Information retrieval based on Explicit Knowledge Representation

Elzbieta Dura and Marek Drejak
Lexware Labs, Göteborg, Sweden

Abstract

The tool which we tested in the present monolingual retrieval task, Lexware®, is based on explicit knowledge representation not on statistic language modeling. In the present task Lexware® indexing seems to be satisfactory while its query builder is not. The system has been tested extensively on indexing of Swedish parliamentary debates with very good results. We are happy that Swedish is finally introduced into CLEF, unfortunately the present test suite is not reliable. Swedish parliamentary debates may perhaps be used instead. They are many, constantly growing and they are thoroughly indexed manually with keywords chosen from a thesaurus of about 4000 items.

Knowledge-based information retrieval

Lexware® is rather unusual among information retrieval tools because it is based on explicit knowledge representation. Its comprehensive representation of Swedish enables it to make sense of Swedish texts, which we believe to be a precondition for obtaining satisfactory precision in information retrieval. Lexware® has about 80 000 vocabulary entries, with relations of hypo/hypernymy, synonymy, derivations, word components. Etymology and content categorization are provided for majority of entries. Supplementary word lists include about 50 000 non-appellatives like names of people, places, organizations, etc. The representation of Swedish inflection corresponds to 800 000 forms. Grammar includes 400 word formation rules, 500 general phrase rules and 700 collocational patterns. Basic glossaries of English, French, German, and Latin are also provided.

A similar monolingual information retrieval task

Lexware® is applied in an information retrieval task which bears much resemblance with the present one. The parliament library (Riksdagsbiblioteket) designed and conducted tests of software in order to determine whether manual indexing of the parliaments' documents can be supplemented or even substituted by automatic indexing. The task was to select proper keywords from a thesaurus specially created for the parliament's documents. Keywords are limited in number, from 1-10 and not only are they supposed to identify the main subject but also do so with a proper level of generality in the thesaurus hierarchy.

Software from Connexor, Lingsoft, Kungliga Tekniska Högskola and LexWare Labs participated in the tests. The evaluation was based on a comparison of keywords assigned manually and automatically by the tested programs. The overlap in keywords assigned manually by two different indexers was 34%. LexWare® proved to obtain the best F-value (2*precision*recall) / (precision + recall)): 36%, Kungliga Tekniska Högskola 32%, Connexor 22%, Lingsoft 19%.

The fully developed application for indexing parliament documents proves to have surprisingly high coverage with full precision. 80% of documents the keywords assigned by LexWare® are the same (50%) or very closely related in the thesaurus to those assigned manually. These results are mainly due to the integration of task specific knowledge - in this case the thesaurus - with LexWare®'s own rich language representation. Thesaurus terms are identified as occurrences of terms closely related in the thesaurus or in the lexicon (synonyms).

Swedish parliamentary debates are indexed by Lexware® not only with the thesaurus terms but also with all content words, all of which are made available for querying. This retrieval can be checked on www.lexwarelabs.com. The enclosed picture shows a list of matches in a query ”kompetens” (”competence”). The list is produced after the initial query in order to facilitate a precise statement of a follow up query. The lexical character of the system can be appreciated among others in that all synonymous expressions (”behörighet”, ”befogenhet” besides ”kompetens”) and compounds are automatically present in the query. For each matched word on the proposed list the number of documents matching is provided in parenthesis.

The present monolingual task

Articles are indexed with content words, phrases and proper names. It is important to remember that text occurrences are collected per vocabulary item, not per each inflectional form. A query is built in a similar way: content words, phrases and proper names are fished out from a query text. For each query item all related items of the vocabulary are added to the query. This is to ensure that the search is content oriented rather than dependent on the specific wording present in the query text. Each item in a query is associated with a relevance weight. Total relevance of an article for a given query is calculated as a sum of the weights of each query item matched in the index. Only articles that pass a threshold of the total relevance weight are selected.

Building of a query

How a query is built from the description of the topics is explicated with an example. An English version of the original query text is
<title> Sex in advertising </title>
<desc> Look for articles on the kind of advertising that attracts attention by sexual allusions, and for articles on the social consequences of such advertising. </desc>
<narr> Relevant documents take up the problem of excessive use of openly sexual allusions in advertising. That such advertising can indirectly be responsible for sexual abuse or violent actions is also relevant. </narr>

A translation of a counterpart Lexware® query ”Sex in advertising”:
exists:
    15 advertising
not_exists:
plus_intersection:
    25 sex life
    10 sexism
    25 sexual
    15 erotic

15 erotism
1 violence
10 violent action
25 woman abusive
1 aggressive
1 aggressivity
1 attack
10 allusion
10 allusive
10 alluding
25 sex-related
25 sex
10 woman
1 problem
1 society

minus_intersection:

A Lexware® query is basically a list of weighted keywords subdivided into the following groups:
”exists” – must occur in an article,
”not_exists” – must not occur in an article,
”plus_intersection” – contributes to relevance,
”minus_intersection” – counteracts relevance.

Proper names and non-general content words of the title part of the query text are classified as ”exists-class”. Other words in non-negative context constitute ”plus_intersection”. In the present example neither of the negative groups are filled, because there are no restrictions in this particular query text. ”Minus_intersection” is present in 12 of the 50 queries in the present task. Lexware® parses each sentence of the query text and checks it for presence of phrases characteristic of a negative group when classifying words. The set of such phrases is limited and Lexware® may miss an unusual formulation of a negative condition.

Each query text is analyzed linguistically. For vocabulary items recognized in a text as valuable content words all their synonyms, derivations and other related words are added. Rich non-general content nouns are most ”valuable”, function verbs are least ”valuable”. Weights depend mainly on where in the query text the word appears: in the title, description or narration part.

Evaluation of the evaluation

There is a minor bug in the evaluation program. Whenever Lexware® retrieves 0 documents the program counts 1 document as retrieved and 0 documents as relevantly retrieved (queries: 91, 105, 110, 111, 121). 0 articles was retrieved for query 109 both in the test suite and by Lexware® – this result is totally missing in the result list. We are happy that Swedish is brought into CLEF but in order for the tests to have any significance at all a test suite must have an acceptable reliability, which we doubt it has in the present shape.

There were only 50 queries so it was easy to go through the results of Lexware® query builder manually. We could see directly which of the queries were good and which were almost worthless. Therefore we were surprised by poor results reported for good queries, which in turn made us check manually some of the results. Our findings show that this year’s Swedish test might not be reliable at all.

It is difficult to determine relevance intersubjectively and therefore we have gathered only extremely obvious cases: when an article on the topic was missed by obvious mistake or when an article without a slightest mention of the topic was qualified by mistake. We have also noted cases when a sheer mention of some keyword of a query is sufficient to qualify an article for a query, and when there is a mention but an article is not qualified. For instance, Pisa tower is mentioned in TT9495-950921-238402 but the article is not qualified for a query about Pisa tower, while mention of extradition to Peru in article TT9495-940706-175258 on Swedish refugee policy qualifies the article for query on human rights abuse in Latin America.

In a manual check of some of the results we have found that 53 articles were qualified by mistake, 19 fully relevant articles were left out by mistake. A mention of the query subject was sufficient to qualify 40 articles as

relevant, and at the same time 32 articles mentioning the query subject were not qualified as relevant. The number of different articles is lower than the number of article texts because some texts are repeated - we encountered 3 pairs of the same articles. After checking the results for about half of the queries the numbers are following. All relevant articles are 1219, retrieved by Lexware® 463 and relevant retrieved 281.

Evaluation of query building

Even if query building and indexing are evaluated jointly in the present task, it is fairly obvious that it is the quality of queries that is lacking in Lexware® and not the quality of indexing.

Some of shortcomings resulted from the focus of our test. Our objective was to check how much content the system can fish out from the texts. Information about articles, e.g. dates, was neglected, which caused some overgeneration. Many articles were rejected because of the priority given to highest possible precision. For instance, all of relevant articles on asthma were found by Lexware®, but none of them was included because of the restriction of the query to bronchial asthma, not to asthma in general.

One obvious miss detected in Lexware® queries depends on its lack of knowledge, but this is rather easy to repair. External thesaurus can easily be introduced and integrated in Lexware®. For instance if the query about reports from Amnesty International in Latin America included all countries and cities of the continent it would have been trivial to retrieve relevant articles.

Not all misses in query building can be repaired so easily. When queries involve a kind of meta-language - general instructions for retrieval, there is no easy way to translate this with the help of simple knowledge of vocabulary. For instance, one query requires that the reasons for some event should be mentioned. Another non-trivial problem are very general concepts. For instance, what keywords should be chosen for ”immaterial property”? Yet another hard problem is the fact that some of our language knowledge lies in default senses associated with words, seldom explicitly represented in vocabularies. For instance, Lexware® retrieved an article on gold medals in the Paraolympics in Lillehammer as relevant for the subject of gold medals in the Olympics in Lillehammer. This article was not qualified as relevant in the test suite (TT9495-950122-277443).

Evaluation of retrieval

Lexware® has almost no elaboration of statistics, which is its weak side when it comes to judging relevance weights and establishing a relevance threshold. For instance, query 121 on Ayrton Sennas was extremely easy to find because it was enough to identify this proper name in the text. Lexware® retrieved the relevant article but it was rejected by the relevance threshold. High precision is set as the main objective of Lexware® and it proved that the relevance threshold was often set too high. In cases when articles involved not one but many topics, e.g. articles on various foreign affairs in short, the relevance weight of query items was too low to pass the threshold. This lack of flexibility of the relevance threshold was responsible for excluding of all articles with many subjects.

Too high threshold was also responsible for excluding with only a mention of a query subject. In the test suit an article seems to have qualified for a query even on a mention if there were not many articles available otherwise. Lexware® did not have this kind of flexibility in its judgment. For instance, there were 5 articles assessed as relevant for the query on Eurofighter in the test suite. Lexware® rejected two of these articles because they were not on Eurofighter: one mentioned the plane as an example, another as parenthetical information. On the other hand when there were more articles on the subject, a mention was not this was not recognized as sufficient. For instance there were 20 articles qualified for query ”Sex in advertising” and an article in which the Pope condemns sex in advertising (TT9495-950710-228275) did not qualify.

Negative contexts in articles constitute a problem as well. For instance, Lexware® selected an article in which it is stated that Germany refuses to provide armed forces for a mission abroad (TT9495-950119-203346) for the query on German forces in foreign assignments. This article was not selected in the test suite. A similar example is an article in which it is assessed that cellular telephones cannot be used by people with pace makers (article TT9495-950428-218385), which Lexware® qualified as text on possible uses of cellular phones. If the query is on whether and how a spy affair had an impact on Soviet-US relations, does a text stating explicitly that the

affair was not taken up during some top meeting state also that the impact was weak or none (TT9495-940314-158787)?

The notorious problem of overgeneration caused by metaphors is not easy to solve for a strictly knowledge based system. For instance Lexware® qualified a text about a divorce between Renault and Volvo as an article on divorces.

Conclusions

Evaluation is perhaps the most important tool in improvement of retrieval tools. Even if the test suite may have had too poor quality for checking the strong side of Lexware®, namely indexing, but it was very helpful in checking the weak side of the system, namely its building of queries. Some of the conclusions are rather obvious. The more knowledge is provided the better a knowledge based system gets. For instance, if a query mentions Latin America the system must know what states and main cities belong to the continent. This complement is easy to introduce because Lexware® is designed to incorporate an external thesaurus into its knowledge base.

There are queries which are notoriously difficult for the Lexware® approach. In some cases there can be no meaningful expansion of a query with content words that are lexically related to the ones present in the query. Integration of Lexware® with a system based on statistic language modeling might be helpful not only in query building but also in establishing total relevance weight. Negative contexts and silent defaults seem to constitute a difficulty in any approach.