An Evaluation of Linguistically-motivated Indexing Schemes

Avi Arampatzis Th.P. van der Weide C.H.A. Koster P. van Bommel

Technical Report CSI-R9927, December 1999, Dept. of Information Systems and Information Retrieval,
University of Nijmegen, The Netherlands.
{avgerino,tvdw,kees,pvb}@cs.kun.nl

Submitted to BCS-IRSG 2000

December 14, 1999

Abstract

In this article, we describe a number of indexing experiments based on indexing terms other than simple keywords. These experiments were conducted as one step in validating a linguistically-motivated indexing model. The problem is important but not new. What is new in this approach is the variety of schemes evaluated. It is important since it should not only help to overcome the well-known problems of bag-of-words representations, but also the difficulties raised by non-linguistic text simplification techniques such as stemming, stop-word deletion, and term selection. Our approach in the selection of terms is based on part-of-speech tagging and shallow parsing. The indexing schemes evaluated vary from simple keywords to nouns, verbs, adverbs, adjectives, adjacent word-pairs, and head-modifier pairs. Our findings apply to Information Retrieval and most of related areas.

1 Introduction

The purpose of an automated information seeking system is to process information sources, and provide users with the information they need. The particular nature of an information seeking process is determined by the characteristics of information needs and information sources, such as the change rate. For instance, Information Retrieval assumes a one-time user request and a static collection of information objects, while Information Filtering assumes a long-term user interest and a dynamic collection. What all seeking processes have in common is a technique for representing needs and sources. Representation makes possible to automate a process of computing comparisons of relevance between needs and sources. The process of building such representations is widely know as *indexing*.

Representations are usually derived from the contents of objects. In case of textual objects (documents), words taken directly from the document's text are augmented with weights and traditionally used to form a bag-of-words representation, disregarding the linguistic context. This bag-of-words representation presents inadequacies which have been identified by many researchers. The most obvious inadequacies originate from *linguistic variation* at the morphological, syntactical, and semantical levels of a natural language.

Linguistic variation is responsible for many possible alternative formulations of a single meaning. Briefly, morphology allows affix changes in words as a result of syntax. Additionally, syntax determines if and how words are associated. Lexico-semantics are about words which can be used in more than one senses, or conversely, senses which can be expressed using different words. Many researchers have developed techniques to deal with linguistic variation, nevertheless, empirical evaluations have presented mixed results leaving the matter unsettled.

In this article, we describe an experiment which was performed as a first attempt in a long way to validate a linguistically-motivated indexing model. This model incorporates various techniques dealing with the types of linguistic variation which are most relevant to information seeking tasks, in a single indexing and matching scheme. The approach taken here is based on a Part-Of-Speech (POS) tagger and syntactic pattern matching. First, we experimented with representations based on combinations of different POS categories. These representations combine the category of nouns with these of adjectives, verbs, and adverbs. The different representational choices are compared to the baseline of using all keywords as index terms. Then, we experimented with composite terms which were built, firstly, using a simple criterion like *word adjacency* and secondly, using syntactic structure like *word modification*. We also investigated the effect of morphological normalization by means of *lemmatization*, which can be seen as POS-directed stemming. Evaluation is done in a classification environment using precision and recall.

The rest of this article is organized as follows. First, we briefly present in section 2 the linguistically-motivated indexing model we want to validate with this line of research. In section 3 we summarize and justify our representational choices. In section 4 we describe the experimental system, algorithms used, evaluation measures, the dataset and pre-processing applied to it. In section 5, experiments and results are discussed. Conclusions are drawn in section 6 and directions for further research are identified.

2 A Linguistically-motivated Indexing Model

The indexing model we would like to validate is based on the Phrase Retrieval Hypothesis [1]. The idea of using phrasal indexing terms is not new and can already be found in [5, 10, 17]. It has been explored by several researchers in different ways and with mixed results, e.g. [18, 19, 21]. However, this approach tries to incorporate various techniques which deal with linguistic variation in a single phrase-based indexing scheme. In this section, the underlying model is briefly described. For a more detailed description the reader may refer to [2].

According to the linguistic principle of *headedness*, any phrase has a single word as a head. This head is the main verb in the case of verb phrases, usually a noun (the last noun before any post-modifiers) in noun phrases. The rest of the phrase consists of modifiers. Consequently, every phrase can be represented by a *phrase frame*:

$$PF = [h, m]$$

The head h gives the central concept of the phrase and the modifiers m serve to make it more precise. Conversely, the head may be used as an abstraction of the phrase, loosing precision but gaining recall. Heads and modifiers in the form of phrases can be nested: $[[h_1, m_1], [h_2, m_2]]$. The modifier part might be empty in case of a bare head. This case is denoted equivalently by $[h,]$ or $[h]$.

To deal with the sparsity of phrasal terms, *linguistic normalization* is introduced. Its goal is to cluster different but semantically equivalent phrases (figure 1). We distinguish between three kinds of normalization which can be

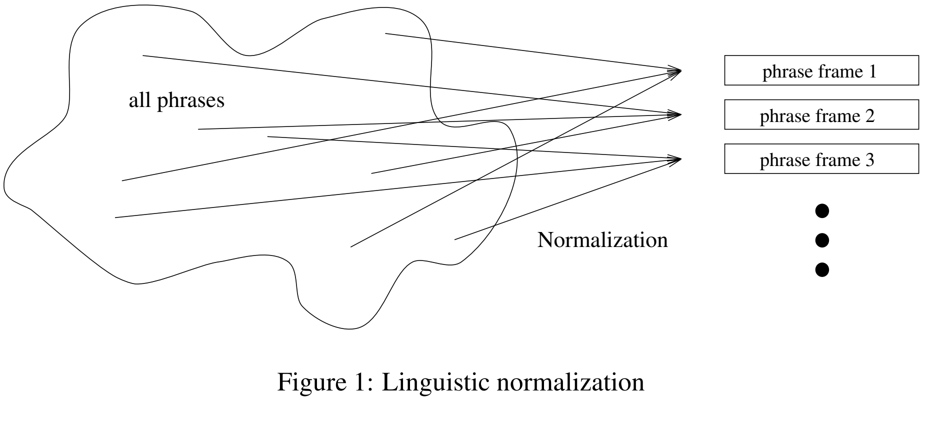

Figure 1: Linguistic normalization

seen as recall enhancement techniques when phrases are used for indexing. These are *syntactic*, *morphological* and *lexico-semantical normalization*.

Phrase frames, by their definition, incorporate the notion of syntactic normalization, that is the mapping of semantically equivalent but syntactically different phrases onto one phrase-class representative, the phrase frame. For instance, both *retrieval of information* and *information retrieval* are mapped to `[retrieval,information]`.

Morphological normalization is applied by means of *lemmatization* to account for morphological variants of the keywords. Verb forms are reduced to the infinitive, inflected forms of nouns to the nominative singular, comparative and superlative of gradable adjectives to the absolute.

Lexico-semantical normalization matches different phrases which are semantically (almost) equivalent by exploring certain relations that can be found between the meaning of individual words, like synonymy, hyponymy, meronymy, etc. This normalization may be implemented either by means of lexico-semantical clustering, or by incorporating in the matching function of phrase frames a semantical similarity distance function between words (*fuzzy matching*).

Parts of this linguistically-motivated indexing model are still under investigation and development [9]. Consequently, to perform these experiments we had to make some assumptions and give some quick solutions which we describe next.

2.1 Current Implementation

In order to simplify the structural matching of phrases, and also to raise recall, we currently follow the strategy of *unnesting* all complicated phrase frames [9]. A composed term like $[a,[b,c]]$ is decomposed into two frames $[b,c]$ and $[a,b]$ using b as an abstraction for $[b,c]$. When this decomposition is applied recursively, it results in *binary terms* (*BT*'s). As an example, consider the sentences

A student visits a conference on software engineering.
The software engineering conference is visited by some students.

from which, due to syntactical and morphological normalization, the same two frames are initially constructed for both sentences:

$$BT_1 = [\texttt{student}, \texttt{visit}],\ PF_1 = [\texttt{visit}, [\texttt{conference}, [\texttt{engineering}, \texttt{software}]]].$$

PF_1 is further unnested to

$$BT_2 = [\texttt{visit}, \texttt{conference}],\ BT_3 = [\texttt{conference}, \texttt{engineering}]$$

$$\text{and } BT_4 = [\texttt{engineering}, \texttt{software}]$$

Of course the unnesting makes it all the more important that a syntactical analyzer should be able to deduce the right dependency structure in complicated phrases.

In the current phase of our experimentation, phrase frames are constructed only from noun phrases, taking into account only prepositional phrase (PP) post-modifiers of nouns starting with the preposition *of*. These PP's are more likely to modify the preceding noun than others for which the PP-attachment problem has to be solved. However, we were able to disambiguate the modification structure of complicated noun phrases by applying statistical methods (described in section 4.5). We did not yet apply any lexico-semantical normalization.

3 Representational Choices

The different indexing sets we experimented with are summarized below. The acronyms will be used to refer to these choices in the rest of the article.

w (**w**ords): All word-forms found in the text.

Sw (**S**temmed **w**ords): All word-forms stemmed by a Porter stemmer. This a traditional indexing scheme and serves as the baseline in order to compare the effectiveness of the rest of indexing schemes.

Lw (**L**emmatized **w**ords): The same as **w**, except that all word-forms are lemmatized with respect to their POS category. In all the following choices, lemmatization is applied as standard.

Of course, for all **w, Sw** and **Lw** we eliminate words of low indexing value by using a *POS stop-list* (see section 4.5).

Ln (**L**emmatized **n**ouns): Nouns and proper nouns are well-known to be important in retrieval. What happens if we omit all other keywords?

Lnj (**L**emmatized **n**ouns and ad**j**ectives): The combined effect of using the union of nouns and adjectives is investigated in this experiment. These two categories cover most of the words occurring in noun phrases.

Lnv (**L**emmatized **n**ouns and **v**erbs): We investigate the combined effect of using the union of nouns and verbs.

Lnjv (**L**emmatized **n**ouns, ad**j**ectives and **v**erbs): This experiment serves as an indication of what might happen if we include to the indexing language only linguistic entities which are extracted from noun or verb phrases. Moreover, the impact of using adverbs for indexing can be measured indirectly by comparing **Lnjv** with **Lw**, since the indexing set **Lnjv** can be constructed from **Lw** by removing the adverbs.

Lap (**L**emmatized **a**djacent word-**p**airs, extracted from NPs): These word-pairs consist of the nouns and adjectives of **Lnj**, associated to form 2-word phrases by using the adjacency criterion. The hypothesis for this experiment is that adjacent words can be considered semantically related because of their proximity and be taken as one term. We use an extended notion of adjacency by accepting non-adjacent words as adjacent if the in-between words belong to certain POS categories (e.g. determiner, article, or preposition). For instance, the phrase *pollution of the air* gives the adjacent pair `pollution_air`.

This is an important experiment because in comparison to **Lbt** (described next) should measure the effect of syntactical normalization on performance.

Lbt (**L**emmatized **b**inary **t**erms (**Lbt**, extracted from NPs): These binary terms consist of the nouns and adjectives of **Lnj**, associated to form 2-word phrases by using the term modification criterion, i.e. head-modifier pairs. The head-modifier pairs are computationally more expensive than adjacent pairs since syntactical normalization is required, however, binary terms are syntactically canonical, e.g. both phrases *air pollution* and *pollution of the air* are mapped onto the same head-modifier pair, `[pollution,air]`.

4 Experimental Setup

Our main concern is to evaluate different indexing schemes. Document classification, categorization, or routing environments provide a good test-bed for such evaluations. In such environments, given a pre-classified corpus, the measurement of recall is straightforward, while for classical retrieval systems it involves expensive human judgments.

The experimental system is based on the vector space model, terms are weighted in a *tf*× *idf* fashion, and classifiers are constructed automatically using Rocchio's (original) relevance feedback method. We are aware that the choices of retrieval model, weighting scheme, and classifier construction method may have a fairly significant impact on performance, but since we are more interested in the *comparative* performance of the different indexing schemes we settle for widely accepted and proven methods.

The same goes for the evaluation measures. Instead of making a binary decision to either assign or not a document to a class, we allow the system to return a traditional *ranked list* of documents for every class: most relevant first, least relevant last. Thus, evaluation is done with 11-point interpolated recall-precision and average precision on a dataset from the Reuters-21578 text categorization test collection. Next we discuss in more detail the methods used, the dataset and pre-processing applied to it.

4.1 The Vector Space Model

In the Vector Space Model [15] documents are represented as vectors of weights:

$$D_i =< d_{i1}, d_{i2}, \cdots, d_{ik}, \cdots, d_{iN} > \quad (1)$$

where d_{ik} is the weight of the kth indexing term in the ith document, and N is a the number of indexing terms being used. Indexing terms are assumed to be *stochastically independent*. An indexing term may be a word, phrase, n-gram, or some other linguistic entity. The weight of a term for a particular document is a function of the number of times the term occurs in that document, the number of documents the term occurs in, and other information. Of the variety of weighting methods possible, we chose the Cornell *ltc* weighting commonly used in text retrieval [4]:

$$d_{ik} = \frac{tf_{ik} \times \log(N_D/n_k)}{\sqrt{\sum_{j=1}^{N} (tf_{ij} \times \log(N_D/n_j))^2}} \tag{2}$$

where N_D is the number of documents, n_k is the number of documents in which term k appears, and tf_{ik} is:

$$tf_{ik} = \begin{cases} 0 & \text{if } f_{ik} = 0 \\ \log(f_{ik}) + 1 & \text{otherwise.} \end{cases} \tag{3}$$

where f_{ik} is the number of occurrences of term k in document i.

Classification queries (or classifiers) are represented in the same manner, e.g. for a topic t

$$Q_t =< q_{t1}, q_{t2}, \cdots, q_{tk}, \cdots, q_{tN} > \tag{4}$$

is the corresponding vector (using the same set of terms as for the document vectors).

To compute the similarity between a query Q_t and a document D_i we used the dot product formula

$$S(D_i, Q_t) = D_i * Q_t = \sum_{k=1}^{N} d_{ik} * q_{tk}. \tag{5}$$

4.2 Classifier Construction

Classifiers were constructed automatically by applying Rocchio's relevance feedback method [13] on a pre-classified set of documents (training set). Rocchio's algorithm is a well-known algorithm in the IR community, traditionally used for relevance feedback. Classifiers based on Rocchio have proven to be quite effective in filtering [16] and classification [12, 8] tasks.

When training documents are to be ranked for a topic, an *ideal* classifier should rank all relevant documents above the non-relevant ones. However, such an ideal classifier might just not exist, therefore, we settle for a classifier that maximizes the difference between the average score of relevant and the average score of non-relevant documents.

If the similarity between topics and documents is calculated by the cosine measure (equation 5), and document vectors are weighted and length normalized such as $|D_i| = 1, \forall i$, then Rocchio specifies that the *optimal* classification query Q_t for topic t should have term k weighted as

$$q_{tk} = \frac{1}{|R_t|} \sum_{i \in R_t} d_{ik} - \frac{1}{|N_t|} \sum_{i \in N_t} d_{ik}, \tag{6}$$

where R_t and N_t are respectively the sets of relevant and non-relevant to t training documents, and $|.|$ denotes the number of elements in a set. After training, negative weights are usually set to zero.

4.3 Term Selection

Term selection (also called *feature selection* or *feature reduction* in classification) is an important task. Training data usually contain too many terms, thus it is not uncommon to end up with thousands of terms in the indexing vocabulary. Applying feature reduction techniques to text classification tasks was found not to impair classification accuracy, even for reductions up to a factor of ten [20, 12]. This is also economical in time and space.

The answer to the question of how many terms are sufficient for a topic representation is rather topic dependent and should be determined experimentally. However, assuming that relevant documents are likely to look more like each other (even in their lengths) than non-relevant documents do, a sensible number of terms would be a number proportional to the expected average number or unique terms in relevant documents:

$$N = c \times \textit{average number of unique terms in relevant documents.} \tag{7}$$

The constant c is topic dependent.

The average number of unique terms in relevant documents was calculated on the training data. The value of the constant c was estimated experimentally for our dataset. After a few tuning experiments, we set $c = 0.4$. Only the terms with the top N (equation 7) Rocchio weights were selected for every classifier and the rest were removed. This resulted in a great reduction in the number of terms without significant drop in performance. It should be noted that c was tuned for **Sw**, and assumed to hold for all other experiments as well, suggesting that the **Sw** experiment may be favored over the rest.

4.4 The Dataset

We evaluated on a dataset from the Reuters-21578 (distribution 1.0) text categorization test collection, a resource freely available for research in Information Retrieval, Machine Learning, and other corpus-based research[1].

We produced the Modified Apte (*ModApte*) split (training set: 9,603 documents, test set: 3,299 documents, unused: 8,676 documents). The ModApte split is a subset of the Reuters documents which are about economic topics, such as *income*, *gold* and *money-supply*. [7] discuss some examples of the policies (not always obvious) used by the human indexers in deciding whether a document belongs to a particular topic category.

Because Rocchio's algorithm requires each training document to have at least one topic, we further screened out the training documents with no topics category. Of course, we did not remove any of the 3,299 test documents, since that would have made our results incomparable with other studies. Documents can have assigned more than one topic, i.e. *multi-classification*. We used only the topics which have at least one relevant training document and at least one relevant test document (90 topics in total).

Since there is a large variation in the numbers of relevant training documents for topics, we evaluate separately on small and large topics. As small topics are considered the ones which have ten or less training documents (32 in total), and the rest (58 in total) are considered as large.

4.5 Pre-processing

In order to obtain for every experiment the appropriate indexing terms from the dataset, we applied some pre-processing. The pre-processing was performed in six steps:

1. **Tokenization** (script written in PERL): Detection of sentence boundaries followed by division of sentences into words.

2. **Part of speech tagging**: Brill's rule-based tagger[2] [3] was employed to obtain POS information for the contents of the dataset. The tagger comes with a lexicon derived from both the Penn Treebank tagging of the Wall Street Journal (WSJ), and the Brown Corpus. Conveniently, the WSJ articles are, like the Reuters documents, about economic topics and this increased the reliability in tagging the Reuters corpus.

3. **Shallow parsing and term extraction** (script written in PERL): Syntactic pattern matching on the POS tags to extract noun phrases for the **Lap** and **Lbt** experiments. For the **Lbt** experiment, the extracted noun phrases were further syntactically normalized and unnested, while for the **Lap** they were just broken down to adjacent word-pairs. For the rest of the experiments, the corresponding terms were extracted based on the POS tags.

4. **POS stop-listing** (only for **w, Sw**, and **Lw**): It is well-known that the use of a stop-list improves the quality of an indexing set. Traditionally, a stop-list is constructed by taking a predetermined *list* of function words (articles, prepositions, etc.) Since our approach is based on a POS tagger, we used a *POS stop-list* to remove all words belonging to the following categories: determiners (e.g. *a, the, all*), prepositions and subordinating conjunctions (e.g. *in, to, of*), pronouns (e.g. *I, yours*), the infinite marker *to*, modal verbs (e.g. *would, must*), the verbs *to be* and *to have* and coordinating conjunctions (e.g. *and*).

[1] For more information, the collection and its documentation is available from: `http://www.research.att.com/~lewis/`.

[2] Eric Brill's tagger V1.14 and a description are available by anonymous ftp from: `ftp://ftp.cs.jhu.edu/pub/brill` in the `Programs` and `Papers` directories.

5. **Disambiguation of the NP structure** (only for **Lbt**, PERL script): Noun phrases with more than two words can be structurally ambiguous. To disambiguate the modification structure we applied statistical methods. First we collected frequency information from the corpus for all noun phrases with two words. Then all 3-word noun phrases were disambiguated by assigning to them the most possible structure based on the frequencies of 2-word noun phrases. Gradually, this was applied up to n-word noun phrases based on the frequencies of all previously disambiguated k-word noun phrases ($k < n$). Where not enough frequency information had been available, *left-dependence* was assigned since it is the most probable modification structure in the English noun phrase.

6. **Morphological Normalization**: Lemmatization was performed according to the POS information by using WORDNET's v1.6 [11] morphology library functions[3].

   For **Sw**, words were stemmed using the Porter stemmer of the `Lingua::Stem` (version 0.30) extension to PERL.

5 Experimental Results and Discussion

Table 1 summarizes the average precision results of all experiments and their percentage change with respect to the baseline of **Sw** the traditional indexing approach.

	small topics		large topics	
run	av. prec.	change	av. prec.	change
w	0.525	-2.2%	0.696	+0.4%
Sw	0.537	baseline	0.693	baseline
Lw	0.547	+1.9%	0.693	0.0%
Ln	0.559	+4.1%	0.678	-2.2%
Lnj	0.563	+4.8%	0.695	+0.3%
Lnv	0.540	+0.5%	0.683	-1.4%
Lnjv	0.548	+2.0%	0.694	+0.1%
Lnj+Lap	0.633	+17.9%	0.730	+5.3%
Lnj+Lbt	0.620	+15.4%	0.732	+5.6%

Table 1: Average precision results

5.1 Stemming vs. Lemmatization

The experiments with unstemmed, stemmed and lemmatized words (**w, Sw** and **Lw**) as index terms showed no significant differences in average precision ($< 5.0\%$). That was not expected, since it is well-known that stemming improves performance in retrieval environments. However, this does not seem to be the case in classification environments. Classifiers can been seen as long queries. While retrieval queries contain usually 2-3 keywords, the average length of our classifiers for these experiments were 28.9, 26.1, and 26.1 keywords respectively. An automated method for building classifiers like Rocchio's, given sufficient training data, will identify and include all potential morphological variants of significant keywords into a classifier. That makes any form of morphological normalization in such environments redundant. Nevertheless, when no sufficient training data are available (like for the small topics), differences in performance grow larger. In this case, lemmatization is slightly better than stemming which is slightly better than no stemming at all.

The results suggest that for short queries (like in text retrieval), or for insufficient training data (like at the beginning of a text filtering task), morphological normalization will be useful, and lemmatization will be more beneficial for

[3] Specifically, we called the `morphstr()` function which tries to find the base-form (lemma) of a word or collocation, given its part-of-speech. WORDNET is created by Cognitive Science Laboratory, Princeton University, 221 Nassau St. , Princeton, NJ 08542. It is available for anonymous ftp from `clarity.Princeton.edu` and `ftp.ims.uni-Stuttgart.de`.

effectiveness than stemming since it is less error-prone. For long and precise queries (like classification queries derived from sufficient training data), morphological normalization has no significant impact on effectiveness. In any case, morphological normalization reduces the number of terms an information seeking system has to deal with, so it can always be used as a feature reduction mechanism.

5.2 Part-Of-Speech-based Indexing

The experiments based on indexing sets derived from combinations of part-of-speech categories (**Ln, Lnj, Lnv**, and **Lnjv**) presented, as well, no significant improvements over the baseline of stemmed words. Of course, all these experiments included, at least, the category of nouns. When we tried to exclude nouns, performance degraded greatly, confirming the importance of nouns for indexing.

If we were allowed to draw a weak conclusion from these results, we could have said that the union of nouns and adjectives (**Lnj**) performs best, while the addition of verbs reduces performance, and adverbs do not make a difference (we should remind that the difference between the indexing sets **Lnjv** and **Lw** is that the former does not include adverbs). The poor performance of verbs may be related to a limited or poor usage of them in the Reuters data, or to some bad interaction between between nouns and verbs. A confusion between nouns and verb arises from the fact most nouns can be verbed (e.g. *verb* $\rightarrow$ *verbed*) and verbs can be nominalized (e.g. *to visit* $\rightarrow$ *a visit*). This issue requires a further investigation.

Despite the non-significant differences in average precision, part-of-speech information may be used to assist term selection mechanisms, like morphological normalization may do. Table 2 gives a comparison of the number of distinct terms our system had to deal with in different experiments. It can be seen that the lemmatized union of nouns and

run	distinct terms	reduction
w	34030	baseline
Sw	27205	20.0%
Lw	29377	13.7%
Ln	23039	32.3%
Lnj	26952	20.8%
Lnv	24997	26.5%
Lnjv	28804	15.3%

Table 2: Distinct term occurrences

adjectives **Lnj** consists of 20.8% less indexing terms than the indexing set of all keywords **w**, while it preserves the effectiveness (it actually even improves it). Such a POS-based feature reduction mechanism has already been seen in [14] where nouns and adjectives were assumed to be most vital in representing document contents, but no comparative empirical evaluation was given.

5.3 Composite Indexing Terms

Since the best performance was presented by **Lnj**, we decided to add to this run composite terms in the form of adjacent pairs **Lnj+Lap**, or binary terms **Lnj+Lbt**.

Both experiments led to significant improvements ($> 5.0\%$) in average precision. Considering **Lnj** as the baseline, the improvement was 12.4% (small topics) and 5.0% (large topics) for adjacent pairs, and 10.1% (small topics) and 5.3% (large topics) for binary terms. Figure 2 gives the 11-point interpolated recall-precision curves. We did not use a special weighting scheme for composite terms. Composite terms were simply mixed up with single terms and weighted using the same *ltc* weighting formula (equation 2). This clearly violates the *term independence* assumption of the vector space model. In order to compensate for this, when single and composite (phrasal) terms are indexed together, composite terms are traditionally weighted lower [6], something we did not do. This suggests that there is margin for even better performance assuming a proper weighting scheme.

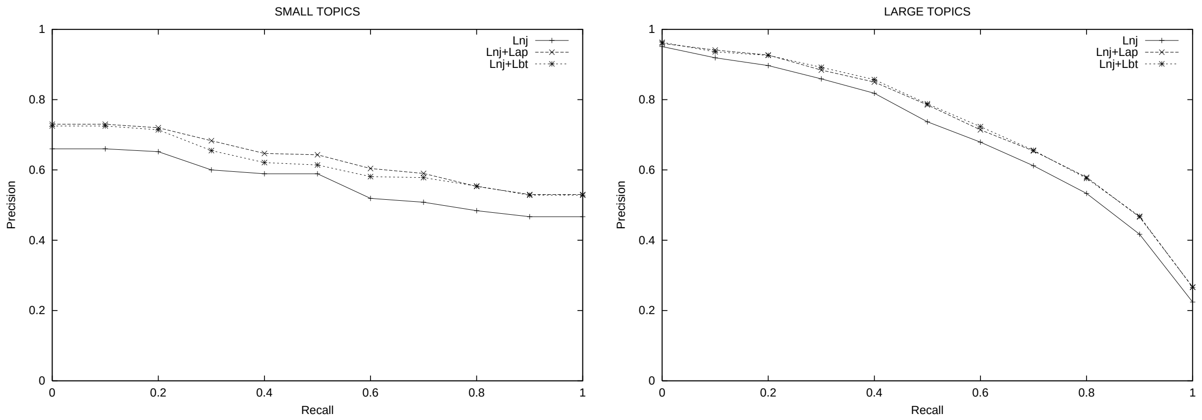

Figure 2: The impact of adding composite terms to the indexing set of nouns and adjectives

Unfortunately, binary terms did not prove more effective than adjacent pairs. That was unexpected, since the syntactically canonical nature of binary terms was thought to outperform word adjacency criteria. In a further investigation, first we measured how effective the syntactical normalization had been. Figure 3 (left) shows the comparative growth of binary terms and adjacent pairs as the dataset grows in documents. In the whole dataset, the total distinct

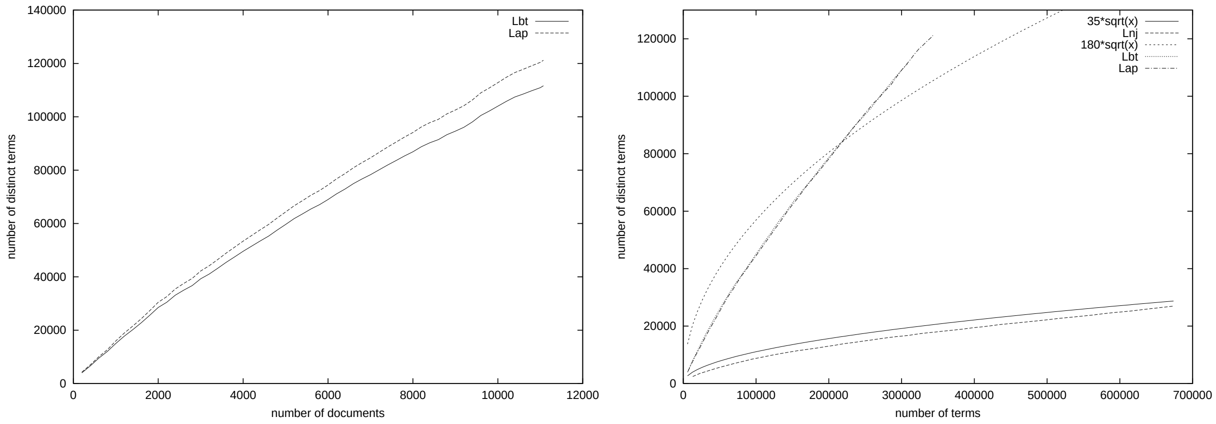

Figure 3: Number of distinct terms as a function of the growing dataset (left) and as a function of the total term occurrences (right). In the right figure, two square-root curves are shown for comparison purposes. The curves of **Lap** and **Lbt** are overlapping. Obviously, the growth of composite terms cannot be approximated with a square-root.

adjacent pairs were 121,185, while the binary terms were 111,631 (7.9% less). Clearly, our syntactical normalization had some effect, but not as extended as we expected.

How limited was the syntactical normalization is more clear in figure 3 (right). It is well-known that the number of distinct words in a growing document collection grows with the square root of the total number of word occurrences. It is obvious from this figure that this extends also to the subset of **Lnj** for our dataset. One could expect that the same holds for composite terms, but the number of such enriched terms grows even faster. We expected that the syntactically canonical nature of binary terms would have resulted to a less steep curve than this of adjacent pairs, but obviously it did not.

6 Conclusions and Directions for Further Research

Our experimental results using linguistically-motivated indexing terms suggest that part-of-speech information is beneficial to indexing. We found that a traditional keyword-based indexing set can be reduced to retain only its nouns and adjectives without hurting effectiveness, even slightly improving it.

Augmenting indexing sets with composite terms resulted in significant improvements in effectiveness for both adjacent pairs and head-modifier pairs. Nevertheless, head-modifier pairs have not proven better than adjacent pairs despite their syntactically canonical nature. The natural language processing techniques used were very limited, but the investigation suggests that using better linguistic tools would improve performance.

A comparison of lemmatization to stemming was not found to produce significant improvements, although lemmatization is considered less error-prone. In fact, both of these forms of morphological normalization were found not to improve significantly the effectiveness of information seeking environments characterized by relatively complete and accurate information needs, such as classification, categorization, or routing given sufficient training data. However, it still seems beneficial for incomplete and imprecise information needs, such as short retrieval queries or near the bootstrapping of filtering tasks. In any case, morphological normalization as much as part-of-speech information may be used to assist feature reduction techniques.

Our current research effort is aimed at several issues. We intend to, first, develop more extended syntactical normalization techniques, second, to replace the temporal solution of unnesting phrase frames with some kind of structural matching, third, to develop a proper weighting scheme for phrase frames, and fourth, to incorporate also lexico-semantical normalization. The overall goal is to break out of the traditional and long-survived bag-of-words paradigm. This goal may seem rather ambitious but not impossible.

References

[1] A. T. Arampatzis, T. Tsoris, C. H. A. Koster, and Th. P. van der Weide. Phrase-based Information Retrieval. *Information Processing & Management*, 34(6):693–707, December 1998.

[2] A. T. Arampatzis, Th. P. van der Weide, C. H. A. Koster, and P. van Bommel. Linguistically-motivated Information Retrieval. In *Encyclopedia of Library and Information Science*. Marcel Dekker, Inc., New York, Basel, 2000. To appear. Currently available on-line from `http://www.cs.kun.nl/~avgerino/encyclopTR.ps.Z`.

[3] E. Brill. Some advances in rule-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, Wa., 1994.

[4] C. Buckley, G. Salton, and J. Allan. The Effect of Adding Relevance Information in a Relevance Feedback Environment. In W. Bruce Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 292–300, Dublin, Ireland, June 1994. ACM Press.

[5] Joel L. Fagan. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and Non-Syntactic Methods*. PhD thesis, Department of Computer Science, Cornell University, Ithaca, New York 14853-7501, 1987.

[6] Norbert Fuhr, Ulrich Pfeifer, Christoph Bremkamp, and Michael Pollmann. Probalistic Learning Approaches for Indexing and Retrieval with the TREC-2 Collection. In Donna K. Harman, editor, *The Second Text REtrieval Conference (TREC-2)*, pages 519–526, Gaithersburg, Maryland, August 31 – September 2 1993. Department of Commerce, National Institute of Standards and Technology (NIST) Special Publication 500-215.

[7] Philip J. Hayes and Steven P. Weinstein. CONSTRUE/TIS: a system for content-based indexing of a database of news stories. In *Second Annual Conference on Innovative Applications of Artificial Intelligence*, 1990.

[8] D. J. Ittner, D. D. Lewis, and D. D. Ahn. Text Categorization of Low Quality Images. In *Symposium on Document Analysis and Information Retrieval*, pages 301–315, Las Vegas, NV, 1995. ISRI; University of Nevada.

[9] C. H. A. Koster, C. Derksen, D. van de Ende, and J. Potjer. Normalization and Matching in the DORO System. In *Proceedings of the 21st BCS-IRSG Colloquium on Information Retrieval*, 1999.

[10] D. P. Metzler and S. W. Haas. The Constituent Object Parser: Syntactic Structure Matching for Information Retrieval. *ACM Transactions on Informations Systems*, 7(3):292–316, 1989.

[11] G. A. Miller. WORDNET: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[12] Hein Ragas and C. H. A. Koster. Four Text Classification Algorithms Compared on a Dutch Corpus. In W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 369–370, Melbourne, Australia, August 1998. ACM Press, New York.

[13] J. J. Rocchio. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System — Experiments in Automatic Document Processing*, pages 313–323, Englewood Cliffs, NJ, 1971. Prentice Hall, Inc.

[14] Stefan M. Rüger. Feature Reduction for Information Retrieval. In Ellen M. Voorhees and Donna K. Harman, editors, *The Seventh Text REtrieval Conference (TREC-7)*, pages 409–412, Gaithersburg, Maryland, November 9–11 1998. Department of Commerce, National Institute of Standards and Technology (NIST) Special Publication 500-242.

[15] Gerard Salton. A Vector Space Model for Information Retrieval. *Communications of the ACM*, 18(11):613–620, November 1975.

[16] R. E. Schapire, Y. Singer, and A. Singhal. Boosting and Rocchio Applied to Text Filtering. In W. Bruce Croft, Alistair Moffat, C. J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 215–223, Melbourne, Australia, August 1998. ACM Press, New York.

[17] P. Sheridan and Alan F. Smeaton. The Application of Morpho-syntactic Language Processing to Effective Phrase Matching. *Information Processing & Management*, 28(3):349–369, 1992.

[18] A. F. Smeaton, R. O'Donnell, and F. Kelledy. Indexing Structures Derived from Syntax in TREC-3: System Description. In Donna K. Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, pages 55–67, Gaithersburg, Md. 20899, 1994. Department of Commerce, National Institute of Standards and Technology (NIST) Special Publication 500-225.

[19] T. Strzalkowski and J. P. Carballo. Natural Language Information Retrieval: TREC-4 Report. In Donna K. Harman, editor, *The Fourth Text REtrieval Conference (TREC-4)*, pages 245–258, Gaithersburg, Maryland, 1995. Department of Commerce, National Institute of Standards and Technology (NIST) Special Publication 500-236.

[20] Y. Yang and J. Pederson. Feature Selection in Statistical Learning of Text Categorization. In R. Engels, B. Evans, J. Herrmann, and F. Verdenius, editors, *International Conference on Machine Learning '97 (ICML 97)*, pages 412–420, Vanderbilt University, Nashville, TN, July 1997.

[21] C. Zhai, X. Tong, N. M. Frayling, and D. A. Evans. Evaluation of Syntactic Phrase Indexing — CLARIT NLP Track Report. In Donna K. Harman and Ellen M. Voorhees, editors, *The Fifth Text REtrieval Conference (TREC-5)*, pages 347–358, Gaithersburg, Md. 20899, 1996. Department of Commerce, National Institute of Standards and Technology (NIST) Special Publication 500-238.