Extracting More Relevant Document Descriptors using Hierarchical Information

Antoine Doucet[1,2]

[1] University of Helsinki,
Department of Computer Science,
P.O. Box 26 (Teollisuuskatu 23),
FIN-00014 University of Helsinki, Finland,
antoine.doucet@cs.helsinki.fi
[2] Université de Caen,
Département d'Informatique,
Campus Côte de Nacre,
F-14032 Caen Cedex, France,
antoine.doucet@info.unicaen.fr

Abstract. The emergence of a constantly growing quantity of semi-structured documents has provided considerable hierarchical information. However, most of the document processing techniques are still developed without regard to the hierarchical structure of the documents. This is especially true when the structure is only partial, in which case the preprocessing may simply consist in pruning the structure !

We present in this paper a method for adding structural information to text content descriptors. The resulting document descriptors take the form of word sequences, to which structural information has been added. The result of this process is a mapping from each document of the collection to a (possibly empty) set of descriptive text phrases. Following this ground work, many applications can be considered based on the extracted information.

This method was implemented and tested using a collection of French geographic articles. One example application was then tested, using the new set of descriptors: the discovery of co-occurring text phrases, based on classical algorithms for association rules discovery. An empirical comparison was then made between the results obtained via the "classical" and the proposed "structure-aware" method.

1 Introduction

Since a few decades ago, computer systems and networks have been exponentially growing. So has the amount of digital information stored in these systems: using automated data collection tools, massive amounts of data have been collected and stored in databases. Among this rising quantity of data, the proportion

consisting of semi-structured text documents has exponentially increased. This growing number has provided numerous contextual information. Unfortunately, most text processing systems tend to ignore hierarchical information issued from semi-structured documents. Indeed, in most cases, a majority of the structure is irrelevant to the user (e.g., metadata), and it is therefore often entirely ignored. But since a few years ago, the emergence of the simple encoding standard XML [2] has brought valuable semantic information to the path leading to a text element.

In the case of very large text document collections, applying data mining techniques is a recent idea. Appearing in the second part of the nineties, the concept of data mining in text, known as *text mining*, has been of growing interest. It is especially appropriate when the user does not exactly know what he (or she) is looking for. In this context, the Doremi[3] research group (in the University of Helsinki, Finland) has developed a method for attaching compact content descriptors to full text documents: their maximal frequent sequences [4, 5]. A maximal frequent sequence is a sequence of words that occurs frequently in the document collection and moreover, that is not contained in any other longer frequent sequence. This permits the extraction of very compact content descriptors from the documents of the collection. Unfortunately, the method and experiments were based on plain text documents, rather than structured ones.

Nowadays, an increasing part of the newly created textual data is either semi- or fully structured. Hence, a natural improvement has been to meliorate the relevance of the results by taking the extra information provided by the document structure (be it full or partial) into account. This technique was then recently adapted to include hierarchical information, with a small influence on the algorithm's performance. The evidence of the empirical results has shown the extracted descriptors to have a better relevance. Once each of the documents has been associated with a bag of descriptors, the applications are numerous. In the result section we sketch one experiment we led, based on the discovery of association rules [3]. In this data mining application, we considered the sets of descriptors as the frequent itemsets, and discovered various relations between the descriptors.

In the remainder of this paper, we will summarize the existing method based on plain text documents in section 2. Section 3 will describe the new method: "Word to Pathword (W2PW)", adapted so as to take the documents' hierarchical structure into account. Then we will compare the data we extracted using the plain text method and the data we extracted using the enhanced method (section 4). In section 5, we sketch a comparison between association discovery results from the classical and from the proposed method. Finally, result analysis and a brief discussion will conclude this paper in section 6, opening issues regarding different ways to further develop this work.

[3] DOcument management, information REtrieval, text and data MIning

2 Maximal Frequent Sequences

The technique of extracting *Maximal Frequent Sequences* (MFS) from a document collection is extensively described in [4]. It exhibits details of the method used by the Doremi group for extracting document descriptors from a large plain text document collection. We will hereby summarize the main steps of that method and later remind of its specific strengths.

2.1 MFS: Definition and Extraction Technique

Three main steps are processed within the Doremi method. The general idea fits the main phases of KDD (Knowledge Discovery in Databases); that is, selection and cleansing of the data, followed by the use of core mining techniques, and a final postprocessing step intending to transform and select the results into an understandable knowledge.

Definition of MFS. Assuming S is a set of documents, and each document consists of a sequence of words...

Definition 1. *A sequence $p = a_1...a_k$ is a* subsequence *of a sequence q if all the items a_i, $1 \leq i \leq k$, occur in q and they occur in the same order as in p. If a sequence p is a subsequence of a sequence q, we also say that p* occurs *in q.*

Definition 2. *A sequence p is* frequent *in S if p is a subsequence of at least σ documents of S, where σ is a given frequency threshold.*

Note that only one occurrence of a sequence within a document is counted: whether a sequence occurs once or several times within the same document does not change its frequency.

Definition 3. *A sequence p is a* maximal frequent (sub)sequence *in S if there does not exist any sequence p' in S such that p is a subsequence of p', and p' is frequent in S.*

Phase 1: Preprocessing This first phase basically consists in 'clearing' the data of its useless information. Of course, whether an item is useful or not strongly depends on the intended use of the results. Usually, special characters (including, for example, punctuation and brackets) are pruned away. To avoid processing uninteresting items, a stopword list is also created. It includes articles, pronouns, conjunctions, common adverbs, and common forms of non-informative verbs (e.g., *is, are, be*). All of its elements are removed.

Typically, the two following text fragments:
`...President of the United States Bush...`
`...President George W. Bush...`

would result in:
`...President United States Bush...`
`...President George Bush...`

Phase 2: Extraction Technique (an overview).

Initial phase: Collecting all Frequent Pairs. In this initial phase, all pairs of words, such that their frequency is greater than a given threshold, σ (10 in the experiment), are collected. Two words form a pair if they occur in the same document, and if their distance is less than a given maximal gap. A gap of 2 was used in the experiment, which means that at most 2 other words can appear between the words forming a pair. Also, note that the pairs are ordered, i.e. the pairs (A,B) and (B,A) are different.

Expanding the frequent pairs to MFS's. For each step k, $Grams_k$ is the number of frequent sets of length k. Hence, the frequent pairs found in the initial phase form $Grams_2$. MFS's are found bottom-up by combining shorter frequent sequences (of length k) to form longer sequences (of length $k+1$). Each step also includes various pruning stages, so as to ensure computational efficiency.

Further details about the different phases of discovery would be highly technical and irrelevant for this paper's purpose. Please refer to [4] if you want to know more.

Phase 3: Postprocessing. However, by computing maximal frequent sequences, and by increasing the length of the selected sequences, the corresponding frequencies naturally tend to decrease toward the minimum frequency threshold. Sequences that are both shorter and more frequent have not been selected, even if they might carry more valuable information. This might be especially true if the sequence's frequency was much higher, by taking a few words out of it.

Therefore, more sequences are added to the final set of content descriptors (based on [6]), which will be used in the next phases. For each maximal frequent sequence, any of its subsequences responding to both of the following criterions will be selected:

- its frequency is bigger than the corresponding maximal frequent sequence's
- it is not the subsequence of some descriptive sequence having an equal frequency

This way, in the resulting sets of descriptive sequences, the sequences' sizes are optimized by the maximal frequent sequences, and the sequences' frequencies are optimized by the subsequences added afterwards.

Finally, as a result, an (eventually empty) list of content descriptors is attached to each document of the collection.

2.2 Main Strengths of the Method

The method efficiently extracts all the maximal frequent word sequences from the collection. From the definitions above, a sequence is said to be maximal if and only if no other frequent sequence contains that sequence.

Furthermore, a *gap* between words is allowed: in a sentence, the words do not have to appear continuously: a parameter g tells how many other words two words in a sequence can have between them. The parameter g usually gets values between 1 and 3.

For instance, if g = 2, a phrase "president Bush" will be found in both of the following text fragments:
`...President of the United States Bush...`
`...President George W. Bush...`
Note: Articles and prepositions were pruned away during the preprocessing.

This allowance of gaps between words of a sequence is probably the strongest specificity of the method, compared to the other existing methods for extracting text descriptors. This greatly increases the quality of the phrase, since processing takes the variety of natural language into account. The method is *style tolerant*. Even deficient syntax can be handled (and that is fairly common in newswires, for example).

The other powerful specificity of the Doremi technique is the ability to extract maximal frequent sequences of any length. This offers a very compact description. By example, by restricting the length of phrases to 8, the presence, in the document collection, of a frequent 25 words long phrase, would result in thousands of phrases representing the same knowledge as the one maximal sequence.

3 The "Word to PathWord" Method (W2PW)

The main idea of this structure-aware method is to modify the notion of a word. Each word of the collection is bound to its corresponding path in the structure's hierarchy. For example, in a recipes book, by applying this modification, potato may become either "/book/chapter/appetizers/recipe/ingredients/potato" or "/book/chapter/side_dishes/recipe/ingredients/potato". The point is that these 2 occurrences of the same word will be considered different, because they occur within different paths. This might be particularly helpful for homonyms (i.e., words sharing the same spelling but a different meaning).

The bad aspect of this approach is that it may be misleading in some cases: Thinking of an HTML document for example, being written under the style "bold" or with a different style (or without any) would make a word different. A difference on a lower level of the structure tree would also make a difference: A word would be considered different, depending on whether it is referred (by the structure) directly in a chapter or in a subchapter of a main chapter.

StopList of tags. Based on this example, if "potato" could indeed be considered different whether mentioned as an appetizer ingredient (e.g., crisps) or as a side-dish, on the other hand, one can complain that being included within a subchapter or not shouldn't make any difference:
"/book/chapter/appetizers/recipe/ingredients/potato" should be equivalent to "/book/chapter/subchapter/appetizers/recipe/ingredients/potato".

Therefore, the ability to create and use a stoplist of structure tags was naturally needed to separate relevant and non-relevant hierarchical information. This way, tags not carrying any important information, such as <chapter>, <subchapter> or <book> will not be considered. That pruning list also solves the "text style" problem: if we consider that the HTML tag <B> (for bold) brings a misleading information by making "<B>potato</B>" differ from "potato", then "<B>" should simply be included in the *"stoptag" list*.

Of course, as in any pruning step, the selection of the tags that should be included in the stoplist is strongly related to the intended use of the data (e.g., the font style might actually be what the user is interested in !). This feature makes the method highly adaptive to the user's interests.

Further Details. The switch from word to "path+word" is made by reading the data from the standard input and:

- storing the current path.
- printing each word preceded by the current path to the standard output.

This is clearly of linear complexity, since it only requires one pass over the data, and a single structure to store the current path. But we also need to take care of non-closed tags, such as
 (i.e., a newline in an HTML document).

To get rid of these single tags, a step was added to the preprocessing phase. Its principle is when a closing tag is met, then it is checked to determine whether it equals the last part of the *current_path*. If not, then this last part (i.e., the last opened tag) will never be closed, since we assume that the data is well-formed. Therefore, a backward search is made, so as to delete that never-closed tag.

This part of the algorithm can be considered as problematic in the case where numerous empty tags are present. But this is untrue in practice, since empty tags should be deleted during the preprocessing, via the stoplist of tags (as those empty tags cannot occur in any final result anyway). This process is thus done in linear time. This is why this backward search remains safe in practice: It is only used to deal with rare irregularities of the data.

4 Experiments

The Word to Pathword (W2PW) method was implemented in Perl. We tested it by comparing the result obtained by the former method (plain text) and the proposed W2PW method. This was done by using the French-written semi-structured document collection of the Cross-Channel Atlas [9], a 205 documents collection created through electronic sharing of geography articles, focused on the

transborder relations between France and Great Britain, and fed by researchers from both countries. A sample of one of these semi-structured documents is shown in figure 1. The experiments focus on the content descriptors obtained via both techniques, comparing their amount, size and relevance.

```
<HEAD><TITLE>Situation des liaisons maritimes passagers (hiver 1998/1999)
</TITLE><DATE>1999-3-1</DATE><AUTHOR><NAME><LAST>Buléon
</LAST><FIRST>Pascal</FIRST></NAME><EMAIL>buleon@mrsh.unicaen.fr
</EMAIL></AUTHOR><AUTHOR><NAME><LAST>Lefevre</LAST><FIRST>Samuel
</FIRST></NAME><EMAIL>samuel.lefevre@criuc.unicaen.fr</EMAIL>
</AUTHOR><DESCRIPTOR>Fusion</DESCRIPTOR><DESCRIPTOR>compagnies
de ferries</DESCRIPTOR><DESCRIPTOR>liaisons maritimes</DESCRIPTOR>
</HEAD><BODY><FONT SIZE=2><P ALIGN="JUSTIFY"><CENTER>
<IMG SRC="lecteur.php?base=atlas&cmde=image&refimage=212">
</CENTER></P></FONT><P ALIGN="JUSTIFY"></P><P ALIGN="JUSTIFY">Les
<linkid>liaisons maritimes</linkid> passagers ont connu une
nouvelle évolution depuis l'automne 1998. Celle-ci est une
nouvelle conséquences de la <linkid>fusion</linkid> P&O et
prolonge celles survenues dans le cours de l'année 98.
(...)
```

Fig. 1. A sample of a document from the cross-channel atlas

4.1 Discovery of Content Descriptors

The stoplist of tags contains the common non-closed tags from the document collection, which are
 and <IMG>. It also includes those tags, that are not carrying relevant information, with respect to our interest. In this experiment, we considered the style of a document as irrelevant information. Therefore, such tags as <FONT>, <P>, <HEAD>, and <BODY> have been pruned away.

The whole preprocessing phase condensed the data from 74,754 words to 40,714 words. Note that each opening and closing tag is here counted as a word.

Number of maximal frequent sequences discovered. Table 1 gives a summary of the number of maximal frequent sequences discovered, depending on the method and frequency threshold.

Since the W2PW method makes some words different from each other, we could have expected that the number of content descriptors would be lower for this method, compared to the classical one, based on the same frequency threshold. However, it is actually quite similar. Besides that, one can observe that the number of content descriptors found by W2PW declines at a slower pace when the frequency threshold raises.

The reason for this difference is that many highly frequent words are found within the header of the documents, which is pruned away in the classical

Frequency Threshold	2	3	4	5	6	7	8	9	10	15	20
Classical Method	3,780	1,096	456	236	132	81	50	32	27	8	5
W2PW	3,816	1,137	493	260	162	113	75	53	42	20	9

Table 1. Number of maximal frequent sequences depending on method and frequency threshold

method. This fact compensates the average frequency reduction due to these newly different words. Confirmation occurs when the frequency threshold becomes unreasonably high. The number of maximal frequent sequences extracted through W2PW then becomes even larger than through the classical method. Indeed, the more the frequency threshold rises, the more those highly frequent words, occurring within the header, get proportionally numerous within the frequent ones.

During the next steps of the experiment, 4 was chosen as a frequency threshold. A larger number would lead to a lack of material for the following steps, whereas a lower number might select irrelevant descriptors, since we only consider a 205 documents collection.

Length of the Phrase	2	3	4-6	7-10	11-15	16-25	26+
Classical Method	444	9	1	1	1	0	0
W2PW	426	17	21	10	9	2	1

Table 2. Classical vs. W2PW, number of maximal phrases of various length

Length of the maximal frequent sequences. Looking at table 2, the main observation is that the longest sequences are found via the new method. This is due to the fact that long sequences are mostly found in the headers, and that a few words always occur together: for instance, the same last name with the same first name and the same email address. If a group of persons is often co-writing documents about similar themes, then the descriptive phrase can easily become quite long. For example, the phrase in figure 2 was found in 7 documents of the collection.

```
/date/NR  /author/name/last/loew-pellen  /author/name/first/frédérique
/author/email/loew@mrsh.unicaen.fr  /author/name/last/winter
/author/name/first/ansgar  /descriptor/emploi  /descriptor/indicateurs
/descriptor/population  /descriptor/ville  /some  /linkid/regional
/linkid/indicators  /about  /population  /and  /employment  /with
/map  /of  /travel  /to  /work  /areas
```

Fig. 2. An instance of a long descriptor (size 24).

It shows that Frédérique Loew-Pellen and Ansgar Winter are frequently co-writing documents, that they described by "emploi", "indicateurs", "population", and "ville" (entrance descriptors). These documents include links (i.e., exit descriptors) through the phrase "regional indicators". The plain text sequence "about population and employment with map of travel to work areas" then occurs within the body of the document.

5 An application: the Discovery of Associations

From these content descriptors, we can compute the association rules discovery, in a similar way as in [6].

This was done with various confidence and support threshold values. For both methods, the main difficulty was to pick a correct support threshold. Indeed, the collection of documents is not extremely large, and a 1% variation within the support differs the number of associations rules produced from 20 to thousands! This is also due to a large amount of phrases, occurring with similar frequencies, of type "region, city", such as "Hampshire, Southampton". Indeed, one of these typical association rules is:

```
/table/fareham_/table/hampshire
    =>
/table/hampshire_/table/southampton (1.00,0.04)
```

...meaning that when the phrase *(fareham, hampshire)* occurs in a document, then the phrase *(hampshire, southampton)* always occurs in the same document (confidence = 1 = 100%). Both of the phrases occur together in 4% of the documents of the collection. Also, observe that this association rule stems from the W2PW method. This offers additional information: these phrases were found in a table. The same association rule occurs, using the classical method. The only difference is that one piece of information is missing: nothing shows that this data is related to a table.

And indeed, this is the main difference between the 2 methods: W2PW effectively adds information to the results.

5.1 Extra Information is added...

Below is an even better example of what this hierarchical extra information can bring, comparing the classical method's association rules to the new one's.

- Classical method:
  `buléon_pascal => transport_régional (0.86,0.03)`
  From this association rule, based on the classical method, the extracted information is that: *"Buléon Pascal" and "regional transport" are somehow related.* But we do not learn anything about the relation itself.

- Word to PathWord:
  `/author/name/last/buléon_/author/name/first/pascal`
  `=>`
  `/descriptor/transport_/descriptor/regional (0.86,0.03)`
  Looking at the association rule based on the new method, the extracted information is now that: *Pascal Buléon is an author, "Pascal" is his first name, and "Buléon" his last. In addition, 86% of the documents he wrote are dealing with regional transportation.*

5.2 ...but a counterpart may exist

We can also notice that the support and confidence may differ between results related to the same phrases. For example:

- Classical method:
  `jersey_guernesey => îles_anglo-normandes (0.78,0.05)`
- Word to PathWord:
  `/jersey_/guernesey => /îles_/anglo-normandes (0.88,0.04)`

This means that one or more occurrences of "/îles_anglo-normandes" were found within a different path, and thus not counted together (because it is not considered as the same entity). A consequence is that some descriptors will not be found, using the W2PW, even though they were extracted by the classical method. This happens in the cases where that phenomenon pushes the frequency of a phrase below the threshold.

But it is not certain that the consequently unextracted descriptors should be regretted. Actually, it relies on the quality of the stoplist of tags. If the latter is appropriate, then those identical words that are differing only due to their paths should not be mixed, indeed. Based on a good stoplist of tags, this frequency lowering is actually very valuable.

6 Results and Discussion

We have successfully added structural information to the content descriptors of the documents, via the maximal frequent sequences discovery technique. A new method, the Word to PathWord, has been presented, implemented and tested. Some important observations can be formulated.

One of its strength is that it does not require the documents to be valid. In other words, no DTD is required. The only assumption is that the documents must be well-formed, that is, no overlap is allowed.

Also, hierarchical knowledge is efficiently added to the data, in the way that the structural information gives further details about the content descriptors, e.g., their structural context (table, item-list...), style context (bold, italic, emphasized...), or any other hierarchical information from the data.

As already underlined, the quality of the stoplist of tags is a crucial element, both for a better computational efficiency (by pruning empty tags) and for a

better relevance (by dropping irrelevant tags). The latter implies that building the list requires a good knowledge of what the end user is interested in (i.e., what is relevant *to him/her*). That's why any use of the method on another collection of documents requires particular attention in building the stoplist of tags.

However, many further developments and ameliorations shall be considered. Among these, a way to take advantage of the information from the tag's attributes has to be added to the method features. Also, the computational efficiency of the method could be improved, by deepening the techniques, if they were to be adapted to specific data or for a specific purpose, whereas in this paper, the data spectrum was kept as broad as possible. The lack of a *very large* and publicly-available semi-structured document collection has also been problematic. Fortunately, such a collection has recently been released, through the "Initiative for the Evaluation of XML retrieval" (INEX) [8, 1]. This new dataset will permit further testing from a more significant amount of information.

It is clear that the extraction and binding of a set of content descriptors to each document of a collection gives various applicative prospects. These sets of descriptors representing the documents may be used so as to measure distances between the documents and thus form clusters of documents. Another possibility is to use these descriptors to dynamically create hyperlinks between the documents [7].

Acknowledgements

This work is supported by the Academy of Finland (project 50959; DoReMi - Document Management, Information Retrieval, and Text Mining).

References

1. Initiative for the evaluation of xml retrieval, Available at http://qmir.dcs.qmw.ac.uk/INEX/. [Cited 23 September 2002].
2. Extensible markup language (xml), Available at http://www.w3.org/XML/. [Cited 23 September 2002].
3. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining, Menlo Park, California, USA*, pages 307–328. AAAI Press, 1996.
4. H. Ahonen-Myka. Finding All Frequent Maximal Sequences in Text. In *Proceedings of the 16th International Conference on Machine Learning ICML-99 Workshop on Machine Learning in Text Data Analysis, Ljubljana, Slovenia*, pages 11–17. J. Stefan Institute, eds. D. Mladenic and M. Grobelnik, 1999.
5. H. Ahonen-Myka. Discovery of frequent word sequences in text. In *The ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining, Imperial College, London, UK*, pages 180–189, 2002.
6. H. Ahonen-Myka, O. Heinonen, M. Klemettinen, and A. I. Verkamo. Finding Co-occurring Text Phrases by Combining Sequence and Frequent Set Discovery. In *Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI-99*

Workshop on Text Mining: Foundations, Techniques and Applications, Bled, Slovenia, pages 1–9. ed. R. Feldman, 1999.
7. B. Crémilleux, M. Gaio, J. Madelaine, and K. Zreik. Discovering browsing paths on the web. In *Third International Conference on Human-System Learning, CAPS 2000, Paris (France)*. Europia, 2000.
8. N. Fuhr, N. Goevert, G. Kazai, and M. Lalmas. Inex: Intitiative for the evaluation of xml retrieval. In *ACM SIGIR Workshop on XML and Information Retrieval, Tampere, Finland*, 2002.
9. M. Gaio, M. Smurzlo, L. Thomazo, and C. Turbout. A collaborative editing system. In *GisPlanet'98, International Conference and Exhibition on Geographic Information, Lisbon, Portugal*, 1998.