Using Linguistic Knowledge in Information Retrieval
Technical Report

Wessel Kraaij
TNO-TPD
Postbus 155
6200 AD Delft
The Netherlands
tel. +31-152692259
fax. +31-152692000
email:kraaij@tpd.tno.nl

Renée Pohlmann
OTS/STT
Utrecht University
Trans 10
3512 JK Utrecht
The Netherlands
tel. +31-302536064
fax. +31-302536000
email:Renee.C.Pohlmann@let.ruu.nl

Project coordinator:
Prof. Ir. S.P.J. Landsbergen (OTS/IPO)

Researchers:
Ir. W. Kraaij (TNO-TPD)
Drs. R.C.M. Pohlmann (OTS/STT)
Drs. H. Ruessink (OTS/STT)

Contents

Abstract

The current practice in Information Retrieval is largely based on statistical techniques. These techniques are reasonably successful but many researchers believe that statistical techniques have reached their upper bound. Some recent research in IR is aimed at investigating whether Natural Language Processing techniques can be used to improve the performance of existing retrieval strategies. In the UPLIFT project (Utrecht Project: Linguistic Information for Free Text retrieval) we want to investigate whether the addition of linguistic information will improve the performance of a statistical retrieval engine for the Dutch language. During the first phase of the project, which is now completed, we concentrated on morphological and semantic information (synonymy relations). Morphological information can be used during document indexing. The variation of index terms is reduced by using stems instead of word forms as the basis for indexing. Many algorithms have been developed to reduce word forms to their 'stem', ranging from simple non-linguistic truncation algorithms to dictionary-based linguistic algorithms. Previous research on stemming has shown both positive and negative effects on retrieval performance. In this report we will describe experiments in which several linguistic and non-linguistic stemmers were evaluated on a Dutch test collection. Results show that linguistic stemming can yield a significant improvement in Recall over non-linguistic stemming, without causing a significant deterioration in Precision. Besides testing morphological algorithms, we also experimented with a synonym database. This database was used to expand query terms with synonymous expressions. Results of our experiments show that synonym expansion is potentially useful but disambiguation of query terms is essential.

1 Introduction

The aim of the UPLIFT project[1] is to investigate whether the addition of linguistic information will improve retrieval performance for the Dutch language.

In the UPLIFT project we intend to investigate the effect of adding the following types of linguistic information:

- Morphological information
- Semantic information
- Syntactic information

During the first phase of the project, which is now completed, we investigated whether the addition of morphological and semantic information would have an effect on the performance of a statistical retrieval engine[2]. The results of this investigation are described in this report. In the next phase of the project we will investigate whether the addition of syntactic information will improve retrieval quality.

The report starts with a detailed description of the linguistic tools that were tested and a comparison with other approaches in the field (sections 2 and 3). Subsequently, we describe important aspects of the set-up of the experiments that were carried out: test collection, measuring procedures, test hypotheses and statistical validation (section 4). We conclude with a presentation and discussion of the results of the experiments (sections 5 and 6).

2 Morphological information

One of the techniques employed in Information Retrieval (IR) to improve performance is morphological analysis of index and query terms. The idea behind this is that Recall (i.e. the ratio 'number of relevant

[1] UPLIFT is sponsored by the NBBI, Philips Research, the Foundation for Language Technology (OTS), the Dutch Ministry of Education and Science and the Dutch Ministry of Economic Affairs.

[2] The retrieval engine used in the UPLIFT project is the TRU vector space engine developed by Philips Research [2].

articles retrieved'/'all relevant articles') will increase when morphological variance of terms (e.g. singular - plural) is reduced. Several different techniques have been proposed to achieve this goal. One of the simplest of these techniques, suffix stripping, uses a list of frequent affixes to reduce words to their base form or 'stem' e.g. [15], [19]. Suffix stripping algorithms are very efficient because they do not involve dictionary look-up, but due to their lack of linguistic information, e.g. about word class, they frequently introduce mistakes. Words can be reduced to the wrong stem resulting in conflation with semantically unrelated words (overstemming errors) or semantically related words are not reduced to the same stem (understemming errors). Furthermore, the 'stems' yielded by suffix-stripping algorithms are not necessarily linguistically correct so that further linguistic processing (e.g. translation, synonym substitution) is impossible. Based on an evaluation experiment with several different suffix-stripping algorithms, Harman [6] concluded that suffixing is not effective, at least not for English. Other researchers [14] [24] have reported favourable results using more linguistically motivated stemming algorithms. We will discuss their approach and the approach chosen in the UPLIFT project in sections 2.1, 2.2 and 2.3 below.

2.1 Suffix-stripping

Harman [6] compared three well-known suffixing algorithms for English: the S–stemmer, the Lovins stemmer [15] and the Porter stemmer [19].

Harman contrasted these suffixing algorithms with a baseline of no stemming at all. After a detailed evaluation Harman reached the conclusion that none of the stemming algorithms consistently improve performance. The number of queries that benefit from the use of a stemmer is about the same as the number of queries that deteriorate.

Popovič and Willet [18] investigated whether suffix stripping would be effective for a morphologically more complex language like Slovene. They developed a Porter-like algorithm for the Slovene language and tested this algorithm on a small Slovene test collection. Their experiment shows a significant improvement in Precision. Popovič and Willet's study also included an interesting control experiment. The Slovene test corpus was translated to English and the experiment was repeated. The results of this control experiment confirmed Harman's conclusion that Porter-like stemming does not improve retrieval for English documents. They therefore conclude that the effectiveness of stemming is determined by the morphological complexity of a language.

2.2 Linguistically motivated algorithms

Krovetz [14] developed a morphological analysis method for English which closely resembles suffix stripping but uses a dictionary to validate the result of rule application. He started with a modified version of the Porter algorithm that checked the existence of the resulting stem prior to applying the corresponding rule. However, the suffix rules of the original Porter proved to be incompatible with this approach (i.e. they do not always render a linguistically correct stem). He subsequently developed a completely new morphological rule component based on information about inflectional and derivational suffixes in the Longman's Dictionary of Contemporary English (LDOCE). The morphological rules cover the most frequent suffixes in English and they are only applied if the resulting string corresponds to an entry (i.e. a base form) in the LDOCE. They are not applied, however, if a morphological variant is listed as a separate entry in the lexicon. Krovetz assumes that if a variant is listed separately in the dictionary, its meaning may not be conflated with the meaning of the base form. This hypothesis was tested with a number of quantitative experiments and detailed qualitative performance analyses. A few problem areas were identified:

- inconsistency/incompleteness of the dictionary
- spelling errors in test corpus

- proper nouns
- hyphenation variation (e.g. on-line, on line, online)

Almost two thirds of the derivational variants in the test corpus were listed in the lexicon. According to Krovetz, 40% of these should not have been listed separately (i.e. the meaning of the derivational variant can be derived from the meaning of the base form). His experiments show, however, that although errors still occur, this stemming technique does result in improvements in performance compared to Porter stemming, especially with short documents.

Savoy [24] developed a slightly different morphological analysis module for French using syntactic information.

The French module is based on a two-stage process:

1. Inflectional suffixes are removed using a dictionary and a declension table. If a word does not occur in the dictionary, characters are removed one-by-one from the end of the word until they match an entry in the declension table. The declension table and the dictionary are linked to yield the dictionary entry and its syntactic category.
2. Subsequently, (possible) derivational affixes are removed with a stripping algorithm that is based on syntactic constraints (e.g. the derivational affix *-ique* only combines with nouns to form adjectives). This algorithm was refined by adding a module which generates slight spelling adjustments if the resulting stem is not present in the dictionary. This enables correct stemming of slightly irregular derivations.

A stop list (i.e. list of frequent (mostly function) words which are not suitable for indexing) based on grammatical categories (instead of surface forms) was also implemented. Since there are no standard test collections for French, Savoy tested his morphological analysis module on specially prepared word lists using an evaluation method developed by Paice [17]. Using this method a mean error rate of 16% was measured. Careful analysis of the data showed that errors are mainly due to highly irregular derivations and incorrect application of the spelling adjustment rules.

2.3 Morphology in UPLIFT

In the UPLIFT project we wanted to investigate whether stemming in general would be effective for the Dutch language and, moreover, we wanted to contrast linguistic stemming techniques with suffix stripping. We started with the development of suffix stripping algorithm for Dutch based on the Porter algorithm. Our version of the algorithm closely resembles its English original and consists of 98 rules which fully cover Dutch regular inflectional morphology and partly cover derivational morphology[3].

We subsequently developed two linguistic stemmers (inflectional and derivational) using a computer readable dictionary, the CELEX lexical database [3]. Using CELEX, two separate files were created which relate stems to their inflectional and derivational forms respectively. To avoid unnecessary overhead, not all possible forms were included in these files but only those forms which actually occurred in our test collection. In the case of ambiguity, i.e. a particular string can be related to two different stems (e.g. *kantelen* can either be related to the noun stem *kanteel* ('battlement') or the verb stem *kantelen* ('to turn over')), we simply selected the most common interpretation based on frequency information provided in the CELEX database.

Instead of creating separate indexes for each stemming variant, we used a method which was also used by Harman in her evaluation experiment [6]. Before the actual execution of a query by the retrieval engine, query terms are 'expanded' with related terms using the dictionary files. This technique allows the

[3] For a more detailed description of the Dutch Porter algorithm we refer to [12].

developer to vary the depth of morphological analysis (e.g. inflection only, inflection and derivation etc.) without having to create a new index for every possible variant. It also creates the possibility to manipulate the query vector (e.g. increase/decrease the 'weight' (i.e. importance) of certain forms, interactively remove unwanted terms etc.). The expansion method has one drawback though. The Vector Space Model (VSM) relies on the assumption that the n concepts (i.e. index terms) spanning up an n-dimensional vector space are uncorrelated [21]. This simplification reduces the query-document similarity computation to the inner product of their corresponding term vectors. The query expansion method, however, is a less optimal approximation of this assumption because morphological variants of the same concept are treated as independent base vectors. Harman corrected for this defect by modifying the similarity computation procedures: document frequencies for morphological variants of the same term are 'grouped'. This has the effect that morphological variants are mapped on a single concept in the vector space. We have experimented with a number of different variants of this correction procedure. We also ran a control experiment to compare a system where stemming is 'emulated' by query expansion with a system where the stemmer is used during the indexing process, i.e. the index contains stems instead of word forms. This index was built with the Dutch Porter stemmer. The result of these experiments will be described in section 5.2.4 below.

We will illustrate the query expansion method by means of a (simplified) example:

Consider the following query:

Ik zoek recensies van klassieke concerten die in het Muziekcentrum in Eindhoven zijn gehouden
(I am looking for reviews of classical concerts held at the Music Centre in Eindhoven)

After removal of stop words[4], the following query terms are left:

recensies
klassieke
concerten
muziekcentrum
eindhoven

Using the inflectional database only, these query terms are subsequently expanded with the following variants[5]:

recensies	*recensie* (singular)
klassieke	*klassiek* (non-inflected positive form) *klassiekst* (superlative)
concerten	*concert* (singular)
muziekcentrum	
eindhoven	

The derivational database yields the following additional query terms:

recensies	recensie	*recensent* (reviewer)
klassieke	klassiek klassiekst	
concerten	concert	*concerteren* (to perform a concert)
muziekcentrum		
eindhoven		

Careful analysis of the document collection used in the UPLIFT project (see section 4.1 for details), revealed that of a subset of approximately 50,000 unique word forms ± 20,000 were not included in CELEX. We examined a random sample of ± 2,500 of these words to establish why they were not in the dictionary. The results of this analysis are summarized below:

[4] Besides the dictionary modules we also developed a Dutch stop word list, a tokenizer which extracts individual words from the texts by recognizing word boundaries, punctuation characters etc. and a small morphological rule component which contains rules for some of the most frequent omissions in the CELEX database (e.g. '-tje' (diminutive), '-baar' (-able), '-heid' (-ity)).

[5] Remember that only those variants which actually occur in the document collection are added to the query.

46%	proper names
37%	compounds
10%	spelling mistakes
3%	other language
3%	morphological variant not in CELEX
1%	stem (and variants) not in CELEX

The majority of words not included in CELEX are either proper names or compounds. We anticipated that compounds would be a problem case. In Dutch, compounds are generally formed by concatenating two (or more) words to create a single orthographic word, e.g. *fiets* ('bicycle') + *wiel* ('wheel') → *fietswiel*. As compounding is a very productive process in Dutch, every dictionary is necessarily incomplete in this respect. To handle this problem, some stemmer versions were extended with a compound analyser, the 'word splitter' developed by Theo Vosse for the CORRie (grammar checker) project[29]. The word splitter will try to split a compound into its components (stems) on the basis of word combination rules for Dutch and a lexicon. If the splitter is unsuccessful, the word is left unchanged. The following results were obtained with the compound splitter using a random sample of ± 1,000 compounds not included in the CELEX dictionary[6]:

5%	no analysis
3%	incorrect analysis
92%	correct analysis

The compound splitter was used to create a separate compound file consisting of stems and compounds containing the stem. This file was used in a slightly different way than the inflectional and derivational databases. At first we experimented with adding all compounds which contain the stem of a query term to the query. For the example above this would result in the following expansion:

recensies	*boekrecensie* (book review)	*filmrecensies* (film review)	*recensiewerk* etc. (review work)
klassieke	*klassieke-muziekliefhebbers* (classical music lovers)	*Elvis-klassieker* (Elvis classic)	*popklassieker* etc. (pop classic)
concerten	*popconcerten* (pop concerts)	*live-concerten* (live concerts)	*concertgangers* etc. (concert goers)
muziekcentrum			
eindhoven			

After some initial experimentation we concluded that this form of query expansion was too inaccurate and needed to be refined. Too many terms (some stems proved to be very productive and yielded more than a hundred compounds) which were too far removed in meaning from the original terms, were added to the query, resulting in very poor retrieval performance. We subsequently considered a reduced version of the expansion where only those compounds are added where the original query term is the head of the compound (in Dutch, most compounds are right-headed, i.e. the right element of the compound determines the basic meaning of the whole, the left element is a modifier). This version, however, still performed very poorly. We finally implemented two very restricted forms of query expansion using the compound database. In one variant, compounds already present in the query are split into their components, which are subsequently expanded and added to the query. For our example, this would yield the following additions:

recensies		
klassieke		
concerten		
muziekcentrum	*muziek* (music)	*centrum* (centre)
eindhoven		

In the second version, new compounds are constructed using elements (stems) already present in the query. Query stems are paired and the resulting compound is subsequently validated in the compound database. For our example, this would lead to the addition of one compound only: *concertrecensie* (concert review).

[6] Some frequent compounds are included in the CELEX dictionary.

3 Semantic information

Another source of linguistic information which has been used in information retrieval is semantic information. The idea behind this is that adding knowledge about the meaning of words and using this knowledge to disambiguate words in context and to identify relations between words, will improve retrieval quality. Several different techniques have been proposed to use semantic information in the retrieval process. At this point we need to make a distinction between IR systems developed for a specific domain and Free Text retrieval systems[7].

3.1 Domain specific systems

IR systems developed for a specific domain (e.g. law, medicine etc.) rely on the assumption that certain word senses do not occur (or are at least very unlikely) in the domain. In medical texts, for instance, a word like 'dressing' will probably only occur meaning 'bandage' and not 'salad dressing'. In domain specific IR systems this assumption can be exploited by (manually or automatically) constructing thesauri which represent the concepts of the domain and their relations (e.g. synonymy, antonymy, hyponymy etc.) and using these thesauri for indexing of documents and query expansion (i.e. adding related terms to the query). Examples of these kinds of systems are numerous, for a detailed discussion of the techniques involved we refer to [21].

3.2 Free text systems

3.2.1 General

Free text systems cannot resort to the techniques described above. Instead of *global* disambiguation (i.e. excluding certain word senses based on domain knowledge), *local* disambiguation techniques (i.e. disambiguating words in their immediate context) are applied. Several ways have been devised to achieve local disambiguation. One way is to match the context of an ambiguous word against a description of the different senses in a dictionary and choose the right interpretation on the basis of some sort of similarity computation. Large, general purpose thesauri like, for example, WordNet [28], have also been used in free text systems. In this case, however, disambiguation is frequently left to the user and is not performed automatically, although some attempts in this direction have been undertaken [11].

3.2.2 Semantics in UPLIFT

In the UPLIFT project we have also experimented with a general purpose thesaurus for Dutch (EUROGLOT database, developed by Linguistic Systems). The EUROGLOT database consists of a large number of 'synonym groups' for Dutch which can be accessed through each of the members in a group. The synonym database is structured in the following way: a distinction is made between 'synonym rings' (strict synonymy) and 'synonym groups' which also contain less strictly related words like hyponyms and hyperonyms. A synonym group may contain one or more synonym rings corresponding to different senses of a particular word. The synonym database is also used for query expansion (cf. section 2.3). Every query term is looked up in the synonym database and related terms (strict synonyms, i.e. rings, when available, otherwise less strictly related words) are added to the query. No real attempt is made to disambiguate query terms in a principled manner. All query terms are first processed by the morphological analysis module (cf. section 2.3), and if the query term is ambiguous, e.g. *fietsen* (noun or verb), the morphological analysis module will select the most common interpretation on the basis of frequency information provided by CELEX. This choice is subsequently maintained for synonym look-up. It is evident that this procedure does not

[7] We will not discuss purely statistical methods like, for instance, clustering and Latent Semantic Indexing, since they fall beyond the scope of this report.

always disambiguate correctly. In the case of purely semantic ambiguity, e.g. *buis* (noun meaning a.o. 'tube' or 'television set'), disambiguation is impossible and synonyms of both senses of the word are added. In the next phase of the UPLIFT project, we hope to improve disambiguation by adding more linguistic information to our system. We plan to incorporate a tagging procedure which will at least eliminate (or drastically reduce) syntactic category ambiguities like the *fietsen* (noun or verb) case. For a detailed description of the performance of the different versions of the UPLIFT system that were created using the synonym database, we refer to section 5.2.3 below.

4 Design of the evaluation experiment

IR evaluation is not an easy task, because great precision and care are required to make a valid statement about whether the data supports or falsifies the hypothesis of the experiment and more important: to ensure that results can be generalised. Important factors are: collection scope and size, evaluation method and statistical validation. IR evaluation publications sometimes suffered from insufficient precautions on one of these aspects. As a consequence, experimental results could not always be compared. This made it difficult to make significant advances in the IR field, because less fruitful approaches were not pruned. The TREC initiative ([7, 8, 9]) is an improvement in this respect as it provides researchers with standard test corpora including relevance judgements.

We decided to base the quantitative analysis of the different UPLIFT versions on the traditional Recall & Precision measures. There are several other measures such as Van Rijsbergen's E-measure or the average search length, but we decided to adhere to the emerging standards of IR evaluation as set by the TREC experiments which in turn are largely based on SMART evaluation procedures. Although current IR evaluation research is almost exclusively based on English test corpora[8], we hope that this will increases the level of significance of our work for the international IR community. In the following subsections we will describe the preparations for the final experiment and some important issues concerning the evaluation of results.

4.1 Test collection

Since the UPLIFT project aims at developing domain independent full text retrieval stategies, we considered the following candidate texts for our document collection: articles in newspapers, encyclopedias, weekly magazines etc. One of the major Dutch publishers of regional newspapers (VNU) kindly offered us a copy of a subset of their electronic database: 59,608 articles[9] published in *Het Eindhovens Dagblad*, *Het Brabants Dagblad* and *Het Nieuwsblad* in the period January-October 1994. We examined a sample of the VNU corpus and (roughly) classified the articles on the basis of key words assigned to them by the journalists[10]. We concluded that the corpus provided a sufficient variety of articles to be useful for our experiment. Some general statistics for the document collection are given below:

Total number of documents	59,608
Total number of words (tokens)	26,585,168
Total number of terms (types)	434,552
Max number of words per document	5,979
Av. number of words per document	446
Max number of terms per document	2,291
Av. number of terms per document	176

[8] Although TREC3 did have a Spanish track.
[9] This is comparable in size to the individual test corpora used in the TREC evaluation experiments.
[10] These key words were of course not used for document indexing during the experiment.

4.2 Test subjects and test environment

4.2.1 Set-up of the experiment

The test subjects for the experiment were recruited among staff and students of Utrecht University. Care was taken to ensure that subjects were not familiar with the details of the UPLIFT project (e.g. the specific hypotheses being tested in the experiment). After some brief instruction (a short manual describing the task and some details about the document collection) subjects were asked to formulate a query in normal Dutch sentences. We collected 36 queries from 25 different test subjects.

Instead of testing system versions separately, a method was devised to test all versions in one run. A query is processed by all (n) versions, resulting in n ranked lists of documents of length 1000 (cutoff point). Subjects do not see these separate lists, instead they are presented with a list that consists of a merge of the top 100 documents from each list, with duplicates removed. This results in a list ranging from 150 - 600 documents, depending on the query. This list is ordered on document number and presented to the subject for relevance judgement. This merging and ordering method effectively hides the source of the document (i.e. the particular system version that retrieved it). Secondly, this design enables a statistical analysis that separates run effects (the factor we are interested in) from query effects (cf. 4.4). The average number of documents that were judged relevant by the subjects was 29.4.

4.2.2 Description of a session

A subject session runs as follows:

1. The subject is asked to read a manual which provides general background information about the experiment without giving away details about the different system versions. The manual describes in detail what is expected from the subject:
2. The subject must fill in name and run number, start and stop time are logged.
3. The subject must enter a search query which must satisfy the following conditions:
   - The query must be stated in normal Dutch sentences.
   - The query must contain at least 15 words. (This is a heuristic to try to ensure that a sufficient number of content words is present in the query.)
   - The query must aim at a *collection* of relevant documents.

   The manual contains a list of keywords related to topics in the database to give an idea of the scope of the database. Queries are not restricted to these topics.
4. The system will perform a test retrieval run and uses a heuristic to test whether the user's query will yield enough relevant documents. If not, the user is asked to reformulate his query or to make up a new one.
5. If the query has passed all tests, the n retrieval runs are performed and a merged list (cf. 4.2.2) of potentially relevant documents is presented to the user. The tedious relevance judgement task is facilitated by a special application program which provides easy control and prevents errors. The interface (cf. figure 1) consists of a scrollable list of document numbers. A separate window contains the text of the selected document number. The relevance judgement tool starts with the first item selected, and by pressing the **y** and **n** keys (corresponding to relevant and not relevant) the list can be traversed. Correction is possible and judgements appear in the document list.

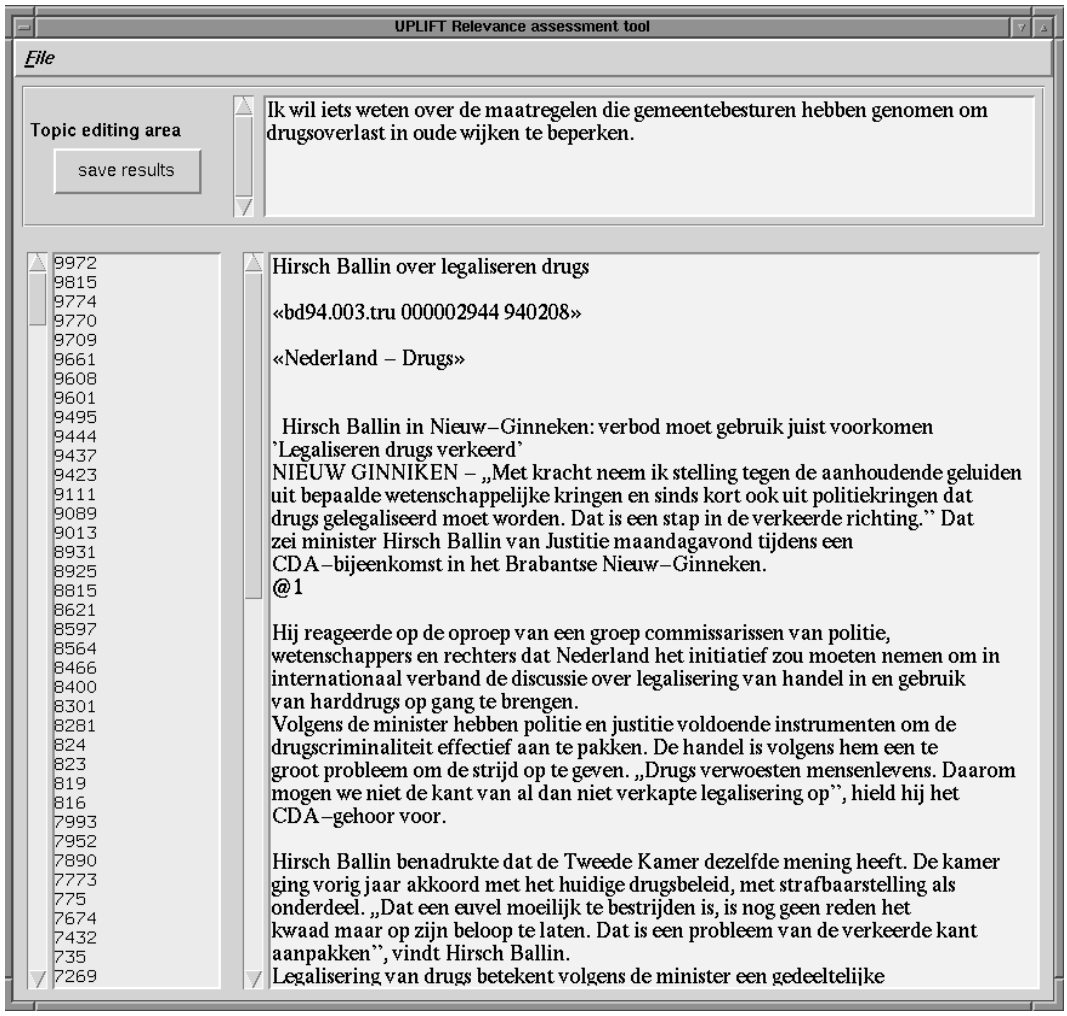

Figure 1: UPLIFT relevance assessment tool

4.3 Measuring procedures

4.3.1 Measuring Recall

The computation of Recall is a traditional problem in IR evaluation. Recall for a certain query is defined as the ratio of the total number of relevant documents retrieved by a certain system as opposed to the total number of relevant documents in the database. This last number is difficult to estimate for large databases, without doing relevance assessments for nearly the complete database [25]. We have decided to use the 'Pooling method' which is employed in TREC. This method computes *relative* Recall values instead of *absolute* Recall. The method is based on the assumption that if one has a 'pool' of diverse IR systems, the probability that a relevant document will be retrieved by one of the systems is high. So a merged list of document rankings (cf. 4.2) is assumed to contain all relevant documents.

4.3.2 Precision/Recall Plots

Precision and Recall are intuitive parameters for boolean retrieval systems. These systems retrieve a fixed number of documents. Relevance ranked based systems like VSM based systems yield a (partial) order of the complete database which is generally cut off at a fixed number. In principle it is possible to compute Precision/Recall data at each point in a document ranking resulting in a Precision/Recall graph. If n is the document (rank) number, r is the number of relevant documents encountered so far and R the total number of relevant documents then Precision and Recall are defined as:

(1) $$Precision = \frac{r}{n}$$

(2) $$Recall = \frac{r}{R}$$

For each query we have computed Precision/Recall data along the lines of TREC/SMART:

The basis for the computation is the relevance judgement file and the separate document rankings as produced by each different system version. The first step in the procedure is the computation of Precision/Recall pairs.

> **Basic algorithm**
> Start at the top of a document ranking file and look up the relevance of each subsequent document in the relevance judgement file. Each rank produces a new Precision/Recall pair.

A problem arises however when a rank contains more than one document. Document rankings often contain 'ties' between documents: the match level (assigned by the retrieval engine) is equal and the engine falls back on its secondary ordering method: document number. We have corrected for this effect in the following way: If such a group contains relevant documents, they are ordered in the middle of the group. Table 1 shows an example listing, each line consists of: rank number, document number, match level and an asterisk if the document is relevant.

Table 2 show the resulting P/R pairs (suppose R is 10), figure 2 shows the corresponding P/R plot.

It seems counterintuitive that Precision does *not* decrease monotonically with increasing Recall. But intuition is misled: Precision finally converges on zero at high Recall levels but it may increase intermediately with increasing document rank number (cf. [22]).

1	2233	35	*
2	3345	30	
3	1456	28	
4	2487	28	
5	3478	28	
6	7890	28	
7	8900	28	*
8	3982	26	
9	0045	24	*
10	3739	21	

Table 1: Relevance ranked listing

Precision	Recall
1.0	0.1
0.5	0.1
0.33	0.1
0.25	0.1
0.4	0.2
0.33	0.2
0.29	0.2
0.2	0.2
0.33	0.3
0.3	0.3

Table 2: Precision Recall points

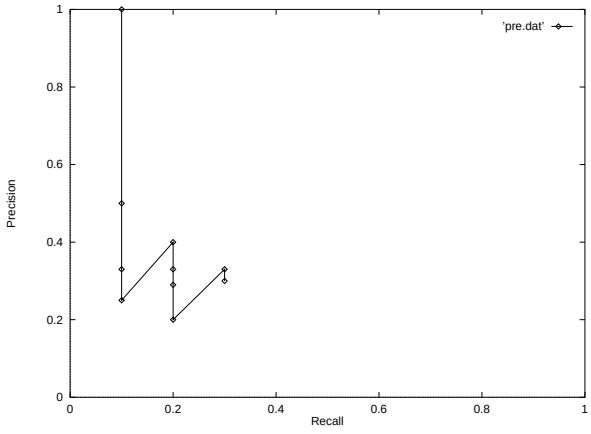

Figure 2: Non-interpolated P/R plot

4.3.3 Precision at fixed Recall levels

If we want to average Precision values over a set of queries (We eventually want to generalize our conclusions to the set of all possible queries of a certain class), we must interpolate Precision values at fixed points of Recall. We have used the same interpolation algorithm as SMART/TREC: at each Recall point the interpolated Precision is defined as the maximum Precision at Recall points greater than the Recall value in question. This yields a graph (cf. figure 3) exclusively composed of horizontal and vertical segments, which can be used to find (interpolated) Precision values at 21 fixed points (0.05, 0.1, ... 1.0). These data can be used to compute averages over queries.

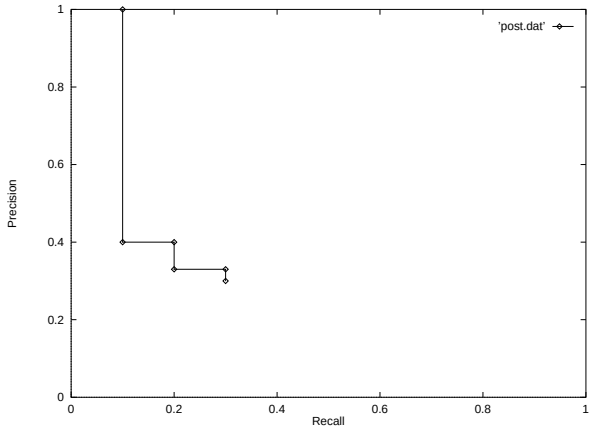

Figure 3: Interpolated P/R plot

4.3.4 Average Precision

However, the interpolation approach has a number of drawbacks, especially when a certain query yields only a small amount of relevant documents. We have therefore also tested a second measure: the average Precision, from the collection of measures assessed in TREC3 [27]. This measure is easy to compute, does not require interpolation and has proven to yield reliable results in TREC3 cross-measure evaluation experiments. Average Precision proved to be a suitable measure to make quick comparisons between a large number of system versions and allowed an easy statistical validation with an analysis of variance.

The average Precision for a certain query and a certain system version can be computed by averaging all Precision values at relevant document positions in the relevance ranking. The example presented in table 1 gives an average Precision of 0.58.

4.3.5 Focusing on Recall

Since the evaluation measures mentioned above focus on Precision values and query expansion is mainly aimed at Recall enhancement, we decided to add two additional measures: Recall at a fixed cut-off point and R-Recall. A Vector Space Model can always achieve a Recall of 100% as it produces a partial order on the total document database. We have chosen to measure Recall after 1000 documents, the same number of documents that is tested for the Recall/Precision pairs. Since Recall measured at document cut-off levels of 200 or more seems only of academic importance and is not interesting for users, we also experimented with document cut-off levels of 25, 50, and 100, but a disadvantage of this method is that Recall at 25 does not seem to make much sense for queries with many relevant documents. The number of relevant documents for the queries in our test collection varied from 3 to 187. This variety motivated us to measure Recall at R documents, where R is the number of relevant documents for a particular query. This measure is more intuitively pleasing since it normalizes over query variance. An ideal system has an R-Recall of 1 and

R-Recall is by definition equal to R-Precision[11]. R-Recall thus provides a singular performance measure in which both Recall and Precision are expressed[12].

4.3.6 Evaluation tools

To facilitate analysis of the data and to check whether the experimental runs really did what was expected, a data inspection environment was created. This environment is based on Tcl scripts (cf. [16]) that generate HTML pages and graphs. HTML files and a WWW browser offer excellent possibilities for quick access to structured data collections which consist of text and graphics.

The results of each query are represented as a separate HTML page, with links to:

- Recall/Precision plot (thumbnail version on page)
- Topic (=query) itself
- Log file
- Individual term scores
- Average Precision, Recall at 1000 docs and R-Recall
- List of documents that were judged relevant, which can be retrieved on a mouse click

The 'Individual term scores' page contains a table of all words that occur in a possible query variant, and the weight of that word in the query for each system variant. It also gives the average number of relevant terms in each variant, an ordered list of relevant terms and an overview of query terms and their variants. Figure 4 shows an HTML page with some results of the final experiments. Table 3 shows an extract of the term scores for one of the test queries[13].

word	n	p2	c1	c1f	c1fow	c2fow	p2pr	sfow	p2ow	c1fcow	c4fow
museum	1	1	1	1	3	3	1	3	3	3	3
musea			1	1	1	1		1		1	1
verbouwing	1	1	1	1	3	3	1	3	3	3	3
verbouwen		1	1				1	1	1		
restauratie								1			
museumuitbreiding											1

Table 3: individual term scores

4.4 Statistical validation

Statistical analysis of IR evaluation data has become increasingly important. Simply calculating means and drawing conclusions on very small differences is not sound from a methodological point of view, especially when there is large variation in the data. Statistical tools are required to test whether differences between means of the observed statistic are significant or should be attributed to chance. Researchers do not agree on the choice of statistical testing methods. Analysis of Variance is the most powerful method but a number of assumptions concerning the data must be checked in advance. Non parametric methods like the Sign test can always be applied but have the disadvantage that they can only decide whether a difference is significant and they do not yield quantitative confidence intervals. Salton [22] does not advocate ANOVA because the R/P data usually do not show a normal distribution. He uses Sign tests which can be applied to the means

[11] This measure was introduced by Chris Buckley (Cornell University) for TREC2.
[12] Although at R, Precision is more dominant than Recall.
[13] Original terms are printed in boldface.

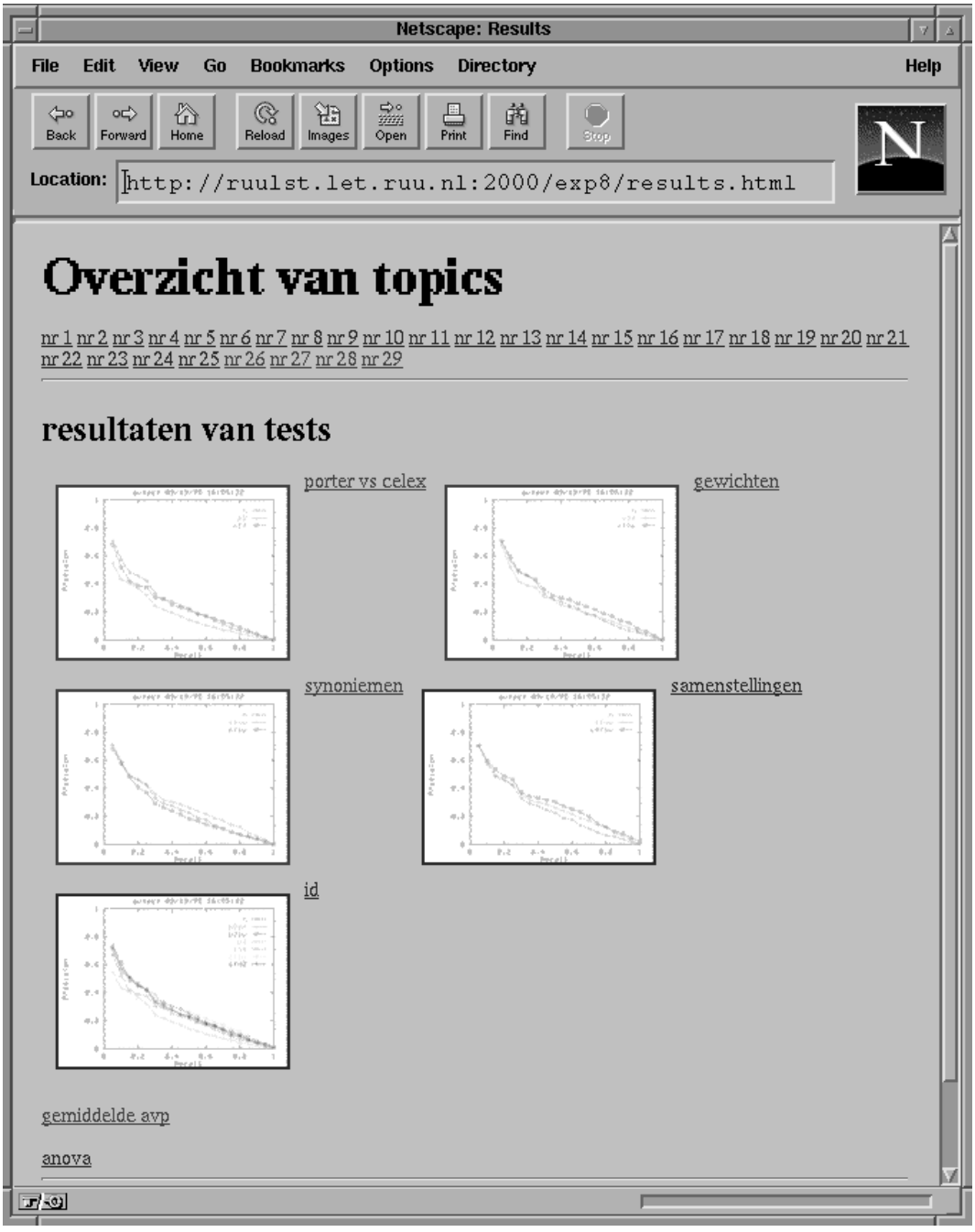

Figure 4: A HTML page with links to experimental results

of two populations without any restriction on the distribution function. Tague-Sutcliffe [25] and Hull [10] state that classical statistical tests like ANOVA can be applied if the population is known to be normally distributed *or* when the data is continuous and the sample size is large. The second category is justified by the Central Limit Theorem of statistics which says that the sample means for a non-normal population will be approximately normal for large populations. A common threshold is 30, we therefore aimed at at least 30 queries for our experiment. Tague-Sutcliffe [27] also shows that average Precision is a reliable performance measure and that it is acceptable to apply ANOVA on TREC3 data. Tague also concludes that arcsine transformations to stabilize the data are not really necessary.

We conclude that it is desirable to run ANOVA tests on data. A query set larger than 30 satisfies the normality condition, but one still has to check whether the distribution of the variances of the means are homogeneous. If not, arcsine transformations can be tried or non-parametric tests like the Sign test or Friedman test can be applied.

We have set up an experimental design and analysis method along the lines of [27] and [26]. The chosen design is a repeated measures single factor design, sometimes also referred to as randomized block design. This design has the advantage that the query effect is separated from the run effect:

$$Y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij} \tag{3}$$

Y_{ij} represents the score (e.g. average Precision) for system variant i and query j , μ is the overall mean score, α is the system version effect, β is the query effect and ϵ represents the random variation about the mean.

The H_0 hypothesis which is tested by the ANOVA is:

> The means of the observed statistic (e.g. average Precision) are equal for all system versions

i.e. the system version effect (α) is zero. If this hypothesis is falsified, we can conclude that at least one pair of means differs significantly. T-tests are subsequently applied to determine which pairs of system versions really show a significant difference.

5 Pilot experiments

The final experiment was preceded by two pilot experiments. One experiment was aimed at getting some insight in user behaviour and the other experiment was used to test hypotheses and select promising system versions for the final experiment.

5.1 User experiment

An UPLIFT retrieval prototype was integrated with an on-line database consisting of articles which appeared in the Colibri newsletter[14]. All actions of archive users were logged to gain some insight in what real users do.

The Colibri WWW interface offers users the choice to do retrieval with dictionary-based query expansion (default mode), traditional OR and AND boolean retrieval, the NOT operator and phrase matching (adjacent AND). Figure 5 shows the Colibri WWW interface. In a period of about 6 months, 1761 queries were processed of which 1247 were expanded (default mode), 236 were OR type, 169 AND and 109 were phrase queries. The majority of the queries consists of one keyword, for example, names of persons or conferences or the name of a certain area of interest: “lexical semantics”, “phonology”. There are of course a number

[14] Colibri is an electronic newsletter and a World Wide Web (WWW) service for people interested in language, logic, speech and information. URL: http://colibri.let.ruu.nl

of requests that try to test the capabilities of the system with a full sentence (mostly first time users). It is hard to judge whether query expansion has really helped the users because relevance judgements are not available. The logs show, however, that users do regularly modify expansions and users do not de-select all added terms, so the query expansion technique is definitely of potential use.

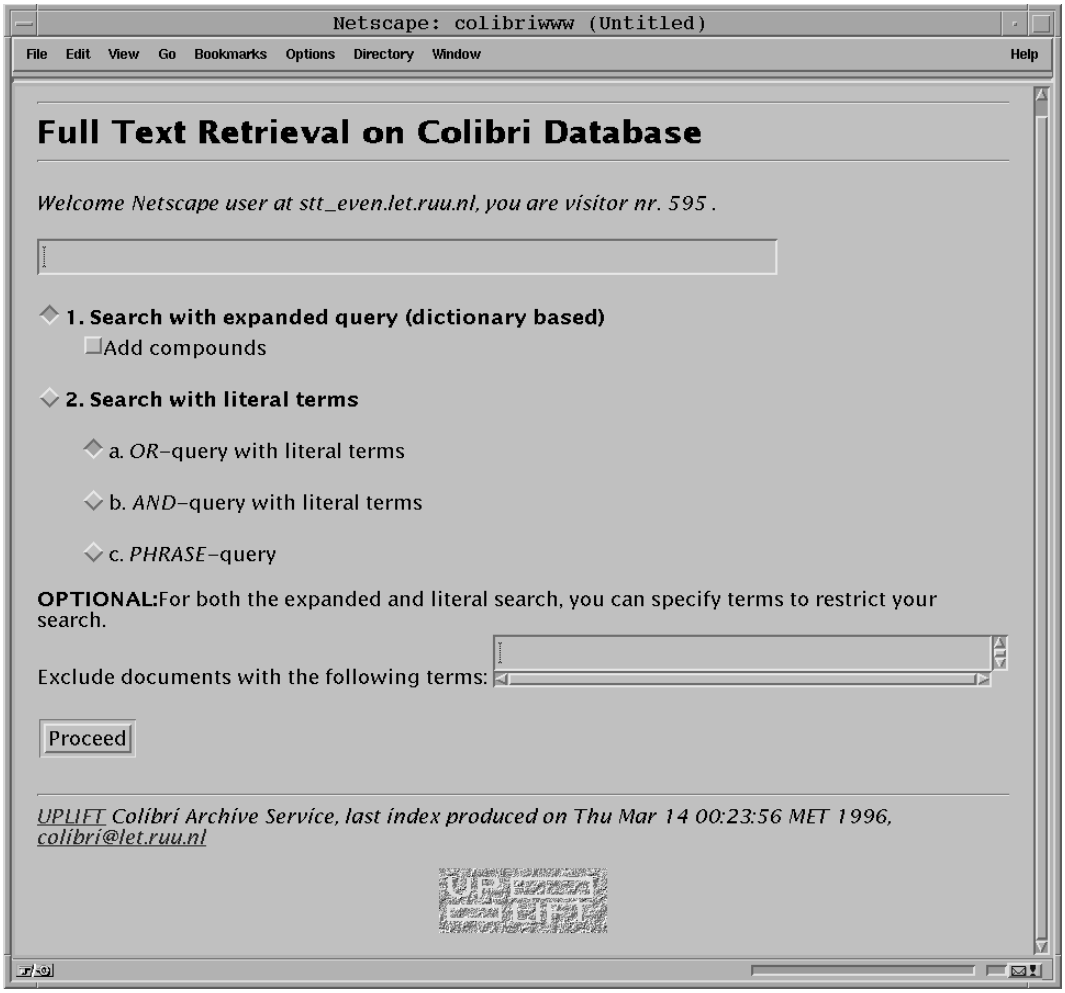

Figure 5: Colibri WWW interface

This experiment revealed that users tend to specify very short queries. VSM, however, performs much better when the queries are longer. On the basis of these results we therefore decided to give the users in the final evaluation experiment some guidance concerning query length and style (cf. 4.2.2).

5.2 System version experiments

In this experiment we used a collection of 8 short test queries (query set 1[15]) and a collection of 12 translated TREC topics (query set 2) and the VNU database. All queries were formulated/translated by the researchers and the results of the retrieval runs were also judged by them.

This pilot experiment was based on a series of sub-experiments:

1. A comparison between CELEX and Porter stemming.

[15] All queries used in the various experiments are listed in Appendix B.

2. A comparison of different versions of CELEX-based stemming using the different expansion databases (cf. section 2.3).
3. The effects of adding synonyms.
4. Influence of changing term weights of the query vector.
5. Experiments with 'ideal' queries and relevance feedback.

5.2.1 Porter vs. CELEX

version	avp	rec(1000)	r-recall
n	**0.340** (0.199)	**0.912** (0.074)	**0.318** (0.150)
p1	0.253 (0.179)	0.838 (0.097)	0.163 (0.111)
c1	0.232 (0.154)	0.863 (0.050)	0.180 (0.098)

Table 4: Porter vs. CELEX, query set 1

version	avp	rec(1000)	r-recall
n	**0.408** (0.153)	**0.885** (0.145)	0.324 (0.154)
p1	0.385 (0.220)	0.818 (0.185)	**0.346** (0.192)
c1	0.370 (0.218)	0.828 (0.211)	0.296 (0.188)

Table 5: Porter vs. CELEX, query set 2

Tables 4 and 5 show the results of a comparison between a reference version of UPLIFT (**n**, no stemming[16]), a version with the Dutch Porter algorithm (**p1**) and a version with a dictionary-based stemmer (**c1**)[17]. The tables show means of the average Precision, Recall at 1000 documents and R-Recall. Variances of these means are given between parentheses. The best results are printed in boldface. For query set 1, the results for c1 and p1 are comparable. For query set 2, p1 seems to be slightly better than c1, but the reference version (n) generally performs better than both p1 and c1. These results seem to indicate that stemming (in general) does not improve retrieval performance. We will investigate this counterintuitive result further in section 5.2.2 and 5.2.4.

In the remaining subsections describing the results of the pilot experiment we will concentrate our discussion on (but not restrict to) query set 2, as it is the largest set and the queries are similar to those used in the TREC evaluation experiments.

5.2.2 Other CELEX versions

We have experimented with small modifications to the CELEX stemming algorithm. The most interesting versions are:

- **c1f** derivational forms are not included in the expansion
- **c1nd** both derivational forms and diminutives are excluded from the expansion[18]
- **c1sp** derivational forms, diminutives and all verbal forms are excluded from the expansion [19]

[16] The abbreviations used for the different UPLIFT versions are summarized in Appendix A.

[17] To guarantee a fair comparison between Porter stemming and dictionary-based stemming, this CELEX version only removes inflectional and derivational affixes and does not use the compound splitter.

[18] A lot of query term expansions introduced diminutives with extremely low document frequencies, making these terms rather important in the total match computation which is based on $tf.idf$ weights [2].

[19] The underlying hypothesis of this version is that verbal forms are less discriminatory index terms. This hypothesis is based on inspection of part of the data.

- **c2** expansion with both inflectional and derivational forms, compound query terms are split and each constituent is added to the query
- **c4** expansion with both inflectional and derivational forms, compound query terms are split and new compounds are composed of query stems (cf. section 2.3)
- **c4f** expansion with inflectional forms only, compound query terms are split and new compounds are composed of query stems.
- **p2** a more restricted version of the Porter algorithm which disregards certain derivational affixes

version	avp	rec(1000)	r-recall
n	0.408 (0.153)	0.885 (0.145)	0.324 (0.154)
c1	0.370 (0.218)	0.828 (0.211)	0.296 (0.188)
c1f	0.469 (0.201)	0.884 (0.150)	0.340 (0.195)
c1nd	0.450 (0.213)	0.884 (0.150)	0.340 (0.195)
c1sp	0.455 (0.246)	0.836 (0.196)	0.349 (0.205)
c2	0.431 (0.193)	0.936 (0.054)	0.349 (0.180)
c4	0.440 (0.212)	0.919 (0.111)	0.359 (0.180)
c4f	**0.481** (0.236)	**0.933** (0.064)	**0.380** (0.223)
p1	0.385 (0.220)	0.818 (0.185)	0.346 (0.192)
p2	0.387 (0.228)	0.844 (0.180)	0.325 (0.206)

Table 6: CELEX variants, query set 2

Restricting stemming to only inflectional affixes (c1f) seems to improve performance. The experiments with even further restrictions (c1nd and c1sp), however, did not improve on c1f. It is remarkable that c1sp, the version without verbal forms, gives such a high score. This confirms the importance of nouns for IR performance (cf. 6). Compound splitting (c2) and compound generation (c4, c4f) seem worthwhile.

5.2.3 Synonyms

In this sub-experiment we examined the addition of synonyms. Because initial trials showed that adding synonyms and all their morphological variants without restrictions had a disastrous effect on Precision, we implemented a restricted form of synonym expansion as described in section 3.2.2. This restricted synonym expansion method is denoted by version **sf**. In order to gain insight into the effect of sense ambiguity we implemented **sf1**. This version only adds synonyms for one (the first) sense in case of ambiguity. **sc4** is a version which is essentially like c4: new compounds are added which consist of original query terms or one of their synonyms. Secondly, this version splits up compounds in the original query and tries to substitute synonyms for one of the constituents. Example: 'belastingontduiking' (*tax evasion*) yields 'belastingfraude' (*tax fraud*) via the synonym relation 'ontduiking' $\rightarrow$ 'fraude'.

version	avp	rec(1000)	r-recall
n	0.408 (0.153)	0.885 (0.145)	0.324 (0.154)
c1	0.370 (0.218)	0.828 (0.211)	0.296 (0.188)
sf	0.289 (0.211)	0.670 (0.295)	0.209 (0.183)
sf1	0.334 (0.192)	0.719 (0.276)	0.250 (0.193)
c4	0.440 (0.212)	**0.919** (0.111)	**0.359** (0.180)
sc4	**0.442** (0.244)	0.856 (0.167)	0.355 (0.198)

Table 7: Synonyms, query set 2

Table 7 shows that the introduction of synonyms generally deteriorates system performance. A qualitative look at the data shows that this effect is mainly due to the absence of sense disambiguation. Even just selecting the first interpretation (sf1) already improves performance. We expect that interactive sense disambiguation would improve results considerably.

5.2.4 Weighting and grouping

During the pilot experiment we developed some extra variants to test the influence of (re)weighting schemes for query terms. One of the reasons for these extra experiments was the fact that the plain reference system (n) performed better than most of the query expansion versions. We therefore wanted to test the hypothesis that added terms should have a lower weight than original query terms.

The variants that were tested are:

- **c1ow**: CELEX inflection and derivation, original terms have triple weight[20].
- **c1fow** CELEX inflection only, original query terms have triple weight.
- **c1iw** CELEX inflection and derivation, all inflected terms have triple weight.
- **c1xw** CELEX inflection and derivation, original compounds have triple weight.

version	avp	rec(1000)	r-recall
n	0.408 (0.153)	0.885 (0.145)	0.324 (0.154)
c1	0.370 (0.218)	0.828 (0.211)	0.296 (0.188)
c1ow	0.450 (0.173)	0.890 (0.142)	**0.395** (0.158)
c1f	**0.469** (0.201)	0.884 (0.150)	0.340 (0.195)
c1fow	0.454 (0.177)	**0.894** (0.148)	0.360 (0.165)
c1iw	0.430 (0.196)	0.884 (0.134)	0.340 (0.188)
c1xw	0.422 (0.170)	0.862 (0.186)	0.354 (0.188)

Table 8: CELEX weights, query set 2

Table 8 shows that favouring original terms can have a positive effect (c1 vs. c1ow), but the effects of restricting stemming to inflectional forms (c1f) and favouring original terms cannot be accumulated (c1fow). Favouring other terms such as inflected variants (c1iw) or compounds (c1xw) does not seem to improve results.

We also tried to re-weight terms in a more principled way, based on Harman's 'grouping' experiments (cf. section 2.3). The idea behind the grouping technique is the following: all morphological variants of a term should count as one and the same term. Harman also experimented with down-weighting term variants, i.e. less weight is assigned to those terms added by the stemmer. She found out that down-weighting with a factor 2 gave a significant improvement over no grouping and that results equal the results for the grouping version.

We will reproduce Harman's results here:

treatment:	grouping	no grouping, no dw	no grouping, dw 1/2	no grouping dw 1/4
average Precision	0.388	0.309	0.387	0.377

Table 9: Down-weighting term variants (Harman)

An interesting hypothesis is whether down-weighting within a group improves performance. This is a combination which Harman did not investigate[21].

[20] The more we increase the weight of the original terms, the more that performance results will resemble the results of n, the reference version. We have done an experiment with varying the weight of the original terms between 1 and 5, 3 turned out to be the best choice.

[21] There is a practical reason why a system equipped with the grouping *plus* down-weighting procedure is not attractive. The hypothetical system would require the instant computation of the weight vector for each document, because the weight of a concept is dependent on which variant is present in the query. This is not very efficient.

The matching procedure[22] as implemented in our text retrieval engine TRU [4] can be defined for a document d with respect to a query q by the following formulae for the weight of a term w and the match factor.

$$wght(w_i,t) = \frac{frq(w_i,t)\cdot \log \frac{N}{n_{w_i}}}{\sqrt{\sum_{w_i \epsilon t}(frq(w_i,t)\cdot \log \frac{N}{n_{w_i}})^2}} \tag{4}$$

$$match\_factor(d,q) = \sum_{w \epsilon q} wght(w,d)\cdot wght(w,q) \tag{5}$$

In these equations, $frq(w,t)$ represents the frequency of word w in a text t, N represents the total number of documents in the document base and n_w the number of documents containing word w.

A Harman-like approach for the TRU match factor would be :

$$wght(c_i,t) = \frac{frq(c_i,t)\cdot \log \frac{N}{n_{c_i}}}{\sqrt{\sum_{c_i \epsilon t}(frq(c_i,t)\cdot \log \frac{N}{n_{c_i}})^2}} \tag{6}$$

Here c_i represents a concept i.e. a term with its variants. When computing term frequencies and document frequencies, the original term and variant terms count as one and the same term.

We have tried to influence the matching procedure of the TRU engine to approximate the grouping effect. Since we did not have access to TRU engine source code we tried to manipulate the matching procedure through the query vector. Unfortunately, UPLIFT's underlying retrieval engine stores the weight vectors for each document in a compact form, so that the individual factors like term frequency are not available anymore. However, it is possible to reconstruct the document frequency of each word after indexing. We have tried to exploit this information to do a form of 'pseudo-grouping'. If we inspect formula 4 we see that the weight is composed of three factors:

1. term frequency
2. log of the inverse document frequency
3. normalization (dependent on factors 1 and 2)

Because for TRU, only the document frequency is available we could try to multiply the weights of each word form in a query with a factor that compensates for the incorrect[23] computation of the idf factor.

The idf factor in TRU is:

$$idf_w = \frac{N}{n_{w_i}} \tag{7}$$

whereas we aim at

$$idf_c = \frac{N}{n_{c_i}} \tag{8}$$

We have done experiments by multiplying the weight of a word in the query with (a) $(\frac{\log idf_c}{\log idf_w})^2$, (b) $\frac{\log idf_c}{\log idf_w}$, (c) $\frac{idf_c}{idf_w}$ or (d) $\frac{\sqrt{idf_c}}{\sqrt{idf_w}}$. The document frequency in idf_c is computed by counting the documents that contain

[22] Our matching procedure is different from Harman's.
[23] Based on terms and not on concepts.

at least one of the morphological variants. We subsequently combined this pseudo-grouping procedure with down-weighting[24]. The experiments were based on c1f (CELEX, inflection only). We will not present the results of all investigated versions here. All versions which combined grouping and down-weighting performed better than the grouping-only variants. The best results were obtained with 'grouping factor' (c).

version	avp	rec(1000)	r-recall
n	0.408 (0.153)	0.885 (0.145)	0.324 (0.154)
c1f	0.469 (0.201)	0.884 (0.150)	0.340 (0.195)
c1faow	0.488 (0.196)	**0.896** (0.148)	0.381 (0.186)
c1fbow	0.478 (0.178)	0.889 (0.152)	0.366 (0.172)
c1fcow	**0.507** (0.196)	0.890 (0.152)	**0.412** (0.211)
c1fdow	0.487 (0.181)	0.892 (0.153)	0.383 (0.191)

Table 10: Grouping & down-weighting, query set 2

We also wanted to test our grouping and weighting approaches against the usual stemming approach, i.e. no query expansion but indexing stems instead of word forms. As a basis for these experiments we used a version of the Porter algorithm (p2).

version	avp	rec(1000)	r-recall
n	0.408 (0.153)	0.885 (0.145)	0.324 (0.154)
p2	0.387 (0.228)	0.844 (0.180)	0.325 (0.206)
p2pr	0.462 (0.220)	**0.899** (0.146)	0.389 (0.222)
p2ow	0.470 (0.166)	0.897 (0.144)	0.369 (0.149)
p2cow	**0.497** (0.178)	0.885 (0.159)	**0.406** (0.167)

Table 11: Grouping v.s. stemming, query set 2

Table 11 shows that **p2pr** (traditional stemming), **p2ow** (favouring original terms) and **p2cow** (grouping factor c and down-weighting) all are an improvement with respect to p2. It is important to note that p2pr does not outperform the query expansion versions, this means that the query expansion method does not perform worse than the traditional stemming approach.

5.2.5 Ideal queries & relevance feedback

This section describes experiments with artificial system variants. These versions exploit the relevance judgements by the user in order to simulate the results for an 'ideal' query. This ideal query should give an idea of the maximum performance that can be achieved for a certain query. It is composed in the following way: for each topic a set of unique query terms is collected from all system version expansions. A program tests for each term whether it yields relevant documents. If so, the term is added to the 'ideal query term set'.

A first approximation was **idow** which also favours terms which were part of the user's original query formulation with a weight of 3. A more detailed analysis was done to investigate the effect of query length (i.e. cardinality) on performance. For this purpose, the list of 'ideal query terms' was ordered by the number of relevant documents that were retrieved by each term. **id1** represents a system version with only the best term, **id2** has the best and second best term and so on (**id3, id5, id10, ida**=complete list). A second sub-experiment was performed with **id3f** which adds inflectional variants and **id3n** which picks the three 'best' terms from the original user query.

It is well known from the literature [23], [1] that relevance feedback can improve retrieval performance drastically. The term 'relevance feedback' traditionally refers to a technique where the user marks one or

[24] We actually implemented a form of up-weighting: the weight of original query terms is increased as opposed to decreasing the weight of the added variants.

more documents in the result list as relevant and where this information is used to refine the query, usually by combining the weight vectors of the relevant documents and the query vector. However, it is difficult to do an experiment with relevance feedback without having to account for differences between users. Therefore we added a very simplistic relevance feedback version: **nrf** which simply is a plain run followed by a second run with the top ranked document as query.

version	avp	rec(1000)	r-recall
n	0.408 (0.153)	0.885 (0.145)	0.324 (0.154)
id1	0.561 (0.241)	0.859 (0.100)	0.367 (0.225)
id2	0.505 (0.239)	0.953 (0.067)	0.429 (0.211)
id3	0.504 (0.209)	0.974 (0.042)	0.447 (0.175)
id3f	0.467 (0.235)	0.974 (0.042)	0.381 (0.196)
id3n	0.575 (0.171)	0.909 (0.163)	0.461 (0.122)
id10	0.617 (0.143)	0.975 (0.036)	0.539 (0.121)
ida	0.634 (0.130)	**0.976** (0.032)	0.555 (0.110)
idow	**0.647** (0.185)	0.974 (0.042)	**0.583** (0.157)
nrf	0.551 (0.207)	0.903 (0.113)	0.466 (0.195)

Table 12: Ideal queries and relevance feedback, query set 2

Table 12 show that all artificial 'id' versions and the nrf version perform better than n. Although Precision decreases a bit at first when going from id1 to ida, Recall increases. It seems worthwhile again to stress terms from the original query: idow vs. ida and id3n vs. id3. It is remarkable that a query with only 1 (the best) term has such a high Recall value: 0.859 for query set 2 (0.763 for query set 1).

We also examined the syntactic category of successful query terms. Not surprisingly, nouns[25] form the majority (58%), adjectives and verbs account for 13% and 29% respectively, other categories are negligible. If we restrict ourselves to the best query term, the percentage of nouns is even higher (84%), verbs account for 8% and adjectives also for 8%.

5.3 Summary of results

The purpose of the pilot experiments was to select interesting versions for the main experiment and to solve potential problems. Tentative conclusions of the pilot experiments are:

- Porter vs. CELEX
  - Dictionary-based stemming and Porter stemming do not perform very differently. Both perform worse than the reference version.
- Other CELEX versions
  - Cautious (i.e. weak) stemming makes sense, not stemming derivational affixes seems to improve results.
  - Verbal forms are less important for retrieval.
  - Compound splitting and compound generation seem worthwhile.
- Synonyms
  - (possibly interactive) Sense disambiguation is necessary to make synonyms useful.
- Weighting and grouping

[25] Including nominal compounds and proper nouns.

- The performance degradation which is a result of the query expansion technique can be overcome by favouring original terms or by pseudo-grouping.

- Ideal queries & relevance feedback
  - Using information about relevant documents (either considered relevant by the user or by the system) improves results.
  - One word can make a difference: it is probably more important to exclude irrelevant variants from the query than to add extra variants. Section 5.2.5 shows that queries with just a few terms (id1, id3) yield very good results. Again, interactive query expansion seems an interesting option to investigate.

6 The final experiment

The purpose of the final experiment was to see whether experimental results obtained during development could be confirmed in an experiment with non-expert users (i.e. not familiar with the contents of the database nor with the differences between UPLIFT versions).

We will discuss the results of the following UPLIFT versions[26]:

1. n : reference
2. c1: CELEX stemming, inflection and derivation
3. p2: revised Porter (less derivation)
4. p2ow: Porter stemming, increased weight (3) original terms
5. c1f: CELEX stemming, inflection only
6. c1fow: CELEX inflection only, increased weight original terms
7. c2fow: CELEX inflection only plus compound splitting, increased weight original terms
8. c4fow: CELEX inflection only plus compound splitting and compound generation, increased weight original terms
9. sfow: CELEX inflection only plus synonyms (all senses), increased weight original terms
10. c1fcow: CELEX inflection only plus 'grouping' and 'down-weighting'
11. p2pr: Porter stemming before indexing (no query expansion)

They were selected for the following reasons: c1 and p2 for the comparison between Porter and CELEX stemming, c1f for the effect of selective stemming (no derivation), cf2ow and c4fow for the effect of compound splitting and compound generation respectively, sfow to represent synonym addition, the pairs c1f - c1fow and p2 - p2ow were added to investigate the effect of increasing the weight of original terms, c1fcow as a version of 'grouping and down-weighting' and p2pr to represent stemming before indexing. The experiments with ideal queries and 'relevance feedback' were not repeated because this would have resulted in considerably longer test run times (caused by an extra retrieval run and the corresponding relevance judgements). Furthermore, the results of the pilot experiment were considered sufficiently convincing and similar results have been reported by other researchers for this type of experiment.

[26] A total of 17 versions were tested in the final experiment. For the sake of clarity we will concentrate on a representative subset.

6.1 Results of the final experiment

In table 13 the test statistics *average Precision maximum Recall at 1000 docs*, R*-Recall*, $2R$*-Recall* and $5R$*-Recall* for each system version are given. Average Precsion and Recall at 1000 docs illustrate Precision and relative Recall performance respectively. NR-Recall should give an impression of system performance when both Precision and Recall are considered important. When R is low, Precision is dominant but when R increases Recall becomes more important.

Figure 6 shows a traditional Recall/Precision graph for a subset of the versions.

version	avp	rec(1000)	r-recall	2r-recall	5r-recall
n	**0.368** (0.225)	0.786 (0.203)	0.287 (0.201)	0.391 (0.221)	0.528 (0.243)
c1	0.280 (0.199)	0.868 (0.153)	0.213 (0.183)	0.346 (0.251)	0.521 (0.258)
p2	0.294 (0.217)	0.836 (0.198)	0.227 (0.185)	0.333 (0.233)	0.499 (0.264)
p2ow	0.360 (0.209)	0.849 (0.191)	0.292 (0.198)	0.420 (0.230)	0.590 (0.226)
c1f	0.333 (0.210)	0.860 (0.175)	0.271 (0.193)	0.391 (0.237)	0.587 (0.238)
c1fow	0.356 (0.206)	0.847 (0.173)	0.296 (0.198)	0.412 (0.219)	0.578 (0.247)
c2fow	0.351 (0.198)	0.881 (0.132)	0.296 (0.201)	0.429 (0.212)	0.605 (0.237)
c4fow	0.365 (0.198)	**0.899** (0.132)	**0.323** (0.200)	**0.447** (0.219)	**0.617** (0.250)
sfow	0.360 (0.216)	0.891 (0.152)	0.285 (0.191)	0.388 (0.218)	0.557 (0.248)
c1fcow	0.362 (0.234)	0.834 (0.184)	0.316 (0.184)	0.417 (0.249)	0.553 (0.273)
p2pr	0.356 (0.233)	0.849 (0.200)	0.287 (0.198)	0.415 (00.244)	0.591 (0.247)

Table 13: query set 3

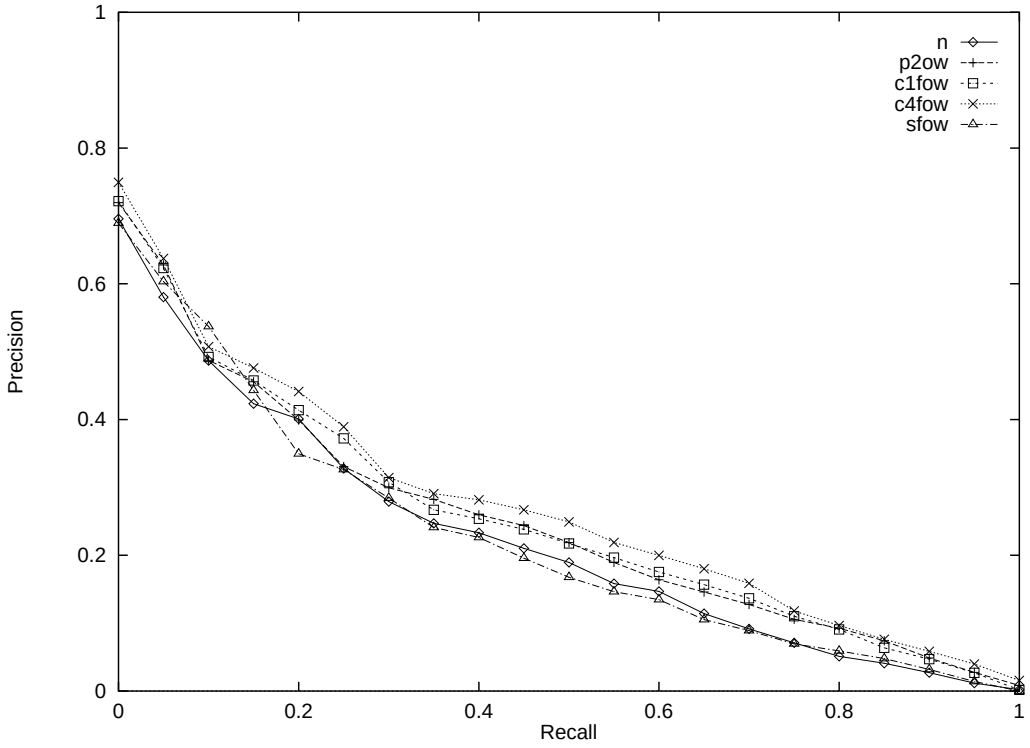

Figure 6: Recall Precision Graph

Tables 14, 15, 16, 17 and 18 show the results of the ANOVA.

The most important figures in the ANOVA tables are the F-values in the rightmost column. The system version effect (Runs row) is significantly different from 0 in all the three cases because the F-values exceed

Source	DF	Sum of Sq	Mean Sq	F val
Runs	10	0.3192	0.0319	4.5351
Query	35	15.6042	0.4458	63.3515
Error	350	2.4631	0.0070	
Total	395	18.3864		
s.e.d. (Runs)	0.020			

Table 14: RESULTS: ANOVA TABLE Average Precision

Source	DF	Sum of Sq	Mean Sq	F val
Runs	10	0.3533	0.0353	5.4768
Query	35	9.7132	0.2775	43.0220
Error	350	2.2577	0.0065	
Total	395	12.3242		
s.e.d. (Runs)	0.019			

Table 15: RESULTS: ANOVA TABLE Recall at 1000 docs

Source	DF	Sum of Sq	Mean Sq	F val
Runs	10	0.4054	0.0405	6.2647
Query	35	12.6229	0.3607	55.7270
Error	350	2.2651	0.0065	
Total	395	15.2935		
s.e.d. (Runs)	0.019			

Table 16: RESULTS: ANOVA TABLE R-Recall

Source	DF	Sum of Sq	Mean Sq	F val
Runs	10	0.4261	0.0426	3.9890
Query	35	17.3359	0.4953	46.3669
Error	350	3.7389	0.0107	
Total	395	21.5009		
s.e.d. (Runs)	0.024			

Table 17: RESULTS: ANOVA TABLE 2R-Recall

Source	DF	Sum of Sq	Mean Sq	F val
Runs	10	0.5087	0.0509	3.1703
Query	35	18.8406	0.5383	33.5488
Error	350	5.6159	0.0160	
Total	395	24.9652		
s.e.d. (Runs)	0.030			

Table 18: RESULTS: ANOVA TABLE 5R-Recall

$F_{.99;10,350}$[27] = 2.32 (i.e. the value required for significance at the 0.99 level), so it is clear that the H_0 hypothesis that means of system versions are equal, can be rejected. The query effect (Query column) is also clearly significant: the F-values exceed $F_{.99;35,350}$ = 1.65. This justifies the choice for a randomized block design (cf. section 4.4). Inspection of a fitted value plot showed that the assumption of homogeneity of variances is confirmed, therefore arcsine root transformations to stabilize data are not required.

Because the ANOVA shows that there are significant differences between system versions, it is necessary to do multiple pairwise comparisons for which we have applied T-tests. The standard errors of differences of means (of runs) are:

Measure	**s.e.d**
Average Precision	0.020
Recall at 1000	0.019
R-Recall	0.019
$2R$-Recall	0.024
$5R$-Recall	0.030

Table 19: Standard errors of differences of means

The SED values are used to discriminate significant different versions in the following way:

(9) $$\mid \bar{x}_1 - \bar{x}_2 \mid > 2 \times s.e.d.$$

Figures 7, 8, 9, 10 and 11 present the results of the multiple comparisons (95% reliability).

The diagrams must be interpreted as follows: if two means are underlined by the same line segment, their difference is not significant.

We can conclude that all versions offer a significant improvement in Recall over the plain (**n**) reference version (cf. figure 8). When Recall increases ($R \rightarrow 5R$) the differences become more pronounced. This confirms our hypothesis that stemming improves Recall. However, all versions seem to degrade Precision a bit, although differences are not significant except for p2 and c1. Most conclusions of the pilot experiment are confirmed.

- Porter vs. CELEX
  - Full CELEX stemming and Porter stemming do not behave very differently.
- Other CELEX versions
  - Our results indicate that restricting stemming to inflectional affixes improves Precision. It would be interesting to investigate this issue more thoroughly. Other researchers e.g. [5], [20] have found that even among inflectional affixes there are important differences.
  - The compound versions (cf2ow, c4fow) score very well. The usefulness (for Dutch) of decomposing compounds in the query and generating new compounds is confirmed.
- Synonyms
  - Surprisingly, sfow, i.e. adding synonyms but with increased weight for original terms, performs rather well. It should be possible to improve on this result with sense disambiguation.
- Weighting and grouping
  - In general, -ow variants perform better than their non-ow counterparts. The more complicated pseudo-grouping and down-weighting scheme (c1fcow) does not significantly improve on this effect.
  - Query expansion is comparable with stemming before indexing.

[27] The subscripts refer to the significance level (1-0.01) and the degrees of freedom.

c1	p2	c1f	c2fow	p2pr	c1fow	sfow	p2ow	c1fcow	c4fow	n
0.280	0.294	0.333	0.351	0.356	0.356	0.360	0.360	0.362	0.365	0.368

Figure 7: Equivalent versions based on multiple comparison of means of AVP

n	c1fcow	p2	c1fow	p2ow	p2pr	c1f	c1	c2fow	sfow	c4fow
0.786	0.834	0.836	0.847	0.849	0.849	0.860	0.868	0.881	0.891	0.899

Figure 8: Equivalent versions based on multiple comparisons of means of Recall(1000)

c1	p2	c1f	sfow	n	p2pr	p2ow	c1fow	c2fow	c1fcow	c4fow
0.213	0.227	0.271	0.285	0.287	0.287	0.292	0.296	0.296	0.316	0.323

Figure 9: Equivalent versions based on multiple comparisons of means of R-Recall

p2	c1	sfow	n	c1f	c1fow	p2pr	c1fcow	p2ow	c2fow	c4fow
0.333	0.346	0.388	0.391	0.391	0.412	0.415	0.417	0.420	0.429	0.447

Figure 10: Equivalent versions based on multiple comparisons of means of $2R$-Recall

p2	c1	n	c1fcow	sfow	c1fow	c1f	p2ow	p2pr	c2fow	c4fow
0.499	0.521	0.528	0.553	0.557	0.578	0.587	0.590	0.591	0.605	0.617

Figure 11: Equivalent versions based on multiple comparisons of means of $5R$-Recall

6.2 Performance differences between query sets 1, 2 and 3

After running some tests with query sets 1 and 2 in the pilot phase, it became clear that there were notable differences between the two sets. For query set 1 most UPLIFT versions performed worse than the reference version. Query set 2, however, showed much better results. Initially, we were not surprised by these differences because query sets 1 and 2 differ considerably when we compare a number of quantitative properties:

	number of queries	average number of content words	average number of compounds
set 1	8	12	2
set 2	11	35	6

Table 20: Query collection statistics, query sets 1 & 2

The final experiment, however, revealed that these properties could not be the source of the observed differences. If we compare query set 3 with the other two sets on these aspects, we would expect to find similarities between query sets 1 and 3.

	number of queries	average number of content words	average number of compounds
set 3	36	7	1

Table 21: Query collection statistics, query set 3

The results for query set 3, however, were comparable to the results for query set 2. This observation caused us to investigate other, more qualitative, properties of the query sets.

Table 22 shows the distribution of successful query terms (i.e. terms present in relevant documents) over system versions. This distribution is fairly similar for all sets of queries, 17-20% of the successful query terms were present in the original query (n), the rest of the good terms are found by expansion versions. It should be noted though that it is not the case that each good term introduces new relevant documents. In most cases only a few terms are required to retrieve all relevant documents (cf. section 5.2.5), the extra terms provide the necessary context to pull relevant documents to the top region of the document ranking so that Precision improves.

set nr	n	c1f	c1	c2	c4	sf	porter
query set 1	17%	11%	12%	12%	6%	36%	6%
query set 2	17%	13%	9%	7%	5%	38%	11%
query set 3	20%	13%	9%	8%	5%	40%	6%

Table 22: Distribution of successful query terms over versions

A more detailed investigation of the successful query terms, however, revealed that the 'best' term for a particular query (i.e. the term that retrieves the highest number of relevant documents) was already present in the original query in 100% of the cases for query set 1 (formulated by the researchers), versus 91% for query set 2 (translated TREC topics) and 70% for query set 3 (formulated by non-expert test subjects)[28]. There may of course still be other factors which we have not considered but we tentatively conclude that the 'quality' of the original query may be of crucial importance for the success of the linguistic techniques used in UPLIFT. If the original query already is almost optimal, query expansion with alternative terms may not improve performance. If the original query is not optimal, because the user is not familiar with the terminology used in the database, for instance, expansion techniques may be useful.

[28] For query set 3, other expansion versions that delivered the best term were: inflection (11%) compound splitting (8%), synonyms (5%), derivation (3%) and porter (3%) (query set 1: no other versions, query set 2: synonyms 9%).

6.3 Evaluation of experimental setup

During the final experiment we used a simple heuristic to test whether a topic would yield enough relevant documents. The following procedure was used:

- the user formulates a query
- a quick retrieval run with 2 different UPLIFT versions results in two ranked output files
- if the match factor for the 50th document in the ranking is below a certain level for both versions, the query is rejected.

Since several queries still yielded only a few relevant documents we had to conclude that this simple heuristic was inadequate. A better approach might be to present the first 25 documents of the plain (n) retrieval output, randomize them and let the user rate them, check whether there are enough relevant documents. If so, let the user assess the complete merged set with the already assessed documents left out.

7 Stemmer evaluation revisited

In [13] we proposed some new measures to compare different versions of stemmers. The measures are:

- MUR: Mean Understemming Ratio
- MOR: Mean Overstemming Ratio
- MMF: Mean Match Factor, singular performance measure
- SW: Stemming Weight, expresses whether a stemmer is 'weak' or 'strong'

The measures are part of a stemmer evaluation method which is not based on relevance judgements but on testing the stemmer on pre-defined groups of morphologically related words. An ideal stemmer would conflate these groups to a single stem. 'Overstemming errors' are those errors which result in the conflation of semantically unrelated words. 'Understemming errors' are those errors where a failure to conflate semantically related words is concerned.

Figure 12 and 13 show that the new version of the Dutch Stemmer (p2) performs worse along these criteria: The mean match factor is lower reflecting the fact that the stemmer is weaker. But the second version of the Porter stemmer explicitly did not stem some infrequent derivational affixes to prevent conflation of terms that are too far removed in meaning. The evaluation method which is discussed in the current report has shown that not stemming derivation in general is beneficial to the performance. We therefore conclude that the validity of the evaluation methods that are described in [13] is strongly dependent on the quality of the grouping process and the definition of what constitutes a group. If a group contains derivational forms than a stemmer which is tuned to remove derivational affixes will have a good performance. In our corpus based experiments, however, we have seen that derivational variants are not always good query terms. Thus a stemmer which removes all derivational affixes will yield good results in an evaluation along the lines of [13] but will perform not so good in the evaluation method which is described in this report. This does not invalidate the former evaluation method however, the definition of a group should be reconsidered. A second difference is that the first evaluation method disregards word-frequency. In a Recall/Precision-based evaluation experiment it is probably not so important that there are some under- or overstemming errors on very infrequent terms.

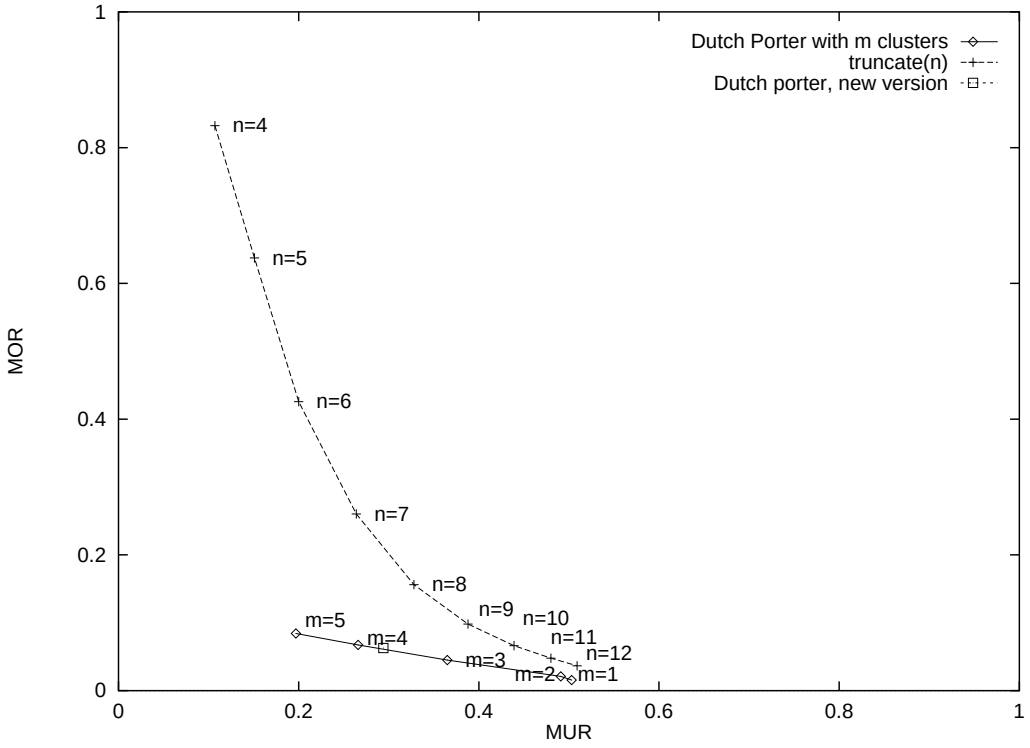

Figure 12: MUR vs. MOR

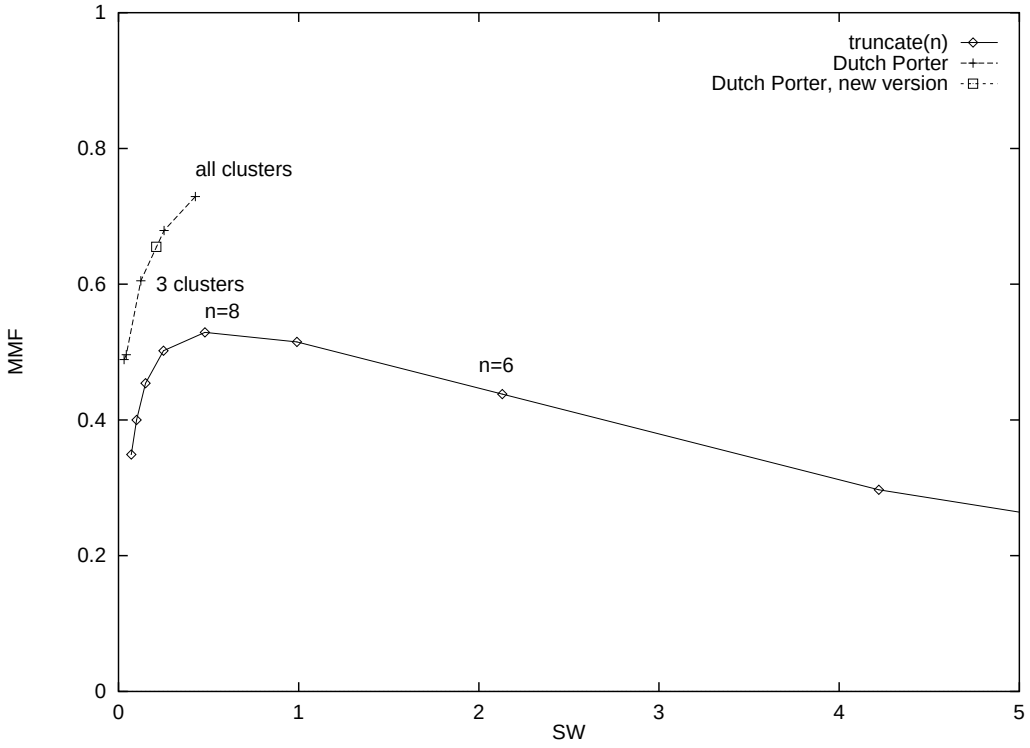

Figure 13: MMF vs. Stemming Weight

8 Overall conclusions

We have tested several linguistic techniques to enhance Recall. The basic method by which different techniques were compared was query expansion. It is obvious that high Recall levels can be reached with massive query expansion, but automatic query expansion tends to deteriorate Precision as well. So the challenge is to find stemming methods which improve Recall without a significant loss in Precision. We found that all but the most simple stemming methods (c1 and p2) satisfy these criteria. Inflectional stemming proved to be most successful "simple" linguistic stemming method. Removing derivational morphology is sometimes useful but in general it reduces Precision too much. The query expansion methods that do more than just conflate the morphological variants and expand the query with new (but semantically related) concepts, namely the methods based on compound analysis and synonym expansion, gave the best Recall results. This shows that automatic query expansion with new concepts can be fruitful when the rules which govern these particular expansion methods and which are aimed at ensuring tight semantic relationships are very strict.

The experiments with ideal queries show that interactive, relevance feedback methods based on selective query expansion have potential for a major improvement in retrieval performance with respect to the methods tested in our experiments.

We also found that query expansion is a competitive method in comparison with the usual stemming before indexing approach. Since query expansion has a number of important advantages for system developers and for use in applications (cf. section 2.3), we consider this an important result. Finally, our results seem to indicate that the linguistic techniques developed for the UPLIFT project are best applied in a context with non-expert users.

Acknowledgements

We would like to thank Wil Roestenburg, Moni Nissen and Hans Kemperman from VNU for the test corpus, Ewout Brandsma and Gerrit Scholl (Philips Research labs) for their TRU engine support, Theo Vosse (Leiden University) for the compound splitter, Jean Tague-Sutcliffe (University of Western Ontario), Peter Defize and Pieter Marres (TNO-TPD) for their comments on statistical test procedures, Denis Girou for the pstchart package and OTS staff and students for their participation in the experiment.

Appendix A

Table 23 explains the system version names.

n	reference system, no query expansion
p1	Dutch Porter version 1
p2	Dutch Porter version 2 (less derivation)
p2ow	Dutch Porter version 2, original terms re-weighted (3×)
p2cow	Dutch Porter version 2, 'grouping factor' c, original terms re-weighted (3×)
p2pr	stemming before indexing
c1	CELEX stemming, inflection and derivation
c1ow	CELEX stemming, original terms re-weighted (3×)
c1f	CELEX stemming, inflection only
c1fow	CELEX inflection only, original terms re-weighted (3×)
c1nd	CELEX inflection only, excluding diminutives
c1sp	CELEX inflection only, excluding diminutives and verbal forms
c1faow	CELEX inflection only with 'grouping factor' a and down-weighting
c1fbow	grouping factor b
c1fcow	grouping factor c
c1fdow	grouping factor d
c1iw	CELEX inflection and derivation, all inflection terms re-weighted (3×)
c1xw	CELEX inflection and derivation, all compound terms re-weighted (3×)
c2	CELEX inflection and derivation + compound splitting
c2fow	CELEX inflection only + compound splitting, original terms re-weighted (3×)
c4	CELEX inflection and derivation + compound splitting + compound generation
c4f	CELEX inflection only + compound splitting + compound generation
c4fow	CELEX inflection only + compound splitting + compound generation, original terms re-weighted (3×)
sf	CELEX inflection only + synonyms (all senses)
sf1	CELEX inflection only + synonyms (1 sense)
sfow	CELEX inflection only + synonyms (all senses), original terms re-weighted (3×)
sc4	CELEX inflection and derivation + compound splitting + compound generation with synonyms
ida	all good terms
idow	all good terms, original terms re-weighted (3×)
id$n$	n best terms
id3f	three best terms + inflectional variants
id3n	three best terms from the user query
nrf	relevance feedback

Table 23: Explanation of system variant names

Appendix B

Query set 1

1. Hoe heette dat museum ook al weer waar er protesten waren tegen uitbreiding of verbouwing en waar men geprobeerd heeft dit tegen te houden door het gebouw op de monumentenlijst te laten plaatsen?

2. Zijn er nog schandalen geweest in de lokale politiek die betrekking hadden op corruptie van ambtenaren of ander overheidspersoneel?

3. In welke gemeenten hebben de centrumdemocraten (CD) of andere extreem-rechtse partijen winst behaald bij de gemeenteraadsverkiezingen?

4. Ik zoek een recensie van een klassiek concert in het muziekcentrum, bijvoorbeeld een strijkkwartet of een symfonie

5. Ik zoek een bericht over een roofoverval waarbij de buit bestond uit een grote som geld of sieraden en waarbij de dieven ontkomen zijn

6. Ik zoek artikelen over plannen die gemaakt zijn om de dijken van de Maas te verhogen of de rivierbedding te verbreden naar aanleiding van de overstromingen in de winter van 93-94.

7. Ik zoek berichten over technieken voor de behandeling van onvruchtbaarheid bij mannen en vrouwen en over de ethische vraagstukken die de toepassing van deze technieken opwerpen. Artikelen over draagmoeders zijn ook relevant.

    1. reageerbuisbevruchting, embryo, foetus, KI (kunstmatige inseminatie)
    2. eiceldonatie, zaadceldonatie, IVF (in vitro fertilisatie), oma-moeders

8. Een relevant document beschrijft protesten van streng christelijke groeperingen tegen bijvoorbeeld het afschaffen van het bidden op scholen, seksuele voorlichting, homoseksuele leerkrachten, seksueel expliciete en gewelddadige televisieprogramma's, abortus, pornografie en satanische teksten in popmuziek.

Query set 2

1. Een relevant document beschrijft een dreigende spoorwegstaking en de omstandigheden die daartoe hebben geleid of een spoorwegstaking die nu aan de gang is. Om relevant te zijn moet het document de plaats van de staking of dreigende staking beschrijven. Bij een dreigende staking moet het document de status van de onderhandelingen, contractbesprekingen etc. beschrijven, om de waarschijnlijkheid van een staking in te kunnen schatten. Bij een actuele staking moet het document de duur van de staking vermelden en de status van de onderhandelingen of bemiddelingspogingen.

    1. NS, spoorwegstaking, blokkade, stremming, werkwilligen, werkonderbreking, wilde staking
    2. spoorwegvakbond, vakbond van spoorwegpersoneel, bemiddelaar, onderhandelaar, spoorwegen
    3. vakbondsvoorstel, besprekingen, schikking, overtollig personeel aanhouden, kostenbesparing

2. Een relevant document beschrijft maatregelen om geweld en sex in bioscoopfilms, televisieprogramma's en videofilms te beperken. Een relevant document kan zowel wettelijke maatregelen als maatregelen vanuit de mediaindustie zelf beschrijven. Berichten over het functioneren van het filmkeuringssysteem of over mogelijke aanpassingen van de filmkeuring zijn relevant. Berichten over advertenties voor bepaalde films die door kranten of tijdschriften geweigerd zijn zijn ook relevant.

3. Een relevant document beschrijft de toestanden in gevangenissen die samenhangen met het cellentekort. Het document geeft aan hoe gedetineerden gedwongen worden om met het cellentekort om te gaan, of wat het gevangeniswezen of de overheid doet of zal gaan doen om het cellentekort tegen te gaan.

4. Een relevant document bevat gegevens die aantonen dat commerciele overbevissing een teruggang in de visstand heeft veroorzaakt. Het document beschrijft wetten of maatregelen hiertegen, zoals beperkingen op de visvangst, verleggen van visgronden, of andere activiteiten die het vistekort tegengaan; bijvoorbeeld, viskwekerijen, het ontwikkelen van alternatieve soorten.

5. Om relevant te zijn moet het document ten minste een specifieke behandeling voor AIDS (Acquired Immune Deficiency Syndrome) of ARC-patienten (AIDS Related Complex) noemen.

    1. Acquired Immune Deficiency Syndrome (AIDS), AIDS Related Complex (ARC)
    2. behandeling, medicijn, farmaceutische industie

3. test, proefpersonen, onderzoek
4. AZT, TPA
5. Genentech, Burroughs-Wellcome

6. Een relevant document beschrijft een zelfmoord waarbij een persoon met een medische achtergrond geassisteerd heeft en de wettelijke gevolgen van een dergelijke daad. Ook gevallen waarbij sprake is van beinvloeding en niet van vrije keuze van de patient zijn relevant. Zelfdoding om gezondheidsredenen waarbij geen sprake is van hulp door medisch personeel is niet relevant.

7. Een relevant document beschrijft tenminste een manier om het begrotingstekort te reduceren, direct (b.v. het verhogen van belastingen, reduceren van overheidsuitgaven) of indirect (b.v. aanpassing van de koers van de gulden, het bevorderen van investeringen). Een document dat alleen de huidige situatie beschrijft maar geen oplossingen noemt is NIET relevant.

    1. begrotingstekort, financieringstekort
    2. begroting Buitenlandse Zaken, defensiebegroting
    3. belastingverhoging, belastingherziening
    4. reduceren van overheidsuitgaven, bezuinigingen, beperken van overheidssubsidies

8. Een relevant document beschrijft een product, b.v. geneesmiddel, microorganisme, vaccin, dier, plant, landbouwproduct, ontwikkeld met behulp van genetische manipulatietechnieken. Het bechrijft ook een mogelijke toepassing zoals milieurampen of gentherapie bij mensen. Het document mag ook standpunten ten opzichte van genetische manipulatie beschrijven.

    1. genetische manipulatie, moleculaire manipulatie
    2. biotechnologie
    3. genetisch gemanipuleerd product: plant, dier, geneesmiddel, microorganisme,vaccin, landbouwproduct
    4. een ziekte genezen, milieurampen bestrijden, verhogen van landbouwproductie

9. Een relevant document beschrijft een protestactie van Greenpeace. Het bedrijf of land dat het onderwerp is van deze actie moet met name genoemd worden, evenals het specifieke doelwit (b.v. een schip, trein etc.) ook het doel dat Greenpeace probeert te bereiken met deze actie moet vermeld worden.

    1. Greenpeace, milieu, groep, activist
    2. protest, verstoren, blokkeren, lastig vallen, confronteren, overtreden

10. Een relevant document geeft ten minste een voorbeeld van acties die voorgesteld of uitgevoerd zijn door publieke overheden ergens ter wereld om het gebruik van tabak door mensen te beperken of te ontmoedigen. Beperkende maatregelen mogen de volgende vorm hebben: beperkingen op de verkoop en reclame, verplichte waarschuwingsteksten, rookvrije zones. Rechtszaken tegen tabaksfabrikanten en anti-rook campagnes van de overheid worden ook gezien als relevante vormen van ontmoediging. Door de overheid gefinancierd medisch onderzoek naar het gebruik van tabak is ook relevant. Prive initiatieven zoals het creeren van niet-rook gedeeltes zijn NIET relevant tenzij direct gerelateerd aan een initiatief van de overheid. Ook NIET relevant zijn maatregelen van de overheid die het gebruik van tabak juist bevorderen zoals prijsverlagingen en export bevorderende maatregelen.

    1. tabak
    2. roken, anti-rook, niet-roken

11. Een relevant document bevat informatie over vitamines die mogelijk kwalen bij de mens helpen voorkomen of zelfs genezen. Informatie over gezondheidsproblemen veroorzaakt door vitamines is ook relevant. Een document dat slechts in algemene zin aan vitamines refereert (b.v. "goed voor de gezondheid", "hebben voedingswaarde") is niet relevant. Informatie over onderzoek dat wordt uitgevoerd maar dat nog niet tot resultaat heeft geleid is niet relevant. Verwijzingen naar vitaminederivaten staan gelijk aan verwijzingen naar de vitamine zelf.

Query set 3

1. Geef mij alle artikelen die betrekking hebben op veeartsen, boeren, en ongevallen of misdaden in zowel Nederland als Belgie
2. Welke verkiezingen vonden plaats? Welke partij heeft het goed gedaan? Welke partij heeft het slecht gedaan?
3. Welke bosbranden hebben mensen het leven gekost?
4. Tegen welke teams speelde het Nederlands elftal gelijk op de wereldkampioenschappen voetbal in de Verenigde Staten?
5. Geef mij alle artikelen over een eventuele fusie van Berlicum met St. Michielsgestel.
6. Geef alle artikelen met verslagen van rechtszaken met betrekking tot financiele compensatie voor medische fouten.
7. Ik ben op zoek naar artikelen over de plannen voor de vorming van een stadsprovincie in de agglomeratie Eindhoven-Helmond
8. Ik ben op zoek naar informatie over de procedure voor het verkrijgen van een verblijfsvergunning
9. Voor welke literaire prijzen werden voornamelijk autobiografische romans genomineerd?
10. Welke bekende personen werden naar aanleiding van het proces tegen Michael Jackson beschuldigd van seksuele perversiteiten met kinderen.
11. Geef me de berichten die handelen over de plannen van het ministerie van onderwijs om de kosten van het promotieonderzoek door onderzoekers en assistenten in opleiding beheersbaar te maken en over de reacties van universiteiten
12. Ik ben op zoek naar recensies van gespecialiseerde restaurants in de streek van Brabant Limburg en Utrecht met name vegetarische Indonesische en Italiaanse.
13. Ik wilde graag wat meer weten over de voor- en nadelen van de verschillende methoden van afvalverwerking (in principe maakt het soort afval mij niet uit: papier chemisch afval en biologisch afval...), zoals daar zijn: verbranding compostering of dumping
14. Wat voor voordelen heeft het gebruik van de electronische snelweg in universiteiten in Nederland opgeleverd?
15. Op welke plaatsen langs welke snelwegen heeft de politie het afgelopen jaar omvangrijke snelheidscontroles uitgevoerd?
16. Geef mij alle artikelen over peacekeeping operaties van de VN vredesmacht in Afrika en Azië.
17. Hebben Tilburgse roeiers successen behaald ?
18. In welke natuurgebieden in Nederland worden paarden en/of runderen gebruikt voor natuurlijk beheer.
19. Geef mij informatie over de activiteiten waar RaRa zich in de afgelopen jaar mee bezig is geweest.
20. Geef mij alle artikelen over de serie inbraken in cafe de Gouden Leeuw in Berlicum.
21. In welke gemeenten zijn er plannen ontwikkeld voor een referendum over gemeentelijke herindeling en/of regiovorming?
22. wat is de invloed van radioactieve straling op het lichamelijk en geestelijk functioneren van de mens?
23. Wat was de mening van Groen Links over het wel of niet toestaan van euthanasie?

24. Wat zijn de aanslagen die gepleegd werden in Israel het laaste jaar door Hamas en Il-Jihad Il-Islami
25. geef een lijst van de aanvallen die Israel heeft gepleegd op zuid Libanon
26. Geef mij een overzicht van de laatste ontwikkelingen op het gebied van de publieke en commerciële omroepen in Nederland
27. Geef mij eens alle artikelen die over computers en talen en hun onderlinge verbanden gaan
28. Het onderwerp moet zijn spoorwegmodelbouw of modelbouw in het algemeen, als hobby, met de nadruk op scenery en voertuigen. Is er een ruilbeurs of een manifestatie geweest?
29. welke maatregelen worden in de biologische landbouw getroffen om energiebesparing te bewerkstelligen
30. geef mij alle artikelen die er verschenen zijn van het satanisme in Noord-Brabant
31. Welke stoffen in chemisch afval beinvloeden de vruchtbaarheid of bootsen oestrogenen na?
32. denken de oostbrabantse veehouders het mestprobleem te kunnen oplossen door verbeterd veevoer of is volumebeleid noodzakelijk
33. Welke landen heeft Beatrix een staatsbezoek gebracht?
34. wat zijn de gevolgen van de overstromingen van de grote rivieren dit voorjaar geweest voor de landbouw en/of veeteelt in het getroffen gebied?
35. in welke gemeente hebben de hindoestanen of andere allochtonen die partij zijn winst behaald bij de gemeenteraadsverkiezingen/?
36. Geef mij alle verslagen van de wedstrijden van het Nederlands elftal op het WK voetbal in Amerika.

References

[1] IJsbrand-Jan Aalbersberg. Incremental relevance feedback. In *Proceedings of ACM-SIGIR92*, 1992.

[2] IJsbrand Jan Aalbersberg, Ewout Brandsma, and Marc Corthout. Full text document retrieval: from theory to applications. *Informatiewetenschap 1991, Wetenschappelijke bijdragen aan de eerste STINFON-Conferentie*, 1991.

[3] R. H. Baayen, R. Piepenbrock, and H. van Rijn, editors. *The CELEX Lexical Database (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia (PA), 1993.

[4] Ewout Brandsma. Tru programmer's documentation - part 1: Reference manual. Technical report, Philips Research, 1995.

[5] Kenneth Ward Church. One term or two? In *Proceedings of ACM-SIGIR95*, pages 310–318, 1995.

[6] Donna Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42(1):7–15, 1991.

[7] Donna Harman, editor. *The First Text Retrieval Conference (TREC-1)*. National Institute for Standards and Technology, 1993.

[8] Donna Harman, editor. *The Second Text Retrieval Conference (TREC-2)*. National Institute for Standards and Technology, 1994.

[9] Donna Harman, editor. *Overview of the Third Text Retrieval Conference (TREC-3)*. National Institute for Standards and Technology, 1995.

[10] David Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of ACM-SIGIR93*, pages 329–338, 1993.

[11] Zeger Karssen, Gemme Swartzenberg, and Joost de Jonge. Understanding conceptual information retrieval. In L.G.M. Noordman and W.A.M. de Vroomen, editors, *Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie*, pages 27–38, 1994.

[12] Wessel Kraaij and Renée Pohlmann. Porter's stemming algorithm for Dutch. In L.G.M. Noordman and W.A.M. de Vroomen, editors, *Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie*, pages 167–180, 1994.

[13] Wessel Kraaij and Renée Pohlmann. Evaluation of a Dutch stemming algorithm. In Jan Don, Bert Schouten, and Wim Zonneveld, editors, *OTS yearbook 1994*, pages 63–84. Research Institute for Language and Speech (OTS), 1995.

[14] Robert Krovetz. Viewing morphology as an inference process. In *Proceedings of ACM-SIGIR93*, pages 191–203, 1993.

[15] J. B. Lovins. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31, 1968.

[16] John K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1994.

[17] Chris D. Paice. An evaluation method for stemming algoritms. In *Proceedings of ACM-SIGIR94*, pages 42–50, 1994.

[18] Mirko Popovič and Peter Willett. The effectiveness of stemming for natural-language access to Slovene textual data. *Journal of the American Society for Information Science*, 43(5):384–390, 1992.

[19] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[20] Ellen Riloff. Little words can make a big difference for text classification. In *Proceedings of ACM-SIGIR95*, pages 130–136, 1995.

[21] G. Salton. *Automatic Text Processing - The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Publishing Company, Reading (MA), 1989.

[22] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Book Co., New York, 1983.

[23] Gerard Salton and Christopher Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 24(4):288–297, 1990.

[24] Jacques Savoy. Stemming of French words based on grammatical categories. *Journal of the American Society for Information Science*, 44(1):1–9, 1993.

[25] Jean M. Tague. The pragmatics of information retrieval experimentation. In Karen Sparck Jones, editor, *Information Retrieval Experiment*, pages 59–102. Butterworths, 1981.

[26] Jean Tague-Sutcliffe. *Measuring Information, an information services perspective*. Academic Press, 1995.

[27] Jean Tague-Sutcliffe. A statistical analysis of the TREC-3 data. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 385–398, 1995.

[28] Ellen M. Voorhees. Query expansion using lexical-semantic relations. In W. Bruce Croft and C.J. van Rijsbergen, editors, *Proceedings of ACM-SIGIR94*, pages 61–69, 1994.

[29] T. G. Vosse. *The Word Connection*. PhD thesis, Rijksuniversiteit Leiden, Neslia Paniculata Uitgeverij, Enschede, 1994.