Expert search in semantically annotated enterprise data: integrating query dependent and query independent relevance factors

Felix Engel, Matthias Juchmes, Matthias Hemmje
Distance University Hagen
Felix.Engel, Matthias.Juchmes, Matthias.Hemmje@fernuni-hagen.de

Abstract

The documentation of processes or employee- and product-related-data in the enterprise does comprehensively contribute to the preservation and future access to acquired in-house knowledge. Sophisticated access to this data is an essential part of successful knowledge management. With the increasing use of semantic web recommendations and technologies in enterprise new challenges and opportunities concerning data access arise. Search for experts in documented enterprise data has been a famous research topic for years. A major reason for this is its beneficial impact on accessing existing enterprise potential. Thus search for experts is a valuable application for the enterprise advancement.

However, recent expert search systems largely implement relevance ranking on basis of topic relevance between a potential expert and the topic of the query. Nevertheless, even though the query topic as relevance evidence source has been proven as one of the most important factors in expert search; it only reflects the relevance between the query topic and the closeness of an expert to the topic. The analysis of further evidence sources is part of researches in the field of expertise seeking. Such research results are rarely taken into account in recent expert search implementations. The provision of comprehensive semantic annotations in enterprises opens new potential and challenges for the implementation of sophisticated expert search systems, taking into account not only topic closeness.

1 Introduction

SMART VORTEX[1] is an integrated project co-financed by the European Union within the 7th Framework Program. The project objective is the provision of an extensive set of intelligent and interoperable tools, methods and services for the management of massive data streams alongside the whole product lifecycle spanning from product idea generation, design, manufacturing and service to product disposal. Within this objective one focus is on supporting collaboration of people involved in the product lifecycle. Part of this objective is the identification of in house experts for collaboration initiation.

Recent expert search systems calculate expert relevance on basis of topic closeness. Beside topic closeness, various additional evidence sources are part of a holistic expert relevance ranking calculation. Those findings are part of studies in the field of expertise seeking. In SMART VORTEX enterprise data encompasses models of various entities such as for instance people, products and their interrelation in the enterprise. As a common basis for modeling and data representation in SMART VORTEX recent semantic web recommendations and tools are applied. In fact the present data represents a semantic information integration of various existing heterogeneous data sources such as ERP systems, PLM systems as well as employee- or product information data bases among others. Various person features are implicit part of this semantic enterprise graph and valuable for the representation of diverse expertise seeking evidence sources.

1.1 Problem statement

Finding experts within an enterprise for any kind of problem is a complex and time consuming task. Few isolated applications for expert finding exist, however the demand for comprehensive solutions keeps on being a non-trivial task. According to Balog et al. [Balog *et al.*, 2012] in the field of expert search a clear distinction between the two fields of expertise retrieval and expertise seeking could be made.

Expertise retrieval includes all content-related approaches that process a document database using information extraction and data mining techniques, among others. The processed data in this case could be searched subsequently with the aid of known information retrieval algorithms. In contrast to the content-related approaches, researches in the field of expertise retrieval analyze all further evidence sources that lead to the decision if the potential expert is relevant from a user perspective. Such evidence sources include for instance the freshness of knowledge, experience, reliability or social closeness. However, recent expert search applications widely realize expertise retrieval approaches and rarely take into account results from expertise seeking researches [Hoffmann *et al.* 2012 and Balog *et al.*, 2012]. Various features relevant for expertise seeking are implicit part of the SMART VORTEX semantic enterprise graph. In order to use these implicitly modeled features in ranking tasks they need to be computable. This could be realized through the application of known algorithms in the field of the semantic search or through simple functions basing on graph functions. Relevance calculations in the field of semantic web are for instance the calculation of popularity, rarity or association length approaches. Generally such features are independent from the query itself. Query dependent calculations similar to Albertonie et al. [Albertonie *et al.*, 2006] that base on the specification of a path in the semantic graph in contrast are query dependent. A query dependent

[1] http://smartvortex.eu/

evidence source could be for instance the number of connections to enterprise roles with specific constraints (e.g. only management or service roles) or the freshness of knowledge given a specific query topic.

However query dependent and independent calculations only make statements about the graph structure. The resulting assessment is dependent on the task at hand and has to be given retrospectively. Furthermore, the assumption that several features are part of the overall relevance assessment lead to the problem of meaningful aggregation of features into one ranking function. Aggregation of relevance calculation constituents is a common problem in retrieval tasks. Learning to rank is an approach that has recently been applied for similar problem statements.

1.2 Objectives

The aim of the work introduced in this short paper is the development of an expert search approach in a semantically annotated enterprise knowledge base. The approach should integrate various sources of expertise evidence beyond content-related proximity. To reach this goal the approach shall take into account the results of various expertise seeking investigations in order to enhance expert relevance calculation in the sense of expertise seeking findings. Calculation of evidence could be dependent- or independent from a query and should take into account existing relevance calculation approaches from research in the field of the semantic web. These various evidence calculations must in the end be aggregated and assessed according to the relevance aspects to be fulfilled.

2 Sources of evidence in expertise seeking tasks

The research areas of expertise seeking and information seeking are closely related. Expertise seeking investigations take a user centric perspective in an expert search task. The focus of these investigations is the analysis of those evidence sources that are crucial for choosing an expert from a user point of view.

Karunakaran et al. [Karunakaran *et al*., 2012] emphasize the physical proximity of an expert, especially under the consideration of the degree of acquaintanceship. Woudstra et al. [Woudstra *et al*., 2008] as well as Helms et al. [Helms *et al*., 2013] consider this finding as part of an access related aspect. Especially the influence of social factors with varying characteristics is part of expertise seeking investigations. Yuan et al. [Yuan *et al*., 2007] emphasize that social closeness between people in particular is valuable for expert search, because user and expert are unbiased in their communication. Woudstra et al. respond in their investigation to quality related factors like e.g. the actuality of acquired knowledge or the reliability of a potential expert.

Some of the mentioned aspects like e.g. the degree of acquaintanceship in a semantically annotated knowledge base could be calculated via famous semantic web techniques such as for instance popularity. Popularity calculates the degree of connectivity in the graph. Such calculations are independent from the query itself. Other sources of evidence cannot be calculated by these well-known relevance measures. In the case of approachability [Woudstra *et al*., 2008] for instance, the relevance of an expert candidate can be calculated by the fact that he is part of the same working group, project or else. This condition is query dependent and could not be calculated by known semantic web relevance measures.

2.1 Query dependent relevance calculations

Query dependent calculations could be characterized by the fact that they could only be calculated based on the query itself. The calculated value in this respect describes a proportion to a query on base of specified basic conditions. Specification of such conditions in a semantically annotated knowledge base demands knowledge about the representation and relation between modeled entities. Since this knowledge is not explicitly part of the model itself, it is external. A considerable similar problem statement and approach has been published by Albertonie et al. [Albertonie *et al*., 2006] in order to calculate the similarity between instances of a semantic knowledge base. Albertonie et al. have applied simple calculation units specifying paths and a similarity function. A query dependent calculation in this sense is for instance the amount of relations between an expert and the topics of the search query.

2.2 Query independent relevance calculations

Plenty of the applied relevance measures in the semantic web community are graph based algorithms. Such relevance measures are inspired by findings in the field of graph theory. A famous measure e.g. is *popularit*y, which measures the amount of in- and outgoing links of an instance. Furthermore, the *association length* analyses the length between instances or *subsumption* which takes into account the taxonomic graph structure. The problem with such measures is that their result is depending on the task at hand. For instance a long *association length* could be interesting because it identifies an unobvious relation between instances. On the other hand shortest paths could be preferred, because they reflect a tight coupling of instances. Same holds true for the *popularity* measure. Here an instance with lots of relations could be relevant because of its high connectivity, but on the other hand an instance with few connections is specific and hence could be relevant. All of these measures are based on the graph structure itself and can be calculated independent from the search query.

3 Rules for configurations of interdependencies between relevance calculations

As stated above, the relevance degree of a query independent measure has to be assessed regarding the search task at hand. The same also holds true for the query dependent measures. In contrast to a general purpose entity search, in the scope of this work it is clear if a high or a low measure value indicates relevance or irrelevance. For instance, if the aim of the search is to find an expert as a course leader it might be of relevance if the potential expert already has course leader experience. This fact could be inferred by counting the number of course leader roles one has already taken. On the other hand it might be better to find a potential expert with few active roles to find someone with appropriate time capacities.

In order to illustrate the approach described above, the example search for a course leader is introduced. Following sources of evidence (SE) are part of the search:

- **SE 1:** How good is the potential experts (P) insight in enterprise processes? *Expertise* [Heath *et al*., 2006]
- **SE 2:** Does the potential expert match the query topic exactly or more specifically? E.g. in a query with the topic ObjectOrientedProgramming, an expert matching this topic exactly will be preferred over an expert with more specific knowledge (e.g. Java), because the course will introduce general concepts of object oriented programming as opposed to concepts specific to Java. *Topic of knowledge* [Woudstra *et al*., 2008]
- **SE 3:** A high number of connections of the potential expert in the enterprise should be preferred, because if the potential expert is well connected in the enterprise, it could be stated that he has a good standing. Nevertheless, besides good standing, a tight coupling between user and expert is of importance. Among others *Familiarity* [Woudstra *et al*., 2008]

In this example the search for an expert shall be evaluated as the sum of the above three evidence calculations. The calculation of these sources of evidence can be implemented as follows. Source of evidence 1 can be calculated by simply counting the enterprise roles a potential expert has already taken. This approach is pretty similar to the *count* function definition by Albertonie et al. The assumption is that the more roles a potential expert has taken the better he knows internal enterprise processes. Source of evidence 2 can be calculated by applying subsumption. In this application a more general result is be preferred. Source of evidence 3 spans two calculations. The degree of a potential expert connection can be calculated by the *popularity* measure. The tight coupling between user and expert is measured through application of the association length measure. In this application shortest paths are preferred.

Based on the above assumptions the search application needs a function to count how often a relation between potential expert and enterprise roles exist. Furthermore, the functions *subsumption*, *popularity* and *association lentgh* are part of the whole calculation. Hence, the above mentioned calculations are aggregated through the definition of the following person feature vector:

$$\begin{pmatrix} \text{feature 1}: count \\ \text{feature 2}: subsumption \\ \text{feature 3}: popularity \\ \text{feature 4}: associationLength \end{pmatrix}$$

Two sample instances of above feature vector could be as follows:

$$P_1 = \begin{pmatrix} 3 \\ 0.7 \\ 0.8 \\ 0.2 \end{pmatrix}; P_2 = \begin{pmatrix} 1 \\ 0.5 \\ 0.9 \\ 0.6 \end{pmatrix}$$

Given these sample instances of feature vectors, it is obvious that the calculated values just express the values of the applied functions. To fully support the source of evidence described above, rules have to be applied in order to make a statement about how well a calculated value supports the relevance of potential experts. In this sample application, a potential expert with a high value related to source of evidence 1 should be preferred. The following rule supports this statement: if $\{feature1_{p1} > feature1_{p2} \rightarrow P_1\}$ else $\{P_2\}$. However, an expert is even more relevant if the value of source of evidence 2 is low. This could be expressed by the rule: if $\{feature2_{p1} < feature2_{p2} \rightarrow P_1\}$ else $\{P_2\}$. The calculation of source of evidence 3 is more complex, because it is composed of two sub calculations. The following rule expresses the required statement: if $\{(feature3_{p1} > feature3_{p2})$ AND $(feature4_{p1} < feature4_{p2}) \rightarrow P_1\}$ else $\{P_2\}$.

The aggregation of these query dependent und independent features via rules apparently is a promising approach to express expertise seeking evidence sources. In fact the application of rules for the assessment of query dependent and independent feature calculation can be regarded as the description of a relevance pattern. To calculate a ranking model from a relevance pattern definition like that defined by above rules, the application of learning to rank is promising.

4 Application of learning to rank for relevance pattern learning

Learning to Rank (LTR) is an application in the research field of machine learning. LTR is used to learn a relevance ranking model of objects that are represented by relevance labeled feature vectors. In fact LTR learns a relevance pattern. Those learned ranking models are coefficients of a ranking function that calculates a relevance value for an object from its feature values. A machine learning algorithm like Support Vector Machines is applied to analyze the training data with the aim to find an appropriate model based on the data. Hence, a good model does not only match the rankings represented by the training data, but can be applied to general search queries not part of the training data set.

Liu [Liu, 2009] distinguishes between the three learning approaches *pointwise*, *pairwise* and *listwise*. The chosen approach influences the structure of the training data, and thus also the machine learning algorithms used to analyze this data. To date LTR is often applied in document retrieval tasks, like in Joachims, 2002 [Joachims, 2002]). Recently, some researches have been made that apply LTR in semantically annotated knowledge bases. Dali *et al.* [Dali *et al.*, 2012] use LTR to learn a ranking model for the aggregation of query-independent relevance measures in semantic databases. Features in this case include *popularity* related calculations. Labels for the test data are gathered by crowd sourcing among others. Fujita *et al.* [Fujita *et al.*, 2012] use LTR to recommend queries that are semantically similar to the original query. Chen et al. [Chen *et al.*, 2011] apply LTR to rank relationships in RDF graphs. In this approach LTR is used in order to learn the user’s preference based on various graph measures like *association length* or *popularity*. However, LTR-techniques include approaches which learn a ranking model based on labeled training data. Hence, critical requirement for each application that make use of an LTR approach is the existence of test data annotated with relevance labels. Generally, relevance labeling is done by experts or collected through crowd sourcing. The disadvantages of these approaches are the high costs and high failure rates.

However, in the application described here the relevance pattern is already known and described through rules (c.f. section 3). Hence, test data labeling in this case doesn't have to be realized by experts or else but by the evaluation of rules.

The following approach is conceivable for test data labeling based on rules as introduced above, in a pairwise LTR application. In a pairwise LTR setting feature vector instances are treated in pairs. Each pair is sorted into one of two classes if possible, depending on which of the vectors is more relevant. If no such decision can be made, the pair is not classified. Thus, algorithms for this approach have to solve a binary classification problem. The above defined rules are evaluated for each pair of feature vector instances as follows: Each possible pair of feature vector instances has to be evaluated given the above described rules. The evaluation result for each rule votes for one of the two feature vector instances. Two results of this voting approach are possible. In the case that one of the two vectors has more votes than the other, the vector with more votes is labeled as more relevant. In case of a tie, both vectors are too similar and thus can't be taken into consideration for the learning process.

The example feature vector instances (P_1, P_2) are evaluated on basis of above rules as follows:

- SE 1: 3 > 1, votes for P1
- SE 2: 0.7 < 0.5, votes for P2
- SE 3: (0.8 > 0.9) AND (0.2 > 0.6), votes for P2

The result of the evaluation is one vote for P1 and two votes for P2. Hence in a pairwise LTR approach feature vector instance P2 is labeled as more relevant as P1. Given a reasonable amount of those test data LTR is able to construct a relevance ranking model that reflects the relevance aspects described through rules.

5 Summary and outlook

This short paper introduced an approach for the integration of query dependent and independent relevance measures in a semantically annotated knowledge base, for the integration of expertise seeking parameters in an expert search task. The described approach aggregates several sources of evidence for the task of expert search going behind pure topic based relevance ranking. The application of rules as specification of a relevance pattern to be learned is the input for an LTR approach that learns a ranking model for unseen queries.

Open questions among others are the evaluation of this approach and hence which expertise seeking parameters can be calculated. Which of these parameters are dependent on a registered user and which can be calculated without registered users? With respect to the LTR application it is crucial to evaluate the dependency between size of database, required amount of training data and dimension of the feature vector.

Acknowledgments

This work has been partly supported by the FP7 EU project SMART VORTEX.

References

[Albertonie *et al*., 2006] Albertoni, R., & De Martino, M. (2006). Semantic similarity of ontology instances tailored on the application context. On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, 4275, 1020–1038.

[Balog *et al*., 2012] Balog, K. Fang Y., de Rijke, M., Serdyukov, P. und Si, L. (2012). Expertise Retrieval. Foundations and Trends® in Information Retrieval, 6(2-3), 127–256. doi:10.1561/1500000024

[Chen *et al*., 2011] Chen N. & Prasanna, V.K., 2011. Learning to Rank Complex Semantic Relationsships Technical Report., (November). Available at: http://www-scf.usc.edu/~nchen/paper/ltr.pdf.

[Dali *et al*., 2012] Dali, L., Fortuna, B., Duc, T. and Mladenić, D. (2012). Query-Independent Learning to Rank for RDF Entity Search. In The Semantic Web: Research and Applications, 484-498

[Fujita *et al*., 2012] Fujita, S., Dupret, G., & Baeza-Yates, R. (2012). Learning to Rank Query Recommendations by Semantic Similarities. *arXiv preprint arXiv:1204.2712*.

[Helms *et al*., 2013] Helms, R., Diemer, D., & Lichtenstein, S. (2011, July). Exploring barriers in expertise seeking: why don't they ask an expert?. In PACIS (p. 77).

[Heath *et al*., 2006] Heath, T., Motta, E., & Petre, M. (2006). Person to person trust factors in word of mouth recommendation

[Hoffmann *et al*., 2010] Hofmann, K., Balog, K., Bogers, T., & de Rijke, M. (2010). Contextual Factors for Finding Similar Experts 1. Journal of the American society for information science and technology, 61(5), 994–1014.

[Joachims, 2002] Joachims, T. (2002). Optimizing Search Engines Using Clickthrough Data. In Proceedings of the Eighth ACM SIG KDD International Conference on Knowledge Discovery and Data Mining, 133-142

[Karunakaran *et al*., 2012] Karunakaran, A. & Reddy, M., 2012. Barriers to collaborative information seeking in organizations. Proceedings of the American Society for Information Science and Technology, 49(1), pp.1–10.

[Liu, 2009] Liu, Tie-Yan. "Learning to rank for information retrieval." *Foundations and Trends in Information Retrieval* 3.3 (2009): 225-331.

[Woudstra *et al*., 2008] Woudstra, Lilian and van den Hooff, Bart. Inside the source selection process: Selection criteria for human information sources. Inf. Process. Manage. May, 2008. Doi 10.1016/j.ipm.2007.07.004

[Yuan *et al*., 2007] Yuan, Y.C., Carboni, I. & Ehrlich, K., 2007. The impact of affective relationships and awareness on expertise retrieval: a multilevel network perspective on transactive memory theory