Making Statistical Data More Easily Accessible on the Web: Results of the StatSearch Case Study

Martin Rajman[1], Martin Vesely[1,2], Ing-Mari Boynton[3], Bert Fridlund[3], Alf Fyhrlund[3], Peter Lundquist[3], Bo Sundgren[3], Helge Thelander[3], Martin Wänerskär[3]

[1] École Polytechnique Fédérale de Lausanne, EPFL Lausanne, Switzerland
{martin.rajman, martin.vesely}@epfl.ch

[2] CERN, CH-1211 Genève 23, Switzerland
martin.vesely@cern.ch

[1]Statistics Sweden (SCB)
Box 24300, SE-10451 Stockholm, SWEDEN
{Ing-Mari.Boynton, Bert.Fridlund, Alf.Fyhrlund, Peter.Lundquist,
Bo.Sundgren, Helge.Thelander, Martin.Wänerskär}@scb.se

CERN-OPEN-2005-022
17 October 2005

Abstract. In this paper we present the results of the StatSearch case study that aimed at providing enhanced access to statistical data available on the Web. In the scope of this case study we developed a prototype of an information access tool combining query-based search engine with semi-automated navigation techniques exploiting hierarchical structuring of available data. This tool enables a better control of the information retrieval, improving the quality and ease of the access to statistical information.

The central part of the presented StatSearch tool consists in the design of an algorithm for automated navigation through a tree-like hierarchical document structure. The algorithm relies on the computation of query related relevance score distributions over the available database to identify the most relevant clusters in the data structure. These most relevant clusters are then proposed to the user for navigation, or, alternatively, are the support for the automated navigation process. Several approaches to automation of the navigation are compared and natural language processing techniques allowing more precise and coherent computation of textual similarities are briefly described.

The resulting StatSearch prototype was evaluated by the Swedish Statistical Office (SCB) on a sample of over 5000 English documents accessible through the SCB web site. The evaluation was based on supervised on-site usability testing and aimed at identification of the prototype's potentials and its added value with respect to information access as objectively perceived by users. The evaluation method and obtained results are presented. The case study was carried out in the framework of the NEMIS Network of Excellence in Text Mining and Its Applications in Statistics (IST-2001-37574).

1. Introduction

The StatSearch case study aimed at providing enhanced access to statistical data available on the Web. An optimized approach to the interactive information access was developed and demonstrated on a prototype of an information access tool. The StatSearch prototype focuses on combination of information retrieval techniques exploiting the state-of-the-art natural language processing, textual similarity computation and comprises an algorithm for semi-automated navigation that optimizes the interactive information access process.

The principle of the algorithm for automated navigation is based on a score distribution on the documents, done typically by textual similarity computation. An upward propagation of obtained scores in the hierarchical structure is performed. Computed scores are then used to guide a faster, partially automatic, downward navigation in the tree. Interactive user intervention at the node selection step is only required for nodes with children corresponding to a score distribution where no clearly good candidate can be identified. Otherwise, the traversal of the tree is performed automatically. Several variants of this approach were compared, including decision rules based on relative minimum best score differences, as well as on information theoretic measures.

The algorithm also allows a more reliable document ranking by providing the user with document filtering functionalities such as the possibility to restrict the search only to the documents dominated by the node corresponding to user's current position in the hierarchy or to the documents matching a limited set of meta-data attributes defined for a data set. Among the various features considered, in addition to the combined access algorithm itself, the computation of keyword contributions to the query-document similarity, semi-automated backtracking during navigation, and document-category associations were tested.

The prototype was validated on a set of selected documents provided by Statistics Sweden (SCB) amounting to over 5000 items in English by internal evaluation. In this evaluation we assessed how well a system meets the information needs of its users in the domain of statistics. When considering evaluation approaches we opted for user-based evaluation by undertaking internal pilot run in five separate sessions aiming at providing a formative feedback on the StatSearch technology. More precisely, we aimed at identifying the potential of the targeted access technology and a precise specification for planned external summative evaluation. The main goal of the evaluation was therefore a first validation of the prototype and of the general evaluation setup.

The rest of this contribution is organized as follows. In section 2, we describe the document processing techniques that were used to extract features out of the documents; in section 3 we discuss the possibility to optimize textual similarity computation by processing computationaly heavy parts of formulas in advance; in section 4 the automated navigation algorithm is described together with adopted decision criteria; in section 5, we describe the internal evaluation of the produced prototype, and; finally in section 6, we provide some conclusions and indicate some future works.

2. Document processing

Information access is generally understood as a process of identifying and presenting information that corresponds to an information need expressed usually in the form of one or several natural language queries [1]. Information is usually expressed by sentences in natural language or such expression may be derived. In this perspective, the focus in document processing is directed towards combining traditional information retrieval techniques with techniques developed in the fields of text mining and natural language processing. Usage of text mining and document processing techniques in the domain of official statistics is in more detail analyzed in [5].

In our approach, several natural language processing techniques are used to obtain semantically coherent document representations in the form of weighted feature vectors.

Selection of relevant features is based on language identification [6], metadata extraction and processing, stemming, part-of-speech tagging, and tf.idf weight computation. More precisely, extracted features are first processed on the basis of the available metadata, and subsequently filtered on relevant morphological category and tf.idf relevance. Features with morpho-syntactic categories irrelevant for the purpose of document content representation are discarded.

In addition we perform a simple form of compound word identification, based on constituent co-concurrencies. For the rest of the processing, the identified compounds are treated both as one single feature and as the set of separate features corresponding to their individual word constituents.

Finally, as far as the *tf.idf* weights are concerned, one option is to use the computed weights as an additional feature selection filter, removing features that did not score well enough in terms of their *tf.idf* weight. This feature selection is performed so that either the N best feature candidates are kept or, since the weights are normalized, by setting a *tf.idf* minimum threshold. We are also experimenting with various other, more sophisticated, weighting schemes, such as the Okapi BM25 formula, that might further increase the quality of the feature selection process.

Another option for the implementation of *tf.idf* based selection is to use the full weighted vectors for the computation of a similarity between the documents and the available queries. This latter approach is more demanding on processing time, but can be performed efficiently with well suited similarity measures as described in the next section.

3. Textual Similarity Computations

3.1 Query-document similarity computation

Textual similarity computation is based on measures that are traditionally used in the field of information retrieval. We considered a variety of similarity measures and finally adopted the standard cosine (see equation (3.1)) as the similarity measure to be included in our model.

$$sim(Q,D) = \cos(Q,D) = \frac{\sum W_{iQ} W_{iD}}{\sqrt{\sum W_{iQ}^2} \times \sqrt{\sum W_{iD}^2}} \qquad (3.1)$$

where Q (resp. D) is a query (resp. a document) represented by the weighted vector (W_{iQ}) (resp. (W_{iD})).

One of the important issues related to the use of similarity measures in our model is the computational cost of the similarity computation. In this perspective, our concern was to use a framework allowing an efficient similarity computation in the case of large data sets.

One possibility is to use a threshold value to replace the tf.idf document vectors by binary vectors: when the tf.idf value exceeds the specified threshold, the feature is assigned the weight of one and a weight of zero otherwise. In such a set up, similarity computation simply becomes a part of the feature selection step.

Another possibility is to use the full weighted document vectors. For efficiency reasons, it is important that, in this case, a part of the similarity computation can be made in advance, once for all. The cosine similarity measure is well suited for this, as it can be decomposed as indicated in equation (3.2):

$$sim(Q,D) = \sum \left(\frac{W_{iQ}}{\sqrt{\sum W_{iQ}^2}} \right) \left(\frac{W_{iD}}{\sqrt{\sum W_{iD}^2}} \right) \qquad (3.2)$$

where the query specific part (which corresponds to a normalization with respect to the length of the query $q = \sum W_{iQ}^2$) is constant for a given query, and the document specific part can be easily obtained if the normalized vector $\alpha = (\alpha_{iD})$, where $\alpha_{iD} = \frac{W_{iD}}{\sqrt{\sum W_{iD}^2}}$, is pre-computed and stored for each of the keyword-documents pairs stored in the used index.

The real time similarity computation can then be reduced to the computation of an inner product vectors α with a vector only dependent on the query.

In the frequent case where the query features are weighted in a binary mode (i.e. $W_{iQ} \in \{0,1\}$), this inner product computation actually reduces to the sum of the relevant α_i's normalized by the query length:

$$sim(Q,D) = \frac{1}{\sqrt{q}} \sum \delta_{iQ} \alpha_{iD} \qquad (3.3)$$

where is δ_{iQ} is the binary weight of the i[th] feature in the query *Q*.

Notice also that we can compute the relevance to a query of a single document or of a set of documents (i.e. a document category), in which latter case the profile used for the document category is either the sum of all underlying document profiles or the most representative of them.

4. Automated Navigation

4.1 Automated Navigation Algorithm

The automated navigation implemented in the StatSearch prototype is based on a modified version of an algorithm previously experimented with in [4,7]. The idea behind this algorithm, as it is in more detail described in [9], is the following: a score is first computed for each of the data items present in the tree. This score corresponds to the relevance of the document with respect to the last query expressed by the user. From these initial scores, a score can then be computed for all the nodes in the hierarchy. The procedure used to compute the scores of the nodes is quite simple: each node is assigned the score corresponding to the best of the scores of its children nodes in the tree.

Once all the node scores are computed, they can be used to guide the automated navigation that relies on the following principle: starting from the current position of the user in the hierarchy, the system automatically follows the path corresponding to the choice, at each node, of the best scored child node, provided that the selected best score fulfils a predefined

optimality criterion that aims at guaranteeing the reliability of the automated decision. Various criteria can be used, and will be described later on in this section; however, they all correspond to different ways of measuring whether the selected choice is "substantially better" than the other competing ones. As soon as a critical decision node is reached, the process stops at the corresponding node, the list of possible navigation choices is displayed and user assistance is required to interactively perform the correct choice.

In short, the idea behind the approach is to let the system automatically perform the navigation decisions as long as enough evidence is available, therefore allowing to the user to faster reach the most relevant data item.

The minimum best score difference criterion. The minimum best score difference is a possible mathematical formulation of the rule that automated selection should be used if the scores of one of the possible next node candidates is "good enough" and "substantially better" than the other ones.

Let s_1 (resp. s_2) be the best (resp. second best) score of the available next node candidates. If a threshold s_{min} is defined to quantify the "good enough" condition ($0 \leq s_{min} \leq 1$), and a threshold d_{min} is defined to quantify the "substantially better" condition ($0 \leq d_{min}$), then, the best node will fulfill the above mentioned rule (and therefore be automatically selected) if:

$$s_1 \geq s_{min} \quad \text{and} \quad 1 - \frac{s_2}{s_1} \geq d_{min} \qquad (4.1)$$

Notice that the formula used in (4.1) for the "substantially better" condition relies on the *relative* best score difference. Alternatively, the *absolute* best score difference can also be considered. In this case, the rule to use for automated decision is that $s_1 - s_2 \geq d_{min}$ should hold, and, because of the normalization of the scores, $0 \leq d_{min} \leq 1$. This rule allows a more conservative approach to automated navigation. In particular, for low values of s_1, this approach may be preferred. Of course, the values used for s_{min} and d_{min} are essential parameters of the automated navigation algorithm and have to be carefully selected. We achieved good results by working with $0.1 \leq s_{min} \leq 0.25$ and $0.33 \leq d_{min} \leq 0.67$.

The entropy based criterion. In this approach, the scores of all the next node candidates are taken into consideration, and not only the two best ones. The entropy *H* corresponding to the normalized score distribution over the next node candidates is computed, and the automated selection takes place when *H* does not exceed a specified threshold H_{max}, i.e. when the uncertainty of making a good automated choice is not too high:

$$H = -\sum p_i \frac{\ln(p_i)}{\ln(N)} < H_{max} \qquad (4.2)$$

where $p_i = s_i / \sum s_i$, s_i is the score associated with the i^{th} node, and *N* is the number of nodes to consider. In the case where $s_i = 0$, the corresponding node is ignored and does not enter in the computation.

We have achieved good results by working with a threshold *k* such that $0.85 \leq H_{max} \leq 0.95$. In addition, the entropy based rule may also be extended with a "good enough" condition of

the form: for all i, $s_i > s_{min}$, where s_{min} is a minimum threshold similar to the above introduced one.

4.2 Hierarchical document structure

In order to allow automated navigation in the available data, the creation of a coherent hierarchical structure over this data is necessary. This hierarchical structure can be created in a variety of different ways, ideally by automated clustering of available data items. Since in this case human-readable descriptions characterizing the artificially produced clusters and sub-clusters need to be created, we have opted for mirroring the access structure as provided on the web site.

Note that in order to allow the production of a consistent tree-like hierarchical structure, each of the nodes of the structure derived from web site has to be associated with a content-related key attribute. In case that the original structure as present on the web site is not entirely based on a content-related key attribute, levels corresponding to an attribute that is irelevant are ignored and semantically identical or similar enough nodes on the next level are merged. In this connection we also considered the possbility to take more than one key attribute associated with a given node which would lead to a creation of several hierarchical structures in parallel.

Another important thing to note is that the nature of automated navigation algorithm implies that the created tree should be ideally as deep as possible, and with nodes corresponding to a quite low branching factor. Unfortunately, since we mirrored an existing web site, it was not possible to influence these parameters as it would be the case when using data clustering and we reached a rather unbalanced tree with lower number of levels and an average branching factor of 9.0.

4.3 Backward navigation

Navigation is usually performed top down, from the tree root to the most relevant leaf or set of leaves. However, there is always a possibility that user "gets lost", either by not giving enough precision in the formulation of a query, or by selecting a wrong node during the navigation process. In such cases it is useful to be able to automatically go back up in the tree when the system recognizes significant differences in the relevance scores of the nodes. In order to implement this idea two possibilities were considered: (i) automatically changing the current sub-tree, and (ii) proposing alternative choices.

Changing the current sub-tree. In order to implement this functionality, two node score computations are performed in parallel. First, within the sub-tree dominated by the current position in the hierarchical structure, and second for all existing nodes. If the best score of some node in the global hierarchy appears to be significantly better than to the best local one, the navigation will allow a "jump", i.e. a path change leading to the best scored node that is not in the current sub-tree. For the jumping decision, we can again use a rule similar to the one presented in (4.1).

This solution provides a rather radical approach to automated decision for alternative navigation. This might lead to significantly better results in the information access, however, it is also important to notice that this procedure may cause frequent jumps in the tree that

could be perceived as an inappropriate behavior of the system by the users. It is therefore desirable to set the d_{min} value for changing the sub-tree quite high, e.g. $d_{min} > 0.5$ and to prefer a decision rule based on the absolute best score differences rather than the relative ones. In addition this approach may be refined including the search history in the computation: the jump is only triggered when the best score in the global tree is computed with respect to the concatenation of all the queries in the search history, appropriately weighted.

Proposing alternative choices. A more conservative implementation of the above mentioned functionality is to suggest the better scored nodes as alternative choices instead of triggering a possible jump. In this way, the system behavior is more stable as the user is not faced with unexpected changes in his/her position in the hierarchy. Alternative choices are displayed one after another in a specific window, allowing the user to scroll down through all relevant choices and perform an explicit choice if desired.

5. Evaluation Framework

5.1 General evaluation setup

User-based evaluation of the StatSearch prototype was performed by Statistics Sweden on a sample of documents accessible through the SCB web site. The evaluation was carried out in the form of on-site usability tests followed by in-depth interviews [10, 11]. When preparing the evaluation specification we built upon experience gained within similar evaluations previously conducted in the scope of the InfoVox project [8], as well as an evaluation of the SCB web site [2] and a self administered web questionnaire relating to the SCB databases [3]. In each evaluation session the context and the objectives of the evaluation is presented to the evaluators before they connect to the system. The interaction is then supervised with a minimum of interference. Finally, supervisors get back to evaluators to perform the in-depth interview and close the session. The overall duration of one evaluation session did not exceed one hour.

At the beginning of each evaluation session the evaluator is first provided with a short description of the prototype for the first orientation as an introductory part to increase the uniformity of the information gathering process. This is done with the help of an instruction sheet that has to be systematically read to the evaluator; in particular, the evaluator is made aware of the fact that the interaction with the system is recorded. In addition information about the purpose and background of the evaluation is given to induce a feedback on the evaluation process itself.

In addition to the qualitative criteria acquired during the interviews, the functioning of the prototype during the interaction with the users was also extensively logged in order to keep some objective information about the interactions. For each evaluation session, the logged information include the duration of the interaction, the total number of navigation/query steps carried out, the total number of documents accessed and the ratio between the time spent by the evaluator with the simple (i.e. search only) and the advanced (i.e. combining search and navigation) interface. The access time (i.e. the total time required for an evaluator to access a document), the number of steps (query/navigation) to access the information, and the success rate as perceived by the evaluator, were considered as the main quantitative measures.

The subjective success rate was acquired by asking the evaluators to indicate the level of correspondence between the information obtained through the interaction and the original information need as described by the search scenario. This was done by displayed the obtained information in a separate frame together with the original scenario, and by requesting the evaluator to tick an appropriate satisfaction level on a predefined scale. The main advantage of this procedure is that it allows to dynamically record the evaluator's reactions during the test, which might thus lead to more accurate feedback information. Notice however that displaying the evaluation frame might be considered as disturbing or intrusive and its usage should therefore be considered with care in the evaluation context.

5.2 Search scenarios

Search scenarios are short stories that aim at simulating a real applicative context in the evaluation environment. In our configuration we based scenarios on categorized information needs as designated by most frequent request types recognized by the preliminary interviews undertaken with specialists in the domain of official statistics. Representative set of real questions addressed to a domain specialists or a general advisors were extracted from recent history and analyzed. Question types that we identified are either factual, asking directly for some statistical information; probing, asking for some statistical meta-information; divergent, relating to a too general scope that does not correspond to a single answer (usually iterative) and methodological, asking for the way some statistical information is obtained or interpreted. Since most frequently the factual questions are asked we also put an emphasis on this type of question at the selection of a representative question sample for creation of scenarios.

5.3 In-depth interview

In the last stage of the evaluation process one of the supervisors interviews the evaluator. Interviews are recorded on tape and transcribed; the resulting transcriptions are subsequently edited and analyzed, and the results are reported in a written form. The interview is based on a topic guide. The following main question topics were identified on when constructing the interview: ergonomics of interaction, general opinion on the interaction, and subjective evaluation of the system performance. In addition, the interview guidelines also contained several sets of questions: introductory questions, questions on the understanding of the prototype functioning, formative questions pertaining to further potential development of the prototype and comparative questions to investigate relations to other information access tools used in the domain of official statistics. Notice that, in the case of the external large-scale evaluation, in-depth interviews are replaced by a satisfaction questionnaire with adequately re-formulated questions.

5.4 Evaluation Results

After each evaluation session, the supervisors got together for summarizing observations and other experiences. The report contains qualitative statements about the potential adequacy of the system for accessing statistical data and some recommendations on the adaptation of the used evaluation framework to evaluation experiments with larger sets of external users.

According to the general views of the evaluators, the main impression of the StatSearch prototype and of the general evaluation setup is positive. On the whole the users approved of the interaction with the StatSearch system as well as the system itself. As previously mentioned we focused on formative evaluation aiming at identifying whether the prototype meets its objectives and validating its new functionalities. We have therefore gahtered two types of feedback, formative feedback on prototype functionality essentially obtained by interviews and feedback on the evaluation specification.

Although we registered 79% of usage for the advanced interface, the simple interface was still subjectively preferred to the advanced one, as being more similar (query syntax, positioning of input elements etc.) to the standard interfaces proposed by traditional search engines. The feedback on the general evaluation setup showed that categories proposed to the internal evaluators should be extended beyond the scope of the ones used by subject domain specialists and should include categories more oriented on users working in related areas of production, presentation and information of statistics, including the web site and the technical systems, as well as the end users of statistics. In this connection some of the evaluators pointed out that both familiarity with the web site and with its structure may have introduced a bias of internal users. Another issue pointed out frequently was the approach to evaluation of new functionalities introduced in the prototype, such as, for example, the keyword relevance computation or correspondence of documents to categories, where presentation of these new features should be more emphasized.

6. Conclusions and Future work

In this paper we presented the results of the StatSearch case study that analyzed practices in information acces to statistical data and demonstrated a novel approach to information access in the domain of official statistics. This has been done by implementing and evaluating a prototype of an information access tool featuring enhanced techniques for statistical data access with focus on combination of information retrieval, exploitation of state-of-the-art natural language processing and textual similarity computation. In addition we presented some extensions to the user interface allowing a better orientation of the user within the information access process where the main extensions include presentation of keyword contributions to the document ranking and a backward navigation in the document tree. The user-based evaluation was performed by the Swedish Statistical Office (SCB) on a set of selected statistical documents where the StatSearch prototype was internally validated.

The future work will focus on several issues namely measurements of textual similarity pre-computations and comparison with other methods of document ranking in a result set, refinement of document processing by including a more sophisticated semantic analysis and generalization of creation of hierarchical structure that should be studied in connection to a possible introduction of document clustering.

References

1. Baeza-Yates R. and Ribeiro-Neto B. Modern information Retrieval. Reading, MA : Addison-Wesley, 513 p, 1998.

2. Boynton, IM., Mujagic, H.: Utvärdering av SCB:s webbplats, Evaluation report, Statistics Sweden, 2002.

3. Boynton, IM., Bäckström, H.: Test av SSD webbenkät, Evaluation report, Statistics Sweden, 2003.

4. Forler E. Intelligent User Interface for Specialized Web Sites. Master thesis, Ecole Polytechnique Federale de Lausanne, 2000.

5. Fyhrlund Alf, Fridlund Bert and Sundgren Bo. Using Text Mining in Official Statistics, *COMPSTAT 2004, 16th Symposium of IASC*, Prague, 2004.

6. Grefenstette G. Comparing two Language Identification Schemes, JADT'95, 3rd International conference on Statistical Analysis of Textual Data, Rome Italy, 1995.

7. Gűhl U. Entwicklung und Implementierung eines UNIX-Assistenten, Tech. Report, Rheinisch-Westfalische Technische Hochschule Aachen, 1995.

8. Rajman M. et al: Assessing the Usability of a Dialogue Management System Designed in the Framework of a Rapid Prototyping Methodology, Acta Acustica, 2003.

9. Rajman M. and Vesely M. Combining Text Mining and Information Retrieval Techniques for Enhanced Access to Statistical Data on the Web. In: 3rd International Conference on Universal Access in Human-Computer Interaction, UAHCI'05 Las Vegas. 2005.

10. Sparck-Jones K. and Galliers J. Evaluating Natural Language Processing Systems. Springer-Verlag, Berlin. 1996.

11. Voorhees E. The philosophy of information retrieval evaluation. In: Evaluation of cross-language information retrieval systems, p.355-370, 2002.