Using Semantic Annotations for Automatic Hypertext Link Generation in Scientific Texts

Massimo Melucci
University of Padova
Padova, Italy

Joe Rehder
NASA Langley Research Center
Hampton, VA, USA

Abstract

The paper illustrates the work-in-progress for the use of ontology-based annotations to enrich the description of document contents and thus enhance the effectiveness of automatic hyperlink generation for scientific text retrieval. We present a methodology based on the vector-space model. Then, we describe the architecture of a prototype that implements automatic hypertext link generation in scientific texts.

Introduction

Scientists are avid users of information retrieval systems and use all the available search capabilities to increase recall and precision.[1] Since the late Nineties, it has been recognized that the integration of hypertext link navigation and free text retrieval permits end users to more effectively satisfy their information needs. This method is especially true for scientists, who are inherently curious and willing to follow hyperlinks to other information. The large amount of available text requires that these hyperlinks be generated automatically (Agosti & Melucci 2000; Allan 1997).

Statistical methods are often used for hyperlink generation because they are efficient. Though they differ to some extent, many of these algorithms compute a similarity function whose arguments are the representations of the texts being linked. These representations use “bag of words” (BoWs) which assigns a set of keywords to each text segment and a hyperlink is generated if high similarity, i.e. set overlap, occurs where similarity is computed using set operators. The BoW representation can be effectively applied across several domains because the knowledge about word meaning is supposed to be absent. However, we would like to be able to not link text fragments that are semantically distant but have keywords in common. In addition, we would like to be able to link text fragments that are semantically close but have no keywords in common. For example, a user of a document base on Mars exploration may be looking at some results regarding the amounts of a particular mineral in the Martian soil and would like to find similar information about another mineral in another planet. Statistical methods would not be able to generate a hyperlink between the two text fragments because the names of the minerals and planets are different. And conventional BoW representations cannot address at all the issue of generating hyperlinks between semantically similar texts that contain no keywords in common.

The advent of Semantic Web technologies makes the representation of semantics for keyword disambiguation and for non-keyword similarity matching feasible, and, hopefully more effective. Ontologies are particularly useful in the case of scientific texts because in any given scientific domain there is generally a common set of agreed definitions and relationships upon which an ontology can be developed. Most of the approaches to integrate Semantic Web technologies and information retrieval lies on indexing and coordinate tags (Shah *et al.* 2002). Annotations [2] can provide a means to deal with keyword ambiguity because they aim at labeling each keyword with ontology classes and thus describing the meaning of words. In this way, annotations make hyperlink generation algorithms capable to deal with meaning mismatch. An algorithm aiming at automatically computing hyperlinks using the BoW representation being extended with annotations can more effectively “understand” if keyword matching as well corresponds to meaning matching. For example, the text A=$\{u, v\}$ is assessed as more similar to B=$\{u, v\}$ than to C=$\{x, y\}$, if set intersection is used as similarity function, even though $u \in$ A, or $v \in$ A, have a different meaning from the same words in B, or x or y have the same meaning as of u or v. If annotations are available, A can be assessed as more similar to C than to B because, for example, the meaning of $u \in$ A is very similar to that of x and is different from that of $u \in$ B.

We propose an extension to the BoW representation in which the original texts are enriched with ontology-based annotations that describe the semantic characteristics of the text. Information about the ontologies and the classes in those ontologies are added to the BoW representation. In the second part of the paper, we describe the work-in-progress to implement a prototype that automatically generate hyperlinks enriched by annotations.

[1] Recall is the fraction of relevant documents that are retrieved, and precision is the fraction of retrieved documents that are actually relevant.

[2] In this paper, annotations are assertions about the membership of text fragments to ontologies classes.

Enhancing the Vector Space Model with Annotations

The vector-space model (VSM) for information retrieval is the most known model based on the BoW representation. The diffusion of the VSM can be explained by retrieval effectiveness, which has been demonstrate by several experiments, as well as by the possibility of easily mapping the notion of vector to that of array, as meant in programming languages. This easy mapping seems to be resulted from the fact that the VSM has not been taken seriously in the information retrieval context – for example, term vector dimensions are very often assumed as orthogonal thus ignoring the semantic relationships among terms. Nevetheless, the VSM can be taken more seriously (Wong & Raghavan 1984). It provides useful methodological tools to face the problem of enriching hyperlink generation with annotations.

Using the VSM, each textual object, e.g. document or query, is described as a vector of scalars that belongs to the subspace spanned by the keyword vectors, which can be thought as a basis for the space. The existence of a vector space implies that we have a system with the linear properties, e.g. the ability to add two (keyword or document) vectors to obtain a new (keyword or document) vector (Salton, Wong, & Yang 1975; Wong & Raghavan 1984). Thus document d is described as

$$\underline{d} = \sum_{i=1}^{n} d_i \underline{t}_i$$

which is the linear combination of n keyword vectors. The scalar product, which we suppose is the measure of correlation between two vectors, of $\underline{d}$ and $\underline{q}$ is

$$\underline{d} \cdot \underline{q} = \sum_{i=1}^{n} \sum_{j=1}^{n} d_i q_j \underline{t}_i \cdot \underline{t}_j$$

Starting from the VSM, different approaches can be designed to represent the meaning of the keywords. These approaches would aim at representing the existence of ontology classes and class relationships.

The first approach would be directly provided by the VSM definition itself. The main requirement that the $\underline{t}_i$s must satisfy is mutual independence, i.e. any keyword vector is not a linear combinations of the others. However, it may be that the correlation $\underline{t}_i \cdot \underline{t}_j$ is not null, thus revealing a relationships between the keywords. In particular, an ontology can be used so that

$$\underline{t}_i \cdot \underline{t}_j \quad \begin{cases} > 0 & \text{keywords are directly related} \\ = 0 & \text{keywords are unrelated} \\ < 0 & \text{keywords are inversely related} \end{cases}$$

For example, two keywords are directly related if they often co-occur in the same classes. Thus, if the document at which the hyperlink is anchored contains a keyword that do not occur in the destination document, a hyperlink can be anyway generated if the former keyword is directly related with a different one that occur in the destination document.

Another approach would consider keywords as linear combinations of classes, whereas class correlations measure inter-relationships. Accordingly to this approach

$$\underline{t}_j = \sum_{k=1}^{K} t_{jk} \underline{c}_k$$

where $\underline{c}_k$ is the k-th class vector. After few linear transformation, $\underline{d} = \sum_k a_k \underline{c}_k$, where $a_k = \sum_j d_j t_{jk}$. The correlation, which we suppose can be used to generate hyperlinks, between two documents d', d'' becomes

$$\underline{d}' \cdot \underline{d}'' = \sum_h \sum_k a'_k a''_h \underline{c}_h \cdot \underline{c}_k$$

The information about the ontology can be mapped to the correlation matrix of the class vectors. For example, two classes c_h, c_k are directly related if c_h contains many keywords of c_k and viceversa.

The investigation of these two approaches is underway, whereas we studied in detail the one we describe in the following Section. The approach considers documents as vectors of sets of classes, i.e. each document keyword is a set of the classes to which it is an element. At present, class relationships are kept apart yet we will consider them at a later step.

An Approach for Hyperlink Generation using Annotations

Each text is assigned a vector of keywords, like the standard VSM. Differently from the VSM, each text keyword is assigned a set of classes, each class describing a meaning of the keyword within the text. Conversely, each text can be assigned a set of class and each class is assigned the set of text keywords belonging to it. As the set of classes corresponding to a keyword can be described as a vector, the result is a cube where the three dimensions correspond to the documents, the keywords and the classes.

This approach differs from others because diverse annotators or ontologies can be employed. Thus keywords are labelled with class names depending on the *document* within which they occur. The dependency of classes on the document, within which the keyword that is element of them occurs, is due to the possibility that the annotator exploits some data related to the document and to the keyword, such as the context provided by the text "window" surrounding the keyword. Thus, different classes can be detected within one document depending on the "windows" containing the keyword. Moreover, a keyword occurring within a document may belong to more than one class if different ontologies and/or annotators are used to annotate the document. An example is displayed in the following matrix.

$$\begin{array}{c|ccc} & w_1 & w_2 & w_3 \\ \hline \underline{d}_1 & \{c_1\} & \emptyset & \{c_1, c_2\} \\ \underline{d}_2 & \emptyset & \{c_2, c_3\} & \emptyset \\ \underline{d}_3 & \{c_2\} & \{c_2\} & \{c_1, c_2, c_3\} \\ \underline{d}_4 & \{c_1\} & \{c_2, c_3\} & \{c_2\} \end{array} \quad = \quad \mathbf{C}$$

The symbol $\emptyset$ at position i, j means that w_j does not occur in $\underline{d}_i$, i.e. class could be assigned. Similarly to preceding matrix, for each document, one set of keywords is assigned to each occurring class (column), as follows:

$$\begin{array}{c|ccc} & c_1 & c_2 & c_3 \\ \hline \underline{d}_1 & \{w_1, w_3\} & \{w_3\} & \emptyset \\ \underline{d}_2 & \emptyset & \{w_2\} & \{w_3\} \\ \underline{d}_3 & \{w_3\} & \{w_1, w_2, w_3\} & \{w_3\} \\ \underline{d}_4 & \{w_1\} & \{w_2, w_3\} & \{w_2\} \end{array} \quad = \quad \mathbf{W}$$

The symbol $\emptyset$ at position i, j means that no keyword belonging to the class c_j occurs in $\underline{d}_i$; if c_j describe the content of $\underline{d}_i$, then there exists at least one keyword that must be present. Note that one can easily map $\mathbf{C}$ to $\mathbf{W}$, and viceversa, without losing information, yet we have preferred to keep them as distinct for computational reasons and for sake of clarity. The role played by $\mathbf{C}$ is explained in the following.

Using the classical VSM, the similarity between two texts $\underline{d}_i$ and $\underline{d}_j$ is as higher as the number of common keywords is higher. In particular, keyword mismatch reduces similarity; in the example, w_1 would not contribute to the similarity between $\underline{d}_2$ and $\underline{d}_3$, whereas it increases the similarity between $\underline{d}_1$ and $\underline{d}_3$, yet with different meanings.

After extending the classical VSM as previously explained, the similarity is still as higher as the number of common keywords is higher, but also is as higher as the number of common meanings is higher, and decreases as the number of common meanings is lower. Thus, the contribution of w_1 to the similarity between $\underline{d}_1$ and $\underline{d}_4$ is higher than its contribution to the similarity between $\underline{d}_3$ and $\underline{d}_4$, because the classes of w_1 match when comparing $\underline{d}_1$ and $\underline{d}_4$, whereas they do not when comparing $\underline{d}_3$ and $\underline{d}_4$. Similarly, class c_1 contribute to the similarity between $\underline{d}_3$ and $\underline{d}_4$ because w_1 and w_3 are both members of c_1.

The similarity between $\underline{d}_i$ and $\underline{d}_j$ might be implemented for example by the following function

$$s(\underline{d}_i, \underline{d}_j) \equiv \sum_{k=1}^{K} c_{ik} \cdot c_{jk} + \sum_{h=1}^{H} w_{ih} \cdot w_{jh} \qquad (1)$$

where K is the number of keywords and H is the number of classes – this is an example of composite matching (see (Salton & McGill 1983)). The first member of 1 might reduce the contribution of keywords co-occurring in both documents but with different meanings – the reduction happens once the keyword occurring in $\underline{d}_i$ belongs to classes being different from the classes to which the keyword occurring in $\underline{d}_j$ belongs. The second member might increase the contribution of mismatching keywords but with similar meanings – the increase happens once the classes of the keywords occurring in $\underline{d}_i$ overlap with those of the keywords occurring in $\underline{d}_j$. The values reported below can be normalized using a cosine-like formula, i.e.

$$\frac{s(\underline{d}_i, \underline{d}_j)}{\sqrt{s(\underline{d}_i, \underline{d}_i)}\sqrt{s(\underline{d}_j, \underline{d}_j)}}$$

to make s independent of text length and of the size of the keyword sets and class sets. As regards to the parameters,

$$c_{ij} \cdot c_{kj} = \left\{ \begin{array}{cl} 1 & c_{ij} \neq \emptyset \wedge c_{kj} \neq \emptyset \wedge c_{ij} = c_{kj} \\ \alpha < 1 & c_{ij} \neq \emptyset \wedge c_{kj} \neq \emptyset \wedge c_{ij} \neq c_{kj} \\ 0 & c_{ij} = \emptyset \vee c_{kj} = \emptyset \end{array} \right.$$

and

$$w_{ij} \cdot w_{kj} = \left\{ \begin{array}{cl} 1 & w_{ij} \neq \emptyset \wedge w_{kj} \neq \emptyset \wedge w_{ij} = w_{kj} \\ \beta < 1 & w_{ij} \neq \emptyset \wedge w_{kj} \neq \emptyset \wedge w_{ij} \neq w_{kj} \\ 0 & w_{ij} = \emptyset \vee w_{kj} = \emptyset \end{array} \right.$$

The parameters α and β measures the degree of keyword mismatch and thus mitigate the effects of keyword ambiguity. In particular, α reduces the contribution of a keyword that occurs in both texts, but that belongs to different classes. Similarly, β gives a measure of the degree to which two texts are about a meaning even if the meaning is not evident by the same keywords in both texts. The computation of s for the example is reported in the following:

	$\underline{d}_1$	$\underline{d}_2$	$\underline{d}_3$	$\underline{d}_4$
$\underline{d}_1$	1	$\beta/2\sqrt{3}$	$(\alpha+\beta)/\sqrt{6}$	$(1+\alpha+2\beta)/2\sqrt{6}$
$\underline{d}_2$		1	$(1+\alpha+\beta)/3\sqrt{2}$	$(1+2\beta)/3\sqrt{2}$
$\underline{d}_3$			1	$(\alpha+\beta)/2$
$\underline{d}_4$				1

Note that $s(\underline{d}_1, \underline{d}_2)$ would have been 0 if a classical VSM were been used because every keyword of $\underline{d}_1$ does not occur in $\underline{d}_2$, and viceversa. The use of classes and an extended VSM permit to increase $s(\underline{d}_1, \underline{d}_2)$ in order to capture the fact that there is a common class, i.e. yet c_2 contains different keywords, i.e. w_2 and w_3. Also, $s(\underline{d}_3, \underline{d}_4)$ would have been 1 if a classical VSM were been used even if the co-occurring keywords had different meanings. The use of classes and an extended VSM permit to reduce $s(\underline{d}_3, \underline{d}_4)$ because the co-occurring keywords belong to different classes.

To make the computation of α and β automatic and independent of end user, their estimation can exploit the data being available from the matrices $\mathbf{C}$ and $\mathbf{W}$ in order to estimate the degree to which a co-occurring keyword belongs to related classes (α), and the degree to which a co-occurring class includes the same keywords (β). If $c_{ij} \neq \emptyset \wedge c_{kj} \neq \emptyset$, such an estimator for α should be $\hat{\alpha} = 1$ if $c_{ij} = c_{kj}$, whereas $0 < \hat{\alpha} < 1$ if $c_{ij} \neq c_{kj}$. A possible definition that satisfies the constraints is

$$\hat{\alpha}_{ikj} = \frac{|c_{ij}|}{|c_{ij} \cup c_{kj}|} \qquad \text{and similarly} \qquad \hat{\beta}_{ikj} = \frac{|w_{ij}|}{|w_{ij} \cup w_{kj}|}$$

Thus, the parameters changes with the matched texts and with keywords, and the table for the example is re-written as follows:

	$\underline{d}_1$	$\underline{d}_2$	$\underline{d}_3$	$\underline{d}_4$
$\underline{d}_1$	1	$1/4\sqrt{3}$	$1/\sqrt{6}$	$5/4\sqrt{6}$
$\underline{d}_2$	$1/4\sqrt{3}$	1	$11/18\sqrt{2}$	$2/3\sqrt{2}$
$\underline{d}_3$	$2/3\sqrt{6}$	$5/6\sqrt{2}$	1	$1/2$
$\underline{d}_4$	$3/2\sqrt{6}$	$5/2\sqrt{2}$	$7/12$	1

Note that the table is no longer symmetric because the estimators are not, e.g. $\hat{\alpha}_{ikj} \neq \hat{\alpha}_{kij}$.

As previously stressed, the model is an extension of the vector space model – the difference is that the vector components are sets and there are two types of component. The common feature is the computational complexity of s which is the same as that of common retrieval functions, and in particular is linear with the product of the number of keywords of the text from which the hyperlink begins by the number of keywords of the text to which the hyperlink ends.

The Prototype

The architecture of a prototype that provides automatic hyperlink generation based on knowledge about the domains to which the documents to be linked is depicted in Figure 1. A *converter* translates a document written in a widely used file format, e.g. Microsoft Word, PDF, text, to the corresponding XML format; a *text segmenter* takes a XML document as input and gives a list of sentences or phrases that are likely to describe the document topics, as output; an *annotator* takes an ontology and a XML document or, alternatively, a list of sentences or phrases that describe the document topics, as input, and gives a list of Resource Framework Description (RDF) assertions as output to represent annotations; a *database management system* (DBMS) with IR functionalities stores documents and annotations as (relational) tables, indexes and retrieves them using SQL or free-text; a *linking engine* takes an annotated XML document as input, filters annotations out the document, exploits these annotations to access the DBMS, retrieves related annotations from the database, computes the similarity values, and generates hyperlinks.

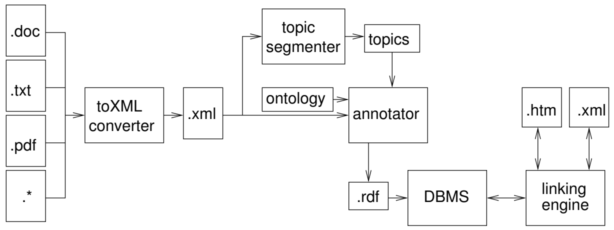

Figure 1: The general software architecture of the prototype.

As the annotations are coupled and originate from an ontology given as input to the annotator, the linking engine is provided with data about the semantics of the hyperlinks. Then, the annotator is able to label the hyperlinks with some words describing the type of relationship; for example, if the hyperlink connects two documents both including a number of entities belonging to the class Organization, the annotation might argue that the hyperlink is about organizations accordingly to the semantics given by the ontology.

When accessing a document, a user can ask the prototype to generate the hyperlinks to the documents that are assessed as semantically related to the current one. Alternatively, the user can highlight a sentence, a phrase, or more generally, a text fragment and ask the prototype to generate the hyperlinks to the documents that are assessed as semantically related to the highlighted fragment. Another scenario is one in which a software agent is navigating a Web of documents and accesses to one of these. To carry navigation on, the agent needs to know which are the documents that are about the current one and that are the semantically closest. The linking engine takes as input the accessed document, retrieves from the DBMS the related annotations, exploits the retrieved tuples to compute the similarity values, and produces a XML or HTML file with the hyperlinks to the linked texts and/or the documents.

At the current stage of design, we think that the most economical process is to re-use as much available software as possible. This design choice permits us to concentrate on the core of our work, i.e. the automatic generation of hyperlinks using knowledge about the semantics of the documents. Therefore, we are using AeroDAML (`http://www.daml.org/tools/#AeroDAML`) to implement the annotator. From its homepage, “Automatically generates basic DAML annotation/markup from text and webpages. The web version is oriented toward novice/infrequent DAML annotators; the client/server version is oriented toward personnel who routinely produce documents or need to annotate legacy documents.”. The latter version can be used as annotation engine for our prototype. The role of DBMS is played by MySQL (`http://www.mysql.com/`) to store, index and retrieve annotations. This DBMS implements the relational data model and is based on ANSI SQL 1992 but a number of extensions or differences. It provides some basic indexing and retrieval of full-text, other than the traditional relational operations to access the annotations using tags.

Future Work

We have illustrated the work in progress on the automatic generation of hyperlinks using ontology-based annotations. The aim is to overcome the problem of ambiguity affecting the classical statistical information retrieval models based on the notion of keyword co-occurrence. We have illustrated a methodology to incorporate annotations in a vector space-like model and then described the architecture of the prototype being in construction that implements the methodology.

In the future, we will further investigate the previously outlined VSM-based approaches to enrich the BoW representation with semantic annotations. We plan to carry out experiments using data from the Text Retrieval Conference. Moreover, we will complete the prototype and show how hyperlink generation and navigation can work. At present, the database includes several hundred full text documents concerning Mars exploration missions described using also graphs and images. The development of a Mars exploration ontology is underway.

References

Agosti, M., and Melucci, M. 2000. Information retrieval techniques for the automatic construction of hypertext. In Kent, A., ed., *Encyclopedia of Information Science*, volume 66. New York: Marcel Dekker. 139–172.

Allan, J. 1997. Building hypertexts using information retrieval. *Information Processing & Management* 33(2):145–159.

Salton, G., and McGill, M. 1983. *Introduction to modern Information Retrieval*. McGraw-Hill, New York, NY.

Salton, G.; Wong, A.; and Yang, C. S. 1975. A vector space model for automatic indexing. *Communications of the ACM* 18(11):613–620.

Shah, U.; Finin, T.; Joshi, A.; Cost, R.; and Mayfield, J. 2002. Information retrieval on the Semantic Web. In

Proceedings of the ACM Conference on Information and Knowledge Management (CIKM), 461–468.

Wong, S., and Raghavan, V. 1984. Vector Space Model of Information Retrieval – A reevaluation. In *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, 167–185.