In Proceedings of the First Workshop on Ontology
Learning, August 2000

Background Knowledge, Indexing and Matching-Interdependencies of Document Management and Ontology-Maintenance

Andreas Faatz[1], Thomas Kamps[2], Ralf Steinmetz[3]

Abstract. This position paper presents an algorithm, which determines similarities between text documents. These text documents are indexed with keywords and further background knowledge-terms from an ontology.The representation of the documents and the evaluation of the algorithm are used to let an ontology learn. This is shown to be one way of improving the results of the algorithm by improving the background knowledge.

1 INTRODUCTION

Consider a human being reading texts from domains, which to a certain extent are familiar to him or her. The reader is capable of the semantics of the text documents. Even if the person is not an expert in any of the domains described in the texts, a minimal comment we expect him or her to state is, weather two texts are similar or not. This kind of judgement also includes text documents, which possess similarities though containing a completely different vocabulary or sharing just a few common terms. Similarities are a part of the intellectual construction of reality [5] and generated by what words and phrases the human mind associates to the actual text.

In a business application grouping documents by their similarity undergoes restrictions: the job has to be done fast, for instance managing the continuous flow of short messages coming in to the editors of a newspaper. Moreover, the document base in use by the newspaper is too large, so an editor is not able to retrieve all similar texts in time.

We apply the above situation to a computer instead of a human reader. Our goal is to express similarities of text documents detected by an algorithm. Hence a semantic matching problem is to be solved. The associations and heuristics recognizing similarities beyond equalities of character strings have to be modeled somehow, otherwise we are restricted to plain full text retrieval [10], like many of the web-based search-engines taking HTML as an input.

1 is at KOM and intelligent views, 2 is at intelligent views, 3 is at KOM and GMD-IPSI
KOM, Technical University Of Darmstadt, Merckstr. 25, 64283 Darmstadt, Germany ||| intelligent views GmbH, Julius-Reiber-r str. 17, 64293 Darmstadt, Germany ||| GMD-IPSI, Dolivostr. 15, 64293 Darmstadt,Germany
email: 1 afaatz@kom.tu-darmstadt.de, 2 kamps@i-views.de, 3 rst@kom.tu-darmstadt.de

The following paper yields some propositions about a process, in which an algorithm obtains a value of similarity from a pair of text documents. Before we describe the algorithm, we take a brief look at how the documents first have to be made readable to the algorithm and in which fashion background knowledge adds further information to the matching process. Then we explain the algorithm: its way of matching documents and the parameters in need. Finally we give some hints concerning the evaluation and improvement of the algorithm. This will be the point, where background knowledge gets affected by our results and we will distinguish objective and subjective influences on the background knowledge.

2 PREPROCESSING THE DATA

We consider a corpus of short text documents to be given. Any document *D* is attached with a vector *v(D)* including a description of its contents. The vector is a result of abstracting a text into descriptors-this can be done either by a knowledge worker or -keeping in mind the constraints from the business application we referred to in the introduction- by an automatic indexing [6,9]. Note that our approach only works in case of a controlled vocabulary of descriptors. Furthermore we discuss a type of background knowledge meeting the requirements of an ontology.

To keep our discourse comprehensive we define an ontology to be a set of terms and their relationships. An example of building such an ontology in an object-oriented fashion can be found in [8], for diverse definitions of an ontology we refer to [11].

To be precise, possible vector entries (index terms) in *v(D)* must represent a controlled vocabulary *V* to keep them computer-readable and capable of comparisons. The index terms of the vocabulary V are

exactly the concepts of a predefined ontology, connected by the ontological relations. The relations we perform with are typed semantic ones like 'is subconcept of', 'is differential of' or 'is associated with'. Example of an index vector: imagine a text-document *D* describing the German chancellor Schröder visiting the U.S., where he meets President Clinton and argues with him about the chair of the IMF. The vectorial representation *V(D)* is:

V(D)=

{THEMES: German foreign policy, Gerhard Schröder, IMF

INDIVIDUAL KEYWORDS: Gerhard Schröder, Bill Clinton, German government, U.S. government, IMF, Caio Koch-Weser

THEMATICAL BACKROUND-KNOWLEDGE: Germany, German government, SPD, international organizations, foreign policy

INDIVIDUAL BACKGROUND-KNOWLEDGE: German government, U.S. government, international organizations, USA, Germany}

The entries on THEMATICAL BACKROUND-KNOWLEDGE and INDIVIDUAL BACKGROUND-KNOWLEDGE depend on the modeling of the ontology, usually there are a more keywords listed. THEMATICAL BACKROUND-KNOWLEDGE refers to the key word from THEMES, INDIVIDUAL BACKGROUND-KNOWLEDGE belongs to the INDIVIDUAL KEYWORDS. Repetitions of keywords are possible, intended to strengthen the importance of a keyword.

3 SEMANTIC MATCHING

3.1 The algorithm

In contrast to classical full text retrieval technology our method provides more structure. As was to be seen from the last paragraph we include background knowledge, which delivers more than synonyms. A first version of the matching algorithm deals with a type of overlap-measuring of the entries of a pair of vectors. We named the measure 'frequency' because of the way its functionality was implemented in the Smalltalk programming language.

Let us define a frequency measure of the similarity of two sets of words as the number of words appearing in both sets (whereby every repetition of a word is extra-counted) divided by the total of all words. An example: (sun, sun, rain) and (sun, sun, snow) have the frequency 4/6.

The output *S(Q,P)* of the matching algorithm is the similarity of a pair of documents. In fact it is a weighted sum of similarities *S(a,f),...,S(b,i)*, where *a,...,d* are the collections of keywords (i.e. the vectorial entries) from the first index vector *V(P)* and *f,...,i* are the collections of keywords from the second vector *V(Q)*. We assume the operation on the *S(a,f),...,S(b,i)* to be a linear one, which means, that a linear regression is able to estimate the participating weights *t,u,v,w.* An estimation is necessary, because we do not know anything about the contribution of each single similarity to the whole. We summarize

$$S(Q,P)=tS(a,f)+uS(b,g)+vS(d,h)+wS(b,i) \qquad (1)$$

with the *t,u,v,w* to be estimated.

How do we get these weights ? We have to take a collection of pairs like *(P,Q)*, in our case we took a sample of size 50, and leave it up to a human to assign the respective similarities *S(Q,P)*. The rest is to be done via a multi-linear regression, minimizing sums of squared errors analogous to the well-known linear regression approach.

3.2 Improvement by feedbacking

Actually the following ideas are independent from guessing the weights *t,u,v,w* itself. Let us return to the environment, from which the regression was implemented. We already explained , that the indexing implying the vectors *V(D)* strongly depends on how far the ontology is developed. Thus the latter fact has also qualitative impact on the results of the matching algorithm. We focus on improving the algorithm by imroving the ontology.

First, a sub-optimal[1] approach for judging an *S(Q,P)* is taking as the value of similarity the percentage of positive answers (given by testing persons) to the question, if *Q* and *P* are similar. From now on we apply a way of grouping keywords, which is inspired by [3], where the authors themselves proposed to include background knowledge in their work. We make use of the 'interestingness'-measure. We want to group keywords, as the clusters with a high rate of interestingness should give hints concerning semantic relations between their participants. The exact semantics then have to be added by human.

Let us define the interestingness [2] of a set of keywords appearing in the same text document as the ratio of the probability of a set of keywords to the multiple of the probabilities of occurrence of the single keywords.

Two starting points of structuring the documents before extracting interesting clusters, a subjective and an objective one, shall finish our reasonings. A subjective pre-grouping follows from what the testing persons percept as similar: we only regard to

1. 'optimal' settings would be in contrast to quantifying individual and subjective judgements

clusters of keywords carrying a high average of interestingness in a collection *C* of similar documents. To find *C*, we must also cluster *the documents.*

On the other hand an objective pre-grouping is introduced by defining *C* via the thematic entries and clustering with respect to the theme. By objectivity in this case we denote selecting a structure given by the themes from the ontology. Here, a theme might consist of several keywords.

The last step is to present the interesting collections of keywords resulting from either grouping to an ontology engineer and to let him or her decide, if he sees a reason why the ontology might be improved by filling in relations he or she associates with the interesting groups of keywords. Note that our approach deals with strictly supervised learning.

4 CONCLUSIONS

From our rather optimistic point of view there clearly exist ideas how to attain at least clues for maintaining an ontology by reuse of the output and evaluation of a matching algorithm. So the feedback of such an algorithm is a human contribution to machine learning- detecting related keywords, which do not have a relation in the ontology yet. Of course the algorithm using background knowledge has to proof its strength- not only in matching documents, but also in case of a growing ontology- is it still exact, when there are many different relations to a keyword ? What are ontologies to master the semantic matching of documents from a special domain properly ?

Within further work would we like to confirm our idea about an interplay of automated retrieval and a human editor, for example by experimenting with a certain amount of new vocabulary, which could be classified to the ontology in our framework more easily.

Another way of improving the results is refining the indexing process by the introduction of an additional qualitative tagging of keywords in our vector representation. For example, if it is obvious, that special semantics of an entry is the only interpretation existing in a document, one cuts off background knowledge, which is not in the sense of the semantics, and gets a better preprocessing.

To end our brief discussion, we mention another field of research, namely the question of how we could derive hints, which point out redundant or even improper ontological. relations.

REFERENCES

[1] S. Borgo, N. Guarino, C. Masolo, G. Vetere: Using A large Linguistic Ontology For Internet-Based Retrieval Of Object-Oriented Components, Proceedings of the Ninth International Conference on Software Engineering and Knowledge Engineering, Madrid, 1997

[2] S. Brin, R. Motwani, C. Silverstein: Beyond Market Baskets: Generalizing association rules to correlations, Proceedings of the 1997 ACM SIGMOD Conference on Management of Data, 1997

[3] C. Clifton, R. Cooley: TopCat: Data Mining for Topic Identification in a Text Corpus, Proceedings of the PKDD 1999

[4] S. McClean, B. Scotney, M. Shapcott: Using background knowledge tn the aggregation of imprecise evidence in databases, Elsevier Journal of Data and Knowledge Engineering, Vol.32/2, 2000

[5] J. Piaget: Biologie und Erkenntnis, Fischer, Frankfurt/Main, 1992

[6] G. Knorz: Automatisches Indexieren als Erkennen abstrakter Objekte, Max Niemeyer Verlag, Tübingen, 1983

[7] M. Minsky (ed.): Semantic Information Processing, MIT Press, 1968

[8] L. Rostek, D. Fischer, W. Möhr: Weaving A Web: Structure and Creation of an Object Network Representing an Electronic Reference Framework, Electronic Publishing 6, 1994

[9] L. Rostek: Automatische Erzeugung von semantischem Markup in Agenturmeldungen, in: Möhr/Schmidt, SGML und XML, Springer, Heidelberg 1999

[10] G. Salton, M.J. McGill: Introduction To Modern Information Retrieval, McGraw Hill, New York, 1983

[11] J.F. Sowa: Knowledge Representation Logical, Philosophical and Computational Foundations, PWS Publishing Company, 1998.

[12] J. van den Berg, M. Schumie: Associative Conceptual Space-based Information Retrieval Systems, technical report, Delft, 1999