Viewing Stemming as Recall Enhancement

Wessel Kraaij
kraaij@tpd.tno.nl
Institute of Applied Physics
Netherlands Organisation for Applied Scientific Research (TNO)
Delft
The Netherlands

Renée Pohlmann
Renee.C.Pohlmann@let.ruu.nl
Research Institute for Language and Speech (OTS/STT)
Utrecht University
Utrecht
The Netherlands

Abstract

Previous research on stemming has shown both positive and negative effects on retrieval performance. This paper describes an experiment in which several linguistic and non-linguistic stemmers are evaluated on a Dutch test collection. Experiments especially focus on the measurement of Recall. Results show that linguistic stemming restricted to inflection yields a significant improvement over full linguistic and non-linguistic stemming, both in average Precision and R-Recall. Best results are obtained with a linguistic stemmer which is enhanced with compound analysis. This version has a significantly better Recall than a system without stemming, without a significant deterioration of Precision.

1 Introduction

One of the techniques employed in Information Retrieval (IR) to improve performance is stemming of document and query terms. By reducing morphological variance of terms (e.g. mapping singular and plural forms of the same word on a single stem) researchers hope to improve the query-document matching process. Several different techniques have been proposed to achieve this goal. One of the simplest techniques, suffix stripping, uses a list of frequent suffixes to reduce words to their base form or 'stem' e.g. [Lovins, 1968], [Porter, 1980]. Based on an evaluation experiment with several different suffix-stripping algorithms, Harman [1991] concluded that suffix stripping does not improve retrieval effectiveness, at least not for English. Other researchers, however, have reported favourable results using more linguistically motivated stemming algorithms for English [Krovetz, 1993] or morphologically more complex languages like Slovene [Popovič & Willett, 1992]. Furthermore, recent research [Hull, 1996] seems to indicate that a more detailed evaluation of stemming algorithms, focusing on Recall, does reveal significant improvement, even for English. In the UPLIFT project [1] we investigated whether suffix stripping is effective for the Dutch language and what effect the use of more linguistically motivated stemming techniques would have. We evaluated our results using traditional Precision/Recall measures and, like Hull, we also looked at the effect of stemming on Recall in more detail. In the next section (2) we will summarize the results of other stemming experiments, we will continue with a description of the set-up of our stemming experiment (section 3) and we will conclude with a discussion of results (section 4).

2 Background

A number of researchers have reported results for evaluation experiments with stemming algorithms. We will discuss some representative results here.

Harman [1991] compared three well-known suffixing algorithms for English: the S–stemmer, the Lovins stemmer [Lovins, 1968] and the Porter stemmer [Porter, 1980].

Harman contrasted these suffixing algorithms with a baseline of no stemming at all. After a detailed evaluation[2], 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 [1992] 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[3]. Their experiment shows a significant improve-

[1] UPLIFT (Utrecht Project: Linguistic Information for Free Text retrieval) is funded by the NBBI, Philips Research, the Research Institute for language and Speech (OTS), the Dutch Ministry of Education and Science and the Dutch Ministry of Economic Affairs. UPLIFT Home Page: `http://wwwots.let.ruu.nl/~uplift`

[2] Evaluation measures used are: average Precision at 0.20, 0.50 and 0.80 Recall, van Rijsbergen's e-measure, number of queries that fail (i.e. 0 recall) at 10/30 documents retrieved and total relevant retrieved at 10/30 documents retrieved

[3] The test collection consisted of approximately 500 documents and 48 queries.

This paper is a slightly revised version of the paper which is to appear in the proceedings of SIGIR96

ment in Precision (at fixed retrieval of the 10 most highly ranked documents). 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.

Krovetz [1993] investigated whether more linguistically motivated stemming algorithms would be effective for English and compared them with the Porter algorithm. Krovetz evaluated the performance of four different stemming algorithms using standard test corpora for English (CACM, TIME, NPL and WEST): Porter, revised Porter (a dictionary is used to check whether the resulting stem really exists), an inflectional stemmer and a derivational stemmer (removes both inflectional and derivational affixes).

Surprisingly, Krovetz finds that all stemmers yield a significant improvement[4] over no stemming. The derivational stemmer generally gives the best results [5]. Krovetz notes that improvements from stemming increase at higher levels of Recall and that derivational morphology is responsible for improvement at high levels of Precision. Document length also seems to be of importance; the best results are obtained with short documents (CACM and NPL collections). It is interesting to note that although both Harman and Krovetz have evaluated the Porter algorithm using the same test collection (CACM) and (almost) the same evaluation measure (AP[0.20,0.50,0.80] vs. AP[0.25,0.50,0.75]), they do not reach the same conclusion. Harman concludes that Porter does not yield a statistically significant improvement over no stemming whereas Krovetz finds that there is a significant improvement.

In a recent article, Hull [1996] argues that current evaluation measures such as average Precision and average Recall are not ideally suited for evaluation of retrieval techniques in general and stemming strategies in particular. Hull claims that average performance figures need to be validated with careful statistical analysis and that detailed analysis of individual queries can uncover important differences that are not found using the traditional measures. Besides the standard average Precision at 11 recall points (0.0,0.10,...1.0) (APR11) which he uses for comparison with other results, he proposes two new evaluation measures, average Precision at 5-15 documents examined (AP[5-15]) and average Recall at 50,60,...150 documents examined (AR[50-150]), which he claims are more suited to estimate performance for shallow searches and more in-depth searches respectively. He subsequently adapts these measures to normalize for query variance by averaging over within-query rank or score. Using these measures, he evaluates the performance of five different stemming algorithms (removes, Lovins, Porter, Xerox inflectional stemmer, Xerox derivational stemmer) using the TREC test collection [Harman, 1993a, 1994, 1995] . Statistical tests are applied and detailed, per-query analysis is carried out to identify probable causes for differences between stemmers. Hull concludes that stemming in general is almost always beneficial, except for long queries (i.e. full TREC queries) at low Recall levels, but he is unable to demonstrate significant differences between suffix stripping algorithms like Porter and Lovins and the linguistic stemming algorithms.

We can conclude that there is a lot of variation in the results of stemming experiments. Quite a number of factors seem to be of importance, e.g. linguistic vs. non-linguistic stemmers, language, query and document length, evaluation measures, etc. It is clear that further research is necessary to clarify inconsistencies.

[4] Figures range from from 1.3 to 45.3% improvement in average Precision at Recall 0.25, 0.50 and 0.75.

[5] Except for the NPL collection where the original Porter algorithm performs best.

3 Design of the UPLIFT evaluation experiment

The research summarized above inspired us to run our own experiment. 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[6]. Besides algorithms comparable in coverage to the suffix strippers and linguistic stemmers that were used in the experiments described above we also developed some additional variants in attempts to optimize our stemming strategy. These variants include stemmers that also handle compounding phenomena and variants based on reweighting schemes for the query vector. In the following section the different system variants developed for the experiment will be illustrated and we will continue with a discussion of a number of other key issues in the design and setup of our evaluation experiment: the test collection, evaluation measures and statistical validation.

3.1 System variants

The retrieval engine used in the UPLIFT project is the TRU vector space engine developed by Philips Research [Aalbersberg et al., 1991]. A plain version of this system (i.e. without a stemming algorithm) was used as a baseline for our experiment.

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[7].

We subsequently developed two linguistic stemmers (inflectional and derivational) using a computer readable dictionary, the CELEX lexical database [Baayen et al., 1993]. 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 when 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 [Harman, 1991]. 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 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 [Salton, 1989]. 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

[6] Besides this experiment we have also investigated the use of synonyms in retrieval. The results of these experiments will not be reported here. For details on these experiments and for an in-depth description of the stemming experiment we refer to [Kraaij & Pohlmann, 1996]. The use of syntactic information will be the subject of the next phase of our project.

[7] For a more detailed description of the Dutch Porter algorithm we refer to [Kraaij & Pohlmann, 1994].

morphological variants of the same concept are treated as independent base vectors. Harman corrected for this defect by modifying the similarity computation procedure: 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 used a different approach and 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.

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[8], 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[9]:

recensies	*recensie* (singular)
klassieke	*klassiek* (positive) *klassiekst* (superl.)
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 give a concert)
muziekcentrum		
eindhoven		

Careful analysis of the document collection used in the UPLIFT project (see section 3.2 for details), revealed that of a subset of approximately 50,000 unique word forms, 40% were not included in CELEX. We examined a random sample of approximately 2,500 of these words to establish why they were not in the dictionary. The results of this analysis are summarized below:

46%	proper names
37%	nominal 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 nominal compounds. We anticipated that compounds would be a problem case. In Dutch, nominal compounds are generally formed by concatenating two (or more) words to create a single orthographic word, e.g. *fiets* ('bicycle') + *wiel* ('wheel') $\rightarrow$ *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 [Vosse, 1994]. 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 approximately 1,000 compounds not included in the CELEX dictionary[10]:

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)	*recensiewerk* etc. (review work)
klassieke	*popklassieker* (pop classic)	*Elvis-klassieker* etc. (Elvis classic)
concerten	*popconcerten* (pop 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).

We also developed some extra stemming variants to test the influence of (re-)weighting schemes for query terms. One of the reasons for introducing these variants was the fact that pilot experiments seemed to indicate that the plain reference system performed better than the stemming variants developed so far. We wanted to test whether the terms added after expansion should have a lower weight than original query terms. The idea behind these versions was that the more the weight of the original terms is increased, the more performance results should approximate the results of the reference version. We experimented with varying the weight of the

[8] 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)).

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

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

original terms between 1 and 5, 3 turned out to be the best choice.

The following stemmers were used in pilot experiments:

- **n**: no stemming
- **p2**: Dutch Porter stemmer
- **p2ow**: Porter, original terms weight 3
- **c1f**: CELEX inflectional stemmer
- **c1fow**: CELEX inflectional stemmer, original terms weight 3
- **c1**: CELEX inflectional & derivational stemmer
- **c1ow**: CELEX inflectional & derivational stemmer, original terms weight 3
- **c2f**: CELEX inflectional stemmer with compound splitting
- **c2fow**: CELEX inflectional stemmer with compound splitting, original terms weight 3
- **c2**: CELEX inflectional & derivational stemmer with compound splitting
- **c2ow**: CELEX inflectional & derivational stemmer with compound splitting, original terms weight 3
- **c4f**: CELEX inflectional stemmer with compound splitting & generation
- **c4fow**: CELEX inflectional stemmer with compound splitting & generation, original terms weight 3
- **c4**: CELEX inflectional & derivational stemmer with compound splitting & generation
- **c4ow**: CELEX inflectional & derivational stemmer with compound splitting & generation, original terms weight 3
- **p2pr**: Porter, control version (no query expansion)

In section 4 we will discuss the results of the final experiment. A representative subset of the versions mentioned above was used in the final evaluation experiment in order to minimize the waiting time for the test subjects.

3.2 Test collection

Since there are no standard test collections available for Dutch, we had to compile our own collection. In order to facilitate comparison with published IR evaluation results we tried to adhere to the standards set by the TREC experiments where possible.

Document collection

Since the UPLIFT project aims at developing domain independent full text retrieval strategies, 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[11] 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[12]. 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	26,572,588
Total number of terms	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

Test subjects, queries and relevance judgements

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. 3.4). The average number of documents that were judged relevant by the subjects was 29.4.

3.3 Evaluation Measures

Precision/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 [Tague, 1981]. For our experiment, we decided to use the 'Pooling method' which is also employed in TREC cf. [Harman, 1993b]. 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. Results for all the different systems are merged into a single list (cf. 3.2) and this list is assumed to contain most of the relevant documents. We think that the pool of UPLIFT system versions tested in the experiment contained sufficiently differing systems to make this assumption acceptable[13].

Precision and Recall are intuitive parameters for boolean retrieval systems. These systems retrieve a fixed number of documents. Relevance ranked based systems like VSM yield a (partial)

[11] This is comparable in size to the individual test corpora used in the TREC evaluation experiments.

[12] These key words were of course not used for document indexing during the experiment.

[13] A total of 17 different versions were used to create the document list, the merged list of the top 100 documents of these 17 versions contains 289 documents on average. Besides Porter- and CELEX-based versions the pool also included versions with synonym expansion. For more details, the reader is referred to [Kraaij & Pohlmann, 1996].

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. A problem arises when a rank contains more than one document. Document rankings often contain 'ties' between documents: the match value (assigned by the retrieval engine) is equal and the engine falls back on a secondary ordering method (e.g. document number). We have corrected for this effect in the following way: If such a group contains relevant documents, we assume they are ordered in the middle of the group.

Average Precision

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. 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 used a second measure: average Precision, from the collection of measures assessed in TREC3 [Tague-Sutcliffe, 1995b]. The average Precision for a certain query and a certain system version is computed by averaging all Precision values at relevant document positions in the relevance ranking. This measure has a number of advantages: it is easy to compute, does not require interpolation and has proven to yield reliable results in TREC3 cross-measure evaluation experiments. For the TREC experiments, Average Precision also proved to be a suitable measure to make quick comparisons between a large number of system versions and allow for an easy statistical validation with an analysis of variance (ANOVA).

R-Recall

Since the evaluation measures mentioned above focus on Precision values and stemming is mainly a Recall enhancement technique, we have also experimented with various Recall measures. Like Hull [Hull, 1996], we started with measuring Recall at fixed document cutoff points (25,50,75,100,200,500 and 1000). A disadvantage of this method is that Recall at 25 does not seem to make much sense for queries with many relevant documents. On the other hand, Recall measured at document cutoff levels of 200 and more seems only of academic importance and is not interesting for users. 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. R-Recall is an intuitively pleasing measure, an ideal system has an R-Recall of 1 and R-Recall is by definition equal to R-Precision[14]. In analogy with Recall at different fixed cutoff points, we also examined 2R-Recall and 5R-Recall, i.e. Recall measured at 2 and 5 times R respectively. At 2R, Recall is by definition equal to twice the Precision, at 5R the ratio is 5 to 1. We think that R-Recall is a suitable measure to normalize the query variety which is present in every IR testing corpus.

3.4 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 [1983] 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 of two populations without any restriction on the distribution function. Tague-Sutcliffe [1981] and Hull [1993] 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 [1995a] 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 [Tague-Sutcliffe, 1995b] and [Tague-Sutcliffe, 1995a]. 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:

$$Y_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij} \quad (1)$$

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.

4 Results

Figures 1, 2 and 3 show Precision/Recall graphs for a representative subset of the versions tested in the experiment.

Figures 4, 5, 6 and 7 present the results of the ANOVAS that were run on the data.

The most important figures in the ANOVA tables are the F-values in the rightmost column, which represent the quotient of the variance in measurements which can be attributed to the effect we are interested in and the variance due to chance. This quotient is of course dependent on the degrees of freedom of the variables in the model i.e. number of system versions and queries. The F distribution shows us that the run effect is significant at the 0.99

[14] This measure was introduced for TREC2 by Chris Buckley (Cornell University).

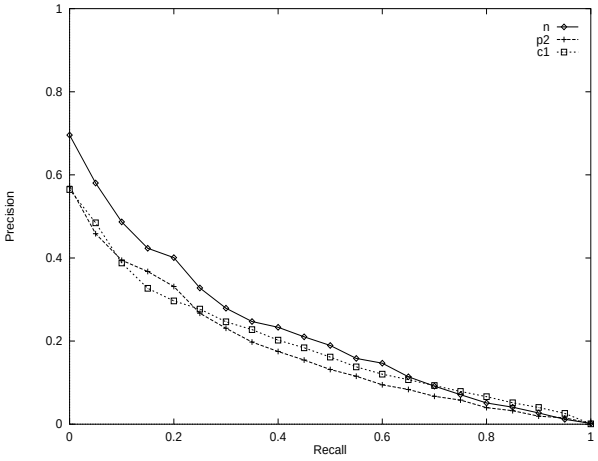

Figure 1: Porter vs CELEX

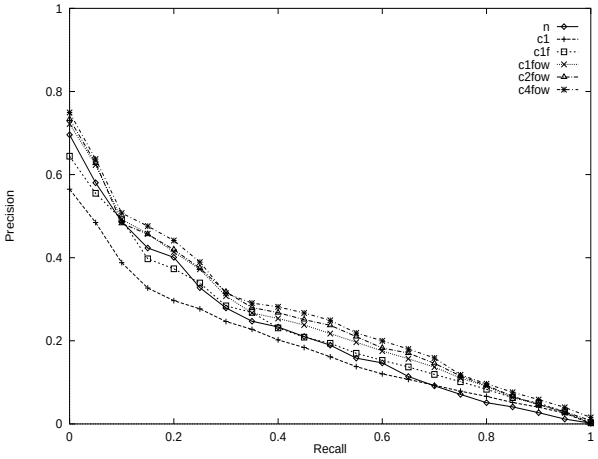

Figure 2: CELEX variants

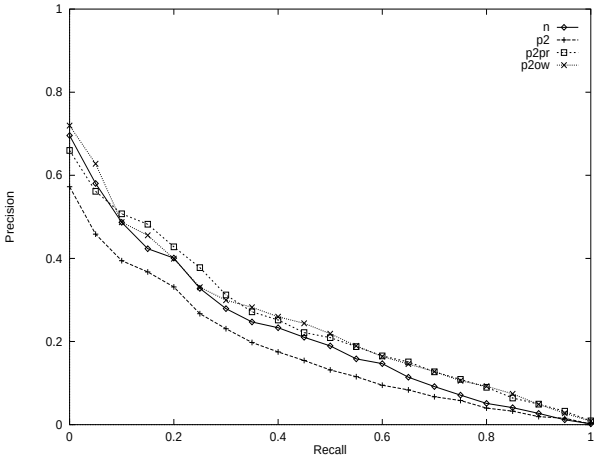

Figure 3: Weighting variants

Source	DF	Sum of Sq	Mean Sq	F val
Runs	8	0.2947	0.0368	6.7321
Query	35	12.8883	0.3682	67.3006
Error	280	1.5320	0.0055	
Total	323	14.7150		
s.e.d. (Runs)	0.017			

Figure 4: RESULTS: ANOVA TABLE Average Precision

Source	DF	Sum of Sq	Mean Sq	F val
Runs	8	0.3105	0.0388	7.6189
Query	35	12.3682	0.3534	69.3692
Error	280	1.4264	0.0051	
Total	323	14.1051		
s.e.d. (Runs)	0.017			

Figure 5: RESULTS: ANOVA TABLE Recall at R

Source	DF	Sum of Sq	Mean Sq	F val
Runs	8	0.4098	0.0512	5.0014
Query	35	14.2546	0.4073	39.7680
Error	280	2.8676	0.0102	
Total	323	17.5320		
s.e.d. (Runs)	0.024			

Figure 6: RESULTS: ANOVA TABLE Recall at 2R

Source	DF	Sum of Sq	Mean Sq	F val
Runs	8	0.4976	0.0622	4.4846
Query	35	15.6642	0.4475	32.2663
Error	280	3.8837	0.0139	
Total	323	20.0456		
s.e.d. (Runs)	0.028			

Figure 7: RESULTS: ANOVA TABLE Recall at 5R

level for all ANOVAS, because the F values of the run exceed $F_{.99;8,280}$[15] = 2.60.
This means that we can reject the hypotheses that the run-effects of the corresponding measures are equal to zero with a certainty of 99%. The query effect (Query column) is also clearly significant: the F-values exceed $F_{.99;35,280}$ = 1.65. This justifies the choice for a randomized block design (cf. section 3.4). Inspection of a fitted value plot showed that the assumption of homogeneity of variances is confirmed, therefore arcsine root transformations to stabilize variances of the data are not required.

Because the ANOVA only shows that there are significant differences between system versions, it is necessary to do multiple pairwise comparisons to detect which specific versions are concerned. The pairwise comparisons are based on the simple method of computing confidence intervals. The T-distribution is used to calculate the standard error of differences of means. These *s.e.d*-values mark confidence intervals of 95 %, i.e. the absolute difference between population and sample mean is smaller than the s.e.d. with a certainty of 95 %.

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

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

[15] The subscripts refer to the significance level (1-0.025) and the degrees of freedom.

Figures 8, 9, 10 and 11 present the results of the multiple comparisons.

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

Summary of results

Generally speaking we can conclude that stemming does improve Recall but at the cost of some Precision. The most salient details are summarized below:

- Porter (p2) and full CELEX stemming (inflection & derivation, c1) show no significant differences for all measures.
- Selective stemming (c1f) is significantly better than full stemming (c1) in both Average Precision (19%) and the Recall measurements (28%, 13%, 11%).
- Splitting compounds (c2fow) and generating compounds (c4fow) seem to improve upon the basic inflectional stemmer, both in Recall and Average Precision, but the improvement is not significant. c4fow, however, is significantly better than the reference version (n).
- In most cases the variants with term weight 3 for original query terms perform better than their non re-weighted counterparts. The difference is significant for the Porter (p2, p2ow) variants.
- Retrieval performance of stemming emulation via query expansion with higher term weights (3) for the original terms is equal to the performance of a system where the stemmer is applied during the indexing process (p2ow vs. p2pr).
- Although it is difficult to say whether comparing our results with TREC results is a valid action[16], we did a quick comparison to get an idea about the performance of our system with respect to TREC3 systems. This comparison shows that the performance of the TRU system with c4fow stemmer ranks among the mid-range of TREC systems.

4.1 A more qualitative analysis of results

In order to qualify the results of our expansion techniques we investigated the level of result that could be achieved by an 'ideal query'. This query should give an idea of the maximum performance that can be achieved with stemming for our system. We tried to approximate such an ideal query automatically. We constructed it in the following way: for each query a set of unique terms is collected from all system version expansions. For each term, a program tests whether it yields relevant documents. If so, the term is added to the 'ideal query term list'. In this way a query is constructed that only contains terms that are included in relevant documents. We also tested another (very simplistic) relevance feedback version: **nrf** which simply is a plain run (n) followed by a second run with the top ranked document as query. Figure 12 shows the results for these versions.

We also looked at the distribution of successful query terms (i.e. terms present in relevant documents) over system versions. Table 1 summarizes results: 20% of the successful query terms were already present in the original query (n), the rest of the good terms are found by expansion versions[17].

[16] Test collections differ, languages differ, but test procedures are comparable.

[17] This table also contains results for the expansion versions with synonyms which are not discussed in this paper. For details on these versions the reader is referred to [Kraaij & Pohlmann, 1996].

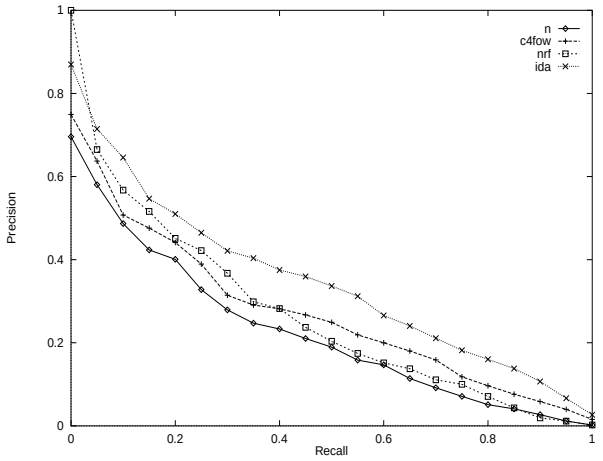

Figure 12: Ideal queries and relevance feedback

n	c1f	c1	c2	c4	sf	porter
20%	13%	9%	8%	5%	40%	6%

Table 1: Distribution of successful query terms over versions

A more detailed investigation of the successful query terms 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 (n) in 70% of the cases[18]. Other expansion versions that delivered the best term were: inflection (11%), compound splitting (8%), synonyms (5%), derivation (3%) and porter (3%). If we look at the syntactic category of successful query terms, we find (not surprisingly) that nouns[19] 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 Overall conclusions

We have compared several stemming techniques focusing on the enhancement of 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. Compound analysis (c4fow) yields the best results, it even seems to improve precision. The experiments with ideal queries show that 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. Further research is necessary to explore the possibilities of interactive use of

[18] This explains why favouring (re-weighting) original terms is so successful.

[19] Including nominal compounds and proper nouns.

c4fow	n	p2ow	p2pr	c2fow	c1fow	c1f	p2	c1
0.350	0.343	0.342	0.340	0.337	0.335	0.312	0.272	0.261

Figure 8: Equivalent versions based on multiple comparison of means of AVP

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

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

c4fow	c2fow	p2ow	p2pr	c1fow	c1f	n	c1	p2
0.447	0.429	0.420	0.415	0.411	0.391	0.391	0.346	0.333

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

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

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

the query expansion technique. The Recall at R measures (where R is the number of relevant documents for a certain query) form a good alternative for the traditional Recall at fixed document cut-off levels which are not suitable for query collections where R shows a lot of variance. 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, we consider this an important result.

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 and OTS staff and students for their participation in the experiment.

References

[Aalbersberg et al., 1991] Aalbersberg, Y. J., Brandsma, E., & Corthout, M. (1991). Full text document retrieval: from theory to applications. *Informatiewetenschap 1991, Wetenschappelijke bijdragen aan de eerste STINFON-Conferentie*.

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

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

[Harman, 1993a] Harman, D., editor (1993a). *The First Text REtrieval Conference (TREC-1)*. National Institute for Standards and Technology. Special Publication 500-207.

[Harman, 1993b] Harman, D. (1993b). Overview of the first text retrieval conference (TREC-1). In *The First Text REtrieval Conference (TREC-1)*, pp. 1–20. National Institute for Standards and Technology. Special Publication 500-207.

[Harman, 1994] Harman, D., editor (1994). *The Second Text REtrieval Conference (TREC-2)*. National Institute for Standards and Technology. Special Publication 500-215.

[Harman, 1995] Harman, D., editor (1995). *Overview of the Third Text REtrieval Conference (TREC-3)*. National Institute for Standards and Technology. Special Publication 500-225.

[Hull, 1993] Hull, D. (1993). Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of ACM-SIGIR93*, pp. 329–338.

[Hull, 1996] Hull, D. (1996). Stemming algorithms – a case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1).

[Kraaij & Pohlmann, 1994] Kraaij, W., & Pohlmann, R. (1994). Porter's stemming algorithm for Dutch. In Noordman, L., & de Vroomen, W., editors, *Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie*, pp. 167–180.

[Kraaij & Pohlmann, 1996] Kraaij, W., & Pohlmann, R. (1996). Using linguistic knowledge in information retrieval. OTS Working Paper OTS-WP-CL-96-001, Research Institute for Language and Speech (OTS), Utrecht University.

[Krovetz, 1993] Krovetz, R. (1993). Viewing morphology as an inference process. In *Proceedings of ACM-SIGIR93*, pp. 191–203.

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

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

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

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

[Salton & McGill, 1983] Salton, G., & McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill Book Co., New York.

[Tague, 1981] Tague, J. M. (1981). The pragmatics of information retrieval experimentation. In Sparck Jones, K., editor, *Information Retrieval Experiment*, pp. 59–102. Butterworths.

[Tague-Sutcliffe, 1995a] Tague-Sutcliffe, J. (1995a). *Measuring Information, An Information Services Perspective*. Academic Press, San Diego (CA).

[Tague-Sutcliffe, 1995b] Tague-Sutcliffe, J. (1995b). A statistical analysis of the TREC-3 data. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pp. 385–398. National Institute for Standards and Technology. Special Publication 500-225.

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