Conceptual Indexing of Documents Using Wikipedia

Carlo Abi Chahine, Nathalie Chaignaud, Jean-Philippe Kotowicz, Jean-Pierre Pécuchet

Conceptual Indexing of Documents Using Wikipedia

Carlo Abi Chahine, Nathalie Chaignaud, Jean-Philippe Kotowicz & Jean-Pierre Pécuchet
LITIS - EA 4108 - INSA Rouen, France
{*carlo.abi-chahine, nathalie.chaignaud, jean-philippe.kotowicz, jean-pierre.pecuchet*}*@insa-rouen.fr*

***Abstract*—This paper presents an indexing support system that suggests for librarians a set of topics and keywords relevant to a pedagogical document. Our method of document indexing uses the Wikipedia category network as a conceptual taxonomy. A directed acyclic graph is built for each document by mapping terms (one or more words) to a concept in the Wikipedia category network. Properties of the graph are used to weight these concepts. This allows the system to extract so-called *important concepts* from the graph and to disambiguate terms of the document. According to these concepts, topics and keywords are proposed. This method has been evaluated by the librarians on a corpus of french pedagogical documents.**

***Keywords*-Document Indexing, Keyword and Topic Extraction, Directed Acyclic Graph, Wikipedia**

I. Introduction

The goal of our work is to help the librarians of a French numeric university[1] to rapidly find the main topics and keywords of pedagogical documents, essentially of scientific or technical nature. It consists in designing and implementing an indexing support system that suggests a set of topics and keywords for a document, retrieved from its textual information. If the *librarians* do not accept them, new ones can be suggested. The relevance of these descriptors depends on their *representativity* (they correctly summarize the document) and their *discrimination* (sufficient to retrieve the document).

Statistical indexing methods (such as Term Frequency-Inverse Document Frequency (TF-IDF) [1] or Latent Semantic Analysis (LSA) [2]) propose features describing a document regardless of its meaning. From a corpus of documents, they compute the discriminating aspect of a word in a document with respect to the other documents.

To give sense to an extracted document descriptor, it is compulsory to use an external knowledge base. According to [3], knowledge base used for semantic and conceptual indexing are either conceptual taxonomies or formal ontologies. For our application, the Wikipedia category network [4] is used as a conceptual taxonomy. A directed acyclic graph is built for each document by mapping as many terms (one or more words) as possible to a concept in the Wikipedia category network. Properties of the graph are used to weight these concepts. This allows the system to extract so-called *important concepts* from the graph and to disambiguate terms of the document.

This article is organized as follows: Section II presents semantic and conceptual indexing using Wikipedia as a knowledge base. Section III introduces the representation of a document as a graph that is the core of our indexing support system. Section IV brings forward a way to compute the *important concepts* of a document, according to which topics and keywords are proposed and disambiguation of the terms of the document is done. Finally, Section V describes the results that are evaluated by the librarians on a corpus of french pedagogical documents.

II. Semantic and Conceptual Indexing Using Wikipedia

Semantic and conceptual indexing consists in finding a concept in a given knowledge base that matches a document term. Using links between concepts, the system is able to infer information. For example, to analyze a sentence that contains the words “Einstein” and “Relativity”, the system infers that these concepts are subsumed by the concept “Physics”, and therefore infers that the sentence probably deals with physics.

However, building a knowledge base, such as a semantic network, requires an effort (in terms of time and people involved) that discourages their use in an information retrieval system [5].

The online collaborative encyclopedia Wikipedia can be used to overcome this time consuming effort. Each article is manually integrated into categories and sub-categories, thus generating a category network. Each important term in an article is linked with another Wikipedia article, building an hyperlink network. In the Wikipedia world, the term “concept” often refers to “article” or “category”.

More precisely Wikipedia has been used for two tasks: firstly, topic/keyword extraction and word sense disambiguation to index documents and secondly, semantic relatedness to retrieve information. Table I summarizes the work cited in the following sections.

A. Topic/Keyword Extraction and Word Sense Disambiguation

The Wikify! system leverages the Wikipedia hyperlink network for keyword extraction [8] and word sense disambiguation (WSD) purposes [9]. Its name refers of matching

[1]UNIT, french acronym for “engineering and technology digital university” - `http://www.unit.eu/`

	Task			Wikipedia		
	Relatedness	Extraction	WSD	Corpus and hyperlink network	Category network	Method
Strube & Ponzetto (WikiRelate!) [6]	✓		✓	✓	✓	ontology based method
Gabrilovitch & Markovitch (ESA) [7]	✓		✓	✓		TF-IDF and ML
Mihalcea & Csomai (Wikify!) [8], [9]		keywords only	✓	✓		Keyphraseness, overlapping and ML
Medelyan, Witten & Milne [10], [11]	✓	topics only	✓	✓	✓	Keyphraseness, TF-IDF and ML
Coursey & Mihalcea [12], [13]		topics and keywords	✓	✓	✓	Keyphraseness, PageRank and ML
Fogarolli [14]			✓	✓		TF-IDF

Table I
COMPARISON OF METHODS FOR SEMANTIC AND CONCEPTUAL INDEXING USING WIKIPEDIA

a term with the corresponding Wikipedia article (the term has been *wikified*), using a keyword extraction method called *keyphraseness*. It computes the probability that a term of a document is a keyword ("the number of documents having the term as keyword" divided by "the number of documents where the term appears"). WSD is achieved in Wikify! by combining a knowledge-based approach (the Lesk-like disambiguation method [15]) and a data-driven method.

Medelyan, Witten and Milne [10], [11] extend this work and propose a Machine Learning (ML) method called "topic identification" that categorizes a document by computing its k-means topics. Among others, it is based on the TF-IDF score of the terms of a document, the length of the terms (number of words) or the number of "children" of a candidate topic (concept).

Coursey and Mihalcea [12], [13] extend Wikify! to enable "topic identification". They also use the Wikipedia category network (instead of the hyperlink network) and they adapt the PageRank algorithm [16] to determine the main topic of a document.

Finally, Fogarolli [14] uses Wikipedia hyperlink network to perform WSD, adding the notion of "strong links" between articles.

Table I summarizes all the methods mentioned above.

B. Semantic Relatedness

Wikipedia is also applied to compute semantic similarity (using hypernymy and hyponymy relations) and semantic relatedness (using all the types of relations).

Strube and Ponzetto's WikiRelate! [6] is a collection of methods to evaluate the semantic relatedness of two Wikipedia concepts using both the article titles and the category network. These methods stem from previous work, where semantic similarity and relatedness are computed using path based measures [17], [18], information content based measure [19], [20] or text overlap based measure [15].

As pointed out in [6], these measures are somewhat inefficient because they are too "Wordnet-centered".

To overvome this fact, Gabrilovich and Markovitch [7] introduce the Explicit Semantic Analysis (ESA) that computes a TF-IDF vector to quantify the links between terms and an article (i.e. a concept). This method compares not only terms (as in WikiRelate!) but also texts.

III. REPRESENTING A DOCUMENT BY A GRAPH

Our document indexing method uses the textual content of a document to be indexed and an external knowledge base (in this case Wikipedia). It consists in representing the document by a graph from the concepts of the base and their hierarchical relations.

The first step of the process is to map as many terms (one or more words) as possible of the document to a set of candidate concepts in the base (i.e. Wikipedia titles). Each of them corresponds to one meaning of the term, for example, the term "Python" matches with two concepts: "Python (Snake)" and "Python (Programming Language)". Preliminarily, the dictionary entries and the document have been lemmatized. The parser extracts terms from the document as follows: a sliding window of n (=4) words scans the text (between two punctuation marks), checking if the n words correspond to an entry of the given base. If the sequence of words does not match an entry of the dictionary, n is decremented and the adjectives, adverbs and empty words are eliminated until an entry is found. Thus, all the possibilities are tested.

Then, the set of the subsuming concepts of each candidate concept is calculated from the knowledge base, via generic-specific relations such as hypernymy or holonymy, until the root of the base is reached.

By merging the graphs of each term, a directed acyclic graph (DAG) representing the document is built: the terms of the document are leaves linked to the candidate concepts found in the base, linked in turn to other parent concepts,

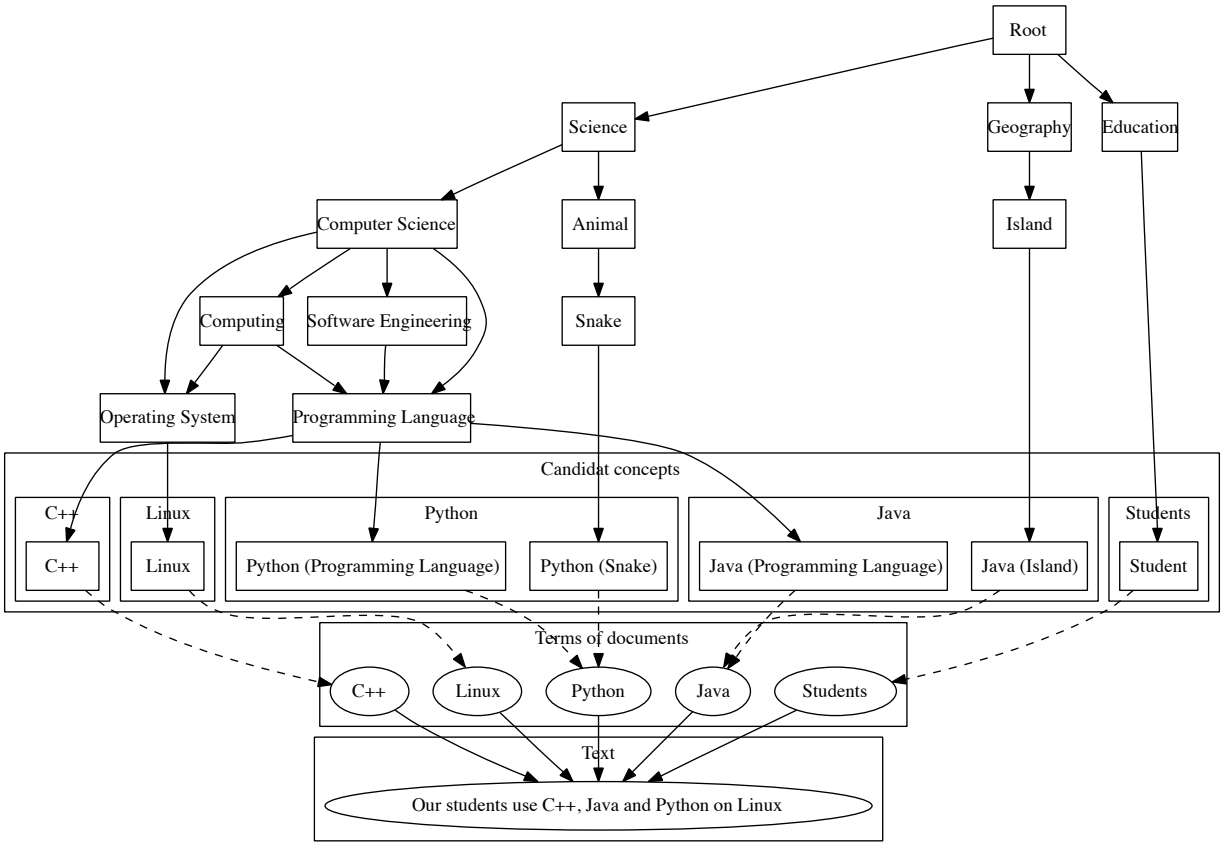

Figure 1. DAG for the sentence *"Our students use C++, Java and Python on Linux."*

and so on until the root. The complete algorithm to build this DAG is described in [21].

As an example, Figure 1 shows the graph associated the sentence *"Our students use C++, Java and Python on Linux."*.

This graph presents several ambiguous terms (Python, Java). For simplicity we will first concentrate on a simpler case (Figure 2).

IV. The Document Indexing Method

Once the graph is built, properties of this graph are used to weight its concepts and to highlight so-called *important concepts*. According to these concepts, main topics and keywords are proposed and disambiguation of terms of the document is done.

A. Important Concept Extraction

Important concepts are akin to Least Common Subsumers (LCS) of a big enough number of leaves (terms of the document). They should not be too specific but not too generic either.

Any concept of the DAG can be linked (directly or not) to several leaves (at least one). The number of leaves subsumed by a concept C is proportional to how often C is dealt with in a document. This number is called the *frequency* of C ($Freq(C)$).

As an example, in Figure 2, "software engineering" occurs once, "computing", "computer science" and "science" twice, and "root" three times. Thus, high level concepts occur more often a low level ones. Having a high number of leaves is necessary but not sufficient for a concept to be important.

A solution is to consider the depth of a concept C in the graph (number of concepts on the path from the root to C). However the notion of depth is ambiguous since several paths can exist from the root to C. For example, in Wikipedia, the depth of a concept can vary from 3 to 10 according to the chosen path. Despite this problem, many measures use the depth to calculate the similarity between concepts [17], [18]. To overcome this problem, we use the notion of *genericity* of a concept coming from the knowledge engineering field. As in [20], we define the *genericity* (Gen)

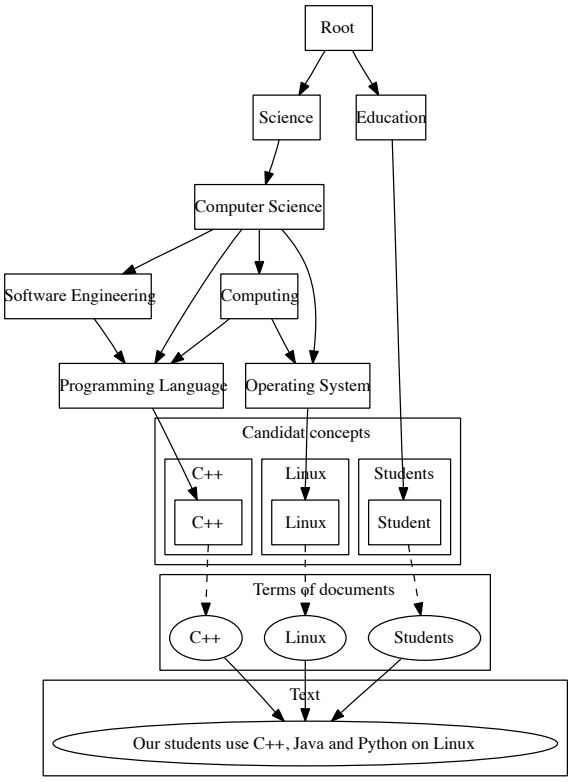

Figure 2. Graph without ambiguous term

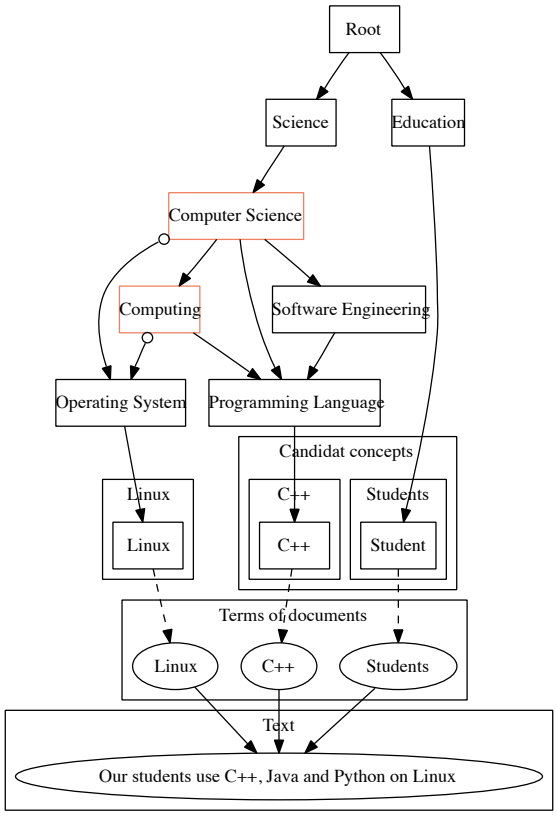

Figure 3. Merging of the graphs generated by the terms “Linux”, “C++” and “Students”.

of a concept C as the number of concepts it subsumes ($Sub^{-1}(C)$) (itself included):

$$Gen(C) = \#Sub^{-1}(C)$$

In Figure 2, $Gen(\text{“}computerscience\text{''}) = 7$ and $Gen(\text{“}root\text{''}) = 11$.

Genericity has problems of its own: a concept having a lot of direct children that are leaves is not necessarily generic but is important. Therefore, we discriminate between vertical genericity, simply called *genericity* and horizontal genericity, called *diversity*. The diversity of a concept does not only measure the number of concepts directly linked to it but also the number of document terms having generated this concept.

To calculate this measure, for each term T of the document, the graph is split in two parts:

- the subgraph generated by T (the leaf and its ancestors),
- and the subgraph generated by all the other terms of the document (and their ancestors).

These two graphs are merged having concepts in common (at least the concept “root”). We add +1 to each of them having different links in the two graphs (all the concepts are initialized to 0). For each concept, the diversity is the calculated sum.

Figure 3 shows the merging of the graph generated by the term “Linux” with the graph generated by the terms “C++” and “students”. The concepts “computer science” and “computing” are common concepts of the two graphs with different links (those marked with a little circle). In this figure, $Div(\text{“}root\text{''}) = 1$, $Div(\text{“}computerscience\text{''}) = 2$ and $Div(\text{“}computing\text{''}) = 2$ (the diversity of all the other concepts are null).

To decide whether a concept C is important, we calculate its score S depending on its frequency $Freq$, its genericity Gen and its diversity Div. The concepts having the highest score are considered as important concepts. This score is calculated as

$$S(C) = \log\left[\frac{Freq(C)}{Gen(C)} + 1\right] \times (Div(C) + 1).$$

The logarithm function reduces the ratio $Freq(C)/Gen(C)$ and the function $+1$ provides only positive scores. Moreover, in order to not penalize concepts

C	$Freq(C)$	$Gen(C)$	$Div(C)$	$S(C)$	$Rank(C)$
C++	1	1	0	0.669	3
Computing	2	5	2	1.176	1
Computer Science	2	7	2	0.934	2
Education	1	2	0	0.452	7
Linux	1	1	0	0.669	3
Operating System	1	2	0	0.452	7
Programming Language	1	2	0	0.452	7
Root	3	11	1	0.602	6
Science	2	8	0	0.283	11
Software Engineering	1	3	0	0.347	10
Student	1	1	0	0.669	3

Table II
SCORE AND RANKING OF THE CONCEPTS OF THE GRAPH OF FIGURE 2

T	$KS(T)$
Linux	0.495
C++	0.452
Students	0.416

Table III
KS FOR THE TERMS "LINUX", "C++" AND "STUDENTS".

having null diversity, we add 1 to it.

Table II gives the frequency $Freq$, the genericity Gen, the diversity Div and the global score S for each concept of the graph of Figure 2.

The concepts "computing" and "computer science" have the highest scores. The concept "science" is not important because it is too generic and not diverse enough.

B. Keyword and Main Topic Extraction

The system is able to extract important concepts from a document. These concepts can be keywords to index a document. But it is also interesting to search keywords that are terms of the document. In this case, a term is chosen as a keyword if it is subsumed by important concepts.

Therefore, we define the keyword scoring measure KS for a term T of a document as the average score of the concepts that subsume T.

$$KS(T) = \frac{1}{\#Sub(T)} \sum_{C \in Sub(T)} S(C)$$

Table III presents the $KS(T)$ values for each term of the document represented by the graph in Figure 2.

Indexing is also achieved using main topics of a document. We define them as concepts that subsume important concepts.

To extract them, we use a measure called Level Scoring LS as the sum of the scores of the descendants ($I \in Sub^{-1}(C), I \neq C$) of a concept C using a shortest path ($SP(C, I)$) not longer than k.

$$LS(C,k) = \sum_{I \in Sub^{-1}(C), I \neq C} \begin{cases} S(I) & \text{if } SP(C,I) \leq k \\ 0 & \text{else} \end{cases}$$

C	$LS(C,1)$	$LS(C,2)$
Computer Science	1.57	2.57
Science	0.66	2.23
Computing	0.66	1.66
Root	0.53	1.69
Programming Language	0.5	0.5
Operating System	0.5	0.5
Education	0.5	0.5
Software Engineering	0.33	0.83
Student	0.0	0.0
Linux	0.0	0.0
C++	0.0	0.0

Table IV
THE LEVEL SCORING OF THE CONCEPTS OF THE GRAPH OF FIGURE 2

Table IV gives the LS for each concept of the graph in Figure 2. In this example, the two main topics are "computer science" and "science" (for both 1-level scoring and 2-level scoring).

C. Term Disambiguation

We propose to disambiguate terms using the DAG model, by selecting only the parent (candidate concept) of an ambiguous term that matches the context of the document. To this aim, the candidate concept having the most important concepts that subsume it will be chosen.

For example, in Figure 1, the concepts "Python (Programming Language)" and "Java (Programming Language)" are the correct candidates concepts because there exists a path between these concepts and the important concepts "computing" and "computer science".

Once disambiguation is done, important concepts, keywords and main topics are computed again, to take into account the disambiguated terms.

V. EVALUATION

We evaluate the efficiency of our measures on a corpus of french pedagogical documents from UNIT and using the french Wikipedia Category Network as a knowledge base. To carry out our experiment, we removed some categories

from the DAG (for instance the administrative categories like "Wikipedia Maintenance", etc.).

A. The Evaluation Protocol

For technical reasons, we can not use the web documents of UNIT, since there are a lot of non-textual documents (video and sound) and textual documents are in different formats (PDF, PPT, HTML, etc). However, each document is embedded with its textual summary and keywords chosen by the librarians. Each document is also manually classified in one or several UNIT categories and subcategories. There are 25 UNIT categories (e.g. Chemistry, Computer Sciences, etc.) and around 200 subcategories (e.g. Organic chemistry, Database, etc.). For each UNIT category and subcategory, the summaries are grouped into one text, forming a document used for the evaluations.

Two evaluations were carried out:

1) We selected 50 random documents corresponding to 50 categories and subcategories, and the system extracted the 5 main topics with the best 15 important concepts and 15 best keywords.
   We use our score for the important concepts and a 1-level scoring for the topics. For each of the 50 categories, we submitted to the two librarians the category name and the 5 main topics (with the concepts and keywords). They had to answer the question: "Are the extracted topics, concepts and keywords relevant for this category?"
2) For 6 categories, we submitted to the two librarians the document made with the summaries. For each ambiguous term of the document, we gave the list of possible disambiguated terms, and the one chosen automatically. The librarians had to answer the following question: "Does the list contain the correct context of the term? Is the automatically chosen term correct?".

B. The Evaluation Results

1) First Experiment: The librarians are not experts in all the fields. The system extracted the best 5 main topics from the 50 categories. If the first topic with the related concepts and keywords are relevant for the librarians, they select it and carry out the same evaluation with the next category. If the first topic is not relevant, they evaluate the second, and so on.

Figure 4 shows the result of this evaluation. For 50 categories, 74% of the time, the main topic (i.e. one of the 5 topics extracted by the system) was successfully retrieved. Conversely, 26% of the time, the system failed to retrieve the topics.

Actually, the overlap of the keywords is calculated in a summary (i.e. the ratio of keywords that are present in the summary). On average, the overlapping does not exceed 10%, which means that the summary alone cannot exhaustibly describe a document.

number of ambiguous terms (1)	282
number of ambiguous concepts proposed (2)	1173
number of correct concepts in the set (3)	185
number of correct automatically retrieved concepts (4)	156
ratio (4)/(1)	0.55
ratio (3)/(1)	0.69
ratio (4)/(3)	0.8

Table V
DISAMBIGUATION EVALUATION

Thus, with a low overlap, most of the time, the system managed to retrieve the important concepts of a set of documents.

2) Second Experiment: The previous evaluation has been carried out without ambiguous term. The next step is to evaluate the capacity of the system to disambiguate terms of a document.

A term is ambiguous if several wikipedia concepts have the form "term (context)". In the first place, we compute the important concepts without the ambiguous terms. Then, we select the best ambiguous concept for each term.

For 6 categories of UNIT, we submitted the set of ambiguous terms to the two librarians. For each term, candidate concepts for disambiguation are suggested and the automatically retrieved one is underlined.

Table V shows the number of ambiguous terms in the 6 category summaries (1) and the total number of possible meanings (2). It also gives the number of times a list of ambiguous concepts contains the good candidate for disambiguation (3) and the number of times the system retrieved the good candidate (4). Finally, we evaluate the ratio of correct disambiguation.

The ratio of successfully disambiguated term is weak (0.55) but using only the Wikipedia Category Network the best we could do is 0.69 since 31% of the proposed concepts did not contain the correct one. Thus, by changing the baseline, we calculated the ratio of correct concepts found on the number of time the correct concept was in the set. 80% of the time, if the set contains the relevant concept, the concept was retrieved correctly.

VI. CONCLUSION AND PERSPECTIVES

Our method of document indexing uses the Wikipedia Category Network as a conceptual taxonomy to build a Directed Acyclic Graph (DAG) model representing a document. Using a generic measure to extract important concepts from a document, three tasks can be performed:

- retrieving the main topics of a document (the topicality),
- finding the keywords of a document,
- and disambiguating the terms of a document.

The evaluation brings very encouraging results for topic extraction, although a great effort is still needed to carry out

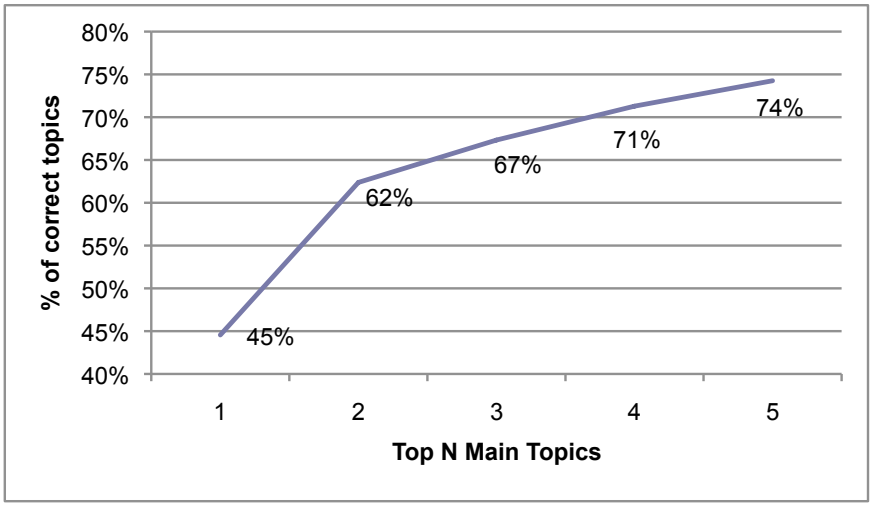

Figure 4. Evaluation of the keyword and main topic extraction.

disambiguation. In fact, by using only the DAG, it is difficult to overlap all the possible ambiguous terms since all the ambiguities could not be solved using the wikipedia form "term (context)". In order to continue the disambiguation process, we need to analyze the "disambiguation pages", that is to say, the content of the article of each ambiguous entry.

The next step is to design an information retrieval system that takes a request as input and retrieves the corresponding documents. The method we propose is to translate the user's query into a DAG. Then, this graph will be compared with each document graph to propose only relevant documents. Thus, we have to propose a measure of similarity between a document DAG, taking into account the score associated to each concept. If a concept is strong in the query DAG and in the document DAG, the document might be relevant for the user. This similarity measure will also be used for two documents, that is to say deciding whether two documents are similar or not.

Finally, the DAG of a document can by sliced into several smaller DAGs, each representing a part of the document. The score of each DAG specifies its relatedness towards the entire document. Moreover, since WSD can only be achieved successfully by extracting the local context where a term appears, slicing the DAG will certainly give a better scoring to disambiguation.

References

[1] G. Salton, *Introduction to Modern Information Retrieval*, ser. McGraw-Hill Computer Science Series. McGraw-Hill, September 1983.

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, K. L. Thomas, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391–407, 1990.

[3] H. Haav and T. Lubi, "A survey of concept-based information retrieval tools on the web," in *5th East-European Conference, ADBIS*. Citeseer, 2001, pp. 29–41.

[4] "Wikipedia:Categorization," retrieved on 03/14/2011. [Online]. Available: http://en.wikipedia.org/wiki/Wikipedia:Categorization

[5] W. Frakes and R. Baeza-Yates, *Information retrieval: data structures and algorithms*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1992.

[6] M. Strube and S. Ponzetto, "WikiRelate! Computing semantic relatedness using Wikipedia," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 2. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 1419.

[7] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 6–12.

[8] R. Mihalcea and A. Csomai, "Wikify!: linking documents to encyclopedic knowledge," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM New York, NY, USA, 2007, pp. 233–242.

[9] R. Mihalcea, "Using wikipedia for automatic word sense disambiguation," in *Proceedings of NAACL HLT*, vol. 2007, 2007.

[10] O. Medelyan, I. Witten, and D. Milne, "Topic indexing with Wikipedia," in *Proceedings of the AAAI WikiAI workshop*, 2008.

[11] D. Witten and D. Milne, "An effective, low-cost measure of semantic relatedness obtained from Wikipedia links," in *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*, 2008, pp. 25–30.

[12] K. Coursey, R. Mihalcea, and W. Moen, "Using encyclopedic knowledge for automatic topic identification," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2009, pp. 210–218.

[13] K. Coursey and R. Mihalcea, "Topic identification using Wikipedia graph centrality," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Association for Computational Linguistics, 2009, pp. 117–120.

[14] A. Fogarolli, "Word Sense Disambiguation Based on Wikipedia Link Structure," in *2009 IEEE International Conference on Semantic Computing*. IEEE, 2009, pp. 77–82.

[15] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone," in *Proceedings of the 5th annual international conference on Systems documentation*. ACM New York, NY, USA, 1986, pp. 24–26.

[16] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[17] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification," *WordNet: An electronic lexical database*, vol. 49, no. 2, pp. 265–283, 1998.

[18] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics Morristown, NJ, USA, 1994, pp. 133–138.

[19] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *International Joint Conference on Artificial Intelligence*, vol. 14. Citeseer, 1995, pp. 448–453.

[20] N. Seco, T. Veale, and J. Hayes, "An intrinsic information content metric for semantic similarity in WordNet," in *ECAI*, vol. 16. Citeseer, 2004, p. 1089.

[21] C. Abi Chahine, N. Chaignaud, J. Kotowicz, and J. Pécuchet, "Context and keyword extraction in plain text using a graph representation," in *IEEE Workshop KARE, SITIS'08*, 2008, pp. 692–696.