Automatic generation of sets of keywords for theme characterization and detection

Mathias Rossignol, Pascale Sébillot

IRISA – Campus de Beaulieu – 35042 Rennes cedex – France – {mrossign|sebillot}@irisa.fr

Abstract

The paper describes a system that automatically detects themes in a textual corpus and characterizes them by sets of keywords, that is, words whose co-occurrence in a paragraph indicates that this paragraph tackles a certain theme. (Pichon and Sébillot, 2000) presents a first version of it where those sets are obtained with the help of the CHAVL hierarchical clustering algorithm, grouping words that have a similar repartition over paragraphs. The weaknesses of the system (quality of the classes highly dependent on manual parameter settings, relevant classes in the classification tree hardly pointed out automatically) are largely reduced here by using a combined classification of the paragraphs based on their lexical cohesion. This new classification first allows to densify the processed data, thus helping CHAVL produce more satisfying classes; it also gives a means to establish an original statistical quality measure that can be exploited both to point out the relevant classes in the tree and to reorganize some of the mergings proposed by CHAVL.

Keywords: corpus theme detection, keywords, hierarchical clustering

1. Introduction

Most linguistic theories agree at least on one point, that has been widely confirmed by many works in the field on natural language processing: the meaning of a text cannot be fully grasped only by short range semantic analysis (such as the detailed study of all its sentences, taken independently), and even a human reader can have difficulties understanding an isolated sentence without any contextual information. A reason for that lies in the fact that words cannot be assigned a meaning before the boundaries of the " semantic universe " in which they may be interpreted have been set, thus contributing to reducing their possible semantic role in the text to a single interpretation. Themes represent such boundaries: stating that a text deals with computers, for example, prevents any further questions about the meaning of " mouse " in that text, and such knowledge can be of great use for detailed semantic analyses of it. Automatic theme recognition thus presents itself as a very interesting starting point for most information retrieval tasks, but also as a way to detect the logical structure of a written document (e.g. for summary generation) and even as a goal in itself for more elementary tasks such as basic web-searching, " intelligent " sorting of messages, *etc.*

The work presented in this article aims at providing simple, easily interpretable elements allowing a reasonably good guess of the topics tackled by the various parts of a text. Due to linguistic and practical reasons, that will be detailed later, the theme recognition method adopted here is fairly simple:

- themes are detected for whole paragraphs of text;
- each theme is characterized by a set of *keywords*, and a paragraph is recognized as dealing with that theme if at least two of these keywords appear in it.

The goal of the system introduced here is to automatically build from a morphosyntactically tagged corpus such sets of keywords characterizing all of the main themes appearing in the text, using statistical data analysis methods and without any preliminary semantic knowledge about the corpus; notably, we have no *a priori* information about the themes addressed in the corpus, their number, and the sets of words that indicate their presence. The core of this system was first presented in (Pichon and Sébillot, 2000); it generates sets of topic keywords by performing a clustering of keyword candidates based on the resemblance of their repartition over paragraphs, using the CHAVL classification algorithm (Lerman, 1991). The results of that first system, although interesting, present two main weaknesses:

- the low density of the processed data (number of word occurrences in each paragraph) makes it difficult to obtain reasonably good classifications; repeated attempts with different parameter settings are therefore necessary in order to produce satisfying results;
- the classification method used yields its results in the form of a classification tree, and the extraction of classes of keywords from that tree remains in great part manual.

The aim of this paper is therefore to show how, with the help of a second classification tree of the paragraphs of the corpus based on the vocabulary they contain, we solve those two problems by statistical means in order to reduce as much as possible the need for human intervention. That classification of paragraphs first permits to densify the data used by CHAVL and to increase the quality of the resulting keyword classification tree; then, it enables us to define an original statistical measure that allows the development of an algorithm responsible for automatically extracting from that tree the " most interesting " keyword classes.

Before developing an explanation of those two points, the first section of this article presents a few existing works addressing theme detection. This short overview enables us to underline what characteristics distinguish our goal from that of those studies, and how those expected features, combined with linguistic theory, justify the choices made here to treat whole paragraphs and detect themes using keywords. Section 3 then exposes the state of the first version of our system, that is, (Pichon and Sébillot, 2000). The strengths and weaknesses of that first study clearly put into light the logic that underlies the proposed solutions, which we detail in section 4. A short evaluation of the achievements reached so far finally concludes this paper.

2. Relative situation

The problem of automatic theme detection has already given rise to many researches. Many of those works simultaneously perform discourse segmentation and theme recognition, with the help of linguistic cues (Litman and Passonneau, 1995) or notions such as lexical cohesion. We only present here two methods of that last family, based on statistical measures, and explain the choices made in the present work in the light of those references.

2.1. Related works

One of the first accomplishments in the field of theme detection, TextTiling (Hearst, 1994), is a tool performing the segmentation of a text in groups of successive paragraphs dealing with the same topic, based on a measure of lexical similarity between consecutive sequences of words. At regular intervals in the text (about twenty words), the algorithm computes the resemblance between the two lists of all words appearing in an interval of some 100 words before and after the focus point. A local minimum of that measure reveals a point of the text showing little lexical consistency, which is considered as an indication that a transition between two themes occurs around the location where the minimum appears. The gap thus detected is then rounded

up to the closest paragraph boundary. The words that have played the most prominent role in maintaining the lexical proximity measure at a relatively high level between two such gaps are used to indirectly characterize the theme of the corresponding region. The drawback of that method is that if two separate regions of the text deal with the same theme, nothing guarantees that the sets of characteristic words retained for each of them will be identical, thus making it difficult to automatically detect thematic proximity between non-consecutive text segments.

Based on a similar idea of lexical cohesion, the work presented in (Ferret and Grau, 2001) performs much finer-grained segmentation, the lexical consistency measure being this time computed for every word of the text, for a window of ten words left and right of the focus point. Since that process uses much less data than above, complementary information is exploited to enrich the computation: a first operation consists in building from a large corpus (about 45 million words) a network of collocations between words, which is then used to perform a first segmentation taking those collocations into account in the lexical proximity measure. Thanks to the results of that first segmentation, the system makes an inventory of " thematic signatures " that form the basis of a second segmentation operation and provide as well an indirect but, contrary to TextTiling, consistent characterization of the themes thus detected.

2.2. Choices

The work we describe here takes place in a more general research aiming at automatically building a semantic lexicon from a raw text corpus using statistical methods (Pichon and Sébillot, 1999), one of the characteristics of the chosen corpus (about eight million words from the archives of the French newspaper *Le Monde Diplomatique*) being that it deals with a wide variety of subjects, from geopolitics to fine arts. For that reason, many words are likely to be used with different meanings in different parts of the corpus, and blindly collecting statistical data about them can only lead to the formation of semantic nebulæ from which no consistent average behaviour can be extracted. To come back to the " mouse " example from the introduction, the choice made to solve that problem was to consider " mouse in the Computer theme " and " mouse in the Biology theme " as two distinct entities. We therefore need to be able, for a given word occurrence, to immediately recognize in what thematic context it appears.

A first consequence of that constraint of immediacy is that we cannot afford to perform, like the aforementioned tools, a linear analysis of the complete text in order to operate discourse segmentation. It is therefore necessary to predefine a text subdivision that will be definitely considered as the atomic theme-consistent unit. Paragraphs immediately seem the fittest for that purpose, mainly because of the role they traditionally play in the writing process, as a hint to the reader — a visual structuration of the text that most often reflects its logical structure. Of course, digressions often appear in the middle of a paragraph, but that does not really represent a danger with respect to our goal, since those modulations seldom differ enough from the original theme to leave possibility for widely different word meanings to appear. It was nonetheless desirable to detect in that case the presence of concurrent themes, hence another constraint put on the theme recognition criterion: that a paragraph may be assigned several themes.

That constraint is satisfied by the principle of co-occurrence of keywords, which is not a completely arbitrary, intuitively chosen criterion, but the algorithmic equivalent of a linguistic concept called *semantic isotopy*. This concept is drawn from *Differential Semantics* (Rastier, 1996), the linguistic theory that was chosen as the meaning representation paradigm for the lexicon building task that was our primary goal. In Differential Semantics, word meanings are implied

by relational markers, or *semes*, denoting the fact that two words share an element of meaning (for example, " means of transportation " can link " bicycle " and " car ") or are distinguished by one (" wide " making a distinction between " river " and " creek "). A semantic isotopy is the repeated presence in a portion of text of a group of such semes between words and characterizes the evocation in that region of the theme underlying that group of semes. For example, the co-occurrence in a text segment of the words " soldier ", " tank " and " attack " clearly denotes a war theme because of the shared semes " army " and " agression ", whereas none of those words alone is sufficient to detect the presence of that theme. Since, as already specified, one of our main requirements is not to rely on any semantic knowledge, making use of semes is obviously out of the question, and we are only able to study the manifestation of semantic isotopies, that is, the co-occurrence of certain words — or *keywords* — in a paragraph.

Given a set of such keywords, it is quite obvious that, being symptomatic of some theme, they will all appear often in paragraphs dealing with it, and therefore have quite similar repartitions over the paragraphs of our corpus. That simple observation forms the basis of a first series of experiments aiming at automatically building sets of keywords, carried on at the IRISA and whose principle still lies at the core of our system: using a clustering algorithm to create classes of words having similar repartitions. The next section details the implementation of this basic idea before highlighting its weaknesses and indicating the directions taken to compensate them.

3. Pre-existent work

Before describing the operating procedure presented in (Pichon and Sébillot, 2000) to generate sets of keywords, we quickly present here CHAVL, the clustering algorithm used to perform word classification, and expose the reasons for that choice. More information about the statistical foundations of that classification method can be found in (Lerman, 1991), and (Peter et al., 1992) provides more details about `chavleps`, a FORTRAN implementation of CHAVL.

3.1. CHAVL*: short introduction*

CHAVL (for *Classification Hiérarchique par Analyse de la Vraisemblance des Liens* — " Hierarchical classification by linkage likelihood analysis ") is a classification method developed by Lerman (Lerman, 1991). Its goal is, given a set of elements and a matrix gathering similarity measures between those elements (*i.e.* numerical values reflecting the degrees of resemblance between all pairs of elements) to build a classification tree organizing those items in growing classes of optimal " density " (see fig. 1). Each node of the tree merges the classes corresponding to its sons, thus creating a more general but also less consistent class, the purpose of the algorithm being to organize those mergings and their order (controlled by the structure of the tree) so as to optimize the consistency of the generated classes with respect to the similarity measures initially provided.

A classical way of reading such a classification tree consists in choosing a level in it (for example, the dotted line on fig. 1), and extracting the classes corresponding to the branches intersecting that level. As we shall see later, this " traditional " process will not be adapted in our case, and that raises the necessity of conceiving a way to read the tree so as to find in it the most interesting classes. One could wonder, meeting that problem, why a clustering algorithm directly yielding classes of elements was not chosen. The main reason lies in the fact that, in order to work properly, those algorithms need to be given initial indications such as the number of classes to produce or the number of elements in each class, and we initially know neither how

many themes appear in the studied corpus, nor whether all of them have an equally important presence in it, which means that the sizes of keyword classes may greatly vary.

Another interesting aspect of CHAVL is that its implementation, `chavleps`, can be used simply as a tool to build classification trees with user-defined similarity values, but also proposes a set of built-in similarity measures defined for various kinds of " classical " statistical data (boolean vectors, collections of numerical values, *etc.*), one of them being particularly adapted for the distribution values we shall be treating.

We now present the results we have obtained by reproducing the *modus operandi* described in (Pichon and Sébillot, 2000), and point out the weaknesses of the system.

3.2. Operating procedure

In a first step, the archives of *Le Monde Diplomatique* are segmented, morphosyntactically tagged and disambiguated using the MULTEXT tools of ISSCO[1]. To facilitate subsequent collecting of data, each of the 80,000 paragraphs of text is stored in its own file. In order to keep the treatments reasonable in terms of computational complexity, only a random excerpt of 10 % of the complete corpus is used (from now on, all mentions made to " the corpus " will refer to that excerpt of 8,000 paragraphs). The words retained as potential keywords are the nouns appearing more than sixty times in that excerpt. The choice to use only nouns was guided by the commonly accepted fact that they constitute the most salient theme indicators, and the constraint imposed on the number of occurrences comes from the necessity of having enough data to allow a successful application of the statistical measures used by CHAVL. With each of the 383 nouns thus selected is associated a vector of 8,000 values representing its number of occurrences in each paragraph of the corpus, and the matrix gathering those 383 vectors is processed using `chavleps`.

3.3. First results

The trees produced by `chavleps` during the experiments we made with this operating procedure were similar to that of fig. 1, although much larger, but also of much lesser quality: managing to get a tree whose nodes corresponded at least for a good part to intuitively correct gatherings of words or classes (as fig. 1, where associating " job " and " unemployment ", and then adding " employment " to that class seems intuitively satisfying for theme characterization) required many attempts and manual modifications of a few arbitrary parameters.

The reason for that difficulty in getting CHAVL to yield satisfactory results was shown to come from a peculiarity of the processed data: the studied nouns appearing on average about a hundred times in the studied excerpt, a simple calculation shows that more than 98 % of the values contained in the matrix treated by `chavleps` are nil. In those conditions, even simple statistical measures, like variance, become extremely small, and the more sophisticated ones defined in CHAVL reach a point where their value can disappear behind the computation approximations inherent to operations on floating-point values. It is therefore necessary, in order to get satisfactory trees to work with, to find a way to " densify " the matrix gathering word repartition data in such a way that as little information as possible is lost in the process, or even better: that singularities in the distribution of words over paragraphs become even more outstanding. That is the first problem we propose a solution for in the next section, but it leaves open the question

[1] For more details about those tools, see `http://issco.unige.ch/projects/MULTEXT.html`

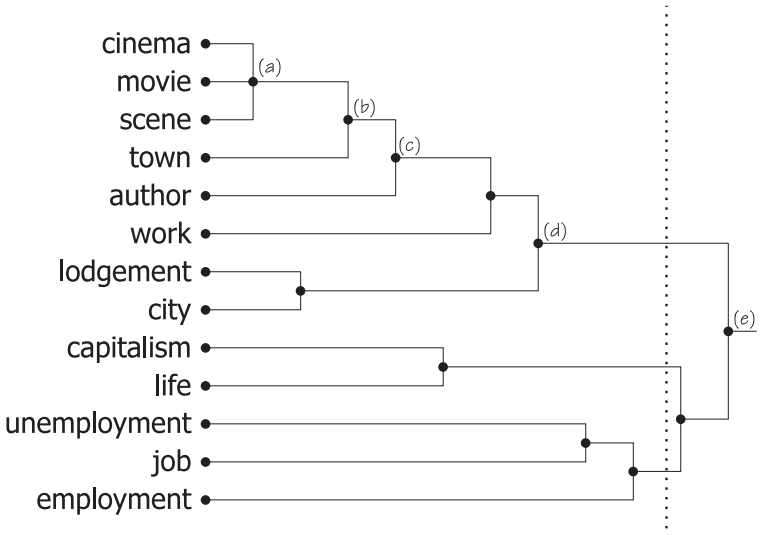

Figure 1: *Small example of a relatively good classification tree built on nouns (translated into English). The dotted line illustrates the most common way of reading this kind of tree, by choosing a level and "cutting branches" there; it gives the following classes: {*cinema*,* movie*,* scene*,* town*,* author*,* work*,* lodgment*,* city*}, {*capitalism*,* life*} and {*unemployment*,* job*,* employment*}.*

of how to extract classes of keywords from the tree we obtain.

As can be seen in fig. 1, the traditional method consisting in choosing a single level of the tree and reading classes along that line is not adapted to the kind of results we get: trying to obtain the (thematically interesting) class {*unemployment*, *employment*, *job*} makes it necessary to choose a level so close to the root that other potentially interesting classes get mixed into a "cinema-city" nebula... A first idea would be to try and find a way to pick at various levels of the tree the nodes corresponding to the most interesting classes of keywords, but a closer look at the example shows that it presents imperfections one could wish to correct: the word "town", instead of acting as a "parasite" at node (b), in the middle of a series of mergings that would lead without it to the keyword class {*cinema*, *movie*, *scene*, *author*, *work*}, could more profitably be added to the class {*city*, *lodgment*}. Therefore, the goal of an "intelligent" tree-reading method will be not only to be able to detect for each branch of the tree the point where subsequent mergings generate inconsistent classes, but also to operate minor re-organizations of that tree so as to obtain the best possible classes.

We now describe our original statistical solution for those two problems, by means of an alternate use of the very same word repartition data we have been treating so far.

4. Implemented solutions

The questions raised by the two problems we intend to tackle all end up reaching a common point, the need for some reference, a first result providing a more "elevated" point of view on the elements we are treating than the raw word repartition matrix. The choice made to obtain such a reference was to keep using the same repartition data we have been exploiting so far, but this time for the reciprocal operation: instead of building a classification tree of nouns according to the paragraphs they appear in, we perform a classification of paragraphs according to the nouns they contain. In order to avoid the problems experienced during our first uses of CHAVL, that were due to the sensitivity of its sophisticated statistical measures to the emptiness

of the studied matrix, a very rudimentary similarity measure is used. That technique enables a partition of all paragraphs into a set of approximately thematic classes of a dozen paragraphs each, and that classification, although of relatively poor quality, still proves reliable enough to:

- significantly improve the quality of the classification of words by simply changing the definition of the data matrix from " the number of occurrences of each word in each paragraph " to " the number of occurrences of each word in each *class of* paragraphs ";
- provide a foundation for the definition of a " quality measure " on sets of keywords based on the idea that those sets of keywords, by detecting themes, perform a thematic classification of paragraphs that should, ideally, match the one we already have.

Since this section alternatively deals with classifications made on paragraphs and candidate keywords, we define a notation that should avoid any ambiguities and make the formulation of what follows much easier: the prefix *k-* will be used to denote operations made on potential keywords (" k-classification tree ": a classification tree built on keyword candidates), and *p-* will characterize operations made on paragraphs (" p-partition ": a partition in classes of the set of all paragraphs).

The rest of this section begins with explaining and justifying the similarity measure used to generate a p-classification, and shortly presenting the results obtained by exploiting it to redefine the word repartition data used for the k-classification. We then spend more time detailing the foundation of the quality measure defined on k-classes and the operating principle of the algorithm we developed to confront it with the k-classification tree.

4.1. Classification of paragraphs

4.1.1. Principle of an alternate similarity measure

The similarity measure we define to achieve simple p-classification is a direct translation of the idea of " lexical proximity " that underlies the works presented in section 2.1, and mainly consists in counting the number of words two paragraphs have in common in order to roughly evaluate how thematically close they may be. As a refinement to that principle, it takes into account the fact that since uncommon words have more chances than others to have a very narrow and specific range of meanings, the possibilities of appearance of each of them are likely to be restricted to very few themes, making the sharing of rare words a particularly outstanding indications for thematic classification. To reflect that observation, the importance of each word in the calculation is inversely proportional to its number of occurrences in the corpus. Finally, the measure is normalized according to the size of the paragraphs compared.

4.1.2. Mathematical expression

Let $A = (a_i)$ and $B = (b_i)$ be the vectors gathering the number of occurrences of each word taken into account for this computation in any two paragraphs, n_i the total number of occurrences of the word i in the corpus, and p and q the numbers of words in the two considered paragraphs. The similarity between those paragraphs is defined by:

$$\frac{1}{\min(p,q)} \sum_i \frac{\min(a_i, b_i)}{n_i}$$

An obvious property of that measure is that, contrary to the one provided by CHAVL, it can easily take into account words appearing very rarely in the corpus. This enables us to use for the computation of the similarity value all nouns with at least two occurrences, thus reaching a

total of about 3,000 nouns and partially compensating the " naiveness " of the measure by the amount of available data.

4.1.3. Generation of classes of paragraphs

An 8,000 × 8,000 matrix (half-matrix, actually, since the similarity measure is symmetrical) gathering the similarity values thus defined between all pairs of paragraphs is used by the `chavleps` program to directly build a p-classification tree without using its own similarity measure. Although it is quite impossible to manually check the validity of that classification, the tree obtained by those means is very well balanced, which is a sign of " smooth " functionning of the clustering algorithm. Classes are extracted from it using the " traditional " method of cutting all the branches of the tree at a given level, chosen so as to provide p-classes whose size is as close as possible to an arbitrary value. An empirically chosen value of twelve proved to be a good compromise between generalisation and thematic consistency of the classes.

We finally obtain a p-partition of 544 classes which we hope group paragraphs dealing with a same theme. One could wonder here why, having achieved some degree of theme detection, we do not consider that p-partition as a result in itself. Four facts can explain that decision:

- a few random checks show that the p-classes we obtain, if they do have a certain level of consistency, are far from being of sufficient quality to be considered as a final result;
- the computation of the similarity measures between all pairs of paragraphs is extremely demanding in memory and cpu, making the technique described here unsuitable for corpora of more than 15,000 paragraphs, whereas the keyword classes we intend to build on a small extract of the corpus should be able to recognize themes in the whole text;
- the p-partition in itself does not provide any information about *what* the theme of the paragraphs of each class is (or at least, should be), that point being one of the main strengths of the technique using keyword classes: an immediately interpretable result;
- finally, nothing has been made to ensure that each p-class groups *all* the paragraphs dealing with a given theme, in other words each theme addressed in the corpus is very likely to concern several p-classes.

For those reasons, the p-partition we have just obtained is used as a " stepping stone " towards other, most satisfying results. Its first role is to improve the k-classification process.

4.2. Densification of the repartition data for k-classification

Coming back to the k-classification task, we can now change the definition of the repartition matrix (*cf.* section 3.2) so that each of its lines represents the number of occurrences of a given word in each class of paragraphs. This new matrix is significantly smaller and denser than the previous one (it now features 544 columns, versus 8,000 before), and its processing by `chavleps` produces a better balanced tree than the ones we had obtained in our first experiments, where more of the mergings suggested by the nodes are intuitively satisfying — although they are still far from being all perfect. A particularly interesting sign showing that the implemented densification process actually managed to increase the " relief " of the repartition data is that, contrary to what happened during the attempts of k-classification made in section 3.3, small variations of parameters now have very little influence on the obtained result.

Having managed to obtain a satisfactory k-classification tree, we can now proceed to the more complicated task of performing an optimal exploitation of it.

4.3. Extraction of k-classes from the k-classification tree

4.3.1. Principles of the k-class quality measure

The quality measure we now define is based on a simple idea born from our initial definition of keywords: if two keywords of the same set appear in a given paragraph, then that paragraph deals with the theme 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 p-partition we have defined in section 4.1, those two classifications should coincide as much as possible (that correspondance 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 theme and each k-class recognizes *all* the paragraphs dealing with a given theme and *only* them, we have the following situation: if a k-class recognizes one of the paragraphs of a p-class, since those paragraphs all tackle the same theme, 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 define now is an attempt to mathematically define a value giving preference to k-classes getting as close as possible to that ideal configuration.

4.3.2. Mathematical expression

Let $\mathcal{K}$ be a k-class and $\mathcal{P}_1, \ldots, \mathcal{P}_n$ all the p-classes defined by the p-partition of section 4.1 (where $n = 544$). 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 $K \in \mathbb{R}^n$, defined by:

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

Each element k_i of 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 $K' \in \mathbb{R}^n$, $K' = (k'_1, \ldots, k'_n)$, a vector containing the same values as 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 getting rather close to the ideal repartition we are looking for (where all values are either 0 or 1): one can notice a clear separation 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 k-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 p-classes, a first idea would be to use a simple measure of variance on the values of K'. The drawback of this is that variance reaches its maximum when *half* of the values are 1 and the others 0, whereas the proportion of p-classes that will be " well-recognized " by a given k-class cannot be known in advance. To avoid that problem, we define a new vector $K'' \in \mathbb{R}^{n-1}$ by:

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

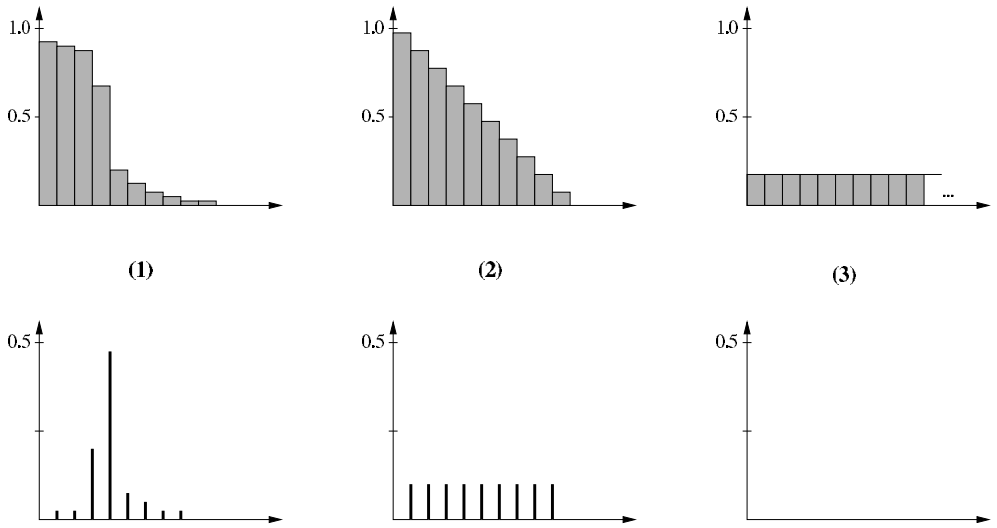

Figure 2: *On top, three graphics showing, for a collection of p-classes (x-axis) what proportion of their paragraphs is recognized by a given k-class (y-axis). The bottom line shows the differences between consecutive values.*

K'' contains the sequence of all differences between consecutive values of K', so to speak its " derivative " (for each example of K' given on fig. 2, K'' is given on the bottom line). The sum of all k''_i, being equal to $k'_1 - k'_n$, cannot be greater than 1, and the variance of the values of K'' can only reflect the " brutality " of the transition between high and low values of K'. $\mathrm{var}\,(K'')$ thus enables us to make a distinction between profiles (1) and (2) on fig. 2, but is equal to zero for both (2) and (3). To correct that behaviour, the extent of K' (that is, $k'_1 - k'_n$) is combined with $\mathrm{var}\,(K'')$, giving the following expression for the " quality function " q:

$$q\,(\mathcal{K}) = (1 + \mathrm{var}\,(K''))(1 + (k'_1 - k'_n)) - 1$$

The q function reflects quite faithfully 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.

4.3.3. Algorithmic exploitation of the quality measure

The algorithmic use of q aims at furnishing a means to enhance the reading and exploitation of CHAVL's initial k-classification tree in two directions: q is used to 1- automatically point out the relevant keyword classes in the tree whatever their level, 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 towards the root checking, for each node met on that way, if the merging suggested by that node between the classes corresponding to its sons allows a progression of q.

- If that is the case, the algorithm executes the merging and keeps going up the tree with the new class resulting from it.
- Else, instead of simply considering that, the node we have just met being a " bad " one, all nodes above it will be as well (which would lead us to stop the exploration of the studied branch at that node), the algorithm keeps going up the tree, but *without* performing the suggested merging: after (b) in fig. 1, we no longer have a k-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 k-classes to find the most interesting ones in terms of the evolution of q:

- In (c): {{*cinema, movie, scene*}, {*town*}} " + " {*author*}
  → {{*cinema, movie, scene, author*}, {*town*}}
- In (d): {{*cinema, movie, scene, author, work*}, {*town*}} + {*lodgement, city*}
  → {{*cinema, movie, scene, author, work*}, {*town, lodgement, city*}}

That way, 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, lodgement, city*}, {*capitalism, life*}, {*employment, job, unemployment*}}.

We cannot give here a more detailed description of the functionning of the algorithm, but complete information about it and the heuristics finally used to try and filter out useless classes (such as {*capitalism, life*} in the example) can be found in (Rossignol, 2001).

5. Conclusions: results and future work

The method described in this article has been implemented as a single program performing all the necessary operations to obtain k-classes from a tagged corpus, from the random selection of the excerpt to work on to the production of the final keyword classes. The results of several executions of the program show that a significant progress has been made since the experiments of (Pichon and Sébillot, 2000), although some weaknesses remain. A complete and thorough evaluation of the system remains to be done, and we only present here an overview of the quality of our results from two points of view: that of the human interpretation of the consistency of the generated keyword sets and that of their efficiency for theme recognition.

5.1. Semantic consistency of keyword classes

The most positive result of our work is the quality of the obtained k-classes from the point of view of an intuitive evaluation of their thematic consistency: on average, out of 35 classes proposed by the program, about 25 can be considered as intuitively satisfying — for example, {*office, centre, teaching institution, research, university, school*} or {*channel, image information, means, media, programme, network, television, event*}. The others are very seldom devoid of any semantic consistency, but most often reflect vague concepts on which their " grasp " is too loose for the class to be considered as reliable for theme recognition (e.g. {*community, dimension, spirit measure, movement*}). This proportion of about two thirds of " good " classes is a significant improvement over the results obtained in (Pichon and Sébillot, 2000), where 45 classes were proposed and 21 retained. Moreover, many of the classes we get operate a more complete covering of the subject than those obtained in the pre-existent work.

Another interesting aspect of the results is that the themes they suggest cover most of the recurrent topics found in *Le Monde Diplomatique*: geopolitics, transnational business, social conflicts, elections, education, social welfare, press, television… It clearly shows that, despite its simplicity, the global statistical approach we have chosen does permit to collect substantial semantic information about a corpus.

5.2. Efficiency of the process for theme detection

The main weaknesses of our system appear when trying to exploit the sets of keywords we obtain: their direct use with the simple keyword co-occurrence criterion we defined for theme recognition only allows to detect the theme of about one third of the paragraphs of the corpus.

That can be partly due to the fact that no keyword class is generated for many minor themes, but, to our knowledge, the detected themes should concern at least 80 % of the corpus. So the most likely reason for that insufficient covering of the corpus is that the sets of keywords we obtain hardly manage to recognize half of the paragraphs dealing with the theme they represent.

5.3. Future works

Several solutions are being developed to overcome the most annoying problem we are facing, that of the insufficient theme recognition rate. A first one consists in taking into account not only nouns, but also verbs and adjectives, but that raises more difficulties than one could think of since those words are more likely than nouns to have relatively " empty " meanings — good, bad, make, act, *etc.* do not provide any kind of thematic information. Another possible solution would be to perform simple multi-paragraph discourse segmentation, by detecting cue words showing the presence of a transition between themes or exploiting general knowledge about the typical structure of a newspaper article.

Experiments are also being made to increase the number of satisfying classes obtained (and hence the number of recognized themes) and, in parallel, reduce the amount of manual filtering to be done. Several executions of the program made using various random excerpts showed that the " common denominator " of the obtained results was a set of highly consistent but overly restrained classes. This observation led to one of the threads of research we are currently pursueing, which attempts to combine the results obtained on different excerpts of the corpus.

Acknowledgements

We would like to thank I.-C. Lerman for his precious help when dealing with statistics, and for enhancing the `chavleps` program in order to better suit our particular needs.

References

Ferret O. and Grau B. (2001). Utiliser des corpus pour amorcer une analyse thématique. *TAL (traitement automatique des langues), special volume Natural Language Processing and Corpus Linguistics, to appear*, 42(2).

Hearst M. A. (1994). Multi-Paragraph Segmentation of Expository Texts. In *ACL'94 (32th Annual Meeting of the Association for Computational Linguistics)*, Las Cruces, NM, USA.

Lerman I.-C. (1991). Foundations in the Likelihood Linkage Analysis Classification Method. *Applied Stochastic Models and Data Analysis*, 7:69–76.

Litman D. J. and Passonneau R. J. (1995). Combining Multiple Knowledge Sources for Discourse Segmentation. In *ACL'95 (33th annual meeting of the Association for Computational Linguistics)*, Montreal, Quebec, Canada.

Peter P., Leredde H., and Lerman I.-C. (1992). *Notice du programme CHAVL*.

Pichon R. and Sébillot P. (1999). Différencier les sens des mots à l'aide du thème et du contexte de leurs occurrences : une expérience. In *TALN 1999 (Traitement automatique des langues naturelles)*, Cargèse, Corsica, France.

Pichon R. and Sébillot P. (2000). From Corpus to Lexicon: from Contexts to Semantic Features. In Lewandowska-Tomaszczyk B. and Melia P. J. editors, *PALC'99 (Practical Applications in Language Corpora), Lodz studies in language*, volume 1. Peter Lang.

Rastier F. (1996). *Sémantique Interprétative*. Presses universitaires de France, second edition.

Rossignol M. (2001). Acquisition sur corpus d'informations lexicales basées sur la sémantique différentielle. DEA report, IRISA / IFSIC, Rennes 1 University.