

Department of Linguistics, Uppsala University
Language Engineering Programme

A Meta Search Approach to
Locating and Classifying Reading Material
for Learners of Nordic Languages

Master's Thesis

Kristina Nilsson
krinil@stp.ling.uu.se

February 27, 2003

Supervisor of Master's Thesis: Lars Borin,
Department of Linguistics, Uppsala University

Abstract

Research in reading as a psychological process indicates that the quality of the processing of target language input can be improved if the reader finds the text personally significant. By providing students with authentic texts in their own subjects, and by allowing them to share the responsibility for their learning, this can be accomplished. This thesis describes the development of Squirrel, a prototype for locating authentic texts in the Nordic languages published on the World Wide Web. It was designed to facilitate the process of producing authentic learning materials for second language learners. The primary target users are teachers and exchange students to Nordic institutions of higher education. A commercial search engine is used for meta searching the Web. Given a HTML document supplied by the user, Squirrel extracts possible query terms from the most frequent content words of the text and from the HTML meta data. The user chooses which terms to use, and the query is sent to the search engine. The returned results are collected and classified according to language and difficulty level. Finally, the user is presented with a list of links to documents that might suit the user's need. Although evaluation involving real users remains to be done, initial results are encouraging. Recall and precision of the automatic language identifier range from 92-100 percent. The methods chosen for query term extraction and readability testing are well-tested, but must be further evaluated.

Acknowledgments

The work described in this thesis was carried out at the Department of Linguistics at Stockholm University as part of the project *Corpus based language technology for computer-assisted learning of Nordic languages*, in the framework of the Nordic Language Technology Research Program 2000–2004, funded by the Nordic Council of Ministers through Nordisk Forskeruddannelsesakademi (NorFA). I would like to thank my supervisor Lars Borin for well-founded suggestions and support throughout this project, and Mats Dahllöf at the Department of Linguistics at Uppsala University for useful suggestions regarding the text. I would also like to thank Sofia Gustavsson-Capcová and Jennifer Spenader at the Department of Linguistics at Stockholm University for their enthusiastic encouragement. A special thank you to Eva Forsbom for suggesting references, and to Stina Åberg and Katarina Wahlund for sharing all the ups and downs. Finally, thank you to my family and all my friends at the Language Engineering Programme at Uppsala University.

Contents

1	Introduction	1
1.1	Purpose	2
1.2	Outline of the Thesis	2
2	Background	3
2.1	Authentic Texts and Reading Comprehension	3
2.2	Basics of Web Information Retrieval	4
2.2.1	Crawling and Indexing	5
2.2.2	Clustering and Ranking	6
2.3	Linguistic Knowledge in Web Information Retrieval	7
2.4	The Nordic Languages on the Web	9
3	Squirrel 1.0	13
3.1	A Session from a User Perspective	13
4	Implementation	17
4.1	Architecture	17
4.2	Meta Searching	19
4.3	Preprocessing	20
4.3.1	HTML parsing	20
4.3.2	Tokenization	21
4.4	Automatic Language Identification	21
4.4.1	Implementation	22
4.4.2	Limitations	23
4.5	Extraction of Query Terms	23
4.5.1	Implementation	25
4.5.2	Limitations	26
4.6	Reading Comprehension and the Readability of Texts	28
4.6.1	Implementation	29
4.6.2	Limitations	30

5	Performance Evaluation	33
5.1	Extraction of Query Terms	34
5.2	Automatic Language Identification	35
5.3	Readability Testing	35
6	Concluding Remarks	39
	References	41

List of Tables

2.1	The largest search engine indices as of December 11, 2001	5
2.2	High frequency words in the Nordic languages	10
2.3	Search results for high frequency words in the Nordic languages	11
4.1	Language model training material	22
4.2	Online news sites used for collecting training material	23
4.3	Term frequency list without stop word elimination	27
4.4	Term frequency list after stop word elimination	27
4.5	Lix text classification	30
5.1	Automatic language identification: Precision and Recall	35
5.2	Readability: Text Characteristics for Readability Testing	36

List of Figures

2.1	The origin of the Scandinavian languages	10
4.1	Squirrel: overview of example text evaluation	18
4.2	Squirrel: overview of meta search and retrieval	18
4.3	Squirrel: the meta search method	19
4.4	The Luhn curve: relating word frequency and word rank order	24
4.5	Example of HTML meta data	25
4.6	Possible query terms: meta keywords and high frequency terms	25

Chapter 1

Introduction

The feasibility study behind this thesis focuses on how to automatically locate and classify text sources published on the World Wide Web, in order to produce authentic learning materials for second language learners of Nordic languages.

Although the Nordic languages are small languages in terms of number of speakers, they are well represented on the Web with millions of documents in each language, and one can safely assume that these numbers will continue to grow. Based on surveys published over the last two years, Nua Internet Surveys¹ estimates that there were 580.78 million Web users as of May 2002 (Nua, 2002). Correspondingly, the number of hosts and Web pages available for these users appears to be growing at an exponential rate, and there is no doubt that the amount of publicly available information on the Web, as well as the number of people looking for information online, continue to grow at a very high rate. Thus, there is reason to believe that the Web can serve as a source of authentic texts for learners and teachers of the Nordic languages, but the task of locating and classifying texts must be facilitated. When the use of authentic material in language learning was first proposed, the lack of availability of genuine written data was a problem. Today, with so much authentic text publicly available on the Web, the pedagogical problem is the opposite: “to evaluate and grade what is available, so that students are not overwhelmed” (Crystal, 2001, 235).

The work described in this thesis was carried out at the Department of Linguistics at Stockholm University as part of the Squirrel Project, or officially: “Corpus based language technology for computer-assisted learning of Nordic Languages”, in the framework of the Nordic Language Technology Research Program 2000–2004, funded by the Nordic Council of Ministers through Nordisk Forskeruddannelsesakademi (NorFA). A paper about the Squirrel project, written by Nilsson and Borin (2002), has been previously published.

¹Nua Internet Surveys <<http://www.nua.ie>>

1.1 Purpose

The aim of this thesis is to describe the development of a prototype for collecting text materials published on the Web, and classifying them according to language and difficulty level. The development of a search engine from scratch lies well beyond the scope of this project. Instead, a commercial Web search engine is used in a meta search approach.

The primary target user group consists of teachers and exchange students to Nordic institutions of higher education. Teachers and students of Nordic languages as a foreign language at institutions outside of the Nordic region might also be added. This target group can be considered to be fairly homogeneous as to educational background and computer literacy.

This prototype is supposed to facilitate the text selection process for the user by providing a collection of links to texts that might suit the user's need. The creation of a user-friendly interface - either in English or in each of the six languages - to such an application is a very important task that must be left for further development. Similarly, a full-scale user evaluation of the prototype is beyond the scope of this thesis, but informal test results are presented, and improvements based on evaluation of these results are suggested.

1.2 Outline of the Thesis

Chapter 2 contains some background information; the use of authentic text in a second language learning context is introduced in section 2.1. In section 2.2 the basics of Web information retrieval are described, followed by a discussion in section 2.3 on methods using linguistic knowledge that might enhance existing information retrieval systems. The representation of Web documents in the Nordic languages is discussed in section 2.4, along with a brief description of the languages in question.

A session with example test results is described from a user perspective in chapter 3. In chapter 4, the implementation of Squirrel 1.0 is presented, with focus on three areas: automatic language identification in section 4.4, extraction of query terms in section 4.5, and readability testing of texts in section 4.6. The methods used for meta searching and text preprocessing are discussed in sections 4.2 and 4.3. Test results are discussed, and improvements and ideas for the future are outlined in chapter 5, followed by concluding remarks in chapter 6.

Chapter 2

Background

The Web is an excellent source of authentic materials in many different languages, even though English is the dominating language at present. However, the huge amount of available information makes search services necessary. Web Information Retrieval (Web IR) is different from traditional IR in that most IR methods have been developed for relatively small, homogeneous and controlled text collections, whereas Web IR methods must handle the constant change in the distributed structure of the Web, as well as its huge, heterogeneous and uncontrolled content.

The following sections introduces the use of authentic texts in language learning, as well as the basics of Web IR. Crawlers and indices, the main components of the commonly used crawler-indexer architecture, are described, as well as standard methods for clustering and ranking of documents. Following this brief outline, methods using linguistic knowledge that could be used to enhance the performance of existing Web IR systems are discussed. Finally, the representation of documents in the Nordic languages on the Web is also discussed, together with a brief description of the languages in question.

2.1 Authentic Texts and Reading Comprehension

Tornberg (1999) defines an *authentic text* as a carrier of a communicative intention, in that it transfers a message in a certain linguistic “outfit”, and similarly Little et al. (1989, 25) define authentic text as “created to fulfill some social purpose in the language community in which it was produced.” For this project, following these definitions, authentic text is defined as as text written to convey information, rather than to supply the reader with language data in order to illustrate particular language phenomena.

Lightbown and Spada (1997) argue that while too difficult reading material might prevent the reader from fully understanding the content, the difficulty level must not be too low either, since this might lower the motivation of the reader. This is not restricted only to the linguistic aspects of the text: In the experience of Alderson and Urquhart (1984, 197)

“educationally sophisticated FL¹ learners tend to be put off by the, for them, inaccuracies and over-generalizations of simple accounts, particularly when these relate to their own subject.” If the students are allowed to share the responsibility for their learning through the use of Web resources such as the prototype described in this thesis, their motivation will hopefully be heightened. Psychological and psycholinguistic research strongly indicate that the quality of reading as psychological processing of the target language input depends on whether the reader finds the text personally significant, that is, if the text relates to the reader’s background knowledge and experiences, interests and information need. According to Little et al. (1989, 6–7) this can be accomplished by using carefully chosen authentic texts: “Precisely because [authentic texts] come complete with the savour, stench and rough edges of life beyond the school walls, they are likely to be markedly more successful in provoking pupil reaction and interaction than the somewhat anaemic texts that one so often finds between the covers of textbooks.”

It is important to note that Little et al. emphasize that authentic texts must be carefully chosen. The linguistic difficulty level of a text can be measured by readability formulas, commonly based on surface linguistic features believed to correlate with syntactic and lexical difficulty. Other aspects that influence the overall level of difficulty, and even more so the suitability of a text, can only be determined by human judges. It seems unlikely that the task of selecting authentic texts suitable for language learners could be completely computational. The final selection must be performed by human judges, in this case either teachers or students. However, allowing the students to share the responsibility for their learning with the teacher does not in any way lessen the importance of the teacher. As Porter (2000, 321) states: “it is not enough simply to make resources available to [the students]; the role of the teacher is crucial in ensuring that real learning happens when students interrogate web resources.”

2.2 Basics of Web Information Retrieval

The World Wide Web is a distributed collection of electronic documents in various formats. Currently, the most frequent document format is the hypertext markup language (HTML). The documents are accessible through a standard hypertext transfer protocol (HTTP), which allows uniform access to the available resources. Each document has a unique address known as a uniform resource locator (URL). There are basically three different ways to search the Web; by exploring its hypertext structure (known as browsing or surfing), by using Web directories which classify documents by subject, or by using search engines that index the documents in a portion of the Web (Baeza-Yates and Ribiero-Neto, 1999).

Most search engines today use the crawler-indexer architecture. Crawlers (or spiders, robots, knowbots etcetera) are software agents which send requests for data to remote

¹Foreign Language

Index	URL	No of Web pages
Google	< http://www.google.com >	1,500 million
FAST Index	< http://www.alltheweb.com >	625 million
AltaVista	< http://www.altavista.com >	550 million
Inktomi	< http://www.inktomi.com >	500 million
Northern Light	< http://www.northernlight.com >	390 million

Table 2.1: The largest search engine indices as of December 11, 2001, according to Search Engine Watch. Sizes are reported by each search engine. (Sullivan, 2001)

servers. These data, mostly Web pages but also images etcetera, are sent to a main server where they are indexed. This index is then used to answer queries, submitted from users on different locations on the Web through a user interface and a query engine. The enormous amount of Web pages, and the rapid increase and frequent updating of these pages, make gathering and maintaining data in this fashion very difficult. Research shows that the search engines available today cover only parts of the Web, and that the contents of their indexes only partially overlap. In order to enhance the results of existing search engines, two or more search engines can be combined in a *meta search*. A meta search engine, for example Search.com² and Metacrawler,³ evaluates which search engines are most likely to give good results to each new query. The query is then sent to the highest rated search engines, and the returned results are merged and ranked, using some kind of statistics related to the query terms in the retrieved results (Baeza-Yates and Ribiero-Neto, 1999).

2.2.1 Crawling and Indexing

There are several techniques to crawl the Web. The simplest is to start with a set of URLs and from these pages extract hypertext links referring to other URLs which are followed recursively in a breadth-first or depth-first fashion. This works well for systems using only one crawler, but may cause problems for major search engines using several co-ordinated crawlers to cover more ground. Another technique is to divide the Web by country codes or Internet names, and assign one crawler to each part (Baeza-Yates and Ribiero-Neto, 1999). As a response to the rapid growth of the Web, methods for goal-directed (or focused) crawling have been proposed, using machine learning techniques for example- and topic-driven Web exploration (Kobayashi and Takeda, 2000). Since crawling involves interaction with numerous Web servers beyond the control of the system, this is the most fragile application in the architecture (Brin and Page, 1998).

Most search engine indices are variants of the inverted file. An inverted file is a list of

²Search.com <<http://www.search.com>>

³Metacrawler <<http://www.metacrawler.com>>

sorted words, each one having a set of pointers to a document where it occurs. Compression and other indexing techniques, such as stop word elimination, can be used to reduce the size of the index. As indicated by Zipf's Law (illustrated by Figure 4.4 on page 24), a stop list of a few dozen high frequency words can significantly reduce the index size (Manning and Schütze, 1999). Most major search engines, however, such as Evreka⁴ and Google,⁵ use full text and hyperlink databases where all the words in each document are used as index terms. However, in order to speed up retrieval some high frequency words might be eliminated, even though a full text approach is used. In the Google full text index some high frequency English function words, for example *is* and *the*, are eliminated.

2.2.2 Clustering and Ranking

There are two similarity measures that must be considered during the information retrieval and ranking process. First, the similarity of two documents in a database, and second, the similarity of a document and a query.

The similarity of two documents is important for identifying clusters, that is, groups of documents which can be retrieved and processed together for a given type of user query. Clustering can be defined as the grouping together of similar documents to speed up information retrieval, where the goal is to organize a collection of unclassified objects into a hierarchy of categories (Luger and Stubblefield, 1999). It has been argued that hierarchical clustering methods are well suited for Web search and retrieval systems where short response times are important, because the result is a binary tree. There are also clustering methods developed specifically for efficient clustering of Web documents, based on for example the hypertext structure of Web documents, the words and hypertext links in the document, and links in other documents referring to the document (Kobayashi and Takeda, 2000).

Information about ranking algorithms used by major search engines is usually not publicly available,⁶ but it is likely that most search engines use term weighting or vector space models. In the simplest retrieval and ranking systems, the attributes of each document are represented by coordinates in a vector and each query is also represented by a vector. The ranking of a document is then determined by the distance between the document vector and the query vector (Kobayashi and Takeda, 2000). A widely used vector space model-based algorithm is Latent Semantic Indexing (LSI),⁷ which handles synonymy and polysemy. This greatly improves clustering of related documents, and also avoids clustering of unrelated documents. LSI is based on the idea that the meaning of a document is not determined by its set of words, but by a latent semantic structure. This

⁴Evreka <<http://www.evreka.com>>

⁵Google <<http://www.google.com>>

⁶The PageRank method used by Google is an exception. Information about the architecture of Google and the principles of PageRank can be found in (Brin and Page, 1998).

⁷Also referred to as Latent Semantic Analysis, LSA, when used in a psycholinguistic context.

structure manifests itself by how each word is used with all other words across the entire corpus, that is, by a set of constraints provided by all the contexts in which a given word does and does not appear. This means that replacing words in this structure with their synonyms does not change the meaning of the document, as long as the latent semantic structure does not change (Katsnelson and Nicholas, 2001; Coccaro and Jurafsky, 1998).

But standard IR methods that work well on controlled collections may not give as good results on the Web. One reason for this is that standard methods, such as the vector space model, often have been developed for text collections that are very different from the Web, where documents found differ in language, vocabulary, size, topic, format, and so on. Also, traditional IR queries often consist of six, seven words, whereas Web IR queries often are very short.⁸ Because Web queries usually consist of only a couple of words, and the vector space model defines both query and document as vectors by their word occurrence, the search often results in very short documents consisting of the query and a few words (Brin and Page, 1998). Baeza-Yates and Ribiero-Neto (1999, 391) argue that in order to improve search results on the Web, the users should specify more accurately what they want by for example adding “all possible synonyms to their query.” It has also been proposed that Web masters should add synonyms to the document meta data, thereby improving search results for their own sites. Brin and Page disagree with opinions such as these, that place the responsibility of search engine usability with the users. They believe that the standard IR work needs to be extended with new methods in order to improve the performance of Web search engines (Brin and Page, 1998).

2.3 Linguistic Knowledge in Web Information Retrieval

The performance of information retrieval systems could be improved by applying methods based on linguistic knowledge, since there are features of natural language that affect both the usability and the performance of such systems. Firstly, words can be ambiguous, and by resolving ambiguity the precision of the system would likely be improved. Secondly, a document can be relevant to a query, even though it does not use the exact same words as those that are provided in the query. By expanding the query terms with synonyms or near synonyms and by finding morphological variants of those terms, there would likely be an improvement in recall.

Word ambiguity can cause documents to be retrieved that are irrelevant to the query. A study examining log files of search engine queries in Norwegian shows that almost 25 percent of the most frequent search terms were ambiguous, resulting in irrelevant search results returned from the search engine. A correlation between meaning and grammatical

⁸An extensive user behavior study shows that 52 percent of the users perform a search using one word, and only 12 percent use three or more words. (Sherman and Feldman, 2002)

category was found in about 90 percent of the examined ambiguous search terms, indicating that sense ambiguity could be reduced by using a tagger for grammatical disambiguation (Holen et al., 2001). By using methods for resolving ambiguity, and thereby separating the unrelated concepts of homonyms and recognizing the related senses of polysemous terms, retrieval performance would likely be improved (Krovetz, 1997). Strzalkowski (1994) claims that single words rarely are specific enough to discriminate accurately between documents. Therefore, it is better to identify meaningful phrases which represent important concepts in the domain. This would require a correct syntactic analysis, because low-level methods for handling complex terms based on co-occurrence and frequency (for example n-grams and collocations) show an inclination to high error rates. Taggers identifying the part of speech of a word might be used for word sense disambiguation, while lexical phrases such as the proper noun *Uppsala University* or technical concepts such as *language technology* can be identified by using a thesaurus (Krovetz, 1997). By handling complex terms, such as idiomatic phrases or common constructions with predicative relations between verb and object, IR systems would gain in precision (Karlgrén, 2000). However, the creation of compound index/query terms makes the matching process between query and index terms more complex, as the structure of the compound terms must be dealt with, for example *information retrieval* is the same as *retrieval of information* (Strzalkowski, 1994).

By conflating similar variants of a word into one term, either by *semantic conflation*, for example by finding sets of synonyms or near synonyms, thereby allowing documents with only minor differences to be described by the same terms, or by *morphological analysis* identifying morphological variants of a lexeme, IR systems would gain in recall (Karlgrén, 2000). Semantic conflation is usually accomplished by using a thesaurus based on compiled lexical knowledge, or by statistical methods such as LSI (Krovetz, 1997). Morphological conflation, that is, methods for identifying morphological variants of a lexeme, is especially effective on short documents, since long documents are more likely to contain more word forms (Krovetz, 1997). Since Web documents often are short, this is particularly interesting from a Web IR perspective.

Many algorithms have been developed in order to reduce word forms to their stem, ranging from simple non-linguistic truncation methods to dictionary-based linguistic algorithms. *Lemmatization* involves the reduction of words to their lemmas through a complete morphological analysis based on grammatical rules and a dictionary. A low-level alternative to lemmatization is *stemming*. This is a technique for conflating inflectional and derivational variants of a word to one stem. A stemmer uses language-specific productive rules to remove suffixes and prefixes, and exception rules to handle stem alternations, thereby grouping semantically related words belonging to different word classes. Overstemming (that is, when unrelated words are grouped together under the same stem) can cause deterioration in precision. Koskenniemi (1996) discusses a more advanced method for morphological analysis based on finite state morphology, where two finite state transducers are used; one for morphological analysis using a fixed

lexicon defining and restricting the set of possible stems, and another transducer for morphological heuristics, dealing with words not found in the lexicon, that is, proper names, new terms, and acronyms.

The benefits of using morphological analysis in IR applications have been questioned, especially for English,⁹ but a number of studies suggest that IR applications for languages with rich morphology can yield improvements in recall by grouping morphological variants (see for example Krovetz (1997), and Kraaij and Pohlmann (1996)). The Nordic languages, especially Finnish and Icelandic, are morphologically rich languages, and thus it is likely that the addition of methods based on linguistic knowledge would improve the performance of IR applications for these languages. Studies show that there are morphological features of the Scandinavian languages, for example compounding, derivation and inflection, as well as semantic features such as homonymy and polysemy that must be considered by high-quality IR systems (Hedlund et al., 2000; Fjeldvig and Golden, 1988).

2.4 The Nordic Languages on the Web

The Nordic languages are:

- (1) official state languages Danish, Finnish, Icelandic, Norwegian-Bokmål, Norwegian-Nynorsk, and Swedish;
- (2) official regional languages Faroese and Greenlandic Inuit;
- (3) officially recognized minority languages Meänkieli (Torne Valley Finnish), Romani, Sami, and Yiddish (for each of which at least one Nordic country has signed the *European charter for regional or minority languages* (Council of Europe, 1992)).

The Scandinavian languages, Danish, Icelandic, Norwegian-Bokmål, Norwegian-Nynorsk, and Swedish, have a common ancestor, Common Scandinavian. (See Figure 2.1 on page 10.) Over time, differences increased between the Scandinavian languages; while Icelandic has retained nearly all the morphological categories of Common Scandinavian, the morphologies of Danish, Swedish, and Norwegian have been significantly simplified. But plenty of similarities remain to this day, and traces of the common origin of the Scandinavian languages can be found among the most frequent words in each language, as found in the corpora collected for this project. (See Table 2.2 on page 10.)

⁹English, the majority language in IR research, is a morphologically poor language, and the consequently poor results of using for example morphological analysis has greatly affected the attitudes toward such methods (Karlgrén, 2000).

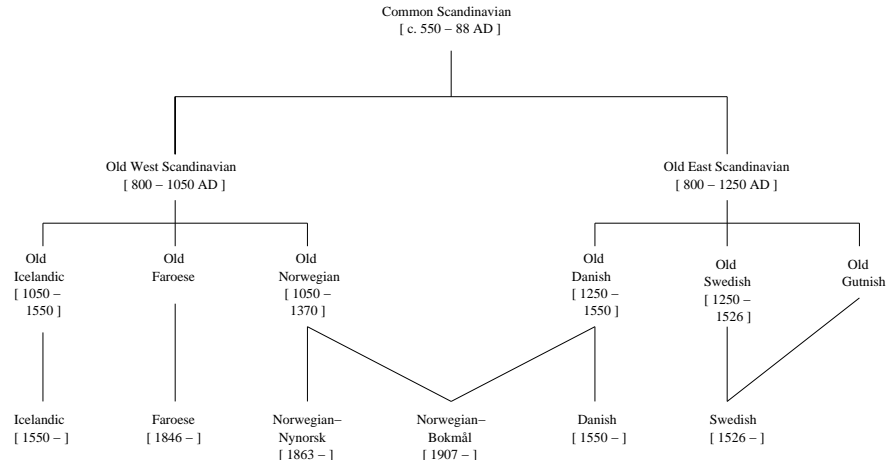


Figure 2.1: The origin of the Scandinavian languages. (Adapted from Haugen (1987))

Finnish (or Suomi) is not related to the Scandinavian languages, but a member of the Finno-Ugric family of languages. As can be seen in Table 2.2, Finnish is quite different from the other major languages spoken in the Nordic countries, even though Finnish vocabulary has been influenced by Swedish due to historical reasons. Finnish is a morphologically rich language; inflectional suffixes have a wide range of grammatical functions. This leads to a significantly larger word length average for Finnish, than the respective averages for the Scandinavian languages, while the average sentence length (measured in numbers of words per sentence) is smaller, and there are fewer one-syllable words in Finnish. (For a thorough description of Finnish, see Sulkala and Karjalainen (1992).)

Danish	Finnish	Icelandic	N-Bokmål	N-Nynorsk	Swedish
i	ja	í	i	i	att
og	on	að	og	og	och
at	ei	á	er	det	i
på	että	og	av	som	det
det	ovat	til	det	er	som
er	myös	um	som	til	är
til	oli	við	til	at	en
for	se	er	å	å	på
en	mutta	sem	en	for	för
af	hän	fyrir	for	ein	av

Table 2.2: The ten most frequent function words in the Nordic languages, as found in the corpora collected for this project.

Although the Nordic languages are to be counted as small languages on a world scale in terms of their number of speakers, at least the official state languages are represented with millions of documents on the World Wide Web.¹⁰ For example, a search for the highly frequent Icelandic function word *að* ('to') (following Ghani et al. (2001)) with the Google search engine returns about 1,950,000 Web pages. The other languages yield similar results, as shown in Table 2.3.

Language	Word	Google	Evreka
Icelandic	<i>að</i> ('to')	1,950,000	1,693,333
Norwegian-Nynorsk	<i>ikkje</i> ('not')	469,000	157,647
Norwegian-Bokmål	<i>ikke</i> ('not')	2,070,000	3,314,504
Danish	<i>ikke</i> ('not')	2,080,000	4,491,186
Finnish	<i>että</i> ('that')	2,550,000	959,919
Swedish	<i>är</i> ('is')	11,500,000	9,356,591

Table 2.3: Search results for high frequency words in the Nordic languages. Search engine language tools for search in specific languages were used to distinguish between the word *ikke* in Norwegian-Bokmål and in Danish. (Search performed September 26, 2002.)

Thus, there are good grounds for believing that the Web could serve as a source of text material for learners and teachers of these languages. Consequently, we need a search service which will locate texts according to their language, topic, and difficulty level.

¹⁰For the officially recognized minority languages, however, the situation is quite different; Sami and Finnish Romani are discussed in Nilsson and Borin (2002).

Chapter 3

Squirrel 1.0

A prototype meta search application has been developed in order to assist users in exploring the Web as a source of authentic material for language learners. In this chapter a session with this prototype, named Squirrel 1.0, is described from a user perspective, and example test results are presented.

3.1 A Session from a User Perspective

- Squirrel is started with the command: `perl squirrel.perl`
- The user supplies an example text, in the form of an URL, as standard input. The example document used in this session is a review in Swedish of the film “Sagan om ringen - Ringens brödraskap” (Lord of the Rings - The Fellowship of the Ring). Below are the first two paragraphs of the example document.

Styrka kan finnas i de minsta av saker. En av 1900-talets populäraste litterära verk, Tolkiens Sagan om ringen, blir nu det nya århundradets största filmhändelse. Ett hett efterlängtat epos och actionäventyr som regisserats av Peter Jackson. Inspelningarna har gjorts i Nya Zeeland's böljande landskap. Sagan om ringen är första delen i trilogin Härskarringen.

(Continued on next page.)

(Continued from previous page.)

Sagan om ringen berättar historien om den unge hobben Frodo som ärver en tillsynes oskyldig ring. Frodo upptäcker att ringens ursprungliga skapare, den onde trollkarlen Sauron, desperat letar efter ringen. För det är en ring med mycket onda krafter som kan göra det möjligt för Sauron att förslava invånarna i riket som kallas Midgård.

- The example text is collected and the total number of words in the text is calculated, as well as a readability score. A short description of the document collected from the HTML meta description is presented to the user, followed by a list of ten possible query terms. These terms are collected from a merged list of the most frequent words in the text and the HTML meta keywords. The following information about the example document is presented to the user:

```
LANGUAGE: Swedish
NO OF WORDS: 739
READABILITY: 43
DOCUMENT META DESCRIPTION: Bio.nu - Filminfo.
POSSIBLE QUERY TERMS (incl meta keywords):
1 ringen
2 sagan
3 peter
4 jackson
5 bio
6 filmen
7 the
8 frodo
9 dess
10 nya
```

- The user chooses which words to use as query terms from the list of possible query terms, and formulates the query by typing the numbers of the chosen terms and finally pressing <enter>. In this example session, the query terms *ringen*, *sagan*, and *frodo* are chosen. This query is sent to the search engine.
- The query results in 60 URLs returned from the search engine, which is the maximum number of URLs retrieved by Squirrel. 15 of these are unreachable, but the remaining 45 are successfully retrieved and evaluated. Of these 45 documents, 38 are in Swedish. The documents presented below are of the same language as the

example document, and of the same number of words or more. They are sorted by readability score in increasing order, that is, the first text is assumed to be the least difficult to read, and the second text to be more difficult, and so on.

```
URL: http://www.moviemix.nu/filmrec.asp?ID=191
NO OF WORDS: 1784
READABILITY: 34
DOCUMENT TITLE: SAGAN OM RINGEN - Moviemix med filmrecensioner från
video, bio och dvd
DOCUMENT META DESCRIPTION: Moviemix har dom färskaste filmrecension-
erna

URL: http://www.amosmagasin.com/ArticlePages/200110/24/Ad200.dbp.
html
NO OF WORDS: 1442
READABILITY: 36
DOCUMENT TITLE: AMOS MAGASIN
DOCUMENT META DESCRIPTION: Man skulle varit fluga

URL: http://mediaarkivet.com/filmrec.asp?typ=1&id=357
NO OF WORDS: 1033
READABILITY: 40
DOCUMENT TITLE: mediaarkivet.com - filmrecensioner och musikrecen-
sioner
DOCUMENT META DESCRIPTION: MediaArkivet.com - Media åt folket!

URL: http://www.allamedia.com/spelfilm/artikel_158.shtml
NO OF WORDS: 944
READABILITY: 41
DOCUMENT TITLE: Sagan om Ringen
DOCUMENT META DESCRIPTION: En av de mest uppskattade filmerna genom
tiderna.

URL: http://vujer.com/recensioner.php?rid=531
NO OF WORDS: 1461
READABILITY: 43
DOCUMENT TITLE: Vujer.com | Sagan om ringen
DOCUMENT META DESCRIPTION: Du är här:
```

- From this list, the user can choose URLs for further evaluation.
- All of the documents presented above are film reviews. Below are the first two paragraphs from the document titled “SAGAN OM RINGEN - Moviemix med

filmrecensioner från video, bio och dvd”, which has the lowest Lix readability score (34) among the documents presented, with 18.8 percent long words and an average sentence length of 15.5 words. The percentage of high frequency words that are likely to be part of a language learner’s vocabulary is 60 percent.

När jag var tio år fick jag höra talas om "Sagan Om Ringen", bläddrade i den och tänkte: "Det här måste jag läsa någon gång." Nästan tjugo år senare har "någon gång" fortfarande inte inträffat, men då har sagan hunnit bli en film istället. Vad har inte sagts redan om "Lord Of The Rings" som inte tål att upprepas? Ingen aning.

Skulle ändå vilja be om att få inleda denna recension med att jag inte håller med om att filmen är gjord enbart bara för att tjäna pengar; visst kostade den enormt mycket att göra och filmbolaget New Line Cinema satsade allt de ägde, på en i och för sig rätt säker hand, men de vågade åtminstone spela högt.

- The document titled "Vujer.com | Sagan om ringen" has a readability score of 43, with 25.5 percent long words and an average sentence length of 17.8 words. The percentage of high frequency words that are likely to be part of a language learner’s vocabulary is 48 percent. Below is the first paragraph of the text.

I ett nu bortglömt Europa ligger Midgård, en värld bortom tid och rum. I Midgård härskar Hober, Alver, Dvärgar, Enter, människor och andra folkslag och varelser. Midgård har dock ett mörkt förflutet, en gång för mycket längesedan smiddes 20 olika magiska ringar med mystiska krafter, att delas mellan de olika folkslagen. Tre ringar för Alv-kungarna, sju för Dvärgherrarna och nio för nio dödliga kungar. Ringarna skulle användas till att skapa fred i Midgård. Men den onde trollkarlen Sauron från landet Mordor smidde ytterligare en ring, en ring som styr dem alla. One ring to rule them all. Med denna härskarring startade Sauron ett krig och täckte Midgård med ett väldigt mörker. Sauron var ohejdbar och spred skräck över hela Midgård. I ett sista försök till motstånd marscherade de samlade folkslagen mot Mordor. Kungen Isildur av Gondor lyckades då ta ringen från Saurons finger.

Chapter 4

Implementation

Squirrel 1.0 has been designed to assist users in exploring the Web as a source of material for language learners through meta searching. Initial results are encouraging, even though the prototype has not yet been formally tested and evaluated by real users.

In section 4.1 the general architecture of the system is outlined. The implementation of the meta search method is described in section 4.2, followed by a discussion of document preprocessing, in terms of HTML parsing and tokenization, in section 4.3. Automatic language identification methods are discussed in section 4.4, as well as the implementation and the limitations of the method chosen for this application. In section 4.5, the implementation of the query term extraction method is described and the limitations of this method are discussed. Finally, in section 4.6, reading comprehension and the concept of readability are elaborated upon, followed by the implementation and limitations of the chosen method.

4.1 Architecture

The Squirrel application has been implemented in Perl 5. Modules in the libwww-library are used to handle Web requests and retrieval, and parsing of HTML documents for links, plain text, and meta data such as a description and keywords provided by the author.¹ In this prototype version of Squirrel, each search is initialized by the user supplying an example text, in the form of an URL. The HTML document this URL refers to is retrieved and the HTML markup of the document is parsed for plain text. This text is evaluated as to language, word count, and readability score. The user is presented with this information, as well as with ten possible query terms extracted from the example text. The user chooses which of the suggested terms are to be sent as a query to the search engine. (See Figure 4.1 on page 18.)

¹The libwww-library and modules can be downloaded from the Comprehensive Perl Archive Network, CPAN. <<http://www.cpan.org>>

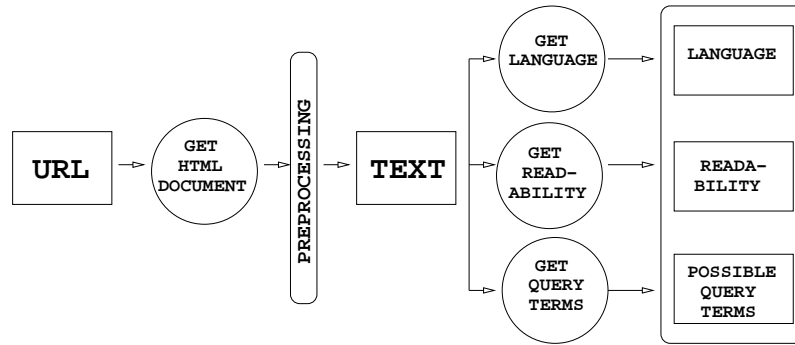


Figure 4.1: Squirrel: overview of language identification, readability testing and query term extraction from the example document.

The query is sent to the search engine, and the result of the search (URLs suggested by the search engine) is collected. The documents referred to by the highest ranking URLs suggested by the search engine are retrieved, preprocessed and evaluated locally one at a time. The language of each retrieved document is compared to the language of the example text, and if they are found to be the same, a readability score of the retrieved document is computed and the result is stored together with additional information about the document, that is, the URL, the total number of words, and the document title and description. When all the documents have been evaluated, the user is presented with a list of possible matches sorted by the readability score in increasing order. (See Figure 4.2.)

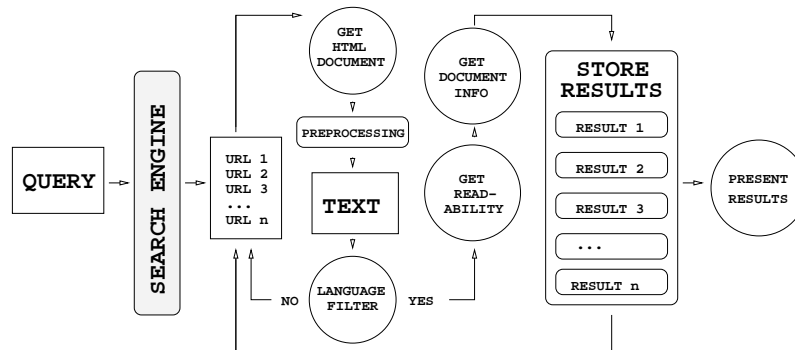


Figure 4.2: Squirrel: overview of the meta search method with the query sent to the search engine, and the language identification, readability testing and information extraction from the retrieved documents.

4.2 Meta Searching

The HTML documents are retrieved by a user agent, an interface layer between the Squirrel application and the Web, through which remote servers are accessed. The user agent handles communication with the remote server by sending a request for the URL and collecting the content, the HTML document.²

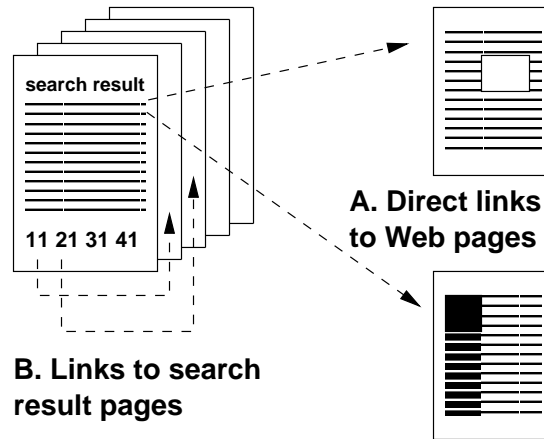


Figure 4.3: Squirrel: overview of the meta search method

A commercial search engine with a full-text index is used for meta searching the Web. The results of a search with the search engine are ranked according to how many times the query terms occur in each document, and where in the HTML structure the query terms are located (for example in the document title). Ranking is also influenced by the number of references from other Web sites.

After a query has been sent to the search engine, the HTML markup of the returned result page is parsed to find all links. Irrelevant links, that is, links that do not refer to search results, are removed and the remaining links are divided into two lists, as illustrated in Figure 4.3:

- A. the direct links referring to the ten highest ranking Web documents, and
- B. the links to the remaining search result pages (if any).

The remaining search result pages in list B are then accessed one by one, and parsed for all direct links. These links are added to list A. This list is sorted in ranking order and, one item at a time, each of the 60 highest ranking document are accessed through the user agent. Each document is retrieved and evaluated locally, that is, the HTML structure and the text content are examined. The cutoff at 60 documents was chosen for two reasons:

²The modules used are LWP::UserAgent, HTTP::Request, and HTTP::Response.

to keep the response times within reasonable limits, and, more importantly, because of the expected decrease in quality of search results with lower ranking. This cutoff can of course easily be modified.

4.3 Preprocessing

The preprocessing of the HTML document is performed in two steps. Firstly, the HTML markup is removed leaving only plain text,³ and secondly, this text is tokenized, that is, the text string is converted to a list of words. Every retrieved document is preprocessed in this fashion.

4.3.1 HTML parsing

Hypertext is text with embedded links connecting related documents, thereby providing a simple framework for documents displayed on screen. This framework contains a small set of elements, markup tags, that serve as basic rules of structuring. In a sense, the markup of a HTML document carries some linguistic meaning, for example tags that indicate the division of text into paragraphs or font-changes that mark titles. But HTML cannot be compared to strict markup languages, since choosing which markup tag to use is a decision of style as well as function. For example, the author might choose to use a couple of line break tags instead of a paragraph tag to mark paragraphs in the document, or to use font tags instead of headline tags in order to create effects in the document. Although this makes the HTML markup a somewhat unreliable source of information about the text, there are still markup elements that can be used to describe the topic of the document. These elements include meta data supplied by the author, and the document title. (See Figure 4.5 on page 25.) If no such elements can be found, headlines and text segments written in bold face can be used to describe the topic of the document.

In order to get plain text, the retrieved HTML document is parsed for the text content, that is, the markup tags are removed, leaving only the text segments. This is done by breaking up the HTML document into different segments much like a browser would, and storing the text segments. The parsing of the HTML markup is quite important, since the quality of the result has a great impact on further processing, such as the language identification and the readability assessment of the text. HTML documents are difficult to parse since the specifications for HTML change continually, and HTML markup is often both incorrectly and inconsistently written. There are many special cases to consider: embedded JavaScript code, frame sets, style tags, and comments extending over multiple lines to name but a few.

³The modules used for HTML parsing and extraction of hyperlinks are HTML::Parser and HTML::LinkExtor.

4.3.2 Tokenization

When the HTML markup has been removed, the preprocessing continues with the *tokenization* of the text, that is, the text is converted from a string of characters into a list of words. As digits usually contribute very little to retrieval results (Baeza-Yates and Ribiero-Neto, 1999), they are eliminated, while hyphenated words are kept intact. The most common punctuation marks are removed, and as no distinction is made between upper and lower case by the search engine, the text is converted to lower case.

When deciding on methods for preprocessing text, one must weigh the potential gain in efficiency against the risk of losing linguistic detail that is expressed by punctuation, hyphenation, and capitalization (Crystal, 2001). Structurally recognizable tokens such as numbers, dates, and acronyms can contain ambiguous punctuation, for example, punctuation that may or may not mark the end of a sentence. The most common ambiguous punctuation mark is the period. It is difficult to decide when a period is a full-stop, part of an abbreviation, or both, but by analyzing the structure of the input strings, some ambiguities can be resolved (Grefenstette and Tapanainen, 1994). Since this prototype must handle six different languages, the tokenization is low-level using only structural information. For language-specific applications, a stop list of frequent abbreviations can be used to recognize tokens containing periods, thereby distinguishing between cases where the period is used as a sentence delimiter, and when it is not.

4.4 Automatic Language Identification

Automatic language identification (language ID) deals with the problem of identifying the language from a sample of text or speech. The approach to language ID is affected by the form of input: written or spoken. There are a number of algorithms for written language ID, based on linguistic features such as common short words, diacritics and special characters, syllable characteristics, morphology, and syntax. Other techniques use statistical measures such as the independent probability of letters and the joint probability of various letter combinations, or n-grams of words or characters (Mathusamy and Spitz, 1996; Rosenfeld, 2000).

Cavnar and Trenkle (1994) report good results (99.8 percent accuracy in longer texts) for a language identification method that compares rankings of the most frequent n-grams to assign a document to a class. The method is based on Zipf's Law, which states that the n-th most common word in a text occurs with a frequency inversely proportional to n in this text. This also holds for the frequency of n-grams, the implication being that documents within the same category should have similar n-gram frequency distributions. The system uses category profiles computed from training sets. To classify a new document, the system computes the n-gram frequency profile of the document, and then compares the profile and each of the category profiles to find the category with the smallest distance measure. (See Cavnar and Trenkle (1994) for more information on this

method.)

This text categorization algorithm has been implemented by van Noord (2001) as TextCat,⁴ a written language identification program. TextCat currently supports 69 natural languages, including Icelandic, Norwegian,⁵ Danish, Swedish, and Finnish, and can be redistributed and modified under the terms of the GNU General Public License. A modified version of TextCat was used as a language filter by Ghani et al. (2001) in CorpusBuilder, a system which automatically generates Web search queries for collecting documents in minority languages. The performance of the filter on Slovenian documents was tested by native speaker evaluation, and the results were very good, with precision at 99 percent and recall at 90-95 percent.

4.4.1 Implementation

One of the advantages of TextCat is that it only requires small quantities of training material. There is no information available as to the size of the training material used by van Noord, but according to Cavnar and Trenkle (1994) as little as 20-120 kB is sufficient. However, as Cavnar and Trenkle observe, the quality of the training material affects the categorization performance. Since the Scandinavian languages share many of the most common words that make up the n-grams in the language model, and furthermore, since TextCat does not differentiate between the two variants of Norwegian, Nynorsk and Bokmål, new language models have been created for this project.

Language	File Size	Tokens	Types	TTR
Danish	139 kB	25,285	5,042	19.9%
Finnish	142 kB	16,182	7,756	47.9%
Icelandic	167 kB	25,215	6,719	26.6%
Norwegian-Bokmål	162 kB	26,008	6,266	24.1%
Norwegian-Nynorsk	171 kB	27,880	6,416	23.0%
Swedish	153 kB	25,627	6,385	24.9%

Table 4.1: The file size in kilo bytes, the total number of words (Tokens), the number of different words (Types) and the vocabulary diversity in percent (Type-Token Ratio) in the language model training material.

These language models were created from collections of texts for each language, consisting of roughly 25,000 words each. (See Table 4.1.) The texts were collected from

⁴TextCat <<http://odur.let.rug.nl/~vannoord/TextCat/>>

⁵Although it does not distinguish between Norwegian-Bokmål and Norwegian-Nynorsk; see below.

major online news sites. (See Table 4.2.) The topics range from domestic and foreign politics over science and economics to culture and sports. The texts were preprocessed using the methods described in section 4.3.

Language	Online news sites
Danish	Politiken, Berlingske Tiderne
Finnish	Helsinki Sanomat, Turun Sanomat
Icelandic	Morgunblaðið, Sudurland, Skessuhorn
Norwegian-Nynorsk	Dag og Tid, Syn og Segn
Norwegian-Bokmål	Aftenposten, Caplex
Swedish	Dagens Nyheter, Svenska Dagbladet, Expressen

Table 4.2: Online news sites used for collecting training material for the language models.

TextCat was integrated into the Squirrel architecture as a subroutine called from the main program. The TextCat output was redirected from standard output to the main program but, aside from the new language models, only minor modifications of the implementation by van Noord (2001) were needed.

4.4.2 Limitations

Since the language identification method proposed by Cavnar and Trenkle (1994) and implemented by van Noord (2001) generates a profile of the entire text, multi-lingual text cannot be separated into different language segments, and thus different languages in one and the same document cannot be identified.

The quality of the language models depends of course on the quality of the training material used, and it is possible that larger training sets spread over more domains might improve the results further. The result of the language identification method is also highly dependent on the quality of the input, which makes the preprocessing of the HTML documents very important.

4.5 Extraction of Query Terms

Terms chosen to represent the topic of a document are called keywords or index terms. When used for information retrieval, keywords are called search terms or query terms. Such terms can either be *extracted* or *derived* from the document.

Extraction methods are often probabilistic methods based on physical properties of written text. The idea of automatically analyzing the topic of a document through word

frequency was first suggested by Luhn (1958), who proposed that the frequency of word occurrence in a document provides a measure of word significance. The basic idea behind Luhn's proposal was that the more frequent a word is in a document, the more significant that word is in indicating the topic of the document. A list of the words in a document, compiled in descending order of frequency, will generally take the form of the diagram in Figure 4.4. The most frequent words of a text are often function words which seem to have little to do with the topic. Because low frequency words are less likely to contribute to the description of the document, a lower cut-off must also be determined in order to capture the topic through the content words used in the text.

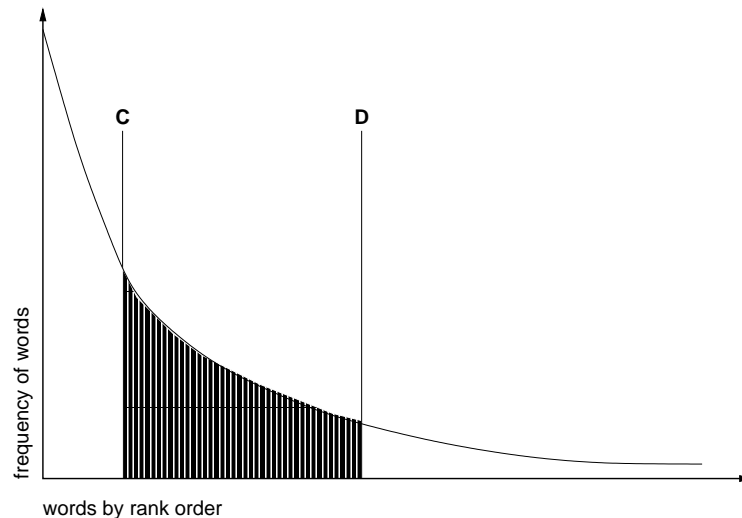


Figure 4.4: A plot of the curve relating word frequency and word rank order. (Adapted from Luhn (1958, 120).)

A frequency of each word in a document can be used if the objective is to find content words which capture the “logical view of the document” (Baeza-Yates and Ribiero-Neto, 1999, 5). This method often includes elimination of function words by comparing the text to a list of stop words in order to find the content words that best describe the topic of the document. Term frequency (TF) can be normalized with respect to document length, that is, it is more significant for a term to occur in an abstract, than in an article. This is called Probabilistic Term Frequency (PTF). PTF is useful when the objective is to compare term frequencies, or to find the terms which capture the common logical view of a collection of documents of different length. The terms that are unique for each document in a collection can be found by computing the weight of each term from the collection frequency and the document term frequency for each term, normalized by the document length. This is commonly done during indexing, as the most useful index terms are those that can select a small number of relevant documents from the many non-relevant documents in the index (Robertson and Sparck Jones, 1997).

Methods for deriving keywords are more complicated, and often rely on morphological analysis to identify morphological variants of a lexeme, by stemming or by a more sophisticated method. Keyword derivation can also be based on semantic knowledge. With for example a thesaurus based on domain-specific lexical knowledge, information about semantic relations such as synonymy and hyperonymy can provide a wider set of keywords. (See section 2.3.)

4.5.1 Implementation

Because of the meta search approach, the choice of query extraction method is influenced by the full-text indexing method used by the search engine. (See sections 2.2.1 and 2.2.2.) The query format is also determined by the architecture of the search engine. *Queries* are made up of sequences of words, or phrases. *Words* are defined as combinations of letters and digits separated by space, without distinction between upper and lower case characters. *Phrases* are defined as words in a specific order, enclosed by citation marks. Queries can be negated by using the expression `-termA`. A query such as `(termA termB termC)` can be used to search for documents where at least one of the terms are found. The default is inclusion, explicitly expressed as `+termA`.

<pre><META name="keywords" content="Sagan om Ringen, filmrecension"> <META name="description" content="En sida om Sagan om Ringen med bilder, recensioner, artiklar, skärmläckare"> <META name="author" content="Fröken Hobbit"></pre>	<pre>POSSIBLE QUERY TERMS (incl meta keywords): 1 ringen 2 sagan 3 peter 4 jackson 5 bio 6 filmen 7 the 8 frodo 9 dess 10 nya</pre>
--	---

Figure 4.5: Example meta keywords, description and author found in the HTML structure of the document.

Figure 4.6: Example of a merged list, with words found in both the meta keyword tag in Figure 4.5 and the word frequency list in Table 4.4 on page 27 presented as possible query terms.

Two sources of information are used in order to extract words that can be used as query terms from the example document supplied by the user: the HTML markup of

the document, and the words that occur in the document. Firstly, the HTML structure is checked for possible meta keywords supplied by the author of the document. (See Figure 4.5 on page 25.) Secondly, the probabilistic term frequency (PTF)⁶ of each word in the document is computed, and the word list is sorted in descending order of frequency.

In order to filter out high frequency words, such as articles, prepositions, and conjunctions, that most likely have nothing to do with the topic of the text, words found in language specific lists of stop words are eliminated. Without stop word elimination, the majority of the words in a text are function words (as can be seen in Table 4.3 on page 27). After stop word elimination, however, a list of the most frequent words gives a better description of the topic of the document. (See Table 4.4 on page 27.)

A ranking of ten possible query terms is produced from these two sources of information, with words found in both the HTML meta data and the PTF list placed in the top rank positions. The remaining positions are filled with the most frequent words from the PTF list. If there are no meta keywords in the HTML structure, the ten most frequent words from the PTF list are presented to the user as possible query terms. (See Figure 4.6 on page 25.) The final selection of query terms from this list is made by the user.

4.5.2 Limitations

This prototype does not make use of the possibilities for expressing relations between query terms, in that the query term extraction method cannot be used to extract phrases. A first step to improve the method would be to extract the phrases found both among the meta keywords and in the text. The information in the document title and meta description tags could also be used, and possibly also headlines and text in bold face.

⁶PTF, rather than TF, is computed in case this prototype should ever be improved in order to handle multiple example documents for a single query.

	Term	TF
1	som	25
2	i	23
3	och	21
4	av	20
5	är	18
6	<i>ringen^a</i>	16
7	om	15
8	att	13
9	en	13
10	det	12
11	<i>sagan</i>	11
12	den	8
13	med	8
14	<i>peter</i>	8
15	på	8
16	för	7
17	<i>jackson</i>	7
18	inte	6
19	till	6
20	du	6

Table 4.3: Example of a term frequency list computed without stop word elimination.

	Term	TF
1	<i>ringen</i>	16
2	<i>sagan</i>	11
3	<i>peter</i>	8
4	<i>jackson</i>	7
5	bio	5
6	filmen	5
7	the	5
8	frodo	5
9	dess	4
10	nya	4
11	äventyr	4
12	ring	3
13	zeeland	3
14	sean	3
15	härskarringen	3
16	scenerna	3
17	liv	3
18	tolkiens	3
19	vissa	3
20	onda	3

Table 4.4: Example of a term frequency list computed after stop word elimination.

^aWords present in both Table 4.3 and Table 4.4 are italicized.

4.6 Reading Comprehension and the Readability of Texts

Reading comprehension depends primarily on a cognitive top-down process, where previously acquired knowledge and experience are used. As well as *language dependent knowledge* such as knowledge of the target language and of other languages (including the first language), knowledge of affixes, compounding, word formation, and syntax in the target language, there is *language independent knowledge* such as knowledge of the subject, background knowledge, knowledge of logic relations, and knowledge of how to interpret for example numbers, diagrams and pictures. To understand a text, one needs not understand the entire language system, since “the comprehension on which effective language acquisition depends is not a matter of word-for-word translation” (Little et al., 1989, 27).

The term *readability* is used to describe text characteristics which can be used to predict how easy or difficult texts are to read and to understand. Discourse structure, complex phrase structures, and the amount of abstract and difficult terminology are examples of such text characteristics (Alderson and Urquhart, 1984). Formulas for measuring text readability are often based on more easily calculated surface linguistic features such as word length and sentence length, which are believed to correlate with these ‘deeper’ characteristics. The Flesch-Kincaid readability metric, which is the U.S. Department of Defense standard, was developed to test the readability of military training manuals. The metric measures the average number of words per sentence and the average number of syllables per word, with manually assigned parameters. Other methods focus on reader characteristics, such as the reading ability of each student represented by the grade average (Mikk, 1995; Mikk and Elts, 1999), and on concept difficulty, based on the hypothesis that this can be captured by statistical language models learned from actual corpora (Si and Callan, 2000).

In determining the difficulty level of a text, one must keep in mind who the intended readers are and in which situation the text is to be used, and the method of analysis should be chosen depending on the linguistic difficulties that are regarded as important. For example, if the lexical complexity is the most important factor, frequency lists can be used. Within the Intelligent Web-based Interactive Language Learning Project (IWILL), a lexical difficulty filter has been designed to filter corpus search concordancing results. The filter uses a frequency list and a function which allows the user to set a threshold level for the filter. Concordance results are thus filtered according to the chosen threshold level and presented to the user, giving the user comprehensible authentic language data. (Wible et al., 2000) Vocabulary diversity is often measured with type-token ratio, which indicates both the size of the vocabulary and the repetitiveness, but this measure must be normalized over large corpora. Such a method is proposed by McKee et al. (2000). Strother and Ulijn (1987) suggest that the focus in second language reading education should be on vocabulary and on the development of reading skills. Their research show

that lexical rewriting, that is, simplifying the vocabulary, can increase reading comprehension in English as a second language used in science and technology education, but that no such benefits can be accomplished by simplifying the syntax. Ghadirian (2002) has implemented a computer-based method for incremental introduction to topic-specific vocabulary. A collection of authentic texts of a certain topic is sorted according to the percentage of high frequency words, thus ensuring that each new text contains a suitable amount of new terms and concept.

The use of readability formulas has been criticized because they only measure surface linguistic features, which is regarded as primitive and in some cases even misleading. One of the reasons for this criticism is that readability formulas have been used prescriptively as a style checker tool for writers, even though they were originally designed for descriptive use. Studies by Karlgren (2000) and Platzack (1973) show that if the objective is to efficiently grade texts according to difficulty, readability formulas work well as long as they are used descriptively.

Platzack (1973) studied the effect of syntactic complexity on reading comprehension, and came to the conclusion that since only a few of all the possible different syntactic structures occur frequently in any given text, one may as well use surface symptoms to grade texts according to readability. This conclusion is supported by a study by Karlgren (2000), where the most important factors in readability testing were found to be word length and sentence length. Among other factors tested in the study were the average number of nouns, adverbs, prepositions, and pronouns.

4.6.1 Implementation

For the Squirrel prototype, the readability formula Lix has been chosen as an easily calculated first approximation of text difficulty level for second language learners. This choice must of course be subject to evaluation (see section 5).

The Lix readability formula, developed by Björnsson (1968), measures lexical difficulty and syntactic complexity through surface linguistic features; the readability score is calculated from the average sentence length and the percentage of long words (see Formula 1, below). This formula has been used for readability testing on a large number of children's books, novels, text books, and newspaper articles in 6 languages: Swedish, German, Danish, English, Finnish, and French. It has also been used in less comprehensive tests on newspaper articles in Norwegian-Bokmål, Norwegian-Nynorsk, Italian, Spanish, Portuguese, and Russian (Björnsson and af Segerstad, 1979).

The average sentence length is calculated by dividing the total number of words in the text with the total number of sentences. This measure correlates with syntactic complexity; the more complex the phrase structure, the longer the sentence (Björnsson

and af Segerstad, 1979).

The percentage of long words correlates with lexical difficulty in that the most frequent words in a language often are monosyllabic, and that words defining abstract or complex concepts often are polysyllabic, and that polysyllabic words generally are perceived as more difficult to read and to understand. In the Lix readability formula, the percentage of long words is the percentage of words with more than a predefined number n of characters. For the Scandinavian languages this number n is set to 6. This is based on research showing that in doing so the difference of the ratio of long words in easy and in difficult text is maximized, thereby giving a good measurement of lexical difficulty. Due to the characteristics of Finnish word formation and orthography (see section 2.4) the predefined number n is set to 9 for Finnish long words (Björnsson and af Segerstad, 1979).

$$\text{Lix Readability Score} = \frac{\text{WordCount}}{\text{SentenceCount}} + \left(\frac{\text{Word} > n}{\text{WordCount}} * 100 \right) \quad (4.1)$$

The Lix readability score, when calculated on running text, is a value ranging from about 15 to 65. According to Björnsson (1968), the Lix readability score can be used to classify texts as in Table 4.5. The Squirrel prototype, however, leaves the final classification to the user by presenting all the documents by readability score in increasing order. (See section 5.3.)

	Lix	
Very easy	20	
	25	Children's books
Easy	30	
	35	Fiction
Average	40	
	45	Ordinary prose
Difficult	50	
	55	Non-fiction
Very difficult	60	

Table 4.5: Lix text classification. (Adapted from Björnsson and af Segerstad (1979, 16).)

4.6.2 Limitations

The Lix readability formula can only be used on running text. Poetry, lists of book titles, lists of URLs etcetera might generate very high - or in the case of poetry: low -

readability scores, that are in no way significant of the difficulty level of the document.

It is important to differentiate between *readability* and *suitability*, since the readability score of a text only measures surface linguistic features which correlate with syntactic complexity and lexical difficulty. Suitability must be determined by a human judge, especially if the text is collected from the Web. Anybody can publish on the Web, and this means that documents can be incorrect as to both information, which can be invalid, mistaken or even misleading, and language, which can be incorrect, with spelling errors and grammatical errors.

Chapter 5

Performance Evaluation

Since Squirrel collects texts according to language, difficulty level, and topic, the overall performance of the system depends on the combined results of the language identification method, the readability assessment method, and the relevance of the documents returned by the search engine in response to the query terms extracted from the example document.

The performance of an IR system depends on both its *effectiveness*, that is, the ability to provide the users with relevant information, and its *efficiency*, that is, the time and resources needed to perform this task. The effectiveness is often more important than the efficiency, especially during development, and since Squirrel is a prototype, efficiency will not be further discussed.

Relevance is usually interpreted as a logical relation between two texts, the query and the document, and consequently, a document is relevant to a query if the content matches the requirements stated in the query. Traditionally, the performance of information retrieval systems has been measured with *recall* and *precision*. Recall measures the proportion of relevant information retrieved, while precision measures the proportion of retrieved elements that are relevant. In order to calculate recall the total number of relevant elements in the search space must be known, which of course is impossible in Web IR. Furthermore, it has been argued that since the concept of recall implicitly assumes that the users want a complete set of relevant documents, it is not well suited for the Web (Nielsen, 1999). In modified form, however, precision and recall can be applied to Web search. Web users are interested in short response times, and the precision and recall of the URLs listed on the first page of retrieved documents, since search engine users seldom view more than the top 10 or 20 results of a search (Kobayashi and Takeda, 2000).

A retrieved document that has been correctly classified as relevant to the query is called a *True Positive* (TP). A *False Positive* (FP) is a retrieved document that has been incorrectly classified as relevant, that is, a document that should not have been retrieved. A *True Negative* (TN) is a document correctly classified as irrelevant, and a *False Negative* (FN) is a document that has been incorrectly classified as irrelevant, that is, a document that

should have been retrieved. Precision is calculated from the total number of true positives divided by the number of true positives and the number of false positives, that is, the total number of retrieved documents. Recall is calculated from the total number of true positives divided by the number of true positives and false negatives, that is, the total number of documents that would have been retrieved in a perfect search.

The document relevance of Squirrel depends on the performance of the search engine, and consequently also on the quality of the query terms used in the search. Therefore, the query term extraction method is discussed in section 5.1. The recall and precision of the language identification method are calculated, and the results are presented in section 5.2. Because the results are presented in increasing readability order, the Lix readability formula used in this project also affects the overall performance. Lix is discussed in section 5.3.

5.1 Extraction of Query Terms

The Probabilistic Term Frequency method used to extract query terms is only indirectly using knowledge of natural language; Since the Nordic languages, in particular Finnish, are languages with rich morphology, the results of the query term extraction method might very well benefit from methods using direct knowledge of language, such as lemmatization or stemming. But when it comes to meta search results, the performance is completely dependent on the architecture of the search engine. If stemming was used, *ringen* and *ring* (and possibly also *härskarringen*) would be recognized as variants of the lexeme *ring*. In theory, this could improve precision. In practice, however, this would not add to the performance of the system, but rather decrease recall because of the full text indexing method used by the search engine.

A problem related to the query extraction method used, as well as the meta search approach, is queries that are identical in different languages. The similarities between the Scandinavian languages (and possibly also the similarities between Finnish and Swedish vocabulary) might cause such problems in query based search. This can be illustrated by the query *Ringenes Herre*. This phrase is identical in Danish, Norwegian-Bokmål and Norwegian-Nynorsk, and this affects the results of the search, especially for the least frequently used language, Norwegian-Nynorsk. Since this query is identical for all three languages, the returned result from the search engine are mainly the same in each search. The consequence is a poor result for Norwegian-Nynorsk. In fact, of the returned results, none of the documents were in Norwegian-Nynorsk. Similarly, queries such as the name of the Danish author *H C Andersen* might generate results in just about any language.

A possible solution to this problem is to allow the user to add language specific inclusion and exclusion terms to the query (following Ghani et al. (2001)). With the addition of the Norwegian-Nynorsk inclusion terms *ikkje*, *eit*, and *ei*, and the exclusion terms *ikke*, *hva*,

and *hvor*, which are frequent in Norwegian-Bokmål and Danish, to the original query *Ringenes Herre*, the search engine returned in total 173 hits for this query. Of the first 60 documents, 42 were in Norwegian-Nynorsk, 4 were in Norwegian-Bokmål, and 14 documents were irretrievable.¹

5.2 Automatic Language Identification

In order to measure the performance of the language identification method used in this application, the search space has been reduced to a sample consisting of the documents returned by the search engine for each query. The results of the TextCat language filter are very encouraging: precision and recall range from 92-100 percent. (See Table 5.1.)

	Danish	Finnish	Icelandic	N-Bokmål	N-Nynorsk	Swedish
TP	44	57	58	65	12	43
FP	2	0	0	1	1	0
TN	56	4	4	19	57	5
FN	4	3	1	1	0	1
P	95.6%	100.0%	100.0%	98.5%	92.3%	100.0%
R	91.7%	95.0%	98.0%	100.0%	92.3%	97.7%

Table 5.1: Automatic language identification: Precision and Recall

The incorrectly classified documents are basically of two types: multi-language documents, and documents containing lists of for example names or book titles. As mentioned in section 4.4.2, TextCat cannot handle documents with large segments in different languages, and names, book titles and the like might consist of infrequent n-grams that affect the classification.

5.3 Readability Testing

The readability assessment method used in this project affects user opinion of the overall performance, because the results are presented in increasing readability order as calculated by the Lix formula.

¹However, the use of inclusion and exclusion terms appears to have a negative affect on the relevance of the returned results. Of the 43 returned documents in Norwegian-Nynorsk, only 8 documents were judged to be relevant to the original query *Ringenes Herre*. The reasons for this problem, as well as the solutions, lie with the search engine used and will not be discussed further.

Some text characteristics that can be used in readability testing are presented in Table 5.2. The texts thus described are the example results from the session in section 3.

	Text 1	Text 2	Text 3	Text 4	Text 5	Example
No of Tokens	1,784	1,442	1,033	944	1,461	938
No of Types	806	791	525	482	792	527
Type/Token Ratio	45.2	54.9	50.8	51.1	54.2	56.2
Stop Words (%)	60	52	62	48	48	45
Avg Sentence Length	15.5	14.5	17.0	17.2	17.8	16.5
Long Words (%)	18.8	22.2	23.3	24.7	5.5	27.2
Lix Readability score	34	36	40	41	43	43

Table 5.2: Readability: Text Characteristics for Readability Testing

The number of tokens is the total number of words in the text, and the number of types is the number of different word forms in the text. The vocabulary diversity, that is, the percentage of different words in the text, can be measured by the type-token ratio. In order to compare the type-token ratio of texts of different length, the texts must be normalized as to the number of tokens. If the sample size is not controlled, as in Table 5.2, the type-token ratio can not be successfully used for readability testing. (For more information on vocabulary diversity, see McKee et al. (2000).)

The percentage of words in the text that are found in a stop list² for Swedish is presented in the column Stop Words. The words in this stop list are highly frequent and therefore more likely to be part of a language learner's vocabulary. As can be seen in Table 5.2, the percentage of stop list words tends to decrease with higher Lix readability scores. By using information such as corpus frequency, the Lix formula could be refined. (See for example Wible et al. (2000) and Ghadirian (2002).)

The average sentence length and the percentage of long words are the two variables that make up the Lix formula. Instead of using the average sentence length as an indication of syntactic complexity, syntax data could be used to refine the Lix formula. Karlgren (2000) describes the use of output from a robust parser as an approximation for syntactic complexity. The parser, which was built for information retrieval purposes, has a built-in timer regulating the amount of time allowed for parsing each sentence. If the parser is unable to parse a clause within the allowed time-frame, it skips to the next token and a note is made in the parse tree. In order to estimate syntactic complexity, the average parse tree depth was calculated, and also the average number of parser skips per sentence. (See Strzalkowski (1994) for more information on the parser.)

²An example of candidate words for a stop list can be seen in Table 2.2 on page 10.

Readability assessment methods are often evaluated through native speaker evaluation, but is not the only text characteristic that must be evaluated by human judges. Quality, suitability, reliability, and similar characteristics which add up to the total document relevance, can only be determined by human judges. The evaluation of characteristics such as these become even more important in Web IR because of the democratic nature of the Web, where anybody can publish anything. At present, this evaluation must be left to the user.

Chapter 6

Concluding Remarks

The Squirrel prototype could be further developed in many ways. The most obvious improvement, perhaps, is the design of a graphical user interface where for example the readability of a text can be illustrated by an appropriate metaphor. But there are several functions aside from an interface that would increase the usability, for example if the user was allowed to give example texts by “cut and paste” or by defining search paths to locally stored text files.

The extraction of query terms could be improved by a query term expansion module that could add semantically related words found in for example a thesaurus or a word net to the list of suggested query terms. The extraction and ranking of the suggested query terms might also be improved by methods for handling morphological features such as compounding, derivation and inflection. A more easily implemented improvement would be to allow the user to edit the suggested query terms, or to add new terms to the query. The cutoff at the tenth most frequent term in the list of suggested query terms could be replaced by a statistical cutoff, allowing for differences in document length. Further flexibility would be added if the user is permitted to change the cutoff if the suggested query terms are unsatisfactory.

The method for readability testing could be further refined by using information such as corpus frequency or syntax data, as mentioned in section 5.3. If the objective is to match the learner with texts at the optimal level of complexity, the learner’s level of knowledge must be measured. Latent Semantic Analysis (see section 2.2.2, above) uses the learner’s own written output for finding texts on a suitable difficulty level, somewhere just above that evidenced in the learner’s own production (Landauer et al., 1998).

The meta search approach could be extended by using more than one search engine, or by making use of search engines specializing in specific countries, such as the Finnish

search engine SUOMI24.fi,¹ the Danish search engine Sol Søg,² and the Norwegian search engine Kvasir.³ By using query expansion methods combined with relevance feedback of retrieved results from the user an iterative meta search could be performed, starting with the example document and iterating until the user's information need has been met.

The prototype describes the retrieved results to the user by meta descriptors found in the HTML markup of each document, but such meta data seem to be used infrequently by Web authors. Instead of describing the topic of the suggested documents by meta descriptors, the context of each query term could be presented, thereby giving the user an idea of how the query terms are used in the document. Another possible addition to the prototype is a module for retrieval of those of the suggested documents that the user finds interesting. From these documents, word lists, KWIC concordances, and other information could be extracted, thus giving raw material for exercises etcetera.

The list of possible improvements could of course be extended even more, but any further development requires user evaluation of this prototype, and interviews with teachers and students. This project has shown that it is possible to locate text sources in the Nordic languages on the Web in order to produce authentic learning materials for second language learners, but the prototype must be tested by real users to determine whether the methods used for classification are sufficient, or if more sophisticated methods are required.

¹SUOMI24.fi <<http://www.suomi24.fi>>

²Sol Søg <<http://soeg.sol.dk>>

³Kvasir <<http://kvasir.sol.no>>

References

- J. Charles Alderson and A. H. Urquhart, editors. 1984. *Reading in a Foreign Language*. Harlow: Longman.
- Ricardo Baeza-Yates and Bethier Ribiero-Neto. 1999. *Modern Information Retrieval*. ACM Press Books, Addison-Wesley.
- C. H. Björnsson and Birgit Hård af Segerstad. 1979. *Lix på franska och tio andra språk*. Stockholm: Pedagogiskt centrum, Stockholms skolförvaltning.
- C. H. Björnsson. 1968. *Läsbarhet*. Lund: Liber.
- Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.
- William B. Cavnar and John M. Trenkle. 1994. N-gram Based Text Categorization. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94)*, Las Vegas, Nevada, U.S.A., pages 161–175, April.
- Noah Coccaro and Daniel Jurafsky. 1998. Towards Better Integration of Semantic Predictors in Statistical Language Modeling. In *Proceedings of ICSLP-98*.
- Council of Europe. 1992. European charter for regional or minority languages. European Treaties ETS No. 148. World Wide Web, <<http://www.coe.fr/eng/legaltext/148e.htm>>, Downloaded November 29, 1999.
- David Crystal. 2001. *Language and the Internet*. Cambridge: Cambridge University Press.
- Tove Fjeldvig and Anne Golden. 1988. Experiments with language-based aids in information retrieval systems. *Nordic Journal of Linguistics*, 11(1):33–46.
- Sina Ghadirian. 2002. Providing Controlled Exposure to Target Vocabulary Through the Screening and Arranging of Texts. *Language Learning & Technology*, 6(1):147–164.
- Rayid Ghani, Rosie Jones, and Dunja Mladenic. 2001. Building Minority Language Corpora by Learning to Generate Web Search Queries. Technical Report CMU-CALD-01-100, Carnegie Mellon University Center for Automated Learning and Discovery. World Wide Web, <<http://citeseer.nj.nec.com/444684.html>>, Downloaded September 4, 2001.
- Gregory Grefenstette and Pasi Tapanainen. 1994. What is a word, what is a sentence? Problems of tokenization. In *The 3rd International Conference on Computational Lexicography*, pages 79–87, Budapest, Hungary. COMPLEX'94.
- Einar Haugen. 1987. Danish, Norwegian and Swedish. In Bernard Comrie, editor, *The World's Major Languages*. London: Croom Helm.

- Turid Hedlund, Ari Pirkola, and Kalervo Järvelin. 2000. Aspects of Swedish morphology and semantics from the perspective of mono- and cross-language information retrieval. *Information Processing and Management*, 37(1):147–161.
- Gordana Ilic Holen, Janne von Koss Torkildsen, and Janne Bondi Johannessen. 2001. On Ambiguity in Internet Searches. In *NoDaLiDa '01. The 13th Nordic Conference of Computational linguistics*, Uppsala, Sweden. Department of Linguistics, Uppsala University.
- Jussi Karlgren. 2000. *Stylistic Experiments for Information Retrieval*. Ph.D. thesis, Department of Linguistics, Stockholm University.
- Yulia Katsnelson and Charles Nicholas. 2001. Identifying Parallel Corpora using Latent Semantic Indexing. In Paul Rayson, Andrew Wilson, Tony McEnery, Andrew Hardie, and Shereen Khoja, editors, *Proceedings of the Corpus Linguistics 2001 Conference, Lancaster University Centre for Computer Corpus Research on Language, Technical Papers*, volume 13 - Special Issue, pages 323–331.
- M. Kobayashi and K. Takeda. 2000. Information Retrieval on the Web. Technical Report RT0347, IBM Research.
- Kimmo Koskenniemi. 1996. Finite state morphology and information retrieval. *Natural Language Engineering*, 2(4):331–336.
- Wessel Kraaij and Renée Pohlmann. 1996. Using Linguistic Knowledge in Information Retrieval. Technical Report OTS Working Paper OTS-WP-CL-96-001, Research Institute for Language and Speech (OTS), Utrecht University.
- Robert Krovetz. 1997. Homonymy and Polysemy in Information Retrieval. In Philip R. Cohen and Wolfgang Wahlster, editors, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 72–79, Somerset, N.J., U.S.A. Association for Computational Linguistics.
- Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284.
- Patsy M. Lightbown and Nina Spada. 1997. *How languages are learned*. Oxford: Oxford University Press.
- David Little, Seán Devitt, and David Singleton. 1989. *Learning Foreign Languages from Authentic Texts: Theory and Practice*. Dublin: Authentik in association with CILT.
- George F. Luger and William A. Stubblefield. 1999. *Artificial Intelligence. Structures and Strategies for Complex Problem Solving*. Addison-Wesley Longman Inc., Reading, MA, U.S.A., third edition.
- Hans Peter Luhn. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159–165. Reprinted in: H. P. Luhn: Pioneer of Information Science. Selected Works. Edited by Claire K Schultz. Spartan Books, New York. 1968.
- Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.
- Yeshwant K. Mathusamy and Lawrence Spitz. 1996. Automatic Language Identification. In Ronald A. Cole, editor, *Survey of the State of the Art in Human Language Technol-*

- ogy. Center for Spoken Language Understanding, School of Science and Engineering at Oregon Health & Science University. World Wide Web, <<http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html>> Downloaded June 11, 2001.
- Gerard McKee, David Malvern, and Brian Richards. 2000. Measuring Vocabulary Diversity Using Dedicated Software. *Literary and Linguistic Computing*, 3(15):323–337.
- Jaan Mikk and Jaanus Elts. 1999. A Reading Comprehension Formula of Reader and Text Characteristics. *Journal of Quantitative Linguistics*, 6(3):214–221.
- Jaan Mikk. 1995. Methods for Determining Optimal Readability of Texts. *Journal of Quantitative Linguistics*, 2(2):125–132.
- Jakob Nielsen. 1999. User interface directions for the Web. *Communications of the ACM*, 42(1). World Wide Web, <<http://citeseer.nj.nec.com/nielsen99user.html>> Downloaded August 14, 2001.
- Kristina Nilsson and Lars Borin. 2002. Living off the land: The Web as a source of practice text for learners of less prevalent languages. In Manuel González Rodríguez and Carmen Paz Suarez Araujo, editors, *Proceedings of the third International Conference on Language Resources and Evaluation*, volume 2, pages 411–418. ELRA - European Language Resources Association, May.
- Nua. 2002. Nua Internet Surveys. World Wide Web, <http://www.nua.ie> Downloaded September 24, 2002.
- Christer Platzack. 1973. *Språket och läsbarheten*. Lund: CWK Gleerup Bokförlag.
- Sarah Porter. 2000. Technology in Teaching Literature and Culture: Some Reflections. *Computers and the Humanities*, 34(3):311–324.
- S.E. Robertson and K. Sparck Jones. 1997. Simple, Proven Approaches to Text Retrieval. Technical report, Department of Information Science, City University & Computer Laboratory, University of Cambridge. Update of 1994 and 1996 versions.
- Ronald Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here? In *Proceedings of the IEEE*, 88(8), 2000.
- Chris Sherman and Susan Feldman. 2002. What tools make site search more effective? IDC bulletin no. 27513, feb 2002. World Wide Web, <<http://www.mondosoft.com/icd-opinion-full.asp>> Downloaded September 24, 2002.
- Luo Si and Jamie Callan. 2000. A Statistical Model for Scientific Readability. In *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM)*. ACM.
- Judith B. Strother and Jan M. Ulijn. 1987. Does Syntactic Rewriting Effect English for Science and Technology (E S T) Text Comprehension? In Joanna Devine, Patricia L. Carrell, and David E. Eskey, editors, *Research in Reading in English as a Second Language*. Washington D.C., U.S.A.
- Tomek Strzalkowski. 1994. Robust Text Processing in Automated Information Retrieval. In *Proceedings of the 4th Confernece on Applied Natural Language Processing*, Stuttgart, Germany. ACL.
- Helena Sulkala and Merja Karjalainen. 1992. *Finnish*. London & New York: Routledge.
- Danny Sullivan. 2001. Search Engine Sizes. The Search Engine Report Newsletter,

- Dec 18, 2001. World Wide Web, <<http://www.searchenginewatch.com/reports/sizes.html>> Downloaded September 24, 2002.
- Ulrika Tornberg. 1999. *Språkdidaktik*. Kristianstad: Gleerups.
- Gertjan van Noord. 2001. Textcat. World Wide Web, <<http://odur.let.rug.nl/~vannoord/TextCat/>>, Downloaded September 27, 2001.
- David Wible, Chin-Hwa Kuo, Feng yi Chien, and Chih-Chiang Wang. 2000. Adjusting Corpus Searches for Learners' Level: Filtering Results for Frequency. In *TALC2000 Austria*, July.