Running head: TOPIC EXTRACTION FOR ONTOLOGY LEARNING

Topic Extraction for Ontology Learning

Marian-Andrei RIZOIU and Julien VELCIN

Laboratory ERIC, University Lumière Lyon 2

Laboratoire ERIC

Université Lumière Lyon 2

5 av. P. Mendès-France

69676 Bron Cedex

France e-mail:{Marian-Andrei.Rizoiu; Julien.Velcin}@univ-lyon2.fr

Abstract

This chapter addresses the issue of topic extraction from text corpora for ontology learning. The first part provides an overview of some of the most significant solutions present today in the literature. These solutions deal mainly with the inferior layers of the Ontology Learning Layer Cake. They are related to the challenges of the Terms and Synonyms layers. The second part shows how the same pieces can be bound together into an integrated system for extracting meaningful topics. Whereas the extracted topics are not full concepts yet, they constitute a convincing approach in concept building and therefore in ontology learning. The chapter concludes by discussing the research done for filling the gap between topics and concepts as well as perspectives that emerge today in the topic learning area.

Topic Extraction for Ontology Learning

Introduction

The last years have seen an intensive research on automatic ontology construction, especially from *natural language texts*. Special attention has been given to texts found on the Web, as they have specific features that we will present later in this chapter. Ontologies can be seen as collections of concepts linked together through relations. Therefore ontology learning is closely connected to concept learning. Buitelaar, Cimiano, et Magnini (2005) divide the process of ontology learning in a chain of different phases, the output of each phase being the input of the following one, as described in chapter (place here your reference to the chapter presenting the Ontology Learning Layer Cake) of this book. An analysis of the state-of-the-art in terms of ontology learning at each of the various phases can be found in Cimiano, Völker, et Studer (2006).

In order to place topic extraction in the context of Ontology Learning process, we propose to take the reader into a descending overview of the inferior layers of the Ontology Learning Layer Cake (Buitelaar et al. (2005)), highlighting the challenges at each step. Beginning from the observation that ontologies are dynamic, and that they keep evolving mainly by means of refining concepts or replacing old concepts with new ones, a special attention must be paid to the "concept" layer. Therefore, automated ontology learning is closely connected to concept learning. As shown in Cimiano et al. (2006), the main approach toward learning concepts and their taxonomy (the hierarchical relations between concepts) is **Conceptual clustering** (Michalsky et Stepp (1983)), an unsupervised machine learning technique closely connected to unsupervised hierarchical clustering. This approach generally outputs a concept tree, each level being more specific than the previous one. At each level, the collection of terms is partitioned around each

concept, using clustering algorithms, thus obtaining partitions of different granularity levels: bigger under the root and smaller as we reach the leaves. Examples of algorithms developed for this purpose are the well-known COBWEB (Fisher (1987)) and the more recent WebDCC (Godoy et Amandi (2006)). While this approach is promising and has shown good results, the resulted hierarchy is still very noisy and dependent on both the quality of extracted terms and their frequency in the text collection. Therefore, researchers have tried to improve the quality by allowing the expert to validate and guide the process. Others touched the field of semi-supervised learning techniques by making the algorithm aware of external information,

Taking into consideration these preliminary observations about the dependency of the superior layers of the cake on the quality of terms, we descend another step into the ontology layer cake. At the *terms* and *synonyms* layers, new challenges arise, such as extracting pertinent, non-ambiguous terms and dealing with disambiguation. Term extraction literature proposes solutions, out of which we mention some recent ones like Wong, Liu, et Bennamoun (2009) and Wong, Liu, et Bennamoun (2008). The purpose of the lower layers of the cake is to extract terms and regroup synonyms under the same concept and finally defining the concepts, both in intention and in extension.

There are other approaches that pass though topics on the way towards concepts. Just like the later (see concept definition in Buitelaar et al. (2005)), topic definition is controversial. While some researchers consider a topic being just a cluster of documents that share a thematic, others consider topics as an abstraction of the regrouped texts that needs a linguistic materialisation: a word, a phrase or a sentence that summarises the idea emerging from the texts. Table 1 presents an example of the topics that can be extracted from text. More details about some experimentation made with this system will be presented later, in section "Combining the two phases into an integrated system for extracting topics". A topic is not a concept since it is an abstraction of the idea behind

a group of texts rather than a notion in itself. While the difference between the two is subtle, evolving a topic into a fully fledged concept is still to be achieved and the reader will find a couple of ideas in the section "Conclusions and Perspectives".

In this chapter we propose to present a method of topic extraction from natural language texts, focusing on **flat clustering** techniques obtaining a partition of the documents at a single level. Basically, by means of Unsupervised Machine Learning, these algorithms divide the input set of texts into groups that are similar in terms of their thematics (politics, economics, informatics etc), meaning that all the texts in a group approach the same domain and there is a visible distinction between them and the texts from the other groups. We chose to present these approaches not only from the point of view of topic extraction, but also regarding their usage at the different layers in the Ontology Learning Cake.

Most of these clustering algorithms present at the output a central point for each of the created groups. This central point is often called a centroid and summarizes the common part of all the documents in the cluster. The centroid can be viewed as an abstract representation of the topic denoted by that group, a prototype. Even if highly rated features in this vector are correlated in the topic, the vector or the distribution itself rarely makes any sense to a human.

That is why it is more convenient to choose a name for it. There are multiple ways of naming a topic: choosing an arbitrary number of high rated words, selecting a document as the representative, assigning it a meaningful, human-readable expression(phrase) etc. In order to facilitate the passage between topics and concepts, we believe that the assigning phrases to the topics could prove to be the most useful, as they would serve later to construct the concepts intention. What makes a good topic name? Roche (2004) presents the problem in detail. One of the first things that must be taken into consideration is that words have the property of **polysemy**, meaning that the same

word can have different meanings in different contexts. For example, each of the words "data" and "mining" have different meanings than the phrase "data mining". Seen from the light of these observations, we would like to allow groups of documents to overlap, authorising documents to be part of more than one group. In this way, a text that talks about the "economical outcomes of a political decision" can be part of both the "politics" groups, as well as the "economics" group.

This second phase gives the topic a linguistic materialisation. It allows to go from an abstract centroid, a prototype that summarises the common part of all documents in its group, to a human comprehensible topic.

State of the Art

Textual Clustering

Given the property of polysemy of terms, an important aspect of the synonyms layer is the identification of the appropriate sense of terms, which determines the set of synonyms that have to be extracted. Buitelaar et al. (2005) present the two main approaches towards finding synonyms :

- algorithms that rely on readily available synonym sets such as *WordNet* or *EuroWordNet* (Turcato et al. (2000); Kietz, Maedche, et Volz (2000));
- algorithms that directly discover synonyms by means of clustering.

The same authors state that "the second group of algorithms, which are based on statistical measures used mainly in Information Retrieval, start from the hypothesis that terms are similar in meaning to the extent in which they share syntactic contexts (Harris (1968))". Therefore, performing textual regrouping on the entire collection of texts, would place texts that share the same content into the same group. This would lead to synonyms to be placed in the same group.

In the following sections, we have divided the textual clustering algorithms into categories based on their ability to create **overlapping groups**. If terms can have different meaning depending on the context (polysemy) it is only natural to allow them to be part of more than one group. In this way, the clustering algorithm would not only find synonyms, but its output could also be used for disambiguation. It is worth mentioning that most of today's word sense disambiguation algorithms, like the one in Lesk (1986), rely on usage of synonym sets.

While some of the solutions presented below were created specifically for text mining (like LDA), others were designed for a general purpose clustering. They partition individuals into groups based on the similarity of their features. All of these methods can be used for textual clustering by representing the documents according to the Vector Space Model (as described in subsection "Vector Space Model")

Crisp solutions. Crisp clustering algorithms regroup the objects in a collection of disjointed classes forming a partition of the dataset (named "crisp" clustering). We present these two principally for didactical reasons. **KMeans** (Macqueen (1967)) is one of the most well-known clustering algorithms. Extensive work has been done and numerous papers proved its accuracy for various tasks. It To do this, the algorithm iteratively optimizes an objective criterion, typically the squared-error function. In the case of text mining and information retrieval, the cosinus distance can be used in order to calculate similarities between texts. **Bisecting KMeans** (Steinbach, Karypis, et Kumar (2000)) is a hierarchical variant of KMeans which has been proved to be more accurate than KM for the task of text clustering. It is based on a top-down algorithm that divides, at each step, the documents into two crisp sub-clusters. For instance, at the first level, the whole corpus is divided into two subsets according to multiple restarting 2-means. For the next level, one of the subsets is chosen (for example, the bigger one) and split: globally, we obtain three text clusters. The process is iterated until a stopping criterion is satisfied, e.g. a

fixed number K of clusters. The final output of BKM can be seen as a truncated dendrogram. **Hierarchical agglomerative clustering (HAC)** in another hierarchical clustering technique used more frequently in Information Retrival. It construct the hierarchy bottom-up and consists in merging at each step a pair of clusters.

Of course, there are many other systems offering clustering solutions, some of them even having topic extraction capabilities, like AGAPE (Velcin et Ganascia (2007)). Their main inconvenience is that they output a crisp partition, where each document can be part of only one group. While, from the point of view of the Ontology Learning Cake, they can be used for regrouping synonyms, they cannot be used for disambiguation. From the topic extraction point of view, they do not allow overlapping for the clusters, forcing a document to be associated with only one topic.

Fuzzy solutions. In fuzzy clustering, each document has a degree or a probability of belonging to all clusters, rather than belonging completely to just one or several clusters. Thus, a document at the edge of a cluster, is associated with it in a lower degree than a document in the center of the cluster. For each document d, we have a coefficient giving the degree (similar to the probability) of being in the kth cluster $u_k(d)$. Still, fuzzy logic clustering algorithms can be adapted to output an overlapping partition by choosing a threshold θ and considering that if $u_k(x) > \theta$ then the document d is in the kth cluster.

Fuzzy KMeans (Dunn (1973)) is an adaptation of the KMeans algorithm to the fuzzy logic. The main differences between the Fuzzy KMeans and the original version are :

- the way the objective function is calculated - every document contributes to the update of the centroid according to the weight associated to that cluster;
- the output of the algorithm - a vector with the probabilities of membership to clusters.

Latent Semantic Indexing (Berry et al. (1995)) is a statistical topic discovery algorithm using Singular Value Decomposition (SVD) as the underlying mathematical

ground. In LINGO (Osinski (2003)), LSI is used for the clustering purpose in conjunction with the Suffix Array (Manber et Myers (1990)) frequent phrase extraction algorithm, which will be detailed in section "Combining the two phases into an integrated system for extracting topics". The main idea of the algorithm is to decompose the term/document matrix in a product of three matrices: $A = USV^T$. U and V are orthogonal matrices, containing the left and right singular vector of A, and S a diagonal matrix, with the singular values of A ordered decreasingly. If we keep only the k highest ranking singular values and eliminate the rest, along with the corresponding columns in U and lines in V, the product $A_k = USV^T$ is also known as the k-approximation of A.

It is well-known that most clustering algorithms take the number of clusters as a parameter, which is arbitrarily set by an expert. The LSI approach allows an automatic approximation of the number of clusters, based on the value of singular values of the original matrix. This is known in literature as a dimension reducing technique. Hence, in LINGO, the Frobenius norm of the A and A_k matrices is used to calculate the percentage distance between the original term / document matrix and its approximation.

The columns in U corresponding to the k highest singular values create an orthogonal basis for the document space. According to the mathematical vectorial space theory, every component of the space, in our case every document, can be expressed as a weighted sum of the elements of the base.

$$d_i = \alpha_1 e_1 + \alpha_2 e_2 + \ ... \ + \alpha_k e_k \tag{1}$$

The elements $e_l, l \in \{1..k\}$ of the base can be considered as the centers of the classes and the formula above is highly similar to the fuzzy approach described earlier, the document d_i having the probability α_j of belonging to the j^{th} cluster.

Latent Dirichlet Allocation (LDA) (D. M. Blei, Ng, Jordan, et Lafferty (2003)) is a probabilistic generative model designed to extract topics from text corpora. It

considers documents as collections of words and models each word in a document as a sample from a mixture model: each component of the mixture can be seen as a "topic". Thus each word is generated from a single topic, but different words in a document are generally generated from different topics. Each document is represented as a list of mixing proportions of these mixture components and thereby reduced to a probability distribution on a fixed set of topics.

LDA is highly related to **probabilistic Latent Semantic Analysis** (pLSA), except that in LDA the topic distribution is assumed to have a *Dirichlet* prior. This point is highly important because it permits to go beyond the main limitations of pLSA: overfitting and the impossibility of making true inferences on new documents (see (D. M. Blei et al., 2003) for details). More precisely, LDA is based on the hierarchical generative process illustrated in Fig. 1. The hyperparameters α and β are the basis of two Dirichlet distributions. The first Dirichlet distribution deals with the generation of the topic mixture for each of the $|D|$ documents. The second Dirichlet distribution regards the generation of the word mixture for each of the K topics. Each topic is then a distribution over the W word of the vocabulary. The generative process is the following: for each word $w_{d,i}$ of the corpus, draw a topic z depending on the mixture θ associated to the document d and then draw a word from the topic z.

Note that words without special relevance, like articles and prepositions, will have roughly even probability between classes (or they can be placed into a separate category). As each document is a mixture of different topics, in the clustering process, the same document can be placed into more that one group, though resulting in a (kind of) **Overlapping Clustering Process**. Learning the parameters θ and z, and sometimes the hyper-parameters α and β, is rather difficult because the posterior $p(\theta, z/D, \alpha, \beta, K)$ cannot be fully calculated, because of an infinite sum in the denominator. Therefore various approximation algorithms must be used, such as variational EM, Monte-Carlo

Markov processes, etc.

This probabilistic approach presents advantages and disadvantages:

• The theoretical framework is well-known in bayesian statistics and well-grounded. It has led to many fruitful researches (see below).

• It is designed to make inferences on new documents: what are the associated topics and with which proportions? What part of the document is associated to which topic? Depending on the likelihood $p(d/\Theta)$, does a new document describe an original mixture of topics or a new, never seen before topic?

• LDA is a complex mathematical model, which considers each document as a combination of possibly many topics. While this may be interesting for describing the documents, in the case of clustering, it could lead to a situation where each document belongs, more or less, to many clusters (similar to a fuzzy approach). An issue is therefore to be able to choose a finite (and hopefully short) list of topics to be associated to the document, beyond setting a simple threshold parameter.

• This method does not present a center for each cluster, but a distribution of the document over the topics. This could make it difficult to associate a readable name to the cluster. Note that recent works relative to LDA are seeking to find useful names using n-grams (X. Wang, McCallum, et Wei (2007)).

• As in the other presented methods, this probabilistic approach does not solve the classical problem of finding the global optimum and choosing the number K of topics. For the latter, some methods are proposed inspired by the works in model selection (Rodríguez (2005)).

Numerous works have followed the way designed by Blei et al. to deal with various related issues: extracting topic trees (hLDA) (D. Blei, Griffiths, Jordan, et Tenenbaum (2004)), inducing a correlation structure between topics (Lafferty (2006)), modeling topics through time (D. Blei et Lafferty (2006)), finding n-grams instead of single words to describe

topics (X. Wang et al. (2007)), etc.

Overlapping solutions. **Overlapping K-Means (OKM)** (Cleuziou (2007)) is a recent extension of the well-known **K-Means**. It shares the general outline of the algorithm, trying to minimize an objective function. It does so by initially choosing randomly k centroids (centers) from the data set, and then iterating in two steps:

1. Assigning the documents to the clusters;
2. Recalculating the clusters, based on the new configuration;

until the objective value reaches a local minimum.

The main difference of the OKM algorithm compared to the K-Means is that a document can be assigned to multiple clusters. If in K-Means each document was assigned to the centroid that was closest to it, in terms of cosine distance (detailed in subsection "Vector Space Model"), OKM calculates an image of the centroids, adding each document to clusters so that the distance between the centroid and its image is minimal. This image is the *Gravity Center* of the assigned centroids.

Therefore, the function that OKM tries to minimize is the distortion in the dataset:

$$distorsion\,(\Pi) = \frac{1}{NK}\sum_{i=1}^{N} d\left(X^{(i)}, Z^{(i)}\right)^2 \tag{2}$$

where $Z^{(i)}$ represents the image of document $X^{(i)}$, N the number of documents and K the number of desired clusters.

OKM inherits from K-Means most of its drawbacks (its powerful dependence on the initialization and the number of clusters that must be arbitrarily specified by the expert) and its advantages (linear execution time, good performance when working with texts). Nevertheless, it outputs directly an *overlapping partition* of the data set, without the need of setting a threshold parameter necessary for fuzzy approaches as those presented before. This is the main reason why it was chosen for the clustering task in the topic extraction algorithm that will be presented in detail in section "Combining the two phases into an

integrated system for extracting topics". This threshold, necessary for transforming fuzzy into overlapping, is highly dependent on the chosen data set.

As stated in the beginning of this section, using an overlapping solution solves 2 problems at the same time:

- synonym terms are grouped together in the same cluster ;
- it addresses the disambiguation problem, allowing terms to be in more than one cluster. This way, terms that have different meanings depending on the context can be regrouped together with their synonyms for each meaning.

Cleuziou (2009) presents wOKM, a weighted version of OKM, that uses weights internally and achieves even better performance in terms of **precision**, **recall** and **FScore**. At the same time, it limits the overlapping in the clusters issued by OKM, which in certain cases can be significant. WOKM is a kind of subspace clustering approach, a review of which can be found in Parsons, Haque, et Liu (2004).

Keyphrase Extraction

The first level of the **Ontology Layer Cake** is the *Terms Layer*. This layer is a prerequisite for all aspects of ontology learning from text (Buitelaar et al. (2005)). Its purpose is to extract relevant terms that unambiguously refer to a domain-specific concept (Cimiano et al. (2006)). Buitelaar et al. (2005) observe that although "the literature provides many examples of term extraction methods that could be used as a first step in ontology learning from text", still "much of the research on this layer in ontology learning has remained rather restricted". Cimiano et al. (2006) also considers that automatic term extraction techniques have not yet reached their maturity, since the "resulting list of relevant terms will for sure need to be filtered by a domain expert."

Topic extraction, on the other hand, shares the same need for relevant, unambiguous terms or phrases to synthesize the thematic of the group of documents

associated to the topic. The algorithm presented in the section "Combining the two phases into an integrated system for extracting topics" has 3 phases for extracting topics: overlapping document clustering, term/keyphrase extraction and cluster-name association. Such a topic name is a complete phrase, that contains all the words that have a special meaning together (like "data mining") and all the prepositions and articles that make sense to the human reader ("of" in "Ministry of Internal Affairs"). A **keyphrase** is "a sequence of one or more words that is considered highly relevant as a whole", while a **keyword** is "a single word that is highly relevant" Hammouda, Matute, et Kamel (2005).

The literature presents several ways of classifying the term extraction algorithms. Hammouda et al. (2005) divides the approaches into two categories, based on the learning paradigm they employ:

- The approaches that **construct** the keyphrases, which is usually a *supervised learning* task, often regarded as a more intelligent way of summarizing the text. These approaches make use of the knowledge of the field expert, demanding him to validate, at each step for the incremental algorithms, the extracted phrases. While in this way the results are less noisy - only interesting collocations will be extracted -, involving a human supervisor can make the whole process slow, expensive and biased towards the specific field (eg. microbiology). These approaches face problems when demanded to process large datasets of general purpose texts. Examples: *ESATEC* Biskri, Meunier, et Joyal (2004), *EXIT* Roche (2004), *XTRACT* Smadja (1991)

- The approaches that **extract** the keyphrases from a set of documents, which is an *unsupervised learning* technique, trying to discover the topics, rather than learn from examples. Not depending on a human expert makes this kind of approaches scale well to large datasets. Still, their major drawback is the almost exponential quantity of extracted phrases, most of them having no real interest for the specific domain of the application, leading to a noisy output. However, there are techniques to ameliorate their precision,

some of them presented later in this section. Examples: *CorePhrase* Hammouda et al. (2005), *Armil* Geraci, Pellegrini, Maggini, et Sebastiani (2006), *SuffixTree Extraction* Osinski (2003).

Other researchers (Roche (2004); Buitelaar et al. (2005); Cimiano et al. (2006)) divide topic extraction algorithms into 3 categories, according to their employed methods: linguistic, numeric and hybrid.

Linguistic approaches. These algorithms take inspiration from terminology and *Natural Language Processing* research. They employ linguistic processing like phrase analysis or dependency structure analysis.

In Roche (2004), three linguistic systems are presented: TERMINO, LEXTER and INTEX & FASTR. All these systems make use of morphological and syntactic informations about the words in the texts. The POS tagger (Part-Of-Speech) tries to recognize whether the word is a noun, adjective, verb or adverb, and tries to characterize it morphologically (number, person, mode, time etc). Based on this information, the lemmatisation process extract the radix of the word (masculine single - for nouns, infinitive - for verbs). With the texts tagged, each system has its own approach toward discovering the keyprases. In TERMINO, a lexical-syntactic analyser is used to describe the sentences, and then certain patterns are used to uncover the keyphrases (ex: <Head> <Prepositional Group> <Adjectival Group>). LEXTER uses the morphological information to extract from the text nominal groups and then searches for dis-ambiguous maximal nominal groups.

Keyphrase extraction methods based on linguistic approaches do succeed in obtaining less noisy output, but they are also vulnerable to multilingual corpora and neologisms. They also have the tendency to adapt to stereotypical texts (texts from a specified narrow field) (Biskri et al. (2004)). In other words, they do not adapt or scale easily to new fields or datasets. This makes them particularly difficult to work with when

dealing with term extraction from texts found on the internet. Documents on the internet do not necessarily have a scientific writing style, nor do they always respect the official spelling.

Also, the use of linguistic methods leads to an almost exponential explosion of the numbers of collocations extracted when the size of the corpus increases. That is why usage of methods based only on linguistic information could prove prohibitive. Nevertheless, this could be dealt with to a certain extent by use of statistical filters (see subsection "Hybrid approaches")

Numerical approaches. These algorithms are based on information retrieval methods for term indexing (Salton et Buckley (1988)) and make use of numerical (statistical) information in order to discover the topics. For each couple of words in the text, the statistical measure is calculated. This allows to quantify the dependency between the two words in the binary collocation, also called bigram. A well-known and used such measure is the *Mutual Information*, which is given by the formula:

$$IM(x, y) = \frac{P(x, y)}{P(x)P(y)} \tag{3}$$

where $P(x)$ and $P(y)$ are the probabilities that the word x and, respectively, y appear in the text, while $P(x, y)$ represents the probability of the words x and y appearing together as neighbours. This allows us to calculate the correlation between two words that are one next to the other or in a window of specified dimensions. In Anaya-Sánchez, Pons-Porrata, et Berlanga-Llavori (2008), a window of dimension 11 is considered around a word (5 words before + word + 5 words after). Once we have the tool for extracting bigrams from the text, some authors (**EXIT** Roche (2004), **ESATEC** Biskri et al. (2004)) propose ways of constructing ngrams, by combining iteratively the bigrams or adding to an existing (n-1)gram another word, trying to obtain longer collocations that have a higher *Mutual Information* score.

Of course, many statistical measures have been proposed to calculate the strength of the relationship between two words. In Anaya-Sánchez et al. (2008) the algorithm first identifies a set of terms that are frequent (over a minimum threshold). Then, a set of pairs of these terms is created, retaining only the ones that score a minimum frequency. The β-similarity is calculated just for these pairs, and the set of documents for which the pair is representative is constructed. In Silva, Dias, Guilloré, et Pereira (1999), Dias, Guilloré, et Lopes (2000) , the authors consider that a special **"glue"** exists between words that make them have a sense when they are placed together. LocalMaxs is used in conjuncture with the **Symmetric Conditional Probability** (SCP) measure to extract **Continuous Multiple Word Units** and with the **Mutual Expectation** (ME) measure for extracting **Non-Continuous Multiple Word Units**. Thanopoulos, Fakotakis, et Kokkinakis (2002) start from the idea that all n-grams can be constructed from bigrams and it is important to know what measure to use. They study the impact of some of the most known and used measures on the algorithm's performance, judging their ability to identify lexically associated bigrams. The measures compared are: **t-score**, **Pearson's χ-square test**, **log-likelihood ratio**, **pointwise mutual information** and **mutual dependency**.

There are other approaches that do not rely on bigram detection and ngram construction. In **CorePhrase** Hammouda et al. (2005) keyphrases are considered to naturally lie at the intersection of the document cluster. The CorePhrase algorithm compares every pair of documents to extract matching phrases. It employs a document phrase indexing graph structure, known as the **Document Index Graph** (DIG). It keeps a cumulative graph representing currently processed documents. Upon introducing a new document, its subgraph is matched with the existing cumulative graph to extract the matching phrases between the new document and all previous documents. The graph maintains a complete phrase structure identifying the containing document and phrase

location, so cycles can be uniquely identified. Simultaneously, it calculates some predefined phrase features that are used for later ranking.

In **LINGO** Osinski (2003), a **Suffix Array** (Manber et Myers (1990)) based keyphrase discovery is used. The algorithm tries to avoid extracting incomplete phrases (like "President Nicolas" instead of "President Nicolas Sarkozy") which are often meaningless, it uses the notion of *phrase completeness*. A phrase is complete if and only if all of its components appear together in all occurrences of the phrase. For example, if the phrase "President Nicolas" is followed in all its occurrences by the term "Sarkozy", then it is not a complete phrase. Starting from this definition, right and left completeness can be defined (the example above is left complete, but not right complete). Using a Suffix Array data structure, the complete phrases can be detected and the ones that occur a minimum number of times (frequent keyphrases) populate the candidate set for the topics. A more detailed explanation of this approach is presented in section "Combining the two phases into an integrated system for extracting topics".

Hybrid approaches. An hybrid system is usually adding linguistic information to an essentially numerical system or adding numeric (statistical) information to an essentially linguistic system. This process usually ameliorates the results.

It is well-known that statistical systems (like those based on Bayesian networks) produce noisy results in the field of Information Retrieval (Biskri et al. (2004)), meaning that among the extracted candidates, most of them pass the frequency threshold and get good scores, but they are uninteresting from the topics point of view. Such expressions can be comprised of common words (articles, prepositions, certain verbs, etc) like "he responded that" or "the biggest part of the", and they bring no new information. Such phrases should be eliminated. For that, linguistic filters are very useful.

Some of the linguistic methods rely on certain keyphrase formats (like <Subject> <Verb> or <Verb> <Adverb>) to construct the result. A morphological and syntactic

tagger could be used as a final phase to filter out the noise from the candidates set resulted from statistical extraction. From such a filter benefits the system **XTRACT** (Smadja (1991)) which is comprised of three phases. In the first phase, bigrams are extracted from a grammatically tagged corpus, using an eleven words window. The next phase consists in extracting longer phrases if they are frequent in the text. These phrases are called *rigid noun phrases*. The third phase is the linguistic phase. It consists in associating a syntactic etiquette to the extracted bigrams (<Noun> <Verb>, <Adjective> <Noun>) and afterwards, for each bigram, it associates together longer phrases containing the n-grams obtained at the second phase.

Combining the two phases into an integrated system for extracting topics

The literature provides many examples of systems that can extract topics from texts. Mei, Shen, et Zhai (2007), for example, see the labeling problem "as an optimization problem involving minimizing Kullback-Leibler divergence between word distributions and maximizing mutual information between a label and a topic model". In this section we will present a topic extraction system constructed by using textual clustering and keyphrase extraction, proposed by Rizoiu, Velcin, et Chauchat (2010). We will follow phase by phase the chain of processing that starts with a collection of texts (on-line discussions, forums, chats, newspaper articles etc) and presents at the output on one hand the topics extracted from the collection, under the form of readable expressions, and, on the other hand, the partition of texts around the topics.

Figure 2 presents a streamlined schema of the topic extraction system. In a first phase, each of the documents in the data set are pre-processed (see subsection "Pre-processing") in order to eliminate words that do not bring any information about the thematic of the text, thus do not help in extracting the topics. At the same time different

inflected forms are brought to their stem in order to increase their descriptive value. After this phase of pre-processing, the documents are represented by the Vector Space Model (subsection "Vector Space Model") using one of the term weighting schemes, in order to render them compatible with the clustering algorithm.

Afterwards, the process of clustering starts, using the OKM algorithm (see subsection "Textual Regrouping"). Some of the reasons why Rizoiu et al. (2010) have chosen this algorithm will be discussed in the following subsections. With the documents now regrouped, we return to the original dataset in order to extract the complete frequent keyphrases, using the a Suffix Array based algorithm. The procedure will be detailed in the subsection "Keyphrase Extraction. Name candidates". The extracted phrases will be the candidates for the name of each topic. In the final phase (detailed in subsection "Associating names to clusters"), the best candidates are chosen to represent the topics, by means of reintroducing them into the Vector Space Model as pseudo-documents.

Pre-processing

Pre-processing is an important part of the algorithm, as the quality of extracted topics is dependent on the quality of the input dataset and the pre-processing process. Its purpose it to augment the descriptive power of the words, limit the size of the vocabulary and eliminate certain words that are known to bring no useful information. It is traditionally composed of two elements: *stemming* and *stopwords removal*.

Stemming is the process through which inflection, prefixes and suffixes are removed from each term in the collection. It is extremely useful especially for languages that are heavily inflected (like the verbs in French) and reduces words to their stems. This guarantees that all inflected forms of a term are treated as one single term, which increases their descriptive power. At the same time, bare stems may be more difficult to be understood by the users, but since the stemmed version of the terms are never

presented to the user, it will not hinder their usage. Stemming is dependent on language, but algorithms have been developed for most of the widely used languages. For English, the most used is Porter's stemmer (Porter (1980)), while for European languages one of the solutions can be that proposed by the CLEF Project[1].

Stopwords (articles, prepositions etc) do not present any descriptive value, as they are not connected to any thematic, so they are of no use for the clustering process. Even more, they only make the corpus dictionary bigger, so that computation is slower. Some term weighting schemes (such as Term Frequency) are especially vulnerable to stopwords, so their elimination is compulsory. This is done using stopword lists for each language.

Stemmed words are hard to read and stopwords improve the overall cluster names quality for a human reader. Therefore keyphrase discovery requires the texts to be in their natural form, so a non-treated version of the documents is also kept to be used for that phase. Pre-processing is the only part of the algorithm that is language dependent. Adding support for new languages is as easy as adding new stemming algorithms and stopwords lists.

Vector Space Model

As mentioned at the section "Textual Regrouping", most of the algorithms presented were not designed specifically for texts. That is why they require the text to be transformed into a specific format before it can be used. This problem has been addressed extensively in the *Information Retrieval* field and various models have been proposed : the *Boolean Model* compares True / False query statements with the word set that describes a document, t*Probabilistic Model* calculates the relevance probabilities for the documents in the set. The model that is most widely used in modern clustering algorithms is the *Vector Space Model* (Salton, Wong, et Yang (1975)).

In this model, each document is represented as a multidimensional vector. Each

dimension is a keyword or a term and its value associated to a document is directly proportional on the degree of relationship between them. There are four major ways of measuring this relationship degree, also known as **term weighting schemes**.

1. **Presence / Absence.** It is also known as **binary weighting** and it is the simplest way to measure the belonging of a word to a document. Its formula is:

$$a_{i,j} = \begin{cases} 1 & if\, term\, i\, is\, found\, in\, document\, j; \\ 0 & otherwise. \end{cases} \tag{4}$$

In Osinski (2003), it is shown that this weighting scheme can only show if a word is related to a document, but it does not measure the *strength* of the relationship.

2. **Term Frequency** It is also known as **term count**. It is the number of times a given term appears in a document. While this is a better measure of the relationship between the term (word) and the document, this scheme has the tendency of favouring longer documents. In order to prevent that, normalization is usually used.

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{5}$$

where $n_{i,j}$ is the number of occurrences of the considered term (t_i) in document d_j, and the denominator is the sum of number of occurrences of all terms in document d_j.

3. **Inverse Document Frequency.** It is a measure of the general importance of a term in the whole corpus. It expresses the idea that a word should be less important if it appears in many documents. In this way very common words, as prepositions, articles and certain verbs and adjectives could be filtered out or, at least, given less importance.

$$IDF_i = \log \frac{|D|}{|\{d \,|\, t_i \in d\}|} \tag{6}$$

where $|D|$ is the total number of documents in the collection and $|\{d \,|\, t_i \in d\}|$ is the total number of documents where the term t_i appears. In practice, IDF is never used alone, as it lacks the power to quantify the relationship between a word and a document. It also

favorises the very rare words, which are most of the time just noise. Instead, IDF is used in conjunction with TF to bring dataset information in the TFxIDF weighting scheme.

4. **TFxIDF.** It is the most used scheme in Information Retrieval. It is the product of **Term Frequency** and **Inverse Document Frequency.**

$$TFxIDF_{i,j} = TF_{i,j} * IDF_i \tag{7}$$

This scheme aims at balancing local and global occurrences. A high weight in TFxIDF is reached by a high term frequency (in the given document) and a low frequency of the term in the whole collection of documents. This weighting scheme, hence, tends to filter out common terms.

Once documents are represented by the **Vector Space Model** using one of the schemes presented above, the similarity between two documents is usually calculated using the **cosine distance**.

$$similarity(a,b) = cos(\vec{a},\vec{b}) = \frac{\sum_{i=1}^{t} a_i b_i}{\sqrt{\sum_{i=1}^{t} a_i^2}\sqrt{\sum_{i=1}^{t} b_i^2}} \tag{8}$$

which can be interpreted as the geometrical angle between the vectors in the multidimensional space.

Clustering

Having the documents pre-treated and represented by the **Vector Space Model** using one the the four measures presented in subsection "Vector Space Model", the dataset is ready to be partitioned. At the beginning of this chapter, we have insisted on the importance of the polysemy of words and the need of term disambiguation for Ontology Learning. We have shown that one solution for addressing this problem would be the usage of an overlapping clustering solution that would allow documents to be part of more than one group. Therefore, the topic extraction system presented in Rizoiu et al.

(2010) clusters the text documents using the **OKM algorithm** (presented in subsection "Textual Regrouping, Overlapping Solution").

The **OKM** implementation used by the authors Rizoiu et al. (2010) respects the original indications of Cleuziou (2007). The main difference is the stopping condition. In the original form, the iteration process comes to an end when the partition composition does not change any more - which means that a local minimum has been reached. While from the clustering's point of view the final result has been found, it does not necessarily mean that centroids do not evolve over the next iterations.

In K-Means, the centroid is computed depending only on cluster's composition. This means that if the clusters do not change between 2 iterations, neither do the centroids. In OKM the centroid update process is a little more complicated. In the documents - cluster assignment phase, OKM does not try to minimize the variance between a document and its centroid. It rather constructs *an image* of centroids to which the document is associated in such a way that the distance document - image is minimal. Therefore, in the phase of cluster update, the centroid is dependent not only on documents in their own group, but also on the other centroids resulted from the last iteration. The update formula for the centroids becomes:

$$c_{j,v} = \frac{1}{\sum_{x_i \in R_j} \frac{1}{\delta_i^2}} * \sum_{x_i \in R_j} \frac{1}{\delta_i^2} . \hat{x}_{i_v}^j \tag{9}$$

where $\hat{x}_{i_v}^j$ in formula 9 has the following expression $\hat{x}_{i_v}^j = \delta_i x_{i_v} - (\delta_i - 1)\bar{x}_{i,v}^{A \setminus \{c_j\}}$ while:

- A is the set of centroids to which the document x_i is assigned;
- R_j is the collection of documents associated to the centroid c_j;
- $\bar{x}_{i,v}^{A \setminus \{c_j\}}$ is the vth component of the image of the centroids to which x_i is assigned, except centroid j;
- $c_{j,v}$ is the vth component of the centroid to be updated.

This dependency means that centroids continue to change even if the cluster composition does not. In the process of general purpose clustering, the centroid is a by-product and the partition is the main result. But since the process of topic name assignment (presented in the next subsection) is dependent on the centroid quality, it is very important to have the centroids computed as exact as possible. That is why the iteration process should not stop when the clusters stop changing, but rather use a threshold ϵ. In this manner, the clustering process ends only when the variance of the objective function between two iterations drops under the threshold.

Keyphrase Extraction. Name candidates

The next processing phase of the topic extraction system proposed in Rizoiu et al. (2010) employs a keyphrase extraction algorithm in order to build a topic name candidate set. Osinski (2003) presents the conditions that a collocation (or a term) must respect in order to be considered a name candidate:

- it appears in the text with a specified frequency. This is based on the assumption that the keyphrases that occur often in the text have the strongest descriptive power. Also, isolated appearances have high chances of being incorrect words (Roche (2004)).
- it does not cross the sentence boundary. Usually, meaningful keyphrases are contained into a sentence, because sentences represent markers of topical shift.
- it is a complete phrase. Complete phrases make more sense than incomplete ones (e.g. "President Nicolas" vs "President Nicolas Sarkozy").
- it does not begin or end with a stopword. Cluster name candidates will be stripped of leading or trailing stopwords, since this is likely to increase readability. Stopwords in the middle of the keyphrase will be preserved.

Both LINGO (Osinski (2003)) and the system presented in Rizoiu et al. (2010) chose the **Suffix Array based** (Manber et Myers (1990)) approach for the keyphrase

extraction task. They motivated their choice by the approach's ability to extract the phrases from untreated text, its language independence, linear execution time and the power to extract humanly readable phrases. Also, both systems were designed to extract topics from texts found on the Internet, which requires a great flexibility and stability towards different writing styles - which can vary from informal discussions to scientific articles - and different languages. These two characteristics make dealing with texts that appear on the web particularly difficult when using non-statistical approaches, like those presented in subsection "Keyphrase Extraction".

Suffix Array based makes use of the property of completeness (defined in subsection "Keyphrase Extraction, Numerical approaches"). The keyphrase discovery algorithm works in two steps: in the first step left and right complete expressions are found. In the second step, the two sets are intersected to obtain the set of complete expressions.

Suffix Array Construction. The algorithm of discovering right complete expressions relies on the usage of Suffix Array. A Suffix Array is an alphabetically ordered array of all suffixes of a string. We note here that in our case, the fundamental unit is not the letter (as in the case of classical strings), but the term / word. For example, having the phrase "we are having a reunion", the Suffix Array for it would be constructed as shown in Table 2. One of the most important problems in the construction of the Suffix Array is the space-time and time-efficient sorting of the suffixes. In Larsson (1998), two approaches are presented: "Manber and Myers" and "Sadakane's algorithm". The paper also makes a comparison of the two, from both the theoretical and practical performance point of view. According to the test results of Larsson (1998), the second approach gives better results in terms of efficiency.

The only thing required for the algorithm is that the terms have a lexicographic order, so that they can be compared. If in the example in table 2, for the sake of clarity, we have used the alphabetical order, in real-case implementation, the criteria used is not

important. The order of term arrival into the collection can also be used. The "Sadakane's sorting algorithm" is a modified bucket sorting, which takes into consideration the unequal dimensions of the suffixes. In Larsson (1998), it is shown that the sorting complexity is $O(n \log n)$, with n the number of suffixes. Considering that a keyphrase can not pass the boundary of a sentence, the implementation in Rizoiu et al. (2010) differs from that proposed in Osinski (2003) in that it constructs the Suffix Array on a sentence based approach, rather than the whole document approach found in the latter. Therefore a suffix identification is given not only by the beginning of the suffix, but also on the index of the sentence.

Complete Phrase Discovery. The general idea behind the right complete keyphrase discovery algorithm is to linearly scan the suffix array in search for frequent prefixes, counting their occurrences meanwhile. Once such a prefix is identified, information about its position and frequency (initially the frequency is 2) is stored along with it.

Once the right complete phrases have been discovered, we also need to discover the left complete phrases. This can be achieved by applying the same algorithm as before to the inverse of the document - meaning that another version of the document is created, having the words in reverse order. While the algorithm finds the right complete phrases in lexicographic order, the left complete set needs another inversion to recover the right order.

Since both sets are in lexicographic order, they can be intersected in linear time. Name candidates are returned along with their frequency. We must note here that the extracted candidates can also be single terms, as sometimes a single word can be enough for explaining the content of the cluster (Osinski (2003)).

The last phase is filtering the candidate set. First, only phrases that appear in the texts with minimum frequency are kept, the rest are eliminated. In Osinski (2003), the value of this threshold is suggested to be between 2 and 5. The relatively low value for it

can be explained by the fact that the most frequent expressions are not necessarily the most expressive, but usually they are meaningless expressions - noise in the output.

Afterwards, candidates are filtered based on one of the conditions that we enumerated at the beginning of this subsection: *not to begin or to end with a stopword*. Using the same methods as in the pre-processing phase, leading and trailing **stopwords** are recursively eliminated from the phrases. As a result some of the candidates disappeared completely (they were composed only from stopwords), while others reduced their form to another one (example: "the president" and "president of" become both "president").

Associating names to clusters

The clustering phase outputs a data partition that regroups documents relatively by their thematic similarity. At the same time, this phase outputs the centres of each class, also called centroids, which can be regarded as abstract representations of the topics. These centres are documents in the Vector Space Model, having a high weight for the terms that are specific to the group, i.e. the words that are characteristic for the specific topic.

On the other hand, the keyphrase extraction phase generates a list of name candidates for the topics. In this last phase, a suitable name is chosen for each centroid in order to label the topics. This 'centroid - name' association is done by taking all the name candidates and reintroducing them into the Vector Space Model document collection as "pseudo-documents". Initially, the same pre-processing as to the original documents is applied, because the keyphrases were extracted from natural language texts so they may contain inflected words and stopwords. Afterwards, they are translated into the Vector Space Model, using the same term weighting scheme as for the original documents of the collection. The last step is to calculate the similarity between each of these

"pseudo-documents" and the centroid of the class. The one that scores the highest is chosen to be the cluster name.

This phase filters out the noise from the keyphrase candidate set. While centroids are the essence of the documents in those classes, choosing the candidates that are closest to them naturally eliminates phrases that are too general. For example: in a document group that talks mainly about politics, the most important terms (measured with a *term weighting scheme*) should naturally be "parliament", "govern", "president", "party", "politics" etc. When calculating the similarity (cosine similarity) between this centroid and the phrase candidates, it is natural that a candidate that contains many of those words would be favoured. The phrase "presidential elections" would clearly score higher than the phrase "as a matter of fact the".

This candidate pruning side-effect resembles the hybrid approaches presented in the subsection "Keyphrase Extraction", without the actual linguistic filter. From such a linguistic-free approach, which is more suitable for text extracted from the web, could surely benefit the Term Extraction layer from the Ontology Learning Layer Cake.

Experiments and Results

In this subsection we will briefly present some experiments and results that can be obtained with this system. The English dataset used in these tests is a sub-partition of the Reuters [2] corpus, composed of 262 documents. The writing style is journal article, containing between 21 and 1000 words. The authors also used in their experiments French forums, to test the performance of their systems with languages other than English and with different writing styles.

Experiments were performed to test both the clustering phase and keyphrase extraction phase. The behaviour of *OKM* in textual clustering is experimented with in Cleuziou (2007), Cleuziou (2009) and Rizoiu et al. (2010). The authors' main approach

towards evaluating the quality of the resulted partition is to use the classical precision, recall and F-Measure indicators on a corpus that has been tagged a priori by human experts. They used a sub-collection of the Reuters corpus that had at least one tag associated with. Using this method of evaluation, the authors concluded that the overlapping approach indeed out-performs the classical crisp algorithms when being used for text clustering.

The evaluation of the topic names extracted with the *Suffix Array* approach is done in Osinski (2003) and Rizoiu et al. (2010). Here, the authors have used an expert based evaluation of cluster names, arguing that there are no widely accepted automatic topic quality measures (see next subsection). Since topic names need to be humanly-readable and they need to synthesize the thematic of a group of texts, evaluating them is like trying to evaluate "human tastes". The experiments showed a rather good acceptance by the users of the extracted topics, especially when using the *Term Frequency* and the *Presence / Absence* term weighting schemes.

Table 1 presents an output example of extracted topics. The algorithm was run on the dataset presented at the beginning of this section. It was demanded to extract four topics. The first column shows the extracted topics: "cocoa buffer stock", "oil and gas company", "tonnes of copper" and "united food and commercial workers". The second column presents, for each topic, the ten words/terms in the texts that have achieved the highest scores. These words are the most important part of the centroid of each class. As they are output directly by the clustering algorithm, they are in their stemmed version. The next two columns present the number of documents covered and three examples of documents that are part of the clusters of each topic. Let's take as an example the most important topic of the dataset, the one that covers the maximum number of texts: "oil and gas company". The first two examples talk explicitly about the economical activities of companies that operate in the business of oil and natural gas (buying oil and natural

gas proprieties in the first case and estimating reserves in the second case). On the other hand, the third document talks about the food-for-oil program between Brazil and Iraq. Despite the fact that the text does not refer to an oil company, as in the first two cases, the document is still placed under this topic. This is because it still touches the thematic of "oil and gas", as it does with the thematic of food. That is why this document is also found under the topic "united food and commercial workers".

We can see in this example how important is the overlapping property of the clustering algorithm. With a crisp approach, this document would have been only under one topic, when in fact it talks about two topics. Still, the extracted topics are too specific. Topics like "food and gas" and "food" would have been more appropriate.

Conclusion and Perspectives

Over the last ten years, several approaches have been proposed in order to regroup textual datasets into homogeneous clusters and, moreover, to label these clusters with topic names. Among these various approaches, some models are able to deal with the overlapping issue. That is an important point because it allows texts to be related to more than one unique topic. Here stands an important dichotomy between a "fuzzy" approach (each text is covered more or less by the topics) and a "crisp" approach (each text is exactly covered by one to several topics). Until now, the litterature does not present a rigorous comparison between different approaches for topic extraction (such as LDA, LSI, BKM, OKM etc.) in terms of assessment of topic names. The main reason is probably that the comparison criterion is difficult to set, which is highly linked to the question of the assessment of topic quality.

In this chapter, we present different approaches in order to extract useful topic names from texts. Even if some works try to avoid such an additional step (X. Wang et al. (2007)), these techniques seem to be an improvement which permits to go beyond a rough

distribution over words. The extracted phrases are often more intelligible than series of single words. They may be a key to fill the gap between topics and concepts (the topic "data mining" is not far from the concept "data mining").

The "State of the Art" section must be read from two points of view. On the one hand, it provides the ingredients for the topic extraction system presented in the second part of the chapter. But on the other hand, all these algorithms can be used at the different layers of the Ontology Learning Layer Cake. The keyphrase extraction algorithms can be used for the term extraction at the Term Layer, while the clustering techniques can be employed at the synonym layer. Here the overlapping issue seems important in the disambiguation task. Allowing terms to be regrouped in more than one cluster means, in fact, letting the different meanings of a term be put together with their synonyms.

The chapter ends with the presentation of a whole integrated system. This system addresses the problem of topic extraction from textual data. The texts we are interested in present some rather challenging particularities, like being multilingual, having very different writing styles and purposes (from informal chats to academic microbiology articles). The main advantage of this system is that it allows overlapping between the clusters of texts, so that a text could be defined by more than one topic, which is an important aspect, especially giving the property of word polysemy.

For the concept learning, this system allows an extraction of terms and phrases involving a statistics-only approach. By means of transforming the name candidates (the extracted terms) into pseudo-documents and injecting them back into the Vector Space Model, the terms can be pruned, actually obtaining a less noisy list of terms. This has a similar effect as adding linguistic filters to statistic methods, but without their language and field dependency.

With the problem of topic extraction partly solved, there still remains the most strategic issue: filling the gap between topics and concepts. For the moment, the

literature does not provide any largely accepted solution. Of course, the simplest way to do it is to have a human expert manually evolving the topics into concepts by adding relations and building the structure of the ontology. But in the long term, the objective is to completely automatize the ontology building process. That is why relations need to be found in a human-independent way. Some of the recent topic extraction algorithms already provide the means. hLDA (D. Blei et al. (2004)) outputs an hierarchy of topics, which can provide, to a certain extent, the hierarchical relation between concepts. Other algorithms, like cLDA (Lafferty (2006)) obtain a correlation structure between topics by using the logistic normal distribution instead of the Dirichlet. Some authors consider that a hierarchy of topics can already be considered an ontology. Yeh et Yang (2008) extract the topics from the text, using LSA, LDA or pLSA. Then they regroup them into super-topics, using a hierarchical agglomerative clustering using the cosine distance. They consider that "because the latent topics contain semantics, so the clustering process is regarded as some kind of semantic clustering". In the end, they obtain an ontology in OWL. Topic to concept passage is also related to other perspectives, such as reconciling the similarity-based dendrograms built by traditional Hierarchical Agglomerative Clustering and the concept hierarchies used in Format Concept Analysis. The recent work of Estruch, Orallo, et Quintana (2008) proposes in this line an original framework to fill the gap between statistics and logic. Part of the solution is to make contributions relative to the assessment of topic quality. Other works are precisely directed towards such issues (Boyd-Graber, Chang, Gerrish, Wang, et Blei (2009)).

Two other important perspectives are related to the question of granularity. The "horizontal" granularity deals with building hierarchies of topics: each level of the hierarchy presents topics which are more general than the topics of the level below. Recently, several works try to address this issue. For instance, D. Blei et al. (2004) ; C. Wang et Blei (2009) build topic hierarchies based on the nested chinese restaurant

process. Such topic hierarchies seem to be more adapted than “flat” topics in the task of concept construction. At the same time, it brings topics closer and closer to concepts, as these hierarchies provide a relation of taxonomy. The “vertical” granularity deals with the evolution of topics through time. Several probabilistic models have recently been proposed (D. Blei et Lafferty (2006) ; C. Wang, Blei, et Heckerman (2008)). It would be of high interest to relate such dynamic models to other works in the field of concept learning, such as those presented by Chen, Wang, et Zhou (2009). This kind of works will certainly help to address the question of automatic ontology evolution.

Key terms

- **topic** - an abstraction of the idea behind a group of texts;
- **topic extraction system** - an algorithm capable of finding the topics in a collection of texts and, eventually, the relations between them;
- **clustering** - a technique that allows regrouping documents based on the similarity of their features
- **overlapping clustering** - a type of clustering that authorises a document to be part of more than one group;
- **keyphrase** - a sequence of one or more words that is considered highly relevant as a whole

Références

Anaya-Sánchez, H., Pons-Porrata, A., & Berlanga-Llavori, R. (2008). A new document clustering algorithm for topic discovering and labeling. In *Ciarp '08: Proceedings of the 13th iberoamerican congress on pattern recognition* (pp. 161–168). Berlin, Heidelberg : Springer-Verlag.

Berry, M. W., Dumais, S., O'Brien, G., Berry, M. W., Dumais, S. T., & Gavin. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, *37*, 573–595.

Biskri, I., Meunier, J. G., & Joyal, S. (2004). L'extraction des termes complexes : une approche modulaire semiautomatique. In *Données textuelles, louvain-la-neuve, belgique), gérard purnelle, cédrick fairon & anne dister (eds). presses universitaires de louvain, volume 1, pp 192201, isbn* (pp. 2–930344).

Blei, D., Griffiths, T., Jordan, M., & Tenenbaum, J. (2004). Hierarchical topic models and the nested Chinese restaurant process. *Advances in neural information processing systems*, *16*, 106.

Blei, D., & Lafferty, J. (2006). Dynamic topic models. In *Proceedings of the 23rd international conference on machine learning* (p. 120).

Blei, D. M., Ng, A. Y., Jordan, M. I., & Lafferty, J. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 2003.

Boyd-Graber, J., Chang, J., Gerrish, S., Wang, C., & Blei, D. (2009). Reading Tea Leaves: How Humans Interpret Topic Models. *Advances in Neural Information Processing Systems (NIPS)*, *31*.

Buitelaar, P., Cimiano, P., & Magnini, B. (2005). Ontology learning from texts: An overview. In P. Buitelaar, P. Cimiano, & B. Magnini (Eds.), *Ontology learning from text: Methods, evaluation and applications* (Vol. 123). IOS Press.

Chen, S., Wang, H., & Zhou, S. (2009). Concept clustering of evolving data. In *Icde* (pp.

pp.1327–1330). IEEE International Conference on Data Engineering.

Cimiano, P., Völker, J., & Studer, R. (2006). Ontologies on demand? -a description of the state-of-the-art, applications, challenges and trends for ontology learning from text information. *Information, Wissenschaft und Praxis*, *57*(6-7), 315-320. Disponible sur `http://www.aifb.uni-karlsruhe.de/Publikationen/showPublikation?publ_id=1282`

Cleuziou, G. (2007). Okm : une extension des k-moyennes pour la recherche de classes recouvrantes. In M. Noirhomme-Fraiture & G. Venturini (Eds.), *Egc* (Vol. RNTI-E-9, p. 691-702). Cépaduès-Éditions. Disponible sur `http://dblp.uni-trier.de/db/conf/f-egc/egc2007.html#Cleuziou07`

Cleuziou, G. (2009). Okmed et wokm : deux variantes de okm pour la classification recouvrante. In J.-G. Ganascia & P. Gancarski (Eds.), *Egc* (Vol. RNTI-E-15, p. 31-42). Cépaduès-Éditions. Disponible sur `http://dblp.uni-trier.de/db/conf/f-egc/egc2009.html#Cleuziou09`

Dias, G., Guilloré, S., & Lopes, J. G. P. (2000, 22-24 March). Extraction automatique d'associations textuelles à partir de corpora non traités. In M. Rajman & J.-C. Chapelier (Eds.), *JADT 2000 - 5èmes journées internationales d'analyse statistique de données textuelles* (Vol. 2, p. 213-220). Lausanne : Ecole Polytechnique Fédérale de Lausanne.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. , *3 3*, 32 – 57. Disponible sur `http://www.informaworld.com/10.1080/01969727308546046`

Estruch, V., Orallo, J., & Quintana, M. (2008). *Bridging the gap between distance and generalisation: Symbolic learning in metric spaces.* Thèse de doctorat non publiée, Universitat Politècnica de València.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Mach.*

Learn., *2*(2), 139–172.

Geraci, F., Pellegrini, M., Maggini, M., & Sebastiani, F. (2006). Cluster generation and cluster labelling for web snippets: A fast and accurate hierarchical solution. In (pp. 25–36). Disponible sur `http://dx.doi.org/10.1007/11880561\_3`

Godoy, D., & Amandi, A. (2006). Modeling user interests by conceptual clustering. *Information System*, *31*(4-5), 247-265.

Hammouda, K. M., Matute, D. N., & Kamel, M. S. (2005). Corephrase: Keyphrase extraction for document clustering. *MLDM*, *2005*, 265–274.

Harris, Z. (1968). *Mathematical structures of language*. Wiley.

Kietz, J., Maedche, A., & Volz, R. (2000). A method for semi-automatic ontology acquisition from a corporate intranet. *EKAW-2000 Workshop Ontologies and Text, Juan-Les-Pins, France, October 2000*.

Lafferty, D. (2006). Correlated topic models. In *Advances in neural information processing systems 18: Proceedings of the 2005 conference* (p. 147).

Larsson, N. J. (1998, jun). Notes on suffix sorting. (LU-CS-TR:98-199, LUNDFD6/(NFCS-3130)/1–43/(1998)).

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Sigdoc '86: Proceedings of the 5th annual international conference on systems documentation* (pp. 24–26). New York, NY, USA : ACM.

Macqueen, J. B. (1967). Some methods of classification and analysis of multivariate observations. In *Proceedings of the fifth berkeley symposium on mathematical statistics and probability* (pp. 281–297).

Manber, U., & Myers, G. (1990). Suffix arrays: A new method for on-line string searches. In *Proceedings of the first annual acm-siam symposium on discrete algorithms* (pp. 319–327).

Mei, Q., Shen, X., & Zhai, C. (2007). Automatic labeling of multinomial topic models. In *Proceedings of the 13th acm sigkdd international conference on knowledge discovery and data mining* (p. 499).

Michalsky, R., & Stepp, R. (1983). Learning from observation: conceptual clustering, in: R. In J. G. C. Michalski & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach, morgan* (p. 331-363). Kauffmann.

Osinski, S. (2003). *An algorithm for clustering of web search results.* Mémoire de Master non publié, Poznań University of Technology, Poland.

Parsons, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, *6*(1), 90–105.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137. Disponible sur `http://portal.acm.org/citation.cfm?id=275705`

Rizoiu, M.-A., Velcin, J., & Chauchat, J.-H. (2010, janvier). Regrouper les données textuelles et nommer les groupes à l'aide des classes recouvrantes. In *10ème conférence extraction et gestion des connaissances (egc 2010), hammamet, tunisie* (Vol. E-19, p. 561-572).

Roche, M. (2004). *Intégration de la construction de la terminologie de domaines spécialisés dans un processus global de fouille de textes.* Thèse de doctorat non publiée, Université de Paris 11. Thèse de Doctorat Université de Paris 11.

Rodríguez, C. (2005). The ABC of Model Selection: AIC, BIC and the New CIC. *Bayesian Inference and Maximum Entropy Methods in Science and Engineering*, *803*, 80–87.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513–523.

Salton, G., Wong, A., & Yang, C. S. (1975, November). A vector space model for automatic indexing. *Commun. ACM*, *18*(11), 613–620. Disponible sur

http://dx.doi.org/10.1145/361219.361220

Silva, J. da, Dias, G., Guilloré, S., & Pereira. (1999). Using localmaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units. *Progress in Artificial Intelligence*, 849. Disponible sur http://dx.doi.org/10.1007/3-540-48159-1_9

Smadja, F. A. (1991). From n-grams to collocations: an evaluation of xtract. In *Proceedings of the 29th annual meeting on association for computational linguistics* (pp. 279–284). Morristown, NJ, USA : Association for Computational Linguistics.

Steinbach, M., Karypis, G., & Kumar, V. (2000). *A comparison of document clustering techniques.*

Thanopoulos, A. N., Fakotakis, N., & Kokkinakis, G. (2002). Comparative evaluation of collocation extraction metrics. In *In proceedings of the 3rd language resources evaluation conference* (pp. 620–625).

Turcato, D., Popowich, F., Toole, J., Fass, D., Nicholson, D., & Tisher, G. (2000). Adapting a synonym database to specific domains. In *Proceedings of the acl-2000 workshop on recent advances in natural language processing and information retrieval* (pp. 1–11). Morristown, NJ, USA : Association for Computational Linguistics.

Velcin, J., & Ganascia, J.-G. (2007). Topic extraction with agape. In *Adma* (p. 377-388).

Wang, C., & Blei, D. (2009). Variational Inference for the Nested Chinese Restaurant Process. In *Nips.*

Wang, C., Blei, D., & Heckerman, D. (2008). Continuous time dynamic topic models. In *The 23rd conference on uncertainty in artificial intelligence.*

Wang, X., McCallum, A., & Wei, X. (2007). Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 7th ieee international conference on data mining* (pp. 697–702).

Wong, W., Liu, W., & Bennamoun, M. (2008). Determination of unithood and termhood

for term recognition. *Handbook of Research on Text and Web Mining Technologies. IGI Global*.

Wong, W., Liu, W., & Bennamoun, M. (2009). A probabilistic framework for automatic term recognition. *Intelligent Data Analysis*, *13*(4), 499–539.

Yeh, J., & Yang, N. (2008). Ontology construction based on latent topic extraction in a digital library. *Digital Libraries: Universal and Ubiquitous Access to Information*, 93–103.

Footnotes

[1] http://www.clef-campaign.org/

[2] http://mlr.cs.umass.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection

Table 1

Example of output of the topic extraction system.

Topic Name	Highest Rated Words	Docs covered	Text Excerpt
cocoa buffer stock	stock, cocoa, buffer, deleg, icco, consum, produc, rule, meet, council	66	The International Cocoa Organization (ICCO) Council reached agreement on rules to govern its buffer stock, the device it uses to keep cocoa off the market to stabilise prices, ICCO delegates said. The date on which the new rules will take effect has not been decided but delegates said they expected them to come into force early next week, after which the buffer stock manager can begin buying or selling cocoa.
			France is to provide Togo with 475 mln cfa francs of aid for a range of projects that include development of the coffee and cocoa industries and reafforestation in the south of the country, official sources said.
			The Coffee, Sugar and Cocoa Exchange amended regulations governing expanded trading limits on coffee, cocoa and sugar contracts to provide uniformity (...) Previously exchange rules required the first three limited months to move the limit in coffee and cocoa. It had required the first two limited sugar deliveries to make such moves for three consecutive sessions.
oil and gas company	oil, mln, ga, year, barrel, billion, lt, compani, reserv, natur	169	Kelley Oil and Gas Partners Ltd said it has agreed to purchase all of CF Industries Inc's oil and natural gas properties for about 5,500,000 dlrs, effective July 1. It said the Louisiana properties had proven reserves at year-end of 11 billion cubic feet of natural gas and 85,000 barrels of oil, condensate and natural gas liquids. Kelley said it currently owns working interests in some of the properties.
			Hamilton Oil Corp said reserves at the end of 1986 were 59.8 mln barrels of oil and 905.5 billion cubic feet of natural gas, or 211 mln barrels equivalent, up 10 mln equivalent barrels from a year before.
			Brazil will export 6,000 tonnes of poultry and 10,000 tonnes of frozen meat to Iraq in exchange for oil, Petrobras Commercial Director Carlos Sant'Anna said. Brazil has a barter deal with Iraq and currently imports 215,000 barrels per day of oil, of which 170,000 bpd are paid for with exports of Brazilian goods to that country.
tonnes of copper	tonn, copper, cent, price, mine, effect, beef, lb, meat, export	100	Mountain States Resources Corp said it acquired two properties to add to its strategic minerals holdings. The acquisitions include a total of 5,100 acres of titanium, zirconium and rare earth resources, the company said. (...)The company also announced the formation of Rare Tech Minerals Inc, a wholly-owned subsidiary.
			Magma Copper Co, a subsidiary of Newmont Mining Corp, said it is cutting its copper cathode price by 0.75 cent to 66 cents a lb, effective immediately.
			Newmont Mining Corp said Magma Copper Co anticipates being able to produce copper at a profit by 1991, assuming copper prices remain at their current levels. In an information statement distributed to Newmont shareholders explaining the dividend of Magma shares declared Tuesday ...
united food and commercial workers	unit, compani, plant, union, beef, lt, offer, contract, iowa, term	93	The United Food and Commercial Workers union, Local 222 said its members voted Sunday to strike the Iowa Beef Processors Inc Dakota City, Neb., plant, effective Tuesday. The company said it submitted its latest offer to the union at the same time announcing that on Tuesday it would end a lockout that started December 14. Union members unanimously rejected the latest company offer that was submitted to the union late last week, UFCW union spokesman Allen Zack said.
			Brazil will export 6,000 tonnes of poultry and 10,000 tonnes of frozen meat to Iraq in exchange for oil, Petrobras Commercial Director Carlos Sant'Anna said. Brazil has a barter deal with Iraq and currently imports 215,000 barrels per day of oil, of which 170,000 bpd are paid for with exports of Brazilian goods to that country.
			European Community agriculture ministers agreed to extend the 1986/87 milk and beef marketing years to the end of May, Belgian minister Paul de Keersmaeker told a news conference. He said the reason for the two-month extension of the only EC farm product marketing years which end during the spring months was that it would be impossible for ministers formally to agree 1987/88 farm price arrangements before May 12. This is when the European Parliament is due to deliver its opinion on price proposals from the EC Commission.

Table 2

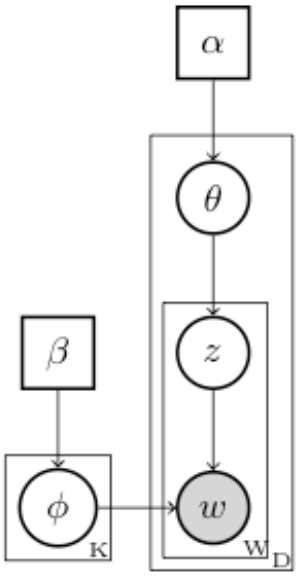
Suffix Array construction for the phrase "we are having a reunion"

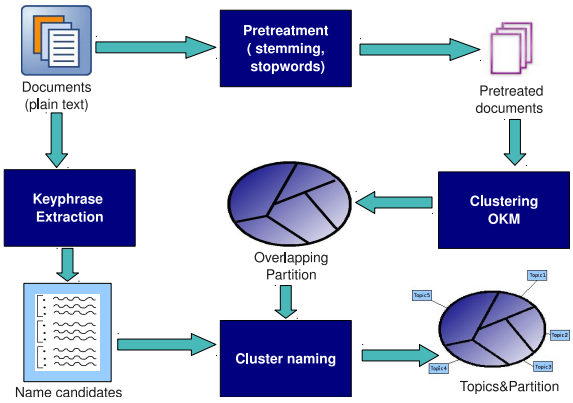
No	Suffix	Start Pos
1	a reunion	4
2	are having a reunion	2
3	having a reunion	3
4	reunion	5
5	we are having a reunion	1

Figure Captions

Figure 1. Schema of Latent Dirichlet Allocation.

Figure 2. Streamlined schema of the topic extraction system.


α
θ
z
w
β
ϕ
K
W
D


Documents
(plain text)
Pretreatment
(stemming,
stopwords)
Pretreated
documents
Keyphrase
Extraction
Clustering
OKM
Overlapping
Partition
Name candidates
Cluster naming
Topics&Partition