LANGUAGE TECHNOLOGY FOR INFORMATION SYSTEMS

Paul Schmidt, Mahmoud Gindiyeh and Gintare Grigonyte
Institute for Applied Information Sciences, University of the Saarland, Martin-Luther-Straße, Saarbrücken, Germany

Keywords: Automatic indexation, Automatic classification, Information retrieval, Language technology.

Abstract: The project presents work carried out in a project funded by the German Ministry of Economy whose goal is to develop tools for information systems on the basis of high quality language technology and standard statistical methods. The paper shows how automatic indexation, automatic classification and information retrieval can be combined to efficiently create a high quality information processing system for expert knowledge in technical engineering.

1 INTRODUCTION

The paper introduces an approach to building a high quality information system for expert knowledge in technical engineering. Methods from language technology (LT), a sophisticated thesaurus, statistical classification, and query processing are combined into an advanced information system. The work relates to (Strzalkowski and Carballo 1998), by using NLP though not on a TREC basis. The paper comes in four sections:

- Automatic indexation with LT
- Statistical classification
- A very short description of a use case
- Summary and conclusion

2 AUTOMATIC INDEXATION WITH LINGUISTICS

IR knows different approaches, vector space models, probabilistic approaches, linguistic approaches. The latter approaches focus on the detection of variants of terms in a document. Vector space models calculate similarity between a query vector and the document vector by a similarity measure. The determination of the document vector is the weighting of the content words of the document. The measure for that is often TF-IDF enhanced with a normalisation that takes the different lengths of documents into account. Another popular IR technique is the „Latent Semantic Indexing“ (LSI) which is also vector space IR with a reduced matrix. Probabilistic IR determines how probable it is that a document is relevant for a query. Our approach is basically a linguistic one. It is combined with a weighting approach which is not based on TF-IDF. The innovation of our approach is the sophisticated linguistic processing.

2.1 A Linguistic Approach to Automatic Indexation

A linguistic approach to automatic indexation tries to discover descriptors in a document. The main challenge is to discover variations of terms denoting the same concept. The main contribution of our system is that variations of terms such as exhibited in fig 1 can be systematically and reliably detected. All the term variations in fig 1 are reduced to the normalised thesaurus term in the box in the middle (“strategies for cost reduction in hospitals”).

The lower half shows how term variants in a query are reduced to the standard thesaurus term. The upper half shows how thesaurus terms are mapped on variants in the document.

A document whose author prefers to use a term like ‘hospital cost reduction strategies’ can still be indexed with the standard term. Linguistic processing reduces the author’s term to the standard term. The user query matches with documents that are about ‘strategies for cost reduction in hospitals’ by using any of the variations exhibited in the figure.

Schmidt P., Gindiyeh M. and Grigonyte G. (2009).
LANGUAGE TECHNOLOGY FOR INFORMATION SYSTEMS.
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval*, pages 259-262
DOI: 10.5220/0002273702590262

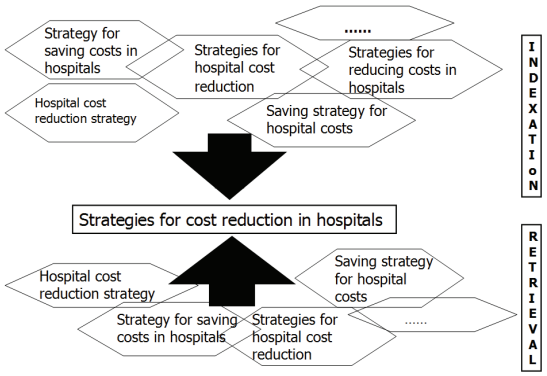

Figure 1: Term mapping.

Automatic indexation that is based on term detection requires the processing chain in fig 2:

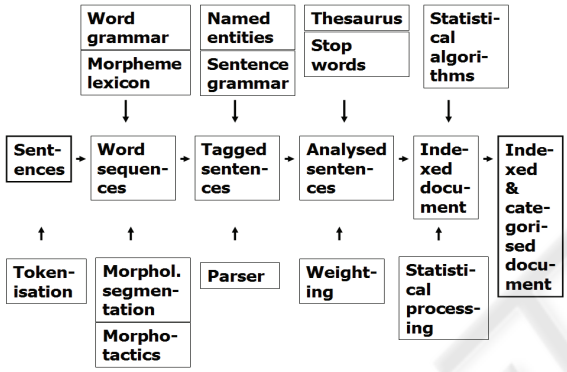

Figure 2: Indexation and classification architecture.

First step is a tokenisation, then a morphological analysis determines word structure delivering a (richly) tagged sentence. A grammatical analysis determines phrases, multi word units, and syntactic variations of compounds. A component not mentioned so far is a named entity recognition (NER) also on a linguistic basis (not be described here). After the detection of terms a weighting component determines descriptors' relevance. LINSearch uses a heuristic procedure (not TF-IDF). The relevant factors: Term frequency (TF): The more frequent a term the more important it is. Linguistic analysis allows for also taking compound parts into account. The weighting takes 'semantic classes' into account, the number of items of a specific semantic class to which a descriptor belongs. The idea is that if a text mainly exhibits a specific semantic class, say e.g. 'institution', then it can be expected that to be predominantly about institutions. Another factor is the place of a term in the text, e.g. in a heading or in plain text. A factor that is under research is how linguistic information is relevant, e.g. 'information structure of sentence' (theme – rheme). A statistical component described in the next section classifies the document.

2.2 Query Processing

The previous section described how term processing is used for indexation. Variants of terms are mapped on standard thesaurus terms. The query may have the variants again so that the same problem for retrieval occurs, namely the mapping on standard terms. This is shown in the lower half of fig1. Promising attempts have been made with query processing. Each query is indexed like an ordinary text document in the database delivering the terms for search in the database, so that 'hospital cost reduction strategy' leads to 'strategies for cost reduction in hospitals'.

Retrieval shows other problems as well. A query corpus exhibited many reasons that lead to 'no hits': For German there is the problem of morphological variants (Haus – Häuser). Then, it often happened that the user applied a wrong language parameter, searching English documents with German terms. Another problem was that the query contained orthographic errors. The first error type will be addressed by using a full analysis of both the query and the data base and do the mapping on the level of 'lexical unit'. A language detection finds out which language the query is. A spell checker provides correction proposals and suggests an automatic correction of orthographic errors. The effect of all the tools is a substantial reduction of 'no hits' (roughly 70%).

2.3 Evaluation

The basis of evaluation is a manual reference indexation of a 500 documents carefully done by 5 (!) experts (to provide consistency). The measure for evaluation is recall (results against the reference) and precision (the proportion of the correct items and the absolute number of items found).

Recall has turned out to be 40% for the reference corpus while precision is 28 % which seems not impressive. If synonyms (that are thesaurus terms) are accepted instead of the descriptors the situation improves to about 45%. Mere recall and precision is not sufficient however. Therefore a 'qualitative' evaluation has been done which is an evaluation by experts in terms of 'qualitative acceptability'. The general result is that automatic indexation is of such a good quality that the system will be used for everyday work.

2.4 Multilinguality

The database of documents to be handled contains German and English documents (to be indexed with German descriptors). A query in German is expected to deliver English documents. The thesaurus has translations of all descriptors and is supposed to function as a 'relay station'. The handling of multilingualism is illustrated in fig 3.

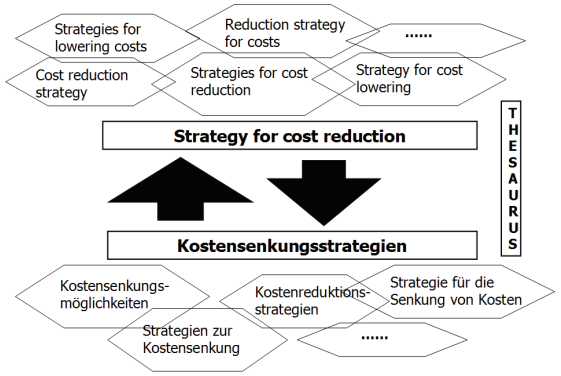

Figure 3: Multilingual term mapping.

3 STATISTICAL CLASSIFICATION

3.1 The Method

Our approach to classification follows standards of machine learning as in (Jurafsky and Martin 2008) and (Manning and Schütze 1999). 'Classification' is the assignment of a domain label to a document in our case e.g. 'semiconductor theory' or 'neuro networks'. Classes cannot be found in the text. A document about Mercury and Saturn does not tell 'I'm about astronomy'. Classes must be learned from terms (co-)occurring in documents of specific classes (determined manually). Thousands of documents that are manually classified are available for training a classifier. There is a total of 320 classes to be learned. In our case, the terms used for training are the descriptors of the document. So, the first step in classification is the indexation of the documents of the corpus. The result of the indexation is a set of descriptors with their weights assigned. This allows for representing each document as a vector in a vector space. A representation of two different documents a and b:
Document *a*: computer system[100], operating system[35], network[20].
Document *b*: network [100], system message[56], microprocessor[45].

$$\mathbf{a} = \begin{pmatrix} 100 \\ 35 \\ 20 \\ 0 \\ 0 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} 0 \\ 0 \\ 100 \\ 56 \\ 45 \end{pmatrix}$$

Figure 4: Document vectors.

The calculation of vector similarity is done on the basis of correlation:

$$\mathrm{Corr}_{XY} = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)}\sqrt{\mathrm{Var}(Y)}}$$

Figure 5: Correlation.

The covariance gives the direction of relations between vectors, but it tells nothing about its strength. Correlation standardises this to a scale between -1 (perfect contradiction) and +1 (perfect match).

The next step is to represent classes as vectors, consisting of the terms and their mean weights. The weight of a term in a class vector is the mean of all weights in its relation to all documents of this class.

$$\bar{g}_{ij} = \frac{1}{N}\sum_{k=1}^{m} g_{ik}$$

Figure 6: The mean weight of term t_i in class C_j .

N:: The number of documents of class C_j
m: The number of documents where t_i occurs
g_{ik}: The weight of term t_i in document d_k So: The new mean class of class C_j (as vector) is

$$\mathbf{C_j} = \begin{pmatrix} t_1 = \overline{g_{1j}} \\ t_2 = \overline{g_{2j}} \\ \vdots \\ t_i = \overline{g_{ij}} \\ \vdots \\ t_s = \overline{g_{sj}} \end{pmatrix}$$

S: The number of terms in class C_j

Figure 7: The new class C_j as vector.

There are two methods for building the class vector: Either all terms are taken that build a class or only the best terms per class. i.e. those that fulfil a relevance criterion ($(Sort\_t_{ij}) \geq q$; q: threshold*)*

$$Sort\_t_{ij} = \bar{g}_{ij} * p_{ij}^{\,2}$$

Figure 8: Sort value.

Where: $\overline{g_{ij}}$ mean weight of term t_i in class C_j,

p_{ij}: probability of term t_i in class C_j

The advantage of this approach is that the probability of term t_i in class C_j has a higher effect in the sort value than the mean weight. The classification of new documents is done according to its probability to belong to a class C. The document is first converted into a vector. Then, the similarity to class vectors is calculated. The most similar class (vector) is considered the most probable class C for the document to be classified.

Considering the 500 reference documents and taking the 6 most probable classes (of automatic classification into account we have a recall of 75%.

3.2 Descriptor Disambiguation

Classification can be used to disambiguate descriptors as it is delivered independently of the indexation. There are ambiguous descriptors like: TCO (Total cost of ownership) - TCO (transparent conductive oxides).

One might think that it is sufficient to determine that the descriptor is 'TCO' without disambiguation. However, if a user is searching for information about 'TCO (transparent conductive oxides) (s)he would not like to grind through a large set of documents including the second reading of 'TCO'. The first reading is about financial management, the second about electrical engineering. The recipe for disambiguation is to determine the classification which is then used for disambiguation. So, for the second reading we have 3ELB (direct energy conversion), 3KXE (magnetic material properties). This is different from 3 AD (financial management).

4 APPLICATION SCENARIO

This section introduces the application into which the system is integrated, a specialist information system for technical engineering. The database production system is a web based system which allows for an integration of LT into the overall work flow at two places as shown in figure 9.

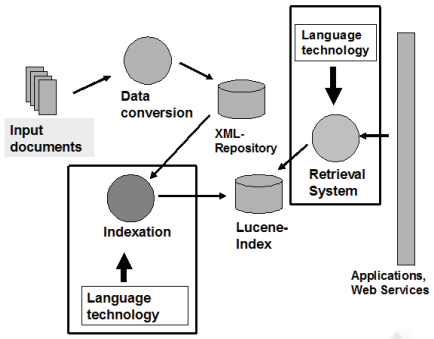

Figure 9: Overview of the application.

5 CONCLUSIONS

This paper has presented an approach that uses high quality language technology and combines it with standard statistical models. The approach is integrated into the workflow of a specialised information provider and provided satisfactory results to an extent that manual indexation could be replaced and retrieval substantially improved. Improvements of the system go into different directions: One is to go deeper into linguistic structure such as taking 'information structure of sentence' into account. Another direction is to integrate the system with a structured ontology. An extensive evaluation through a retrieval experiment is on it s way.

REFERENCES

Jurafsky, D., Martin, J.D. 2008. Speech and LanguageProcessing, Upper Saddle River, New Jersey.

Manning, Ch.,Schütze, H. 1999. Foundations of Statistical Natural Language Processing, MIT Press Cambridge, MA.

Salton, G.; McGill, M. J. 1983. Introduction to Modern Information Retrieval. McGraw-Hill, New York.

Strzalkowski, T., Carballo, J.P. 1998. Natural Language Information Retrieval: TREC-5 Report. In Text RetrievalConference.