



UPPSALA
UNIVERSITET

Query expansion using search logs and WordNet

Lisa Hunemark

Uppsala University
Department of Linguistics and Philology
Språkteknologiprogrammet
(Language Technology Programme)
Master's thesis in Computational Linguistics
25th March 2010

Supervisors:
Joakim Nivre, Uppsala universitet
Pablo Belin, Picsearch

Contents

Contents	2
List of Figures	4
List of Tables	5
1 Introduction	6
2 Background	8
2.1 Extracting document terms	8
2.2 Mining user logs	9
2.2.1 Collaborative Filtering	10
2.3 Semantic relations	11
3 Finding relations	12
3.1 Source 1: Mining user logs	12
3.1.1 Find neighbours in a search log	13
3.1.2 Refining selection	14
3.1.3 Narrow/expand	15
3.2 Source 2: Using WordNet	15
3.2.1 Selection within WordNet relations	16
4 Evaluation method	19
4.1 Quantitative evaluation	19
4.2 Qualitative evaluation	19
4.2.1 About automatic evaluation	19
4.2.2 About manual evaluation	19
4.3 Evaluation procedure	20
5 Results	22
5.1 Quantitative measuring	22
5.2 Qualitative measuring	24
5.2.1 Specific and broad rewriting	24
5.2.2 Narrowing and expanding	25
5.2.3 Top and bottom decile	25
5.2.4 Names	25
5.2.5 Mean rate and example queries	26
5.3 Spelling suggestions	27
5.4 Improvements	28
5.4.1 Adaptation of original query and database	28

5.4.2 Selection	29
6 Conclusion	31
Bibliography	32

List of Figures

3.1	Noun quality. Counting the number of suggestions within each rank to compare the relevance of different WordNet categories.	16
3.2	Verb quality. Counting the number of suggestions within each rank to compare the relevance of different WordNet categories.	17
5.1	Suggestions per query, counting all queries, even those without suggestions.	23
5.2	Suggestions per query, counting only queries given at least one suggestion by the source in question.	23
5.3	Comparing the share of specific rewriting for both sources, and for both data sets. Log mining results as a whole as well as split into narrowing and expanding suggestions.	24
5.4	Comparing the share of broad rewriting for both sources and for both data sets. Log mining results as a whole as well as split into narrowing and expanding suggestions.	24

List of Tables

3.1	WordNet concepts evaluated with the development set.	16
3.2	The four levels of the ranking system by Jones et al (10)	17
4.1	Share of different suggestion categories within the top and bottom decile.	20
5.1	Query/suggestion examples with ratings	27

1 Introduction

The aim of this project is to study methods for providing related queries in a search engine for images, with the help of two different sources, to evaluate which source produces the best results, and to present possible improvements. The work has been carried out in the context of an existing system at Picsearch, a Swedish search engine for images, where currently search results are the documents matching the query according to a global analysis system. Since users are not experts, their search phrases are often askew or insufficient and the search engine user's vocabulary and that of the author of an Internet document are not necessarily similar. Consequently, when searching for information, although using a perfectly suitable term, relevant documents might be missed only because their author was using a synonym. Also, too many or too few documents might be retrieved by a query, causing a need for narrowing or expansion of the query. By expanding the vocabulary of the user, the search result could be improved considerably. Automatic query expansion, i.e. the search engine providing suggestions for queries, more or less related to the original query, could compensate for differences between the original query and the relevant documents.

Suggestions will be extracted from search logs of the search engine and from WordNet respectively. Regarding search logs, the two main issues are firstly to identify sessions, within which relations might be found, and secondly to eliminate noise, such as phrases common enough not to have any significance or unrelated phrases co-occurring by coincidence. Since the logs are the result of thousands of different users randomly entering search phrases, it is not possible to ensure uniformity, so the collection will be a random mix of singular and plural, and spelling variants. The user queries are similarly mixed, which might increase the possibility of getting results each time, but makes it difficult to identify all relevant relations. Regarding WordNet, previous research shows that using a general thesaurus might introduce irrelevant relations and the lack of ranking makes it difficult to sift out the most interesting phrases or the ones fitting the actual scope of the search. Also, modern celebrities and other new phenomena are not even present. The phrases in the thesaurus are uniform, all entered in their basic form and with the correct spelling, making WordNet a stable source. The user queries, on the contrary, are far from uniform, which might be a source of conflict.

The methods used in this study will be tried with two different test suites: one with common queries and one with rare queries, since these two groups are expected to render quite different results. It is desirable that the search engine always provides suggestions, and we need them to be of a reasonable quality. Two aspects will therefore be of interest: the quantity, i.e. the number

of suggestions provided by each method, and the quality of these suggestions, i.e. the rate, using an existing ranking scale presented by Jones et al (10). The results will be evaluated manually by rating all suggestions and calculating the mean rate of all suggestions from each method, as well as the number of suggestions within each rate. Considering the results, possible improvements will be identified.

2 Background

In recent years, research in the field of related queries has been very intense. Search engines surpass each other in clever features in order to attract visitors, and web shops and news portals refine their systems to foresee their clients' wishes in detail. In spite of, or thanks to, the increasing commercial interest in the technology of related search, there is little published research on the relative performance of various algorithms used in commercial systems. A good example of efficient query expansion can be seen in the search engine *Ask* (ask.com) where the user is given word prediction while entering a query, and along with the results the opportunity to choose between narrowing and expanding as well as related names, in case they search for a name. Furthermore, the suggestions are not the same for an image search as for a web search, which indicates that the results are tailored to fit different information needs.

The general source for search engines and other systems aiming to make a relevant selection of related phrases setting out from one original phrase, is the vocabulary of documents available in the search engine through *global* or *local analysis*. Using the information from user logs, e.g. *log mining*, is often more computationally efficient, and it takes into consideration the vocabulary provided by the users and not by the documents. Putting the focus on end-users rather than documents might seem more intuitive, assuming that the preceding users have provided enough information. *Semantic relations* is yet another way to provide related terms by using ontologies, i.e. lexical databases of semantic relations. By searching them horizontally possibly useful synonyms will be presented while a vertical search might present either expansion or narrowing of the original term.

2.1 Extracting document terms

The most common way of extracting term correlations is through the vocabulary of the document set. The idea that the context of terms can determine their similarities (i.e. terms occurring in the same or similar contexts are somehow related), motivates using the document set available in a search engine. Using the entire document set available is called *global analysis*. This requires some sort of summary and the most common is a thesaurus built on term co-occurrence. Though it is a massive task, the thesaurus is built only once, which might make this effort worthwhile. After all, this is how most search engines are built in the first place. *Term clustering* (11) and *latent semantic indexing* (LSI) (10) are part of the global analysis methods.

Global analysis considers all available data, while *local analysis* is only using

a subset. Some selection has to be made as to which documents should be part of the analysis, which is the drawback to this method. If selection is made by the user, a heavy burden is put on the user, who is often reluctant to take part in surveys etc. If selection is made by the system, the relevance might be questionable. For polysemic terms (i.e. same word, several meanings) the risk of drift is quite high due to their ambiguity. If the number of queries addressing a certain instance of the term does not agree with the number of documents addressing this instance, the search engine might give a high rank to the instance with the larger result set instead of the instance with a greater number of searches, which would be what the users would rank higher. SEO by the Sea (15) illustrates the concept of drift with an example from Yahoo! search engine. Methods sorting under local analysis are *relevance feedback* and *pseudo-relevance feedback* which are both costly and might lead to query drift. (10)

2.2 Mining user logs

Mining user logs is a way of extracting user judgements on a large scale, without interaction. It does not consider the content of the documents, but merely summarize user activities, such as querying, query reformulation and click-throughs. It has been proven "useful for improving retrieval effectiveness, especially for short queries on the web" (5). Jansen et al (7) analysed the transaction logs of a set of 51,473 queries posed by 18,113 users of the Excite search engine. Data were collected concerning queries, sessions and terms. Silverstein et al (14) present an analysis of a 280 GB AltaVista search engine query log consisting of approximately 1 billion entries for search requests. They also make an analysis of individual queries, query duplication and query sessions. Furthermore they present results of a correlation analysis of the log entries, studying the interaction of terms within queries. According to the review by Jansen and Pooch (6) it appears to be "the most reasonable and non-intrusive means of collecting user-searching information from a large number of users". It may be considered somewhat limited since hardly any information about the context is provided. However, the valuable information about the formulation of the search, the search strategy and the delivery of results, might be enough to understand the user interactions and the system.

Jones et al (10) present a technique, primarily aimed at sponsored search, though applicable to general web search, "based on typical substitutions web searchers make to their queries". Their suggestions are based on pre-computed query and phrase similarity, combined with an automated ranking of the proposed queries. They also define a ranking for evaluation of their query substitutions. This is also a system well suited for use in this project. If not to say the only one, as well as being the only result available for comparison, using manual evaluation. Their evaluation is not based on manual evaluation, but on an automatic system, modelling human judgement. We shall use their ranking system to compute the qualitative result, and, where applicable, compare our quantitative as well as qualitative results with the numbers presented in their study.

2.2.1 Collaborative Filtering

Collaborative filtering or *recommendation systems* are built on the assumption that a good way to find interesting content is to find other people who have similar interests, and then recommend titles liked by similar users. (1)

Collaborative filtering (CF), which is a specific log mining method, is a domain close to, or even overlapping, the one of query expansion, which suggests the use of CF ideas within query expansion applications. The basic idea is that observations from a user's history may tell you something about their future likes and dislikes. The concept bears quite a resemblance to the idea of mining user logs in order to extract related queries. The main use is recommending, for example, music to a user, in an interactive system based on their ratings of records or artists, and in a non-interactive system, based only on purchases or visits. This may be applied to e-commerce as well as news portals and other web services. To identify recommendations for a user, information is collected about which items the user has shown interest in, with or without rating. This collection is compared to the collections of the other users to find the nearest neighbours with respect to interests. Items from those neighbour collections missing in the original collection are used as recommendations (18). A new approach, increasing in popularity, is an item-based analysis. Instead of identifying users with similar patterns, similar items are used as the recommendation basis. This might help to overcome the major limitations of CF, namely the limited amount of rating history, which lowers accuracy, and high computation costs, which is a scalability problem (9). There is a conflict of interest between collecting large amounts of information for each item to achieve high quality recommendations, and limiting the amount of information, in order to improve scalability (12).

In Breese et al (1) various CF methodologies are described and compared: memory-based algorithms based on correlation coefficients and *vector similarity* as well as model-based, such as *cluster models* and the *Bayesian network model*. Their work addresses an active user with a history, posting votes regarding items of interest, as opposed to a search engine where the active user has got no history and makes no voting. Similarities remain, however, the voting is simply binary instead of ranked, and where the experiments look at active users with less data available, the information reassembles what we know of the search engine user.

Weiss and Indhurkya (18) have developed a lightweight algorithm for recommendation systems, which has been used for recommender systems based on earlier purchases by a customer, and recommendation of web pages, based on earlier visits made by the user. It is based on binary information (i.e. no ranking), and thanks to the simplicity in calculations and data storage, it is efficient, scalable and requires no training of the system. This makes it stand up very well against other, heavier systems, that might render slightly better results, but at a considerably higher cost. This algorithm is an interesting and simple way of structuring the information of the user logs in this project, to make relations easily accessible

2.3 Semantic relations

Rather than finding statistically based relations in an unordered collection, one might rely on the established semantic relations in an ontology, or thesaurus, such as WordNet. WordNet is a lexical database of English, developed at Princeton University. Nouns, verbs, adjectives and adverbs are linked together respectively in sets of cognitive synonyms. The sets, in turn, are ordered hierarchically in specific concepts. WordNet "is considered to be the most important resource available to researchers in computational linguistics, text analysis, and many related areas. Its design is inspired by current psycholinguistic and computational theories of human lexical memory." (4)

Query expansion with terms collected from a domain specific thesaurus has shown good results (8), while general thesauri, like WordNet, have been found less reliable as a source of expansion (17). Research shows a better result for short queries than for longer ones and the result is often dependent on manual extraction of relevant terms from the suggestions found in the thesaurus. Most automatic query expansion systems using thesauri, use the extracted terms along with the original one in a new query, unsupervised by the user. In a search engine, the terms might be suggested to the users, giving them the opportunity to pick the one most relevant to their information need.

3 Finding relations

To provide related queries to the search engine, two separate sources will be used and compared: one of them is user logs and the other one is WordNet. For best performance, a combination might be an idea, as these sources are very different from each other. A development set, containing common as well as rare queries found in the search logs, is put together for minor tweaking of the process, before the actual evaluation is performed using another, larger, test set.

The top ten from the log mining suggestions, split into two categories, i.e. suggestions containing the original query or not, and the top five from the WordNet suggestions, are presented to the user. The more suggestions we keep, the better the recall. On the other hand, there is a limit to when a number of suggestions is useful or just distracting. However, the limit will not be reached for all queries, due to lack of material in the logs or to the query not occurring in WordNet.

3.1 Source 1: Mining user logs

The query logs are a very valuable source of information on which to perform data mining, and could therefore be used to identify and extract the subjects that interest our end users. Rather than using one of the local or global analysis methods on the document set, for generating related queries, an approach similar to the ones for recommendation systems might be faster and potentially provide more accurate results. "An earlier study showed that in an IR environment where more than one recommendation is made, simpler algorithms can be surprisingly effective." (18) Since no rating has been made in the log system, we may use the *collaborative filtering* method by Weiss and Indurkha (18), instead of one considering a weighted relation between items.

This is the lightweight *collaborative filtering* algorithm described by Weiss and Indurkha.(18)

1. Find the k nearest neighbours to the new (test) case
2. Collect all attributes of these neighbours that don't occur in the test case
3. Rank these attributes by frequency of occurrence among the k neighbours.

We have used the Picsearch search logs for one year and only from the UK server, to limit the vocabulary to mainly English. The logs amounted to

several GB of information, though not purely queries. From these, individual sessions were identified and each of them is entered in the *session dictionary* along with its queries. Each unique query was also entered in a *query dictionary*, along with the ID for all sessions containing the specific query. These two dictionaries could then be used to find neighbouring queries and for simple statistical calculation.

3.1.1 Find neighbours in a search log

How do we know which queries are neighbours? It is likely that closely related subjects will exhibit a temporal correlation in the web query logs. A first approach is to regard queries from the same search session as related. Separating a collection of queries into limited sessions is not trivial. How do we know that queries appearing after each other in the logs, are posted by the same user? And how do we know how long the user has been searching for the same thing? No matter which method is used, there is always a risk that several computers are connected to the same IP or even that several users work at the same computer. Especially schools and institutions can be expected to generate "false sessions". Although cookies are by far the best way to distinguish one user from the other, many users have disabled cookies. As an alternative, an identical IP, possibly combined with an identical Web browser, may indicate an identical user.

Also, the time span has to be limited. A user rarely searches for the same topic for an entire week or even for a whole day. Thus, we must choose a time window where a user is likely to address the same kind of information before changing directions. In some papers the time window is one single day (10) while others expect the user to keep to the same subject only if the next query is posted within 5 minutes from the previous one (14). Considering that this project is addressing a search engine for pictures and not documents in general, maybe a time window of 10 minutes would be enough. Users not finding what they are looking for within 10 minutes are more likely to give up than rewording their question yet again.

However, too short or too long sessions will bias the result. Cutting a long session into several small ones, might make repetitions look like several users agreeing on a relation when there is in fact only one user making this connection. Also, we might lose connections if we cut one long session into several smaller ones. On the other hand, sessions that are too long could make it look as if co-occurrences found in several sessions are actually all in the same one, hence only counted as one connection when there are in fact many. Or we might connect unrelated sessions and create faulty co-occurrences. We might also consider discarding big sessions, above some threshold value (5 or 10) if we want to avoid noise from NAT and proxies. This problem will not, however, be addressed. To simplify, the IP is used to identify the user, even in the case where cookies are enabled. The timestamp is used to limit the time window, which will continue as long as a new query is posted within five minutes of the previous one.

3.1.2 Refining selection

A straightforward approach will render a lot of irrelevant documents for popular queries. The user might have found what was asked for in the first query and begun looking for something new, or for some other reason the same subject is not addressed in the queries. Thus the initial list of "related" queries must be refined gradually.

Co-occurrence threshold We may require that more than one user have made the same rewording during a session. If queries co-occur more than once, they are more likely to have a relation. For this, it is important to find a suitable threshold. Too low a threshold will incur a loss of precision while too high a threshold will not leave us with any interesting co-occurrences to work with, i.e. the recall will suffer. Initially the suggestions require two or more co-occurrences. Looking at the results from the development set, some queries seem to benefit from increasing the threshold from two to three co-occurrences. The question is whether the number of queries that benefit from this increase is greater than the number of queries that lose out. The question is whether this increase is beneficial or otherwise. A quick evaluation of the development set shows that approximately 40% of the queries in the set contain suggestions with only two co-occurrences, 75% of which are useful. Consequently, the threshold remains at two. However, for a more extensive log material, we might reach a point where we will benefit from increasing this threshold. Or a more sophisticated threshold could be introduced, taking into consideration not only the co-occurrence number but also the number of sessions containing the original query or the number of sessions containing the suggestion in question.

Edit distance and misspellings Among the suggestions, some are often misspellings or merely the plural form of the original, thus rarely relevant. In correctly spelled queries, the misspellings are most often found by the end of the suggestions list, thus rarely blocking the more common ones from being displayed. But it is not unusual that an alternative number is one of the more common co-occurrences, though a uncommon and unique suggestion would be of greater interest. We assume that words with an edit distance below a certain threshold are likely to be versions of each other rather than separate words. An attempt to sift these out is made by removing suggestions with a Levenshtein distance of less than 3, from the original query. The Levenshtein distance is one of several algorithms for calculating edit distance. It counts the minimal number of operations needed to transform one word into the other. Operations in this case refer to insertion, deletion and substitution of a character. This way redundant suggestions are removed in favour of somewhat less frequent, though more relevant ones.

The cases where we do want the similar suggestions to be displayed, is when the original query is inclined and a more common form would be of help, or even more importantly, when the original query is misspelled. Two problems occur due to the sifting: the problem of making out the correct word in a list of more or less misspelled suggestions, and the case where the suggestions are too short for the edit distance to carry any relevance.

If the original query itself is misspelled, the correct spelling might occur within the suggestions, hence being removed. How do we distinguish which of the versions is correct and which is misspelled? The most intuitive approach is just checking the number of occurrences in the logs for each word. For common queries, the number of misspelled suggestions is quite high. Still, we assume that the correct one will be the most common. For some rare words, this might be an erroneous assumption, though not so disturbing as needing to be addressed at this stage. Alternatively, we may compare the number of hits presented by the search engine, according to its statistics. That way we do not depend on the spelling skills of the users, but on the quality among the documents that form the search results. According to the development sets, counting search results and extracting suggestions with high a hit rate provide more suggestions, though with a considerably lower precision. This indicates that relevant pictures are often found in a misspelled context. The number of cases where the log counts evoke these kinds of suggestions is not overwhelming. However, the precision among them is quite high, according to the test sets. In the model, the suggestions with a higher occurrence in the search log count will be presented as a separate category, as possible suggestions for spelling corrections, while those with a lower hit rate are simply removed from the suggestions list.

When the query is too short, the measure of edit distance is irrelevant, as many of the suggestions, whether relevant and correctly spelled or not, will find themselves within a short edit distance (cat →rat, bat, car, hat). A simple *ad hoc* solution for this application is to skip the sifting part for queries shorter than four characters. A more sophisticated solution would be to relate the distance to the phrase length.

3.1.3 Narrow/expand

The list of suggestions is sorted, to be able to match different kinds of requirements from the user. The suggestions containing the original query are regarded as narrowing the original search. Those not containing the original one are considered as expanding it. (10)

3.2 Source 2: Using WordNet

The alternative method of finding related phrases will be using a thesaurus. Synonyms may be found using the WordNet synset of a query. To expand the query further, using a superset or a subset might be useful (16). According to Voorhees (17), query expansion using a general thesaurus like WordNet, would in many cases decrease precision as well as recall, or, in the best case, not offer any significant improvement. This, however, assumes a random use of selected terms, independent of user preferences. If possible expansion terms are only presented, instead of being automatically included in the search, irrelevant terms may be ignored, not letting them affect the search result.

Jones et al (10), point out that WordNet will not allow us to find suggestions for "new concepts such as products, movies and current affairs". We can expect that to be true for the greater part of the names appearing in a

Table 3.1: WordNet concepts evaluated with the development set.

hypernym The generic term used to designate a whole class of specific instances. Y is a hypernym of X if X is a (kind of) Y .

hyponym The specific term used to designate a member of a class. X is a hyponym of Y if X is a (kind of) Y.

synset A synonym set; a set of words that are interchangeable in some context without changing the truth value of the preposition in which they are embedded.

instance A proper noun that refers to a particular, unique referent (as distinguished from nouns that refer to classes). This is a specific form of hyponym.

entailment A verb X entails Y if X cannot be done unless Y is, or has been, done.

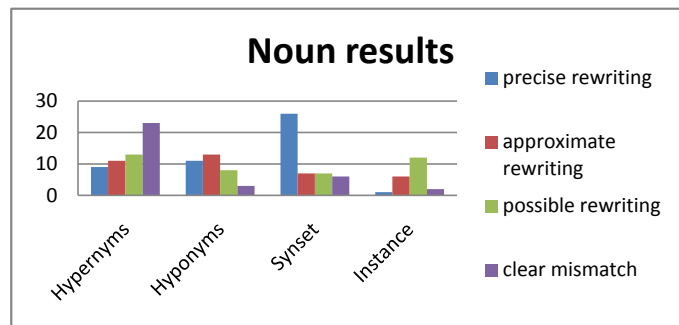


Figure 3.1: Noun quality. Counting the number of suggestions within each rank to compare the relevance of different WordNet categories.

standard search engine, since WordNet is not updated whenever new celebrities emerge. That cannot be solved with WordNet alone but requires a second source.

3.2.1 Selection within WordNet relations

To decide which WordNet relations supply the most relevant suggestions, the development set is consulted. If the query is found in WordNet, its related terms are evaluated with respect to relevance. The concepts in table 3.1 will be evaluated with respect to quantity and quality. The classes of suggestion relevance, table 3.2, from Jones et al (10) are used, to create a common standard of ranking.

Looking at the tables displaying the ranking of the suggestions for each synonym group, figure 3.1 and figure 3.2, the following conclusions can be made. The suggestions from *synset* are more often relevant than any of the other sets.

Table 3.2: The four levels of the ranking system by Jones et al (10)

1	precise rewriting: the rewritten form matches the user's intent, allowing for extremely minor variations in connotation or scope	e.g. frog - tree frog, leptodactylid
2	approximate rewriting: the rewritten form has a direct close relationship to the topic described, but the scope has narrowed or broadened or there has been a slight shift to a closely related topic	e.g. frog - toad, frog prince
3	possible rewriting: the rewritten form either has some categorical relationship to the initial query or describes a complementary product, but is otherwise distinct from the original user intent	e.g. frog - amphibian, snake
4	clear mismatch: the rewritten form has no clear relationship, or is nonsensical	e.g. frog - ant, eva longoria

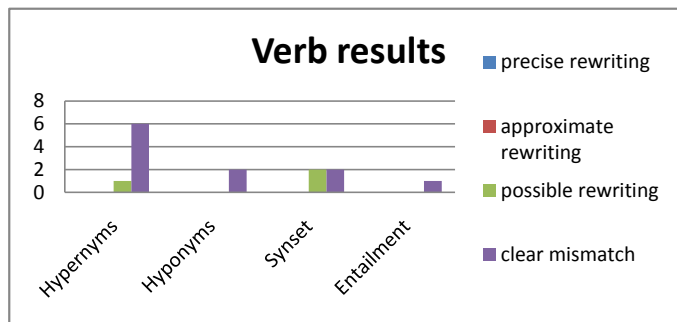


Figure 3.2: Verb quality. Counting the number of suggestions within each rank to compare the relevance of different WordNet categories.

Unfortunately, they do not exist for all queries. Another good category would be *hyponym*, considering that both 1 and 2 are higher than 3 and 4, where 4 is particularly low. Accordingly, these will also be taken into consideration. The set of hyponyms is often large, calling for some kind of limitation. By checking the cache count for the search engine, it is possible to retrieve the number of hits for each of the hyponyms, given that they have been queried recently enough to remain in the cache. This way they may be ranked according to number of hits in the search engine database. (However, this method rendered no hyponym suggestions at all to the test sets, and we need a better method for sifting.) In a live version, suggestions from the different synonym sets may be merged, weighted and suggested accordingly. Whenever the terms are identified as verbs, which is quite rarely, the relevance is too low to be considered a useful suggestion. Hence, synonyms for verbs will not even be looked up.

4 Evaluation method

Two aspects will be evaluated, the quantity and the quality of the suggestions. Quantity refers to the actual number of suggestions each source is able to provide, while quality is a measure of how relevant each suggestion is, with respect to the original query. There are two ways of evaluating the quality of the suggestions given by the two sources: manual evaluation or automatic evaluation of user clicks. Due to the limited size of this project, a manual evaluation is considered to be the most suitable.

4.1 Quantitative evaluation

The quantitative evaluation is the question of how many suggestions the sources are able to provide. This evaluation will be done manually, simply by counting the number of suggestions provided in our test runs.

4.2 Qualitative evaluation

4.2.1 About automatic evaluation

The most accurate evaluation might be obtained by letting real users judge, i.e. evaluating user clicks. If users click the suggested related queries, these may be considered relevant. If they also click the documents resulting from the related query, this is an even better result. This is true for a general search engine, but in a search engine for images, clicks are not naturally signs of relevance, but just as often a sign that the user found the image fun, strange or sexy. We might get relevant clicks for the suggested related queries, but should not count on the user rates when considering image clicks. Furthermore, only 10% of users even bother to click. (2) Thus, in a small context, gathering enough information might take very long time, and the result might not even be relevant. Also, this requires the application to be made more or less public, which might not be feasible in practice for a small project like this.

4.2.2 About manual evaluation

The more simple approach is a manual evaluation by creating a test set containing a balanced sample of queries. The resulting suggestions for related queries are graded manually to make the results comparable. For a broader feedback, several users could be asked to participate in this kind of evaluation. Perhaps some kind of pooling method is useful. Selecting a number of genres and a

Table 4.1: Share of different suggestion categories within the top and bottom decile.

	top decile	bottom decile
plural	16%	43%
singular	84%	57%
single word	40%	32%
multiword	16%	48%
proper name	44%	21%
misspellings	-	0.05%

number of queries within these, counting the number of relevant suggestions per query as well as per genre is a simple, yet a sufficiently exact method to measure precision, though not recall, according to Sato and Sasaiki (13) who used five genres and ten queries each.

4.3 Evaluation procedure

For this project, the evaluation is done manually and four different users are given one of two sets of queries. The sets (containing between 60 and 70 queries each) are based on common queries and rare queries respectively. Common queries are collected from the list of top queries of one month (i.e. march/april 2008) from the Picsearch search engine logs, by splitting the list randomly in two, creating development and test sets. Rare queries are collected from the logs, selecting only those qualifying within the limit for rareness (i.e. fewer than 50 hits in the logs for one year). These sets can be said to correspond to the top and bottom decile of queries, according to Jones et al (10). The two sets of queries differ not only in how common the queries are, but also in how they are phrased and what they address. These details might give a hint as to the kind of queries for which it is more or less easy to find suggestions. The sets differ with respect to number, multiword queries, proper names and misspellings, as can be seen in table 4.1. Multiword queries are a lot more common in the set of rare queries, names and singular queries are more common in the top set, and misspellings only exist in the rare set (i.e. no misspellings are frequent enough to make it to the top queries). We will run both test sets through the recommendation systems based on logs and WordNet, and evaluate all suggestions.

The users are given a set of instructions for their evaluation, to make sure they all understand the criteria of the four different ranking scores by Jones et al (10), shown in table 3.2, and how to use them in different contexts such as names of actors or artists, movie titles, animals or abstract nouns.

Depending on the nature of the search, different levels of rewriting might be acceptable. Either only specific rewritings of the original query are of interest, or broader associations might help in finding relevant documents or images. Thereby, in different contexts 1+2 or 1+2+3 might be acceptable results, in which case either 3+4 or just 4 count as irrelevant. The authors refer to these as *specific rewriting* and *broad rewriting* respectively. Though the evaluation method is taken from Jones et al (10), our results are not entirely

comparable. Firstly, they are using a machine learned classifier for the ranking, while we are using manual ranking, pooling the results, i.e. both sets are graded twice, by two different users, and the results for each set are merged by calculating the mean value. Hence, each suggestion has been given two, possibly different, rates. Secondly, their queries are divided into 10 deciles (each containing 10% of the total number of unique queries), according to the query search rate, while our two suites might be said to cover the top decile and a few of the lower deciles respectively.

5 Results

Again, two types of results are of interest: quantity and quality. We will first look at the quantitative results and then at the results of the manual quality evaluation.

5.1 Quantitative measuring

How many queries produce suggestions? Is there a pattern as to which types of queries get more or fewer suggestions?

In the system devised by Jones et al (10) about 50% of the queries produce at least one suggestion, and they suggest data sparseness, adult queries (which usually make up 5% of all input and are eschewed to preserve "family friendliness") and the *novelty constraint* (i.e. rejection of "suggestions deemed to be identical to the original query", which we try to find and sift out by measuring the Levenshtein distance) as reasons for lack of results. Their top decile result is just below 80% and the bottom decile yielded suggestions for more than 10% of the queries, though only by phrase substitution. In this project 88% of all queries get at least one suggestion from the log, 88% of the top decile and 89% of the bottom decile. This difference does not seem to be significant for our limited amount of data, though no statistical tests have been performed. Apparently the log contains enough data to provide common as well as rare queries with suggestions, at least as long as we only look at quantity and not quality. If we look into the actual number of suggestions, shown in figures 5.1 and 5.2, the difference is more substantial. WordNet provides suggestions for 15% of the evaluation set queries, 19% of the top decile and 13% of the bottom decile. This difference looks big enough to suggest that WordNet finds rare queries rather more difficult.

The coverage is a lot better using log mining than WordNet. It would appear that this can be partly explained by the multiword queries, which will never be found in WordNet, and the names, with which, WordNet can never be kept up-to-date. If we strive always to provide the user with at least one suggestion, WordNet alone is not enough.

As shown in figure 5.1 the log mining provides on average 2.7 suggestions per query, but the number varies greatly between deciles and reaches 3.7 for the top decile of queries, and 1.5 for the bottom decile. The result for rare queries is substantially sparser, confirming that sufficient log data is a condition for generating suggestions, not to mention relevant ones. WordNet only provides 0.5 suggestions, i.e. only one every second query. Counting deciles separately, the numbers are 0.6 for the top decile and 0.3 for the bottom decile.

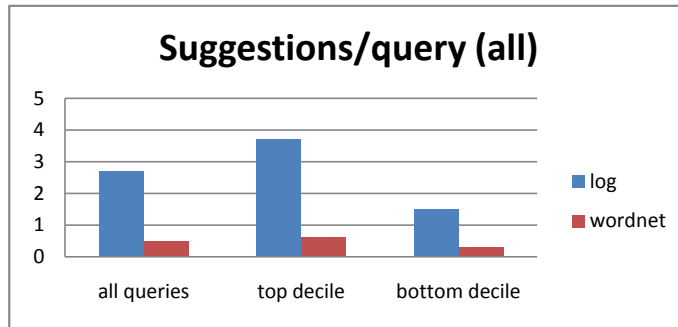


Figure 5.1: Suggestions per query, counting all queries, even those without suggestions.

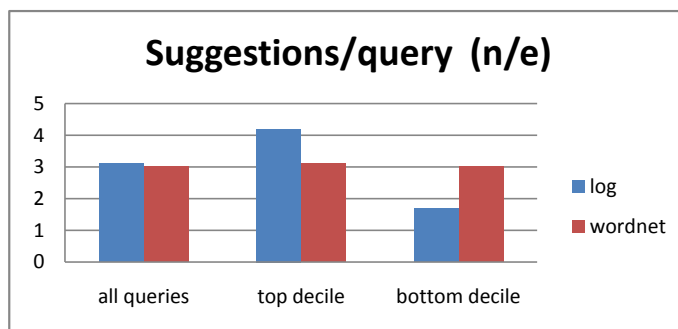


Figure 5.2: Suggestions per query, counting only queries given at least one suggestion by the source in question.

This is the overall result, splitting all suggestions between all queries, even those with no suggestions. Figure 5.2, on the other hand, shows the statistics for all non-empty queries, i.e. those that were actually found in the log mining dictionary or WordNet respectively. The difference indicates that whenever a query is found in WordNet, it can very well compete with log mining, with respect to quantity. We need to improve the matching of the query with the WordNet entries, either by modifying the query to match the basic form of the database, or by partial matching of the multiword queries. The lack of names in WordNet remains a problem and can only be addressed by adding another source. Data sparseness seems to be the bigger issue for log mining, since the bottom decile, i.e. rare queries, produces less than half the number of suggestions, compared to the top decile. Perhaps the solution is the same as for WordNet, since rare queries often are plural forms or multiword queries containing common words or phrases found in the log mining dictionary, see table 4.1.

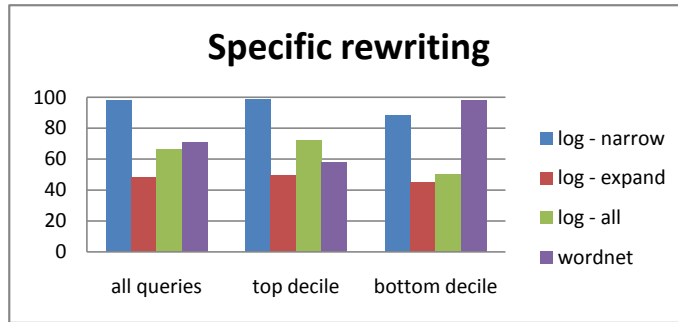


Figure 5.3: Comparing the share of specific rewriting for both sources, and for both data sets. Log mining results as a whole as well as split into narrowing and expanding suggestions.

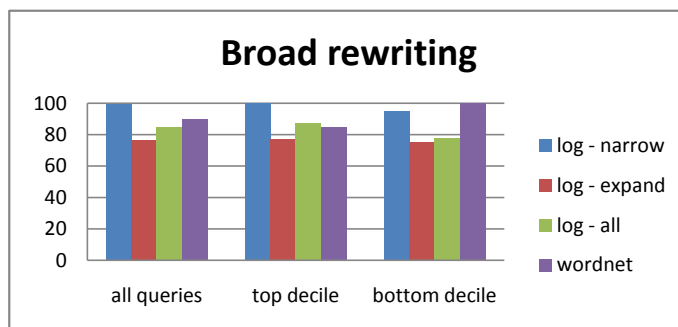


Figure 5.4: Comparing the share of broad rewriting for both sources and for both data sets. Log mining results as a whole as well as split into narrowing and expanding suggestions.

5.2 Qualitative measuring

The rating statistics for the suggestions within each group measures the quality of the two methods. Figures 5.3 and 5.4, show the results for WordNet and for mining user logs. The log based suggestions, are also split into *narrowing* suggestions, i.e. containing the original query, or *expanding* suggestions, i.e. not containing the original query. The reason for this is the difference between the two groups, which makes it a bit misleading to treat the log mining results as one homogenous group.

5.2.1 Specific and broad rewriting

According to the *specific rewriting* table, (i.e. suggestions rated 1 or 2), figure 5.3, the share of *specific rewriting* does not differ much between the two methods, comparing the result for the whole set of queries: 66% for log mining and 71% for WordNet. The number for *broad rewriting* (i.e. suggestions rated 1, 2 or 3), figure 5.4, is 85% for log mining and 90% for WordNet. The overall result is a little higher for WordNet, which might indicate that these connec-

tions agree with those of the users, and might be useful as long as the user is given the chance manually to select the relevant query expansion suggestion. Counting only pure *broad rewriting* (i.e. only suggestions rated 3), the number is 19% for both log mining and WordNet. The two methods show a surprisingly similar performance with respect to rating. However, there is a difference regarding the remaining suggestions, the *Clear mismatches*, which are made up of 15% of the log mining suggestions, and 11% of the WordNet suggestions.

5.2.2 Narrowing and expanding

When the log results are evaluated apart, the *narrowing* and the *expanding* suggestions give quite different rates. 98% of all the *narrow* suggestions are considered *specific rewriting*, and so are 99% of top queries. This lies in the definition of narrowing: just adding words to a phrase makes the change of meaning quite limited. It is rather a clarification of the original meaning, or a chance to focus on one of many meanings within an ambiguous phrase.

Regarding the *expanding* suggestions, only 48% are rated *specific rewriting*, and 76% of the suggestions fall under *broad rewriting*. Considering that these suggestions should be expanding, as the label indicates, this difference can be regarded as positive. We want to expand the focus, step outside the original result set and possibly find alternative meanings for the original phrase. These last 28% might be the most interesting ones, from an expansion point of view.

5.2.3 Top and bottom decile

Not surprisingly, the quality of the result differs between the two test sets. Log mining gives 72% *specific rewritings* in the top decile and only 50% in the bottom decile. Apparently, the latter is significantly more difficult to match. The result evens out somewhat for *broad rewriting*, to 87% for the top decile and 78% for the bottom decile. *Clear mismatch* is, consequently, 13% of the top decile suggestions, and 23% of the bottom decile suggestions. WordNet gives 58% *specific rewritings* in the top decile and 98% in the bottom decile, and as many as 85% *broad rewritings* in the top decile and 100% in the bottom decile. The 15% *clear mismatches* for the top decile suggest that the WordNet associations are not in sync with the user associations and we might have benefitted from a domain specific thesaurus. The problem is which domain to use for a general search engine. Examples in table 5.1 show where WordNet matches the word, but fail to match the most recognized meaning. The most common use of *easter* is as the holiday, not the point of the compass, which make up most of WordNet's suggestions, and the most common intention of *fantasy*, at least in a search engine for images, is the genre, not the more general concept "imagination".

5.2.4 Names

Looking only at names, which make up a large portion of both test sets, log mining reaches 68% *specific rewriting* and 84% *broad rewriting* while WordNet suggestions are too few to be significant. Only two names were found in the WordNet database, giving a total of 7 suggestions, all of them rated *specific*.

For log mining these numbers are close enough to suggest that the portion of names in the evaluation sets is high enough to have a considerable effect on the general result. A large portion of the *specific* suggestions are the *narrow* ones, which have only been extended with a date or a more or less significant specification (e.g. ronaldo 2007, eva longoria bikini), which is related but not ground-breaking.

5.2.5 Mean rate and example queries

The overall relevance is measured by calculating the mean value for all suggestions within a certain group. The results differ greatly between *narrowing* and *expanding* the search. *Narrow* might be considered "safe" with a mean rate just above 1, but might not always be useful, considering it always contains the original query. The query `werewolf`, see table 5.1, produces only one narrow suggestion: `werewolf transformation` which might change the scope, but not significantly where images are concerned. The *narrow* suggestions for the query `chicken` actually touch three different topics (i.e. cartoons, food and chicken run), which all must be considered relevant within their respective scopes, and all rated as *specific rewriting*. *Expanding* is far trickier, according to the higher rates, just above 2.5. In the `werewolf` example, all suggestions are rated 2 or 3, i.e. *broad rewriting*. This may be considered positive since an expansion of the query is supposed to hit outside the original search, otherwise it would not be an expansion. As regards the expanding result for `chicken` three of the five suggestions are "not relevant", thus rated 4. These suggestions are outside *broad rewriting*. This is an example of where the rewriting might have been a bit too broad, unless the user is looking for "food in general".

WordNet suggestions are generally rated lower, i.e. 2 for the top decile and 1 for the bottom decile, but since it only provides 0.5 suggestions per query altogether, or three suggestions per query found in the WordNet database, this is not a reliable source. The high proportion of low-rated suggestions indicates that the suggestions might be of help when the user needs rewording within the same topic, but of less assistance for the user who wants to expand the search and find related topics. Many examples in the test set support this assumption. All WordNet suggestions for `werewolf` are pure rewordings of the original query without any change of the scope. For `chicken` three of five suggestions are pure rewording, still meaning no more or less than "the animal used for food", while two suggestions change the scope from the animal to human characteristics and are consequently higher rated (i.e. *specific rewriting* or *broad rewriting*) by the test users.

Examples in the query table, table 5.1, show how narrow suggestions focus on different scopes of the original query, rated *specific* or *broad* by users. The query `bugs` may be narrowed into such different topics as `lady bugs` or `bugs bunny`. Expanding suggestions are not all considered *broad*. A suggestion not containing the original query might very well be *specific* if the suggestion is a synonym or otherwise closely related (e.g. hypernym or hyponym), as is the case for the suggestions `insects`, `spiders` and `bee`. The suggestions for `heart` are uniform within the two groups, where all narrowing suggestions are *specific rewriting* and all expanding suggestions are *broad rewriting*.

WordNet suggestions are mainly semantically rather than conceptually re-

Table 5.1: Query/suggestion examples with ratings

query	log - narrow	log - expand	WordNet
easter	easter eggs (1), easter bunny (1), easter bunnies (1), easter egg (1), happy easter (1)	christmas (3/2), chicks (2), sun (4), summer (3), halloween (3/2)	easter (1), east wind (4), easterly (4)
fantasy	fantasy art (1), fantasy dragons (1), dark fantasy art (1), fantasy angel (1), fantasy wallpaper (1)	fairies (1/2), angel (1/3), dragons (1), magical creatures (1), dreams (4/3)	synset phantasy (4/1), illusion (4/3), fancy (4)
bugs	lady bugs (1/2), bugs bunny (3/2), microscopic bugs (1), ugly bugs (1), bugs and insects (1)	insects (1), spiders (1/2), tiger (3/4), bee (1/2), south park (3/4)	-
heart	love heart (1), love hearts (1), pink heart (1), human heart (1), pink love heart (1)	love(2), stars (3/4), lips (3), star (3), flowers (3)	nerve (3), kernel (3), nitty-gritty (3), essence (3), warmheartedness (3)
vampire	vampire bats (1), vampire bat (1), scary vampires (1), scary vampire (1), real vampire (1)	Dracula (1), goth (2), bat (2), zombie (2/3), wolf (3)	lamia (1/2)
werewolf	werewolf transformation (1)	dalek (3), wolfs (2), demons (2/3), dragons (2/3), wolf (2)	wolfman (1/2), lycantrope (1), loup-garou (1)
chicken	cartoon chicken (1), roast chicken (1), cooked chicken (1), fried chicken (1), chicken run (1/3)	fish (3), meat (4), cheese (4), fruit (3/4), food (2/3)	poulet (1), crybaby (2/3), gallus gallus (1), volaille (1), wimp (2/3)

lated, and suggestions might be more useful in a context focusing on text rather than one focusing on images. The reason for this might lie in the fact that all suggestions are collected from the *synset* category, while wider categories, like hyponyms and hypernyms, were left out. Still, both *specific rewriting* and *broad rewriting* occur. Not surprisingly, WordNet gives no suggestions for the plural noun, since only singular forms are represented in the WordNet database. This needs to be addressed for WordNet to become an interesting source.

5.3 Spelling suggestions

The suggestions within a Levenshtein distance of 1 or 2 from the original query were kept as spelling suggestions if they were more frequent in the logs than

the original query. This kind of spelling suggestion was found for 12 queries: six in the top query evaluation set and six in the rare evaluation set, of which three were misspelled. Apart from some noise around correctly spelled queries, this actually had the intended effect. For all three misspelled queries, the correct spelling was found. For the correctly spelled queries, six were given the plural form of the singular query, or vice versa, and one was given a more common model of the car in the original query, i.e. not a spelling variant. Two of the spelling suggestions were irrelevant being just nouns with a high portion of characters in common with the original. Both of these were found in the top set.

5.4 Improvements

There are several areas where improvements can be made. We may look into the dictionaries, the algorithm for using them, as well as the selection of suggestions among the, possibly extensive, lists provided by the log mining algorithm. The WordNet method could also benefit from better selection among the suggestions through some kind of sorting, but more importantly, tweaking of the original query by lemmatizing or partial matching so as to better match the contents of WordNet.

5.4.1 Adaptation of original query and database

By lemmatizing the log mining dictionaries as well as the posted query, relations could be found that are now lost due to the usage of inclinations. If one session contains the singular form of a word and another session contains the plural form of the same word, these will be considered totally different queries and will be discarded along with possibly interesting co-occurrences. Especially for WordNet, this is an important issue, as all entries are in the same basic form. For English, a simple stemming algorithm could be used. It looks appears that many of the common queries appear both in the singular and the plural, but the singular is not necessarily the most frequent among users.

One way to address lack of suggestions would be phrase substitution, used by Jones et al (10), i.e. matching and replacing only parts of the original query, typically at the end of the query. Although whole-query suggestions tend to get a higher rank, sometimes, particularly for infrequent queries, phrase substitution might be the only way to generate suggestions. Typically, the last decile needs phrase substitution to generate results, while the top decile gets results from the whole query. (10) This could make a big improvement for the WordNet method, since multiword phrases do not occur in WordNet, but only simple concepts. For proper names, phrase substitution is obviously not applicable, or we end up with another person. Similarly, and probably a lot easier, a partial match, i.e. matching the entire query to parts of dictionary entries, would significantly improve the chances of finding relevant relations. In the current dictionary, only entire phrases can be looked up.

As earlier suggested, the co-occurrence threshold (presently 2) might need increasing as the log material grows larger. Or the threshold should be replaced by a more stable method, considering overall frequency along with the

co-occurrence number. One such method would be the *log likelihood ratio* developed by Dunning (3). The more sessions we collect, the greater the possibility of irrelevant co-occurrences. Similarly, words appearing everywhere do not tell us anything about their context. This is especially relevant regarding frequent but not significant queries, e.g. celebrities massively appearing in media, puppies or nudity. In short, things people search for randomly and thus occurring in almost any session. Several reports agree that the overall most common suggestions might even be discarded due to lack of significance. (10)

5.4.2 Selection

The results for log mining suggest that the collection of suggestions for the dictionaries might not be the major issue, but rather finding the relevant ones among the noise. For a common query, it is not unusual that as many as 600 unique suggestions are produced. Of course, not all of these are relevant.

In the paper by Weiss et al (18), a few methods for giving suggestions different weight are presented that might be applied to our system to improve the selection. Based on the assumption that a co-occurrence from a short session is more likely to be relevant than a co-occurrence from a longer session, each session could be given one point, to be split among the included queries. That way we give a higher weight to suggestions from a short session and a lighter weight to suggestions from verbose sessions. Suggestions might also be weighted according to relative similarity. Queries which are similar to some extent tend to aim at the same information. We do not want them to be too similar, as different inflections of the same query do not provide any new information. But, as the results have shown, suggestions containing the original query tend to get a high relevance rate. There is a greater chance that suggestions containing only parts of the original query are of use. We might give higher weight to suggestions that have an overlap in resulting documents, in relation to the original query, thus considering them possibly related. This is based on the assumption that closely related queries would tend to return a high number of common results and selected documents (19) (2). Since this assumption might not be valid for an image search engine, according to developers at Picsearch, we might try domains instead of documents. That is one way of addressing the lack of ordering between WordNet suggestions. The present procedure of sorting based on number of hits for phrases found in the cache is deceptive, as phrases not used recently will automatically be categorized as low frequent irrespective of the actual number of hits.

In a live version, suggestions from the different synonym sets in WordNet may be merged, weighted and suggested accordingly. The tables from the initial test of WordNet categories indicate that hyponyms as well as frames might be interesting given a better selection. The weight might be based on the total number of synonyms in the particular set, the occurrence of the synonym in the logs or whether the synonym is also suggested by the log-based algorithm. If log mining and WordNet are used together, precision might be improved by giving a higher weight to suggestions given by both. If two different methods agree, this might be interpreted as a better suggestion. Also, a combination of the two sources might compensate for their cases of data sparseness.

Analyzing whether the order of queries in one session is decisive in any way

might be considered an option. Queries from one user during a session should be gradually more relevant, not less. When selecting related queries from a logged user, maybe later queries are of greater relevance than preceding ones. Still, we do not know whether the user needed to generalize or specify their query in order to find a relevant match. However, this might not be an issue if we do not label the suggestions.

In the model, suggestions too close to the original query are treated separately, in order not to suggest versions of the same query. They formed a large share of the suggestions and eliminating them made room for new, unique suggestions. This treatment could be relevant for suggestions that are too close to other suggestions, as they too may block the spot for a new, relevant suggestion. The most common finding is the same suggestion in both singular and plural. In 'too close' we might also include 'query + number', i.e. ronaldo 2007. Removing insignificant 'query + specification' suggestions, like eva longoria bikini, could be trickier, as the specification might reach new, relevant documents. One way might be to remove the suggestion if the hit list is too similar to the one of the original query.

6 Conclusion

The suggestions from the two sources differ, but not to the extent we might expect. The quantitative result is considerably better for log mining than for WordNet, when counting overall number of suggestions. However, when splitting them only between queries rendering suggestions, results even out and WordNet is not far behind. The qualitative results differ a lot less, and a lot of the differences lie between the different types of queries, not between logs and WordNet. The major issues regarding mining user logs is to get rid of the noise, such as suggestions common enough not to bear any significance, to find suggestions also when the data is sparse, and to have the database connect suggestions that differ only by their inflection, in order to find more connections that may be relevant. Introducing a threshold whereby single co-occurrences were rejected was a major improvement. Since the suggestions were sorted by frequency, the low frequent ones were not an issue for the queries with tens or even hundreds of frequent relations. But for the queries with fewer co-occurrences, this noise considerably lowered the mean rate of the suggestions. The WordNet issues are the data sparseness regarding multiword queries and newer proper names, the fact that WordNet entries are always entered in the basic form while search engine input is not, and the difficulty of sorting the suggestions without any direct frequency information such as in user logs. The light weight method seems to be ideal for this type of application, as it is economical with respect to resources, scalable and easily extended. In addition we need to enable searching for parts of phrases to find more relations, e.g. searching for `paper`, we might want to find `newspaper` along with all its co-occurrences not found for `paper`. The other way around, i.e. matching only parts of the original query, might help overcoming the issue of rare multiword queries.

Bibliography

- [1] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann, 1998.
- [2] Adam Berger Doug Beeferman. Agglomerative clustering of a search engine query log. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000.
- [3] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 19(1):61–74, 1993.
- [4] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.
- [5] Jian-Yun Nie Wei-Ying Ma Hang Cui, Ji-Rong Wen. Query expansion by mining user logs. *IEEE Transactions on Knowledge and Data Engineering*, 2003.
- [6] Bernard J. Jansen and Udo W. Pooch. A review of web searching studies and a framework for future research. *Journal of the American Society of Information Science*, 2001.
- [7] Bernard J. Jansen, Amanda Spink, Judy Bateman, and Tefko Saracevic. Real life information retrieval: a study of user queries on the web. *SIGIR Forum*, 1998.
- [8] Susanna Lnnqvist. Expansion av skfragor med svenskt ordnom termka. Master's thesis, Hgskolan i Bor 2006.
- [9] Mingtian Zhou Qilin Li. Research and design of an efficient collaborative filtering predication algorithm. In *Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on Parallel and Distributed Computing*, pages 171–174, 2003.
- [10] Omid Madani-Wiley Greiner Rosie Jones, Benjamin Rey. Generating query substitutions. In *Proceedings of the 15th International Conference on World Wide Web*, 2006.
- [11] Gerard Salton. *Automatic Text Processing*. Addison Wesley Publishing Company, 1988.
- [12] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Reidl. Item-based collaborative filtering recommendation algorithms. In *World Wide Web*, pages 285–295, 2001.

- [13] Satoshi Sato and Yasuhiro Sasaki. Automatic collection of related terms from the web. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 121–124, Morristown, NJ, USA, 2003. Association for Computational Linguistics.
- [14] Craig Silverstein, Monika Henzinger, Hannes Marais, and Michael Moricz. Analysis of a very large altavista query log. Technical Report 1998-014, Digital SRC, 1998.
- [15] Bill Slawski. Seo by the sea. <http://www.seobythesea.com/?p=975>, jan 2008.
- [16] Giannis Varelas, Epimenidis Voutsakis, Paraskevi Raftopoulou, Euripides G.M. Petrakis, and Evangelos E. Milios. Semantic similarity methods in wordnet and their application to information retrieval on the web. In *WIDM '05: Proceedings of the 7th annual ACM international workshop on Web information and data management*, pages 10–16, New York, NY, USA, 2005. ACM.
- [17] Ellen M. Voorhees. Query expansion using lexical-semantic relations. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 61–69, New York, NY, USA, 1994. Springer-Verlag New York, Inc.
- [18] Sholom M. Weiss and Nitin Indurkha. Lightweight collaborative filtering method for binary-encoded data. *Lecture Notes in Computer Science*, 2168:484+, 2001.
- [19] Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. Clustering user queries of a search engine. In *World Wide Web*, pages 162–168, 2001.