Combining Text Mining and Information Retrieval Techniques for Enhanced Access to Statistical Data on the Web: A Preliminary Report

Martin Rajman[1] and Martin Vesely[1,2]

[1] École Polytechnique Fédérale de Lausanne, EPFL Lausanne, Switzerland
{Martin.Rajman, Martin.Vesely}@epfl.ch
[2] CERN, Conseil Européen pour la Recherche Nucléaire
Martin.Vesely@cern.ch

Abstract. In this contribution, we present the StatSearch prototype, a search engine that enables an enhanced access to domain specific data available on the Web. The StatSearch engine proposes a hybrid search interface combining query-based search with automated navigation through a tree-like hierarchical structure. The goal of such an interface is to allow a more natural and intuitive control over the information access process, thus improving the speed and quality of the access to information.

An algorithm for automated navigation is proposed that requires natural language pre-processing of the documents, including language identification, tokenization, Part-of-Speech (PoS) tagging, lemmatization and entity extraction. Structural transformation of the available data collection is also performed to reorganize the nodes in the information space (the Web site) from a graph into a tree-like hierarchical structure. This structural pre-processing (transformation of a graph structure into a tree-like hierarchy) can be done either by document clustering, or, alternatively, derived from existing structure of the document collection by splitting, shifting, or merging of nodes where necessary. The clustering approach is more straightforward but requires that the intermediate nodes in the created tree are assigned understandable descriptions, which corresponds to a difficult task.

Target documents are represented by weighted lexical profiles the components of which correspond to triples of the form (surface form, lemma, PoS). The extracted and normalized terms and entities are weighted using the TF.IDF weighting scheme. Document relevance is computed as the textual similarity between the query and document profiles. Several well known similarity functions from the field of information retrieval have been tested, including the Cosine and Okapi BM25 similarity measures. In addition to the similarity score, the contributions of all the query terms to the computed document similarities are also provided.

The principle of the presented algorithm for automated navigation is to compute a score distribution on the documents (leaves of the tree), and to propagate the obtained scores upwards in the tree. The node scores are then used to guide a

M. Rajman et al.: *Combining Text Mining and Information Retrieval Techniques for Enhanced Access to Statistical Data on the Web: A Preliminary Report*, StudFuzz **185**, 213–222 (2005)
www.springerlink.com

faster, partially automatic, downward navigation in the tree. In particular, user intervention for node selection is only required for nodes with children corresponding to a score distribution where no clearly good candidate can be identified. Otherwise, the (possible partial) traversal of the tree is performed automatically. Several approaches are compared for the automation of the navigation. They include decision rules based on relative (resp. absolute) minimum best score differences, as well as on information theoretic measures. The automated navigation algorithm also allows a more reliable document ranking by giving to the user the possibility to restrict the search to the set of documents dominated by a specific node or to the documents matching a limited set of document types.

The presented hybrid search technique has been implemented in the StatSearch prototype that has been realized in collaboration between EPFL, Statistics Sweden (SCB), and CERN, in the framework of the NEMIS network of excellence. The prototype focuses on domain of official statistics, and currently uses a database of over 5000 full text documents, tables and graphs in English accessible at the SCB Web site.

1 Introduction

Information access is understood as a process of identification and presentation of information that corresponds to user information need expressed by a user query. Since both user queries and document representations are typically based on sentences in natural language, in order to address the performance of the information access we focused on combination of the text mining (TM) and the information retrieval (IR) techniques.

In order to demonstrate our approach we have done a case study on the Web access to information in the domain of statistics. We have developed a prototype of an information access tool integrating querying and navigation. The StatSearch prototype has been implemented and tested on statistical documents provided by Statistics Sweden (SCB) amounting to over 5000 items in English. The SCB Web site can be effectively accessed with this parallel interface that integrates more information about the Web site structure that is not necessarily known by users.

2 Access to Domain Specific Data

Domain specific data introduces several issues to be addressed. Among these the most important ones are (i) the use of domain specific vocabulary, (ii) existing specific metadata, (iii) various information presentations and (iv) various user backgrounds that scale from unexperienced users to highly experienced ones represented by domain specialists and experts [3].

Domain specific vocabulary of statictics has been a subject of research of several initiatives. Several relevant resources might be mentioned as a base for document processing of statistical data. Namely one should cite the ISI Glossary of statistical terms[1], Statistical Data and Metadata eXchange (SDMX)[2] and Metadata Common Vocabulary (MCV).

Statistical information is provided to end user in a document in a predefined form. According to this presentation form we distinguished several document types, most importantly the statistical messages (publications), press releases, statistical database forms, domain portals and individual tables or charts. Domain specific information needs can be categorized and most frequently solved tasks can be identified. These tasks are very often triggered by expected or unexpected events. As an example let us mention documents published by authorities, elections, political declarations, etc. Typical users of statistical information could also be profiled.

In order to reveal the particularities of the domain of statistics we have undertaken a few interviews with the domain specialists at SCB[3]. In summary, we have gathered the following information relevant for our case study:

- Information at SCB is usually requested via Web, other options include requests by e-mail or by phone.
- Typical user profiles correspond to users from academics and research (ca. 40%), users with business background (ca. 35%), journalists and students.
- Majority of requests suffer from ambiguities and have to be often refined by interaction with the user. Often the communication happens to switch to personal mailbox.
- Requests relevant to SCB are redistributed internally to competent specialists, other requests are re-directed to partner institutions. SCB collaborates with another 25 governmental organizations that are entitled to provide official statistics in Sweden (for full list see Appendix A). Other organizations may potentially be contacted, these include banks or research institutes.
- 90% of queries/documents relate to the Swedish national statistics, remaining 10% relate to international statistics.

3 StatSearch Prototype

The access to the domain specific information is demonstrated on a prototype that we implemented with regard to the mentioned requirements of domain specific data. This prototype can be regarded as NLP-based search engine focusing of efficient combination of querying and navigation features of the information access process. The prototype builds upon the work presented

[1] http://europa.eu.int/comm/eurostat/research/

[2] http://www.sdmx.org/

[3] We focused on the domains of National Accounts, Citizen Influence, Labour Market and Prices and Consumption

in [2] focusing on document processing, textual similarity computation, automated navigation and the user interface optimization compliant with the human-computer interaction principles.

3.1 Document Processing

Documents are pre-processed using several NLP techniques in order to obtain semantically coherent document representations and features included in this representations are then indexed. Since we worked with the Web-based documents, we focused on the information extraction from HTML files. The feature selection is done in compliancy with following criteria: (i) feature appears in a relevant HTML tag, (ii) the value of the $tf.idf$ weight is important and (iii) feature belongs to semantically relevant morphological category.

Both original and lemmatized forms of words are used to build the document profile as the document representation. Canonical forms of features were obtained by morphological normalization, language identification and data cleaning.

Morphological Normalization

In order to filter out features with semantically irrelevant PoS with regard to the document representation purpose and we kept only words that belong to morpho-syntactic category of an adjective and a substantive[4]. We also tried to identify word compounds based on their co-occurencies that are then treated both as one feature and as separate features of individual words.

The extracted vocabulary amounts to 2346 non-canonical content bearing words extracted from titles and additional 5574 non-canonical content bearing words extracted from the rest of the document.

Metadata Extraction

According to the principle that content related features should be separated from the ones related with the nature of the documents, selected metadata such as document type or time/space relevance was extracted. Extracted metadata are then used as filtering feature during the information access process. Filtering can be tuned so the documents are either excluded from the result set or shifted in the displayed document rank.

Language Identification

At the feature extraction step we have encountered the need for language identification on the term level. Indeed, documents presented on the Web pages

[4] We used the FreeLing tagger from the UPC of Barcelona (http://www.lsi.upc.es/~nlp/freeling/) and the sylex tagger (http://issun17.unige.ch/sylex/intro.html)

often contain multi-lingual content and since both lemmatization and PoS tagging are language dependent tasks, introducing the language identification step was implied.

Language identification is done for individual terms based on the trigram technique developed previously at Rank Xerox Research Centre (RXRC) France as described by [4]. Since we identify language only for individual terms we have only analyzed trigrams composed of alphanumeric characters, i.e. omitting spaces and puctuation characters. The initial language set contained English and Swedish which were the languages that we needed to distinguish between, where corresponding frequency tables were derived from a text corpus based on the Electronics Texts Center Collections. The Electronic Text Center's holdings include approximately 70,000 on- and off-line humanities texts in thirteen languages, with more than 350,000 related images (book illustrations, covers, manuscripts, newspaper pages, page images of Special Collections books, museum objects, etc.) For our purposes we selected and analyzed ca. 3 Million english words. http://etext.lib.virginia.edu/. Frequency tables were optimized by selection of characteristic (most frequent) trigrams.

As pointed out, in our case we only distinguished between two languages – English and Swedish. Correlation of most frequent trigrams in corpus with the word trigrams was computed as Cramer's Phi (V) coefficient based on the chi-square statistic.

3.2 Textual Similarity Computation

Textual similarity computation is based on coefficients that are traditionally used in the field of IR. We considered a variety of similarity measures, namely the Cosine similarity measure as well as the Jaccard, modified Jaccard and Dice coefficients. The Cosine similarity is currently used as the principal measure:

$$sim(q,D) = \frac{\sum W_{iq} \times W_{iD}}{\sqrt{\sum W_{iq}^2} \times \sqrt{\sum W_{iD}^2}}$$

We have used boolean weighting of document and query vectors. In this scheme the weighting is applied as an additional feature selection filter removing features that do not score well enough in terms of the *tf.idf* measure:

$$w_{iD} = \frac{tf_i}{tf_{\max}} \times \log_N \left(\frac{N}{d_i}\right)$$

The feature selection can either be done so N best features are kept or, since w_i is normalized, by setting a *tf.idf* threshold. We also started to experiment with various weighting schemes that could increase the feature selection performance, such as the Okapi BM25 weighting [5] and its variants.

Keyword Relevance

Individual keywords in the user query and the document profile have different contribution to the computed query-document similarity. That is, we observe how much does the computed query-document similarity changes when particular keyword is left out from the query. More precisely, we compute the query keyword relevance to a document or a document category where the document category profile may be composed as a sum of all underlying document profiles or equalled to the most representative underlying document profile.

The contribution to the similarity may be negative when the query keyword does not appear in the document at all, lowering the computed similarity[5]

Negative contributions of keyword to the similarity are understood as zero keyword relevance. Positive contributions of keyword to the similarity then are normalized. We have experimented with the following approaches computing the contribution of individual keywords: (i) relative contribution of keyword to similarity, (ii) absolute contribution of keyword to similarity, (iii) similarity of keyword to document. We used the relative contribution of keyword to similarity that has been computed as follows:

$$KR(k) = 0 \text{ if } sim(q,D) < sim(q\backslash\{k\},D)$$

$$KR(k) = 1 - \frac{sim(q\backslash\{k\},D)}{sim(q,D)} \text{ otherwise}$$

Alternatively, the similarity of keyword to document could be used directly providing a more discriminative measure:

$$KR(k) = sim(\{k\},D)$$

In a simple example the contribution of each of the relevant keywords in $q = \{1,1,1,0\}$ to the Cosine similarity to the document represented by $D = \{1,1,1,1\}$ are the following:

KR	KR{k}
Relative keyword relevance	0.23
Similarity of keyword to document	0.58

[5] For example having $q = \{1,1,1\}$ and $D = \{1,1,1\}$ we have $sim(q,D) = 1.00$ (we refer to the Cosine similarity unless specified otherwise), whereas adding irrelevant keyword to the query $q = \{1,1,1,1\}$ at $D = \{1,1,1,0\}$ we have $sim(q,D) = 0.87$. The contribution of the added keyword k to the similarity is then -0.13. For us its relevance $KR(k)$ is therefore 0.

3.3 Automated Navigation

Automated Navigation Algorithm

The principle of the automated navigation algorithm is based on the modified Input/Output interpreter described in [6]. It computes a score distribution on the targets in the tree allowing the most relevant node corresponding to the user query to be identified anywhere in the hierarchical structure. This approach allows to skip several levels of the hierarchy and faster navigation based on a given arbitrary threshold. At crucial nodes, the user assistance is required to confirm the following path interactively. Minimum best score difference and the information theoretic approach based on information entropy have been suggested as criteria for decision on automated navigation.

Minimum Best Score Difference

The minimum best score difference seeks for a mathematical formulation of the rule that automated selection should be triggered if some of the scores of nodes in selection is "good enough" and if this score is "substantially better" than the other ones.

Let s_1 (resp. s_2) be the best (resp. second best) score of node in selection and $d_{\min}$ be the minimum best score difference required. The node associated with s_1 is automatically selected if:

$$s_1 \geq s_{\min}$$

and

$$1 - \frac{s2}{s1} \geq d_{\min}$$

Alternatively we have also experimented with the absolute best score difference that allows a more conservative approach to automated navigation. In particular, for low values of the s_1 (particularly when $s_1 < d_{\min}$) this approach may be preferred. The rule of absolute minimum best score difference then requires that $s1 - s2 \geq d_{\min}$ holds.

Values of $s_{\min}$ and $d_{\min}$ are selected arbitrarily, we have achieved good results by working with $s_{\min} \in < 0.1, 0.25 >$ and $d_{\min} \in < 0.33, 0.67 >$.

Information Theoretic Approach

In this approach not only the two best scores are compared but scores of all nodes are taken into consideration for the decision whether automated navigation will take place or not. The information entropy is calculated based on the probabilities derived from the obtained scores. Probabilities p_i are curently derived from similarities s_i as follows:

$$p_i = \frac{s_i}{\sum s}$$

The entropy is then computed and normalized based on these probabilities:

$$H = -\sum p_i \times \frac{\ln(p_i)}{\ln(N)}$$

Where N is number of nodes in selection. The automated selection then takes place when the entropy H does not exceed a specified threshold k, i.e. when the uncertainty of making a good automated choice is not too high. We have achieved good results by working with threshold $k \in (0.85, 0.95)$.

When $s_i = 0$ the node is ignored and does not enter the computation. This rule may be extended to $s_i < s_{\min}$ introduced in previous section.

Creation of Hierarchical Structure

In order to allow automated navigation, creation of a coherent hierarchical structure is necessary. The hierarchical structure can be created for example by clustering or categorization of extracted data items, however, in this case the human-readable descriptions for intermediate nodes would have to be created which corresponds to a difficult task. We have therefore opted for mirroring the existing structure on the SCB Web site, transforming the graph-like structure into tree-like hierarchy.

The SCB sitemap has been analyzed and became a basis for the website structure extraction step[6]. The 23 subject domains have allowed us to produce quite a broad data set with regard to the subject categories covered. As far as content is concerned, domains are clearly identified, whereas sub-domains needed to be extracted from the domain related documents (sub-domains are not explicitly present in the web site structure), and from the output format of the access forms to the publication database, the press archive, and the statistical database.

In order to allow consistent graph-to-hierarchy conversion, identification of a key attribute as a base for the tree-like hierarchical structure composition is necessary. Since our target was to address domain specific data, the choice for this attribute was a content-related subject. Alternatively, in case there would be more then one key attribute associated with the data domain, several hierarchical structures (multiple trees) could be created in parallel. Other attributes, such as document types and time/space relevance as described in previous section, were regarded as metadata allowing the document filtering functionality.

In the prototype we also tried to keep documents with the same granularity on the same level. Ideally the tree would be binary and deep. This is unfortunately not realistic with regard to the nature of data and even with our attempt to keep the tree as close as possible to this vision we arrived at 5 levels with average branch factor of 9.05. It is perhaps interesting to also mention the average branch factors for individual levels (values have been adjusted with respect to leaves occuring in various levels):

[6] http://www.scb.se/templates/SiteMap_2711.asp

Level	Average Branch Factor	Description
1	23	Root
2	7.17	Subject domain
3	8.45	Subject sub-domain
4	9.29	Portal or Statistical message
5	*N/A*	Statistical document or item
all levels	9.05	

Backward Navigation

Navigation is done from the tree root to the most relevant leaf node or set of leaves. However, there is a possibility that user will "get lost" by not precise enough formulation of a query or by selection of a wrong node within the navigation process. In such cases it would be useful to be able to automatically go back up in the tree when the system recognizes significant differences in the relevance scores. In order to implement this idea two possibilities were considered: (i) changing the sub-tree and (ii) alternative selections.

In order to allow this functionality two-level similarity computation was applied. First, on the level of a local sub-tree related to the actual position in the hierarchical structure, second, the global similarity for all existing nodes outside the current sub-tree. If the similarity of the refined query to some node in the global structure proves to be significantly better than to the best local node, the navigation will allow the path correction by alternatively considering the other path to be explored in parallel or instead.

Changing the Sub-Tree

A more pro-active approach is to automatically decide in place of a user to opt for an alternate suggestion anticipating significantly better results in the subsequent information access steps. The best score of the whole tree is compared with the best score of the current sub-tree. In case the two scores differ the change of the current position in the structure is considered. Again, as it is the case for the automated selection algorithm described previously, we compare the difference of the two scores with arbitrarily selected threshold. Note that this option may cause frequent jumps in the tree that may be perceived as an inappropriate behavior of the system by users. We considered the usage of this approach where the search history is included in the computation. That is, the jump is only triggered when the similarity of the global tree with the conjuction of queries in the search history, appropriately weighted retrospectively, is significantly better than the one computed on the local sub-tree.

Alternative Selections

A more conservative implementation of this functionality is to only suggest the better scoring nodes as an alternative selection. This way the user interface is more coherent in a way that user is not faced with unexpected changes in position in the structure.

4 Conclusion

The developped StatSearch prototype is currently undergoing a user-based evaluation. The evaluation methodology was jointly set up by Statistics Sweden (SCB), EPFL, and CERN. The concrete evaluation experiments took place at SCB in early 2005 and will be full reported in [1].

References

1. Ing-Mari Boynton, Bert Fridlund, Alf Fyhrlund, Peter Lundquist, Bo Sundgren, Martin Rajman, Martin Vesely, Helge Thelander, and Martin Martin Wänerskär. Evaluating a system for enhanced access to statistical data on the web: the statseach evaluation experiments. *To appear in the Proc. of the HCI 2005 internaltional conference, Las Vegas, USA*, 2005.
2. Forler E. Intelligent user interface for specialized web sites, Master thesis, Ecole Polytechnique Federale de Lausanne, 2000.
3. Fridlund Bert Fyhrlund Alf and Sundgren Bo. Using text mining in official statistics. *COMPSTAT 2004, 16th Symposium of IASC, Prague*, August 23–27 2004.
4. Grefenstette G. Comparing two language identification schemes. *JADT'95, 3rd International conference on Statistical Analysis of Textual Data, Rome Italy*, 1995.
5. Robertson S.E. and Spark-Jones K. Relevance weighting of search terms. *Journal of the American Society for Information Sciences*, 27(3):129–146, 1976.
6. Guhl U. *Entwicklung und Implementierung eines UNIX-Assistenten.* PhD thesis, Rheinisch-Westfalische Technische Hochschule Aachen, 1995.