Information Retrieval With Language Knowledge

Elzbieta Dura and Marek Drejak

Lexware Labs, Göteborg, Sweden
elzbieta@lexwarelabs.com

Abstract. The introduction of Swedish made it possible for Lexware® to be tested for the first time in CLEF. Lexware is a natural language system applied in an information retrieval task and not an information retrieval systems using NLP techniques, therefore it is interesting to compare its results with other less odd IR systems. We experience that separate evaluation of document description and query building would provide yet better testing for our system.

1 A Natural Language System for Swedish

Lexware is a natural language system applied in an information retrieval task and not an information retrieval systems using NLP techniques, like e.g. NLIR [4]. It can be considered odd also among natural language processing systems, if the latter are assumed to focus on syntactic analysis (c.f. [3]). Text-analysis is shallow and it is not demanding in terms of computing power and storage [1]. The strength of the system is its rich lexicon and the possibility to expand the lexicon with external items without negative impact on access time [2]. The vocabulary of about 80 000 lexical items is richly interconnected by relations of form and content: derivational origin, synonymy, components for complex items, hyponymy. Content words are categorized into about 100 content categories. There are also supplementary word lists which include about 50 000 non-appellatives like names of people, places, organizations, etc. plus basic glossaries of English, French, German, and Latin. 400 word formation rules cope with inflection, compounding and derivation, 500 general phrase rules plus 700 collocation patterns are used to disambiguate and to determine modifier–head roles.

2 Lexware in Another Information Retrieval Task

Lexware has been extensively tested in another information retrieval task. The library of the Swedish parliament – Riksdagsbiblioteket, designed and conducted evaluation of software that could supplement or even substitute manual indexing of the documents of the parliament. The task is to select proper keywords among descriptors in a thesaurus specially created for this kind of documents. Keywords are to be limited in number, from 2-10 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. For

instance, when a document takes up university education the term *university education* and not *education* should be picked from the thesaurus.

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 to the same documents by two different indexers. The overlap in keywords assigned by the two indexers was only 34%. LexWare proved to obtain the best F-value: 36%, Kungliga Tekniska Högskola 32%, Connexor 22%, Lingsoft 19%.

Recent tests of the fully developed Lexware application for indexing of parliament documents proves to have very high coverage with full precision. Keywords automatically assigned were compared with the manually assigned ones in 1 400 documents obtained from Riksdagsbiblioteket. 80% of keywords assigned by LexWare are the same or very closely related in the thesaurus to those assigned manually.

Lexware is also applied in retrieval of documents of the parliament without the thesaurus. All content words are used in document description and are made available for search. The screen dump shows a list of content words presented to the user for the search word *kompetens* (*competence*). Each word is presented with the number of documents it occurs in. Compounds and synonymous expressions for *kompetens* (*behörighet*, *befogenhet*) are included in the list (c.f. http://www.lexwarelabs.com).

This type of two step query is meant to assist an "average user" of a search engine, who according to Google statistics does not look over more than one result screen (85%), leaves the query unmodified (78%), is very "concise" (2.35 terms on average), does not use much syntax (80% queries are without operator).

One role of the lexicon is to provide abstract terms another one is to provide these with precise relevance weights. Each lexical item has its specific semantic load. For instance content words used as part of lexicalised compounds or set expressions are often devoid of all content. Verbs of the kernel vocabulary contribute with little or no content even the ones which are not function words at all. This kind of lexical knowledge used by Lexware (besides corpus statistics) is decisive for precision.

3 The Present Task

The present task is approached in a similar way. Articles are analysed and provided with a description in weighted abstract terms: lexemes, lexical phrases and named items. Each query text is analysed linguistically in a similar way as the articles. Vocabulary items recognized as meaningful content words are included together with items related to them, like synonyms, derivations, etc. Weights are assigned to each item in a query depending also on where in the query text it appears: in the title, description or narration part. The total relevance of an article for a given query is simply calculated as a sum of the weights of each query item matched in the document description. Only articles that pass a relevance threshold are selected.

A query is created as a list of weighted terms 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 meaningful content words of the title part of a query text are classified as *exists*. Other words in non-negating context constitute *plus_intersection*.

4 Evaluation of Query Building

Even if query building is not evaluated separately, it is fairly obvious to us that it is the quality of queries that is primarily lacking in Lexware and not the quality of document description. On the other hand we find it highly improbable that queries in practical applications will soon have the form of queries in the present task.

A query about bronchial asthma is a good example of how trivially easy retrieval is made impossible by an improper query. In the case of *bronchial asthma* the modifier is almost devoid of content because the default use of the word *asthma* is exactly *bronchial asthma* but Lexware excludes articles on *asthma* as not sufficiently specific. A similar example of default use assumption, this time on the part of the test-suite, is the belief that articles on gold medals in the Paraolympics in Lillehammer are not relevant for the subject "gold medals in the Olympics in Lillehamer". It is not obvious at all how default use should be coped with.

Some misses in query construction depend on lack of world knowledge. Some of world knowledge is necessary and not difficult to include in the system. For instance, if country and city names were linked with the name of the continent in Lexware representation no articles about reports from Amnesty International in Latin America would be missed. The disregard of important meta-information about articles, like publication dates, is also an example of Lexware mistakes which are rather easy to repair.

But there are problems that are not easy to eliminate. Queries which involve some kind of meta-information are difficult to cope with lexically. For instance, one query requires that the reasons for an event should be mentioned. How does one expand *reasons* into a list of vocabulary items? Very general concepts are also difficult: which lexical items are counterparts of *immaterial property*?

5 Evaluation of Retrieval

The threshold of relevance was set high in order to maximize precision. Articles were rejected even in cases when only few were retrieved. For instance, proper name identification was not sufficient by itself. Another example of too sharp cutting are articles on many topics, such as "foreign affairs in short", all of which were rejected. The same refers to articles with only a mention of the subject: in parenthesis, or as an example, etc. It would be reasonable to have a flexible relevance metrics dependent on the availability of articles on a subject. This approach is adopted in the test-suite. Articles are qualified for a query even on a mention in cases when only few ones are available otherwise, but not when many articles are available on the subject.

It is difficult to determine whether negative information should be deemed relevant. For instance, Lexware selects an article in which it is stated that Germany refuses to provide armed forces for a mission abroad when German forces in foreign assignments are queried. A similar example is an article in which it is assessed that cellular telephones cannot be used by people with pace makers, selected by Lexware for a query about possible uses of cellular phones. Is an article stating that a spy affair was not taken up during some top meeting relevant for a query about whether and how a spy affair had an impact on Soviet-US relations? Metaphors are also difficult to cope with. For instance an article about a divorce between Renault and Volvo is qualified by Lexware as an article on divorces. This problem borders on the problem of default language use – divorces are for people not for cars.

6 Evaluation of 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 missing in the result list.

Since there are only 50 queries it is easy to go through the results of query building manually. One can see directly which queries are properly constructed and which are

almost worthless. Therefore it was surprising for us to see poor retrieval results when queries were very good, which in turn made us check manually some of the results. We made checks only in cases in which our queries seemed to be properly built, which is results for about half of the queries. The list of our findings can be made available for corrections of the test-suite. Only extremely obvious misses are considered: articles explicitly on a topic not marked as relevant or articles without a slightest mention of the topic qualified as relevant. 53 articles are marked as relevant by mistake, while 19 fully relevant articles are left out. There are less articles than texts because some articles are repeated - we encountered 3 pairs of the same articles. After corrections the total number of relevant articles is 1219, 281 of which were retrieved by Lexware of 463 totally retrieved articles.

7 Conclusions

Evaluation is perhaps the most efficient way to improve both NLP and IR systems but it seems to be appreciated mainly by IR practitioners. It is crucial that monolingual systems for small languages like Swedish can also be tested, even more for NLP than for IR systems. The Swedish test-suite initiated by Jussi Karlgren is extremely valuable to all researchers working with Swedish and it is very important to improve it and develop further.

It would be desirable for Lexware if query construction could be tested and evaluated separately from retrieval. The type of querying that should also find its place in testing are queries closer to the ones stated by an average users of search engines. Testing Lexware in the present task helped us to appreciate the difficulties in query construction, some of which are notoriously difficult for our lexical 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. Perhaps the completion required for Lexware in such cases is statistic language modelling.

Considering deficiencies in our query construction and too highly set threshold the results of Lexware are not bad: the average F-value for all topics is 41% with corrections of test-suite, and 37% without the corrections.

References

1. Dura, E. 1998. Parsing Words. Data linguistica 19. Göteborg: Göteborgs universitet.
2. Dura, E. 2000. Lexicon-based Information Extraction with Lexware. In: PALC99 Proceedings.
3. Sparck Jones, K. 1999. What is the Role of NLP in Text Retrieval?. In: Strzalkowski, T. (ed.): Natural Language Information Retrieval.
4. Strzalkowski, T., Perez-Carballo, J., Karlgren, J., Hulth, A., Tapanainen, P., T. Lahtinen, T. 2000. Natural Language Information Retrieval: TREC-8 Report. Online at: http://trec.nist.gov/pubs/trec8/t8_proceedings.html