,\ v(m_j, m_{j+1}) \leq v(m_{j-1}, m_j)$ and $v(m_j, m_{j+1}) \geq V'$

select $W = \min \{V' \leq V \mid \mathcal{S}_{V'} \neq \varnothing \text{ and } \mathrm{Card}(\mathcal{S}_{V'}) \leq 10\}$

classes $=$ classes $\oplus\ \mathcal{S}_W$

remove all vertices in $\mathcal{S}_W$ from G

until $\forall (a, b) \in \Gamma,\ v(a, b) = 1$

result classes

tent classes of suitable sizes. We associate with each value $V' \leq V$ the set of all vertices that can be reached from a or b by a path composed of edges whose weight is at least V'. If any vertex in that subgraph can be reached by a path of edges of monotonously decreasing weights, then V' is a "legal" potential threshold. Else, it is too low and considered to connect too many vertices, and referred to as "illegal". Figure 4 shows a sample graph illustrating that notion. The selected threshold is usually the lowest legal one, unless it generates too large a subgraph (that can happen at the end of the execution of the algorithm, when what remains of the graph is made up of many loosely connected vertices); in that case, the smallest threshold generating a k-class of 10 words or less is retained.

On our data, algorithm 2 produces about 40 classes of an average 6 words each, such as:

- {*school*, *pupil*, *teaching*, *student*, *teacher*, *university*}
- {*researcher*, *laboratory*, *research*, *science*, *technical*, *technology*}

Compared to those presented in section 4.4, these results point out the interest of this treatment: by making the system more noise-resistant, it lets us take more words into account, greatly increasing the topical precision (words related to education and research used to appear in the same keyword class); moreover, non-significant words like *office* or *center* are removed from the classes. Most importantly, contrary to the previous result where only 25 classes in 35 were estimated valid, all of the 40 produced classes bring together topically related words, identifying 40 precise and distinct topics. Hence, we are at this point in the system free from the need of a human filtering of the obtained results, which is an important step towards our initial goal.

However, these classes are still formed from words originally selected according to a frequency criterion during the first stage of analysis. This is why we consider them as "prototype classes", which we shall expand to fully developed ones in the next section, by adjoining less frequent but topically relevant words to them.

6 Class expansion

The classes we have generated in the previous section are now used to build the foundation of a supervised classification inspired from the works presented in [3] or [2], that we have mentioned in the introduction. We make use of the topic detection ability of the 40 prototype classes (still using the criterion according to which if at least two words of a same k-class appear in the same paragraph, then the paragraph deals with the topic underlying the class) to associate with each k-class the set of all paragraphs where it detects its topics. Each k-class is then expanded by adjoining to it the words having the most exceptional frequency over the text gathered in the corresponding paragraph set compared to their average frequency in the corpus. The collections of paragraphs associated with each class are updated as the class grows to reflect the fact that it can detect more and more occurrences of its topic.

Algorithm 3 Expansion of prototype keyword classes.

function expandClasses - **parameter**: $\{\mathcal{C}_{1...n}\}$ a set of n k-classes

for each $i \in \{1...n\}$, **let** LP_i the list of all paragraphs recognized by $\mathcal{C}_i$

repeat

for each $\mathcal{C}_i$, for each potential keyword M_j, compute

$$r_{ij} = \frac{\text{frequency of } M_j \text{ in } LP_i}{\text{frequency of } M_j \text{ in the corpus}} \tag{5}$$

find I_{max} and J_{max} such that $r_{I_{max}J_{max}} = \max_{i,\,j}(r_{ij})$

$\mathcal{C}_{I_{max}} = \mathcal{C}_{I_{max}} \oplus M_{J_{max}}$

update $LP_{I_{max}}$

while adding $M_{J_{max}}$ to $\mathcal{C}_{I_{max}}$ increases the average number

of words of $\mathcal{C}_{I_{max}}$ per paragraph of $LP_{I_{max}}$

result $\{\mathcal{C}_{1...n}\}$

Given the computational lightness of the procedure, we can work on the whole corpus (100,000 paragraphs) at once and consider as potential keywords all nouns and adjectives counting more than 100 occurrences in it (that is, about 3,600 words). The function presented as Algorithm 3 details the process associating these new words with the already-defined prototype classes, and is used twice: first, it is applied to the classes obtained at stage 2 of the system, and stops considering a class as soon as it reaches 10 words; then, it is used for a second pass starting with these 10-word classes and employing a slightly different topic detection criterion: the detection threshold is raised to 3 keywords in a paragraph, instead of 2—this is reasonably possible now that all classes feature 10 words. That gives us a greater confidence in the relevance of the topic detection, and hence in the contents of the set of paragraphs associated with each class. During that second execution, the expansion of a class stops when adding a new word to it would decrease the average number of its keywords in the paragraphs it recognizes. That means that we require the added words to strengthen the detection on paragraphs where the topic is already detected more than they extend the detection. This is how we

guarantee that new words are actually linked with the original topic of the class, and do not extend it toward a connected but distinct concept.

As can be seen on Algorithm 3, each time a class is modified, the corresponding list of paragraphs is re-computed. That necessity is easily understood when considering the confidence intervals of the computed r_{ij} (the measure of "singularity" of the frequency of word M_j over the set LP_i of paragraphs recognized by the class C_i as defined by equation 5 in the algorithm): at the beginning of the process, they are very rough estimates computed for small collections of paragraphs, but this is compensated by the fact that we are dealing with words very strongly connected with the class ($r_{I_{max}J_{max}} \approx 100$, that is, M_j appears 100 times more often in LP_i than in the whole corpus). Later on in the process, we deal with less "marked" words, ($r_{I_{max}J_{max}} \approx 5$), but since the values are computed for large numbers of paragraphs we can be fairly sure that their r_{ij} are actually high—a r_{ij} of 5 computed on 50 paragraphs would not have much meaning.

Naming classes

In order to ease their subsequent exploitation, the k-classes we build are assigned a "name" in the form of a triplet of words selected from the class. We wish these words to give a clear idea of the topic characterized by the keyword class, but also to give an idea of its "range": when talking about education, are we referring to primary school only, or to the whole educational system? To select the words constituting this designation, we use a simple heuristic choosing three words from the class such that the set of paragraphs containing at least one of these words includes as much as possible of the set of paragraphs where the keyword class detects the occurrence of its topic. That way, we are guaranteed to obtain fairly general words (appearing many times) whose meanings reflect the extent of the characterized topic.

In the next section, we present an overview of the obtained results and evaluate their quality from two points of view: that of their intuitive topical consistency for a human reader, and that of their performance for topic detection.

7 Results

Depending on the use one wishes to make of the obtained classes, two complementary criteria apply to evaluate their quality: in order to obtain a quick overview of the contents, nature and "tone" of a corpus, clearly understandable classes faithfully reflecting the kind of vocabulary employed to deal with each topic are desirable. We show in section 7.1 that the keyword classes we produce fully fulfill that wish. Another use of the keyword classes is for spotting the occurrences of topics in the corpus, as for the task we have mentioned of building topical sub-corpora from paragraphs dealing with a same topic. We show in section 7.2 that our classes attain a fairly good precision for that task, which we hence have totally automated, although their recall is limited by the almost infinite numbers of ways in which a given topic can be addressed.

7.1 Intuitive topical consistency of the obtained classes

This is the most immediate and, obviously, most subjective perception of the quality of the obtained keyword classes; hence the appreciations expressed in this section may be qualified by the reader's individual judgment. However, given our initial aim of topic characterization, it is important to control that our keyword classes are easily human-understandable. From that point of view, our objectives seem to be fully met: all keyword classes correspond to a clearly identifiable topic, quite faithfully reflected by the triplet of words whose selection process we have just described. This judgment is, of course, highly subjective, and to let the reader make his / her own opinion we present here the contents of two classes produced by an execution of the presented system on our corpus, that are quite representative of the kind of results yielded, followed by the "names" of the 38 other classes obtained during that same run:

< **teaching / school / university** > { *alphabetization*, *junior high school*, *diploma*, *graduate*, *school*, *education*, *educational*, *pupil*, *professor*, *teaching*, *taught*, *student*, *studied*, *teacher*, *high school*, *mathematics*, *primary*, *pedagogical*,

academic, *schooling*, *secondary*, *academy*, *university* };

< **producer / agriculture / cereal** > { *farmer*, *nourishment*, *food*, *provisioning*, *self-sufficiency*, *wheat*, *cattle*, *cotton*, *cereal*, *cereal grower*, *foodstuff*, *breeding*, *fertilizer*, *famine*, *fruit*, *grain*, *oil*, *intensive*, *irrigation*, *milk*, *vegetable*, *produce*, *egg*, *bread*, *pesticide*, *plantation*, *producer*, *yield*, *meal*, *rice*, *crop*, *bag*, *season*, *dry*, *seed*, *stock*, *sugar*, *surplus*, *dryness*, *cow*, *meat*, *wine*, *corn* };

< technology / engineer / biological >, < army / colonel / guerrilla >, < computer / satellite / machine >, < election / vote / victory >, < agreement / barrier / customs >, < member / meeting / comity >, < work / job / employee >, < minority / language / ethnic >, < rule / law / article >, < duration / professional / qualification >, < rate / decrease / growth >, < disease / hospital / medical >, < alliance / pact / treaty >, < missile / conventional / arsenal >, < nation / council / international >, < territory / occupation / colonization >, < peace / conference / frontier >, < tax / fiscal / insurance >, < pope / faith / bishop >, < reform / farm / owner >, < crime / Nazi / historian >, < prison / judiciary / inmate >, < enterprise / investment / infrastructure >, < neighborhood / park / tourism >, < campaign / democrat / candidate >, < writer / novel / journal >, < television / image / journalist >, < congress / constitution / supreme >, < deficit / debt / budgetary >, < king / Seoudit / Shiite >, < forest / pollution / drought >, < unification / chancellor / East-German >, < fascism / rhetorical / negation >, < asylum / immigration / stay >, < waste / energetic / plant >, < film / movie / screen >, < finance / bank / credit >, < theater / scene / music >, < oil tanker / gas / barrel >.

From the point of view of a casual reader of the newspaper we have worked on (*Le Monde Diplomatique*), the presented classes perform a fairly good coverage of the topics commonly dealt with in our corpus. In particular, one may notice the variations in "granularity" of the topic detection depending on the importance given to various domains in the considered newspaper: thus, all "performing arts" are gathered within a single class (< theater / scene / music >), whereas economical

or geopolitical questions are extremely detailed in many specialized classes. That observation shows the interest of a study carried on without any *a priori* topics to look for, as opposed to what is done in [2] for example.

Similarly, particularly striking events of the considered historical era acquire, by the amount of text that is dedicated to them, the status of topic: the class < territory / occupation / colonization > refers to the Israel-Palestine conflict, while < unification / chancellor / East-German > is concerned with the unification of Germany. Some classes also reveal the "ideological" positioning of the studied newspaper, either by their nature, like < fascism / rhetorical / negation >, corresponding to the articles published when *Le Monde Diplomatique* took part in the polemics raised by the writings of the French revisionist authors Faurisson and Garaudy, or by their content: it is not incidentally that *propaganda* can be found along with *media* in the class < television / image / journalist >.

The classes we obtain, by identifying the main themes in the corpus and giving indications about the way they are addressed in the text, thus provide a fairly good and informative overview of its contents. This is an interesting result as such, e.g. for the quick exploration of large amounts of textual data. However, this intuitive consistency of the produced keyword classes does not guarantee their performance for topic detection. We present in the next section a second evaluation of the quality of the obtained results from that more applicative point of view.

7.2 Efficiency for topic detection

This section proposes some numerical indicators to evaluate the relevance and completeness of the topic detection performed by our keyword classes. The evaluation criterion is more objective than in the previous section, although it still leaves a certain interpretation margin to the experimenter in charge of deciding whether a given paragraph actually features a given topic.

We have so far made use of the simple topic detection criterion presented in the introduction, that consists in considering that a given paragraph deals with a topic if at least two words from the corresponding keyword class appear in it. A first set

of experiments conducted using that criterion has pointed out that it was not strict enough to allow a precise topic detection, because of polysemous words and the possibility for a keyword of a given topic to be used in the context of another—for example "milk" and "cereals" both belong to the class <producer / agriculture / cereal>, but can be used from a consumer's point of view in a text talking about breakfast, not agriculture.

A simple way to strengthen the detection criterion is to raise the number of required keywords to three instead of two. That evolution proves to greatly increase the precision of the performed detection, but also to decrease accordingly the number of paragraphs in which the topic is detected, including legitimate ones. We have therefore devised a third detection criterion that takes into account the structure of the studied text: being a journalistic corpus, it is naturally split into articles, each of which is likely to address one or two main topics. The new detection criterion relies on that assumption of general topical consistency of articles, and proceeds in the following way:

- if at least two paragraphs of an article contain three or more keywords corresponding to a given topic, then the detection threshold for that topic in that article is lowered from three to two and a new detection is performed on the paragraphs of the article with that less demanding criterion;
- else, the detection of the topic in the isolated paragraph is only confirmed if it features four keywords of the corresponding class.

We present in table 1 three numerical indicators to evaluate the results obtained using these three criteria ("at least two keywords", "at least three keywords", and "at least four keywords if the paragraph is isolated in an article, else at least two"):

- "Coverage" is the proportion of paragraphs from the complete corpus in which at least one topic is detected. This value is also given as the proportion of words of the corpus contained in these paragraphs;
- "Precision" indicates what proportion of the paragraphs detected as dealing with a given topic actually do so. It has been computed by hand on a random

selection of 1,000 paragraphs where at least one topic was detected, and is therefore an average value for all the extracted topics; some experiments conducted on smaller numbers of paragraphs but for a single topic suggest that the given values are subject to a $\pm 5\,\%$ relative variation;

- "Recall" is the proportion of paragraphs actually dealing with a given topic where that topic is detected by our system. This value has only been computed for the topic $<$ teaching / school / university $>$ (class detailed in the previous section) by manually looking in the corpus for 100 paragraphs dealing with it[5] and controlling for each of them whether the corresponding k-class detected the topic or not. As with the precision, other, less thorough evaluations were conducted for other topics, which indicate an evaluate $\pm 10\,\%$ relative variation;

These values are computed on the same data that were used to acquire the keyword classes. Indeed, there is no doubt that a possible use of these classes is to detect topics in new texts of the same domain, for example more recent issues of the studied newspaper. However, our goal has been to make the building of these classes totally automatic so as to make it possible to simply generate new, more adapted classes when studying new texts using a different vocabulary.

First of all, the difference between the figures given in number of words and number of paragraphs for the coverage of the corpus calls for an explanation. That difference is implied by the fact that we do not take the size of a paragraph into account to define the number of keywords needed to perform a detection. Therefore, the longer a paragraph, the more likely it is to be detected as dealing with (at least) one topic. Thus, for the third criterion we have presented, that makes use of the logical structuring of the corpus into articles, we compute that paragraphs where at least one topic is detected are $35\,\%$ longer than others.

The measured precision shows that the first criterion we used, requiring a minimum of two keywords only, performs quite poorly for the detection—that was to

[5] That search required to browse about 4,000 paragraphs of text, which explains why recall has not been evaluated on a larger scale.

be expected, as we have shown with the example of *milk* and *cereal* presented at the beginning of this section. This insufficiency is fully corrected by the two other criteria, that perform extremely good despite their almost naive simplicity.

Recall, having only be computed for one topic, only has an indicative value and is likely to fluctuate from one topic to the other. But it lets us observe that the third criterion we have defined manages to conciliate the recall of the first with the precision of the second, which was our purpose. Our chosen priority is the precision of the obtained topic detection, and this aim is fully met. As has been shown in section 2, comparison with existing works in that area is extremely difficult due to the difference in aims (no segmentation of the text, and a requisite of interpretability of the results) and means (no external knowledge or human intervention) of our approach; however, given the essential complexity of the task and the extreme simplicity of the detection criterion we use (classes of words of a limited size), the precision of the detection is very satisfactory. But the obtained recall, essentially limited by the fact that we are working with finite keyword classes when linguistic creativity is virtually infinite, may not be sufficient for other, more demanding applications. We present, as a conclusion, a few tracks for the improvement of the performance of the system in that area.

8 Conclusion

We have presented a set of methods and algorithms that form together a system to extract from textual data, in a totally automatic way, classes of topical keywords. Most of those techniques have applications outside the field of NLP: we have proposed a general framework and algorithm for class extraction from classification trees performing a partial questioning of the mergings proposed by the tree, in the case where object classes imply a classification of variables (section 4 and Algorithm 1), and a method inspired from random sampling that lets us compute reliable classes from noisy data and remove non-relevant objects from the classification in a totally automated way (Algorithm 2). Finally, we have proposed a simple

an efficient method to expand the result of a classification performed on few objects for which much data is available to many other, more sparsely characterized objects (Algorithm 3).

The generated keyword classes accurately reflect the main topics dealt with in the studied corpus, allow a fairly complete and informative overview of its contents, and let us detect the occurrences of the topics in the text with a good precision. Being fully automatic and domain- and language-independent, our system can have many applications for the processing of large collections of documents: classification, indexation, filtering, retrieval... As a drawback to that total automation, its main limitation remains the recall of the performed detection, that shortcoming being mainly due to the willful simplicity of the detection criterion we employ—indeed, the already important size of the obtained keyword classes suggests that extending them further would only slightly increase the performance of the system.

Two main directions of improvement are to be considered: the first consists in questioning our choice of predefining text segments as whole paragraphs. This leaves the possibility for keywords of a same class to appear in the same text segment but totally unrelated to each other (for example at the very beginning and at the very end of a same paragraph), which could lead to an erroneous topic occurrence detection if we did not use a fairly strict detection criterion (the co-occurrence of three keywords). Performing a first segmentation of the text based, for example, on linguistic clues such as presented in [12] as a preliminary to our system would let us use more "permissive" criteria without any loss in precision. However, that pre-treatment requires the intervention of an expert to select relevant linguistic clues, and is therefore reserved to applications where such an intervention is affordable.

A second direction for improvement, this time maintaining full automaticity, consists in using our results as the training basis for systems such as [3] or [2] that use supervised learning techniques to build language models to characterize the kind of text corresponding to each topic in the corpus. Through these systems, we can reach higher levels of accuracy and recall without losing the main interest of our work: the ability to obtain results adapted to any kind of corpus without any

human intervention.

Acknowledgements

The authors wish to thank I.C. Lerman for his precious help in exploiting the CHAVL method to its full potential, and for working on the adaptation of its implementation to their specific needs.

Many thanks go to the anonymous reviewers, whose comments were a great help to raise the level of technical and formal correctness of this paper.

References

[1] Susan Armstrong. Multext : Multilingual Text Tools and Corpora. In H. Feldweg and W. Hinrichs, editors, *Lexikon und Text*, pages 107–119, 1996.

[2] Brigitte Bigi, Renato De Mori, Marc El-Bèze, and Thierry Spriet. Combined Models for Topic Spotting and Topic-Dependent Language Modeling. In S. Furui, B.H. Huang, and Wu Chu, editors, *1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 535–542, Santa Barbara, CA, USA, 1997.

[3] Armelle Brun. *Détection de thèmes et adaptation des modèles de langage pour la reconnaissance automatique de la parole.* PhD thesis, Henri Poincaré University, Nancy, France, 2003.

[4] Brad Efron and Robert Tibshirani. Statistical Analysis in the Computer Age. *Science*, 253:390–395, 1991.

[5] Olivier Ferret and Brigitte Grau. A Bootstrapping Approach for Robust Topic Analysis. *Natural Language Enginering (NLE), Special issue on robust methods of corpus analysis*, 8(3):209–233, 2002.

[6] Marti A. Hearst. Multi-Paragraph Segmentation of Expository Texts. In *ACL'94 (32th Annual Meeting of the Association for Computational Linguistics)*, pages 9–16, Las Cruces, NM, USA, 1994.

[7] Nancy Ide and Jean Véronis. Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, 24(1):1–40, 1998.

[8] Roland Kuhn and Renato de Mori. A Cache-Based Natural Language Model for Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583, 1990.

[9] Krista Lagus, Timo Honkela, Samuel Kaski, and Teuvo Kohonen. Self-organizing maps of document collections: A new approach to interactive exploration. In Evangelios Simoudis, Jiawei Han, and Usama Fayyad, editors, *Second International Conference on Knowledge Discovery and Data Mining*, pages 238–243. AAAI Press, Menlo Park, California, 1996.

[10] Israël-César Lerman. Foundations in the Likelihood Linkage Analysis Classification Method. *Applied Stochastic Models and Data Analysis*, 7:69–76, 1991.

[11] Israël-César Lerman, Henri Leredde, and Philippe Peter. Principes et calculs de la méthode implantée dans le programme CHAVL (Classification Hiérarchique par Analyse de la Vraisemblance du Lien) – deuxième partie. *Revue de* MODULAD, 13:63–90, 1994.

[12] Diane J. Litman and Rebecca J. Passonneau. Combining Multiple Knowledge Sources for Discourse Segmentation. In *ACL'95 (33th Annual Meeting of the Association for Computational Linguistics)*, pages 108–115, Montréal, Québec, Canada, 1995.

[13] Ronan Pichon and Pascale Sébillot. From Corpus to Lexicon: from Contexts to Semantic Features. In Barbara Lewandowska-Tomaszczyk and Patrick James Melia, editors, *PALC'99 (Practical Applications in Language Corpora), Lodz Studies in Language*, volume 1, pages 375–389. Peter Lang, 2000.

[14] Rodolphe Priam. *Méthodes de carte auto organisatrice par mélange de lois contraintes. Application à l'exploration dans les tableaux de contingence textuels.* PhD thesis, Rennes I University, Rennes, France, 2003.

[15] François Rastier. *Sémantique Interprétative.* Presses Universitaires de France, second edition, 1996.

[16] Mathias Rossignol and Pascale Sébillot. Automatic Generation of Sets of Keywords for Theme Detection and Characterization. In Annie Morin and Pascale Sébillot, editors, *JADT 2002 (Journées internationales d'Analyse des Données Textuelles)*, pages 653–664, Saint-Malo, France, 2002.

[17] Gerard Salton, Amit Singhal, Chris Buckley, and Mandar Mitra. Automatic Text Decomposition Using Text Segments and Text Themes. In *Hypertext'96 Conference*, pages 53–65, Washington D.C., USA, 1996.

[18] Alan F. Smeaton. Using NLP or NLP Resources for Information Retrieval Tasks. In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*, pages 99–111. Kluwer Academic Publishers, 1999.

Appendix A: similarity measure used by Chavl

As has been said, we exploit the Chavl method to classify words according to the proximity of their distribution over a given set of paragraphs. Hence, each object to be classified is defined by a vector indicating its number of occurrences in each paragraph of the selection. By a slight shift of point of view, it is possible to consider the array gathering those vectors as a contingency table crossing two variables (word form, or *lemma*, and paragraph of appearance) measured on a population of word occurrences. Hence, the similarity measure we have chosen to use to bring together word forms is the one implanted in Chavl to process contingency tables. We summarize here the principle of this measure, which is fully described in [11].

Let $L = l_{i,1\leq i\leq m}$ the lemmas of the considered word population, $P = p_{j,1\leq j\leq n}$ the paragraphs composing the studied corpus excerpt, and $k_{ij,1\leq i\leq m,1\leq j\leq n}$ the number of objects (*i.e.* word occurrences) having the lemma l_i and appearing in the paragraph p_j. We write $k_{i\bullet}$ to denote the total number of objects having the lemma l_i, $k_{\bullet j}$ for the total number of word occurrences appearing in the paragraph j, and $k_{\bullet\bullet}$ for the total number of objects.

The contingency table is essentially symmetrical; however, in order to classify word lemmas, we choose to give L the role of the objects set, and P the role of the variables set. With L is associated the point cloud in $\mathbb{R}^n$: $N(L) = \left\{\left(f_P^i, p_{i\bullet}\right), 1 \leq i \leq m\right\}$, where f_P^i is the point in $\mathbb{R}^n$ representing l_i, whose j^{th} coordinate is given by:

$$f_j^i = \frac{k_{ij}}{k_{i\bullet}}, 1 \leq j \leq n \tag{6}$$

and where $p_{i\bullet}$ is the weight of the point, given by:

$$p_{i\bullet} = \frac{k_{i\bullet}}{k_{\bullet\bullet}} \tag{7}$$

For statistical reasons, $\mathbb{R}^n$ is given the diagonal χ^2 metric:

$$\left(\frac{1}{p_{\bullet j}}, 1 \leq j \leq n\right), \text{ where } p_{i\bullet} = \frac{k_{\bullet j}}{k_{\bullet\bullet}} \tag{8}$$

The similarity is computed using φ_j^i values, which result from the centering, reduction and normalization of f_j^i according to the chosen metric:

$$\varphi_j^i = \frac{\left(f_j^i - p_{\bullet j}\right)/\sqrt{p_{\bullet j}}}{\sqrt{\sum_{1 \leq h \leq p} \left(f_h^i - p_{\bullet h}\right)^2 / p_{\bullet h}}} \tag{9}$$

The raw contribution of p_j to the comparison of i and i' is:

$$s_j\left(i, i'\right) = \frac{1}{p} - \frac{\left(\varphi_j^i - \varphi_j^{i'}\right)}{2} \tag{10}$$

and the normalized contribution is computed using the average and variance values taking into account the weights defined by equation 7. The similarity measure is finally obtained by adding the normalized contributions of all variables for each pair, and applying one last normalization to the set of all similarities using the global average and variance.

	Coverage		Precision	Recall "education"
	(paragraphs)	(words)	(paragraphs)	(paragraphs)
Two words	66 %	70 %	55 %	65 %
Three words	32 %	40 %	85 %	35 %
Three words + structure	58 %	65 %	85 %	63 %

Table 1: Coverage, precision and recall of the achieved topic detection, computed for the three introduced criteria. "Three words + structure" refers to the criterion making use of the structuring of the corpus into articles.

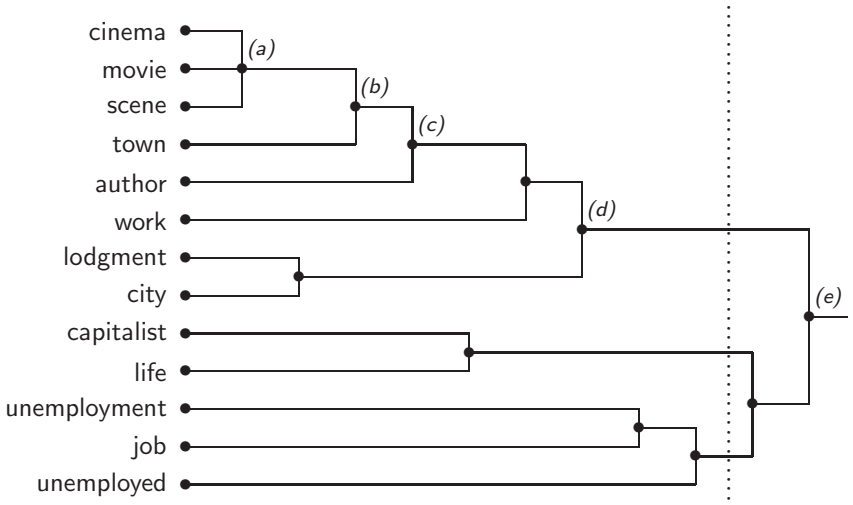

Figure 1: Small example of a classification tree on nouns and adjectives as produced by CHAVL. The dotted line represents one of the "levels" of the tree, commonly used to extract classes by cutting branches.

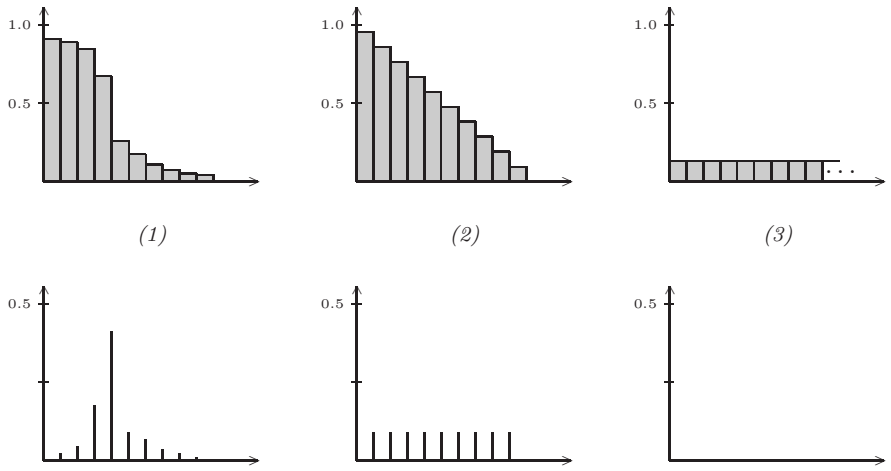

Figure 2: On top, three graphs showing, for a collection of paragraph classes (x-axis), the proportion of their paragraphs that is recognized by a keyword class (y-axis); below are plotted their "differentials".

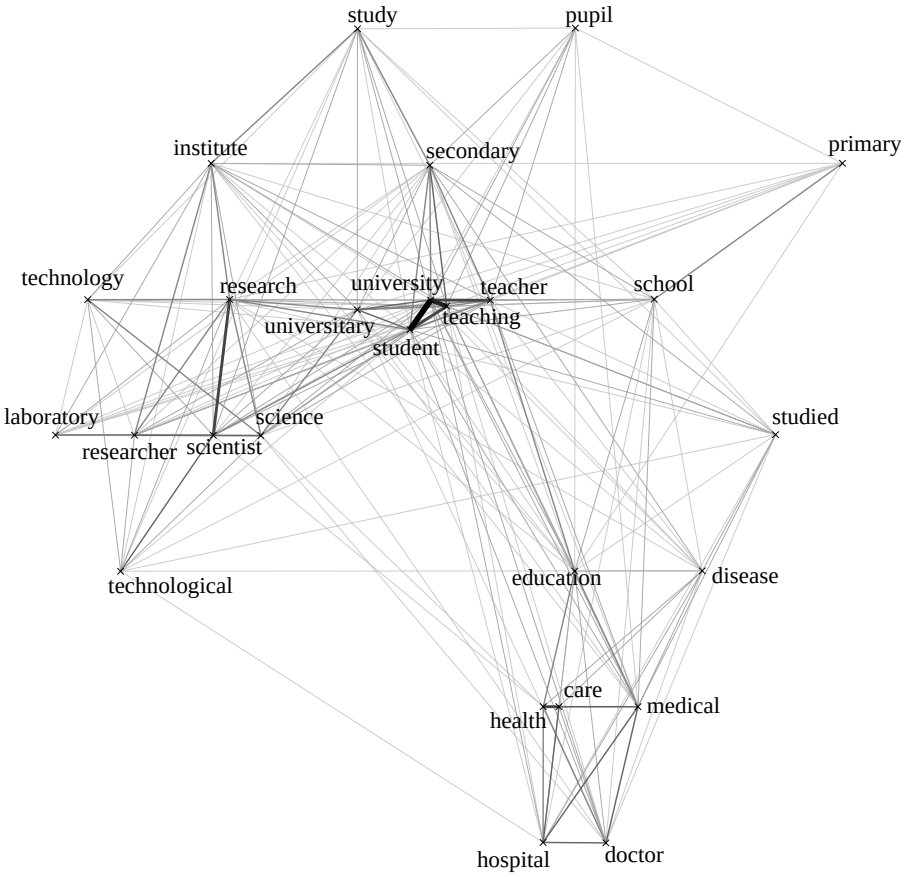

Figure 3: Excerpt from the graph of proximity between potential keywords, drawn for a study taking into account 1,000 nouns and adjectives, represented for a close neighborhood of the noun "university". Word locations have been computed by an algorithm attempting to bring two objects closer as the link between them has an important weight (thick, dark lines on the figure). Links of weight 1 have not been drawn for the sake of clarity.

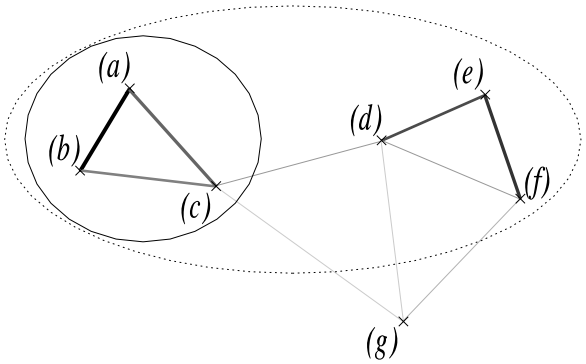

Figure 4: Sample graph illustrating the threshold selection process in the algorithm for class extraction from a graph: (a, b) being the strongest edge in the graph, the continuous line delimits the set of vertices selected by using a fairly high threshold, the dotted one corresponds to a lower threshold (as on figure 3, the thicker and darker a line, the "heavier" the corresponding edge). The latter is "illegal", for no monotonously decreasing path leads from either (a) or (b) to (e).