Structuring Documents from Short Texts
The Enel SpA Case Study

Silvia Calegari and Matteo Dominoni
DISCo, University of Milano-Bicocca, V.le Sarca 336 Building U14 I-20126, Milan, Italy

Keywords: XML, Structured Document, Logical Structure.

Abstract: Nowadays, structured documents are marked-up using XML. XML is the W3C standard that allows to give a meaning about the stored content of a document by the definition of its logical structure. A logical structure can be exploited to have a focused access to structured documents. For instance, in XML Information Retrieval, the logical structure is aimed at retrieving the most relevant fragments within documents as answers to queries, instead of the whole document. The problem arises when it is not possible to automatically define the logical structure of a document by using the methodologies presented in the literature. This position paper takes into account this situation and provides a possible solution adopted in the Enel SpA energy company.

1 INTRODUCTION

Documents usually are based on two main features: (1) content and (2) structure, respectively. The content refers to the textual information of a document, whereas the structure refers to the logical structure of a document that is based on the analysis of the content. There are two types of document structures, i.e., the visible structure and the invisible structure (Bradley, 2002). The former is related to the document's meaning to a human user, e.g., the organization of a book into chapters, sections, and so on. The latter is the use of tags to expose the document's meaning to a machine, e.g., the XML (eXtensible Mark-up Language) (XML, 2014) language for writing documents on the Web.

Our attention is focused on the definition of the invisible structure of documents. The use of a logical structure on documents can provide many advantages. A structured document, for example, allows us to:

- design sites rather than pages
- publish different portions of data according to the context
- navigate documents according to their meaning
- exchange focused information between systems

An increasingly common way to encode the structure is through the use of a mark-up language. The HTML (Hypertext Markup Language) (HTML, 2013) is the first mark-up language used to encode the content available on the Web. Although HTML imposes some structure on a content, this structure is used for presentation purposes and less for representing the semantics of the text (Lalmas, 2009).

Nowadays, XML is the most widely used mark-up language that allows to separate the logical structure of a document from its layout/presentation. In this way, the document logical structure can be exploited to allow focused access to documents. For instance, in the XML Information Retrieval area of research, the document structure is analyzed with the aim of improving the user's request. The aim is to return the most relevant fragments within documents, i.e. returning XML elements, such as sections and paragraphs, instead of whole documents in response to a query. Such focused strategies are suited for information repositories containing long documents, or documents covering a wide variety of topics.

In the last decade, researchers have focused on the study of algorithms and tools devoted to effectively define/access documents marked-up in XML. Since 2002, a campaign called INEX (Initiative for the Evaluation of XML Retrieval) (INEX, 2014) has been established. The goal of INEX is to provide a forum for the evaluation of approaches specifically developed for XML Information Retrieval. Researchears can access the

Calegari S. and Dominoni M..
Structuring Documents from Short Texts - The Enel SpA Case Study.
DOI: 10.5220/0005498800630068
In *Proceedings of 4th International Conference on Data Management Technologies and Applications* (DATA-2015), pages 63-68
ISBN: 978-989-758-103-8

INEX's document repository to validate their strategies. A drawback is that such collections store documents where an XML structure is already defined mainly by expert users. A challenging research topic is on the automatic definition of the invisible structure of a document.

How is it possible to automatically define an XML structure? Which techniques have been defined? This position paper gives a brief overview of some solutions presented in the literature, althougth some open issues still remain (Section 2). What happens in the case of short textual information? To automatically define an XML structure of documents from short texts is not a simple task; it means identifying its topics/sub-topics by overcoming problems of semantics (e.g., ambiguity, polisemy, etc.). The strategies of XML tag extraction are efficient only when long texts are analyzed. This position paper opens a debate on this issue and proposes a preliminary solution.

The paper is organised as follows. Section 2 gives an overview of the problem, whereas Section 3 presents a possible solution tested in the Enel SpA energy company. Finally, in Section 4 some conclusions are given.

2 OPEN ISSUE

The goal of an XML Information Retrieval system is to exploit the invisible logical structure of XML documents to retrieve XML elements, instead of whole documents, in response to a user query. This is useful in the case of information repositories containing long documents, or documents covering a wide variety of topics (e.g. books, user manuals, technical documents, etc.). In fact, if a document is retrieved by a search engine, it does not mean that the whole content of the document is relevant for a user. To discover the relevant paragraphs from long documents can be a time consuming activity. The goal is to identify the relevant content (i.e., the most useful XML elements) within a document in order to support the user search activity.

The general idea is to discover the sequence/list of sub-topical arguments that occur within one (or more) main topic(s), with the indication of the information (i.e., pages, paragraphs) where they occur. Let us consider the example reported in (Hearst, 1997), where a 21-paragraph science news article, called "Stargazers", is considered. The main topic is the existence of life on Earth and other planets. Its content has been described by the following sub-topics (the numbers on the left side indicate which paragraphs are defined):

```
   1-3 Intro – the search for life in
space
   4-5 The moon's chemical composition
   6-8 How early Earth-moon proximity
shaped the moon
   9-12 How the moon helped life evolve
on Earth
   13 Improbability of the Earth-moon
system
   14-16 Binary/trinary star systems
make life unlikely
   17-18 The low probability of non-
binary/trinary systems
   19-20 Properties of Earth's sun that
facilitate life
   21 Summary
```

The ultimate step consists in labelling their subject matter (Hearst, 1997), with the application of subtopics classification algorithms (Lewis and Hayes, 1994).

In the literature, XML Information Retrieval refers to "passage retrieval" and "structured text retrieval" (Lalmas, 2009). In passage retrieval, the aim is to decompose documents into fragments called passages. Passages are then retrieved as answers to a query. In structured text retrieval, the aim is related to the study of models for querying and retrieving from structured text (Baeza-Yates and Navarro, 1996) encoded with the use of mark-up languages, such as XML.

There are several approaches that allow to identify the relevant passages of a document, such as fixed-size text windows of words (Callan, 1994), fixed discourses such as paragraphs (Wilkinson, 1994), text-tiling through the application of a topic segmentation algorithm (Hearst, 1997), or an algorithm that finds chains of related terms via a comprehensive thesaurus (Morris and Hirst, 1991).

These strategies can be summarized into three phases: (1) tokenization into terms and sentence sized units, (2) determination of a score for each sentence sized unit, and (3) detection of sub-topic boundaries from plotting the sentence units against their score, thanks to the adoption of a thesaurus.

A drawback is that these techniques are mainly based on the co-occurrence of words (i.e., approach based on the frequency); thus, they are efficient only for the analysis of long texts. Also the text analysis using a semantic model either based on Latent Dirichlet Allocation (LDA) (Tian et al., 2010) or based on Latent Semantic Analysis (LSA) (Klein et al, 2011) are not efficient in this context as it is assumed the use of a large repository. The

LDA/LSA are "methods for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large collection of texts" (Klein et al, 2011).

What happens when a large collection is not available?

What happens in the case of short textual information?

When there is the need to analyze short texts without the support of a large repository, the identification of the corresponding logical structure can be more complicated. Some aspects have to be considered:

- sub-topic/topic overlapping
- identification of relevant sub-topic/topic
- semantic issues related to polysemy and/or homonymy

Let us consider the analysis of a text extracted from a document in a Power Point format. This particular type of document is mainly defined for each slide only by short texts or images. To apply strategies based on co-occurrence techniques for discovering sub-topics/topics is not the best approach.

There is growing interest in using these types of documents in IT industries. They are used to mainly synthetize the steps of long and complicated business processes (e.g., during the training of new technicians). The presentations of documents in the Power Point format can be very long and, generally, they are composed of many topics/subtopics described only by short texts and images. For this type of documents it can happen that humans manually define a visible logical structure of the presentation, or that the structure is automatically generated by specific software, although the main topics are not always explicitly defined. Generally, the problem arises when there is not a logical structure of the presentation, where the relevant topics/sub-topics and the corresponding pages are indicated explicitly. Let us imagine the scenario of a call center, where a (new) technician has to identify the right text (i.e., content of a slide) faster for answering a user (Section 3). In the case of a long Power Point document the technician is not able to help the user in time. In the literature novel strategies must be defined in order to establish a solution for the analysis of these particular documents.

3 A CASE STUDY

The issue is to establish a strategy in order to retrieve documents and their portions (i.e., passages) with the intent to support the user's searches. In particular, when a user has to find the exact section related to her/his information needs it becomes a time consuming activity. The criticality arises when a long document is composed of short textual descriptions, lists, images, etc. In this case, the classic methodologies applied in the literature are not efficient for defining the logical structure of such documents (see Section 2).

A real case study has been considered with Enel SpA company (Calegari et al, 2014) (Calegari and Dominoni, 2014). The CRM (Customer Relationship Management) is a crucial software product in the Enel SpA, as it is the access point to some key functionalities, such as the creation of energy contracts for new users, and the identification of relevant documents used by technicians for people's practical problems on the phone. Our attention is focused on this last case, where technicians need to discover the contents they are interested in faster and easier. In fact, technicians have a few minutes to answer a user. The problem is that the relevant documents that are useful for the solution to a specific task are very long, and technicians do not have time to read and identify the right portion of the text in 2-3 minutes during telephone assistance. The definition of a logical structure of the analyzed document, as the one reported on page 2 (Section 2), can simplify the technician's activity. Let us consider the scenario where, during a phone call, each technician has the visibility of the topics/subtopics (with the corresponding pages) of each document used to specific business processes; then, thanks to this summarization, the technician is easily supported in her/his work.

In the Enel SpA the whole set of documents used in the CRM module is called "script". A script is a document in the Power Point format made up of (about) 80-100 slides. During a phone call it is a hard task for a technician to identify the right slide in a few minutes, with the consequence that 90% of the requests processed via phone have to be analyzed off-line.

3.1 Additional Information on a "Script" Document

To obtain a logical structure of a script document, we have added some information to each slide. Our attempt is aimed at retrieving the script document related to the critical business process for assistance over the phone and the topics/sub-topics with the corresponding number of pages.

Siebel CRM 8 (Siebel, 2009) is the CRM used in Enel SpA. A new service has been integrated in the native Siebel software in order to create logical communication between the CRM and the search engine owned by Enel SpA (the search engine has been developed within the open source edition of the Liferay 6.1 C.E. portal (Liferay, 2013) by the Lucene library (Lucene, 2013)). The idea is very simple, when a technician is using the CRM for assistance over the phone, he/she in any instant can invoke the search engine to retrieve the document (and its passages) useful for her/his objectives, then the results are displayed in the CRM layout. In order to satisfy this requirement, we have developed a Web Service that is able to logically link the CRM with the search engine by specific SOAP (Simple Object Access Protocol) actions via the WSDL (Web Services Description Language) interfaces, which is the XML format for describing network services (Section 3.1.2).

To allow the search engine to identify the right document and its portions some steps have been performed. Our solution consists in three main phases (Section 3.1.1): (1) addition of specific information on each script document by expert users, (2) the indexing of these 'new' scripts documents, and (3) a procedure able to read the index in order to parse the added text in a proper way for retrieving the passages.

3.1.1 Labelling "Script" Document

Each slide of the script document is labelled with a specific syntax by expert users. When a script document is in write mode, then a specific annotation has to be added in the "page note" section as follows:

```
---BEGIN-ANNOTATION---
---section: section_name---
---title: title_name---
---slide: number---
```

where,

'section_name' is the topic;
'title_name' is the sub-topic;
'number' is the slide number.

If the fields 'section_name' or 'title_name' are empties, then the content of the new slide is related to the previous ones, where topics or sub-topics are explicitly indicated. The insertion of this specific syntax is not a time consuming activity for expert users. Indeed, these fields are filled in with the content related to the slide under examination.

Let us consider the example where a portion of the script 'Request for change in product Enel Gas' is analyzed. The slides are related to the following arguments (but not limited to): 'contractual processes' and 'customer needs', respectively. Then, the related syntax is:

```
---BEGIN-ANNOTATION---
---section: Request for change in
product Enel Gas---
---title: contractual processes---
---slide: 10---

---BEGIN-ANNOTATION---
---section: ---
---title: ---
---slide: 11---
…
---BEGIN-ANNOTATION---
---section: ---
---title: customer needs---
---slide: 25---
…
```

When the writing of a script document is finished, this document is indexed by using the Lucene library, as developed in the Liferay portal. The indexing phase is performed in a standard way without changing the native code. This assumption is possible thanks to the addition, for each slide, of the annotation syntax in the 'page note' section. For the Lucene library the text inserted in the 'page note' section is considered a textual description as well as the one inserted within a slide like content. During a phone call, a technician finds, by invoking the search engine, the script document related to the problem corresponding to the user's criticality. For each retrieved script document the output displayed is its logical structure. By considering the previous example, we have:

```
Title:Request for change in product
Enel Gas
      - contractual processes 10-24
      - customer needs 25-53
      …
```

3.1.2 Web Service: Interfaces of Communications

The link between the CRM and the search engine has been provided by the development of a Web Service via SOA (Service Oriented Architecture) technology. Then, the communication is performed by SOAP messages by using XML interfaces. In detail, the following actions 'searchScript' and 'searchScriptResponse' have been defined (see Figure 1).

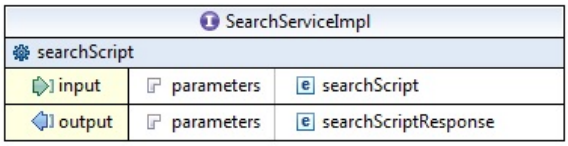

Figure 1: Web Service interface.

The 'searchScript' action sends the request (for a specific user's problem) from the client (CRM) to the provider (search engine); whereas, the 'searchScriptResponse' action sends the answer (i.e., the script document with its portions/passages) from the provider to the client.
The I/O actions satisfy a specific XML syntax. By considering the example of Section 3.1.1, we have:

INPUT

```
…
<SOAP-ENV:Body>
  <ns1:searchScript>
      <userId>a241450</userId>
      <processList>Product Enel
      Gas</processList>
  </ns1:searchScript>
</SOAP-ENV:Body>
…
```

OUTPUT

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/
soap/envelope/">
    <soap:Body>
   <ns2:searchScriptResponse
xmlns:ns2="http://searchws.kbms.org/">
        <return>
<result>
<title>Request for change in product
Enel Gas</title>
<url>http://kbms2.risorse.enel/document
s/10179/87480/
  Request+for+change+in+product+Enel+G
as<url/>
  <section>
    <paragraph>
      <title>contractual
  processes</title>
      <start>10</start>
      <end>24</end>
    </paragraph>
    <paragraph>
     <title>customer needs</title>
      <start>25</start>
      <end>53</end>
    </paragraph>
  </section>
</result>
<error>
 <description>Ok</description>
 <errorNo>0</errorNo>
 </error>
</return>
</ns2:searchScriptResponse>
</soap:Body>
</soap:Envelope>
```

The input phase is related to the user's request during telephone assistance. In detail, a technician (identified by the tag '`userId`') invokes the search engine with the following query: "Product Enel Gas". The query is encoded by the XML syntax defined in the Web Service interface and sent to the search engine. Then the index file is analyzed to retrieve the script documents related to the query. The list of results are encoded according to the XML syntax of the Web Service interface and sent to the CRM. The script document is identified by its title and url (see tags '`title`' and '`url`'), whereas its passages are identified by the tag '`paragraph`'. The definition of the searchScriptResponse in the XML is realized thanks to the annotation presented in Section 3.1.1. In fact, when a script document is considered as relevant for the query, then its content is parsed by identifying the text related to the information of its structure. The tag '`error`' allows to monitor eventual communications problems.
At the end, the XML answer will be parsed and displayed to the technician as presented in Section 3.1.1.

4 CONCLUSIONS

This position paper aims to open a debate related to the definition of the logical structure of a document. A logical structure gives a meaning about the content of a document, and it can be exploited to retrieve fragments of text, instead of the whole document. The problem arises when the standard methodologies, based on frequency analysis or based on LDA/LSA methods, presented in the literature are not efficient for identifying portions of the document, especially in case of the absence of a large collection. It can happen when long documents contain short textual information: it becomes very difficult to establish the topic/sub-topic of the document without having problems of semantics, overlapping content, etc.
In this position paper a preliminary solution has been presented by considering Power Point documents. This type of document is mainly written by short texts; thus, it can be a good candidate for testing the effectiveness of our preliminary approach. We have defined a very simple syntax that can be added during the writing of each slide. The

Power Point document is then indexed with standard methodologies to be available for a search engine.
This solution is nontrivial, but it is a first attempt to solve this complex problem for a real case study where a positive judgement from users has been obtained. In fact, this solution has been tested with Enel SpA energy company, where encouraging preliminary results have been obtained. The scenario considered is a call center, where a technician has to identify the right content in a few minutes during telephone assistance. At the moment, we have conducted some interviews with technicians who are using the new service that implements our solution, receiving positive feedback. In the future, we are planning to validate the preliminary positive feedback with robust analysis.
The work presented in this position paper could open a debate in order to establish new methodologies aimed at solving the problem related to the automatic definition of a structure from documents made up of short textual descriptions, titles, images, etc. The difficulties arise when the standard approaches (based on the frequency of text analysis or based on LDA/LSA methods) presented in the literature are not efficient for this specific task. A discussion to improve the preliminary attempt presented in this position paper that is based on a specific syntax to annotate this particular type of documents could stimulate the definition of new methodologies.

ACKNOWLEDGEMENTS

This paper was written within the Enel SpA Project, and we wish to thank all the people who worked with us on the development of the software.

REFERENCES

Baeza-Yates, R. A., Navarro, G., 1996. Integrating contents and structure in text retrieval. In Newsletter ACM SIGMOD Record, Volume 25, Issue 1, ACM New York, NY, USA, 67–79.

Bradley, N., 2002. *The book*, The XML companion, 3rd edition, In Pearson Education limited.

Calegari, S., Dominoni, M., Panzeri, E., 2014. Towards the Design of an Advanced Knowledge-Based Portal for Enterprises: The KBMS 2.0 Project. In *Proceedings of the 27th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems*, Part II, IEA/AIE, LNCS, Springer, VOLUME 8482, ISBN 978-3-31907466-5, Kaohsiung, Taiwan, pp. 58-67.

Calegari, S., Dominoni, M., 2014. Modeling Ontology-based User Profiles from Company Knowledge. In *Proceedings of the 6th International Conference on Advances in Databases, Knowledge, and Data Applications,* DBKDA 2014, ISBN 978-1-61208-334-6, IARIA, Chamonix, France, pp. 26-29.

Callan, J., 1994. Passage-level evidence in document retrieval. In Proceedings of the 17th annual *International ACM SIGIR conference on Research and development in Information Retrieval, Springer-Verlag New York*, Inc., 302–310.

Hearst, M., 1997. TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages. *In Journal Computational Linguistics. Volume 23,* Issue 1, MIT Press Cambridge, MA, USA, 33–64.

HTML, 2013. http://www.w3.org/html/

INEX, 2014. https://inex.mmci.uni-saarland.de/

Klein, R., Kyrilov, A., Tokman, M., 2011. Automated assessment of short free-text responses in computer science using latent semantic analysis. *In Proceedings of the 16th annual joint conference on Innovation and technology in computer science education (ITiCSE '11). ACM, New York,* NY, USA, 158-162.

Lalmas, M., 2009. *The book*, XML Information Retrieval, In Encyclopedia of Library and Information Sciences. Taylor and Francis Group.

Lewis, D.D., Hayes, P.J., 1994. Special issue of ACM: *Transactions on Information Systems on text categorization, Volume 12,* Issue 1, ACM New York, NY, USA.

Liferay, 2013. http://www.liferay.com.

Lucene, 2013. http://lucene.apache.org/core/

Morris, J., Hirst, G., 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *In Journal Computational Linguistics, Volume 17, Issue 1, MIT Press Cambridge, MA, USA*, 21-48.

Siebel, 2009. http://www.oracle.com/partners/en/knowledge-zone/applications/siebel/default-329117.html.

Tian, Y., Wang, W., Wang, X., Rao, J., Chen, C., Ma, J., 2010. Topic detection and organization of mobile text messages. *In Proceedings of the 19th ACM international conference on Information and knowledge management* (CIKM '10). ACM, New York, NY, USA, 1877-1880.

Wilkinson, R., 1994. Effective retrieval of structured documents. *In Proceedings of the 17th annual International ACM SIGIR conference on Research and development in Information Retrieval, Springer-Verlag New York*, Inc., 311–317.

XML, 2014. www.w3.org/XML/