

Normalization and matching in the DORO system

C.H.A. Koster
Department of Computer Science,
University of Nijmegen, The Netherlands
kees@cs.kun.nl

Submitted to IRSG99

Abstract

This paper is concerned with the use of linguistically motivated phrases as indexing terms in Information Retrieval applications. Apart from the conventional noun phrases, we propose to use verb phrases as index terms for text classification. Techniques for phrase matching through syntactic normalization and semantical matching are described. In particular, we show how to perform syntactic normalization of phrases in order to enhance recall. Semantical normalization is based on lexico-semantical relations, taking into account certain properties of the classification algorithms used.

The ideas described here are being implemented in the Document Routing system DORO, in which statistical learning algorithms are applied to document profiles consisting of phrases. This paper describes the rationale behind work in progress, rather than presenting final results.

1 Introduction

In the Information Retrieval community there is a long standing debate concerning the value of linguistic techniques. Although the use of (linguistically motivated) noun phrases as indexing terms is now commonly accepted, practical Information Retrieval systems using phrases like the CLARIT system [Evans et al., 1993] do not appear to perform consistently better than those based on keywords. There is a widespread conviction that the value of NLP to IR is dubious, even among people who tried hard to make it work [Smeaton, 1997].

In this paper we describe the DORO project, and the techniques developed for the extraction, normalization and matching of phrases. One of the aims of the DORO project is to show that the use of phrases is advantageous in the routing situation, where documents are to be classified on the basis of a superficial analysis of their contents.

The routing (or filtering) situation presents a good setting for the experimentation with linguistic techniques in IR, since the vexing problem of computing Recall is much easier to solve than in the classical re-

trieval situation, using the (preclassified) documents needed for training and testing the classification algorithms. The number of documents needed is much smaller than the number of documents in a realistic database, the number of classes much smaller than the number of possible queries in an IR system, and the classes are known in advance. Furthermore the classification experiments are completely automatic, without human involvement. They can be easily repeated with slightly different parameters. Therefore the routing problem makes it simple to experiment with the effect of other document representations than the traditional keywords.

1.1 Shallow and deep linguistics

A distinction is often made between “shallow” and “deep” linguistic techniques employed in IR. By a shallow technique is meant one which needs few linguistic resources and little linguistic knowledge in order to be used. Many shallow techniques are routinely used in IR:

- *stemming* [Porter, 1980] in order to eliminate morphological variation and thus enhance Recall
- considering function words (closed syntactical categories) as *stop words*
- *expansion* of a query with (semantically) related words.

The most advanced shallow techniques used are noun phrase pickers and Part-Of-Speech taggers. Typically, the resources for shallow techniques can be obtained by a cry for help on a suitable mailing list.

On the other hand, linguistics offers many more concepts and ideas which could be applied, but which are harder to put into a tool and to use without extensive linguistic knowledge. Deriving, representing and searching the meaning of documents is still very far away. But, surprisingly, even a reliable parser for English with a large lexicon, high coverage and acceptable speed is not available in the public domain. It is hard to develop such a parser (outside of limited domains),

not easy to make it work in an IR context and even harder to maintain and extend it.

The very success of the shallower linguistic techniques in conjunction with statistical techniques (see e.g. [Sparck Jones, 1998]) is a barrier for the use of deep techniques in IR: it is hard to justify the development costs.

In the DORO system, a complete syntactical analysis is made of the documents, in order to extract phrases. This requires the development of grammars and lexica for a number of languages and a parser generator system generating efficient and robust parsers. Given the present state-of-the-art in IR, this is deep linguistics.

1.2 The KeyPhrase hypothesis

Most of the commercial Information Retrieval systems are based, explicitly or implicitly, on what we shall call the *KeyWord hypothesis*:

A document is a bag of words.
A query is a bag of words.
The more words the query and the document have in common, the more the document is *about* the query.

Our work in the application of NLP techniques to the classification of documents is based on the *KeyPhrase hypothesis* (see [Arampatzis et al., 1998]):

A document is a bag of phrases.
A query is a bag of phrases.
The more phrases the query and the document have in common, the more the document is *about* the query.

The predictive value of single keywords has its limits. Experiments by others (e.g. [Cohen and Singer, 1996]) have shown that by using combinations of keywords rather than single keywords a better classification may be achieved. This observation has prompted us to take linguistically meaningful phrases as terms: the Noun Phrase (NP) including its modifiers and the Verb Phrase (VP) including its complements.

The Noun Phrase describes a complicated concept, in the form of a head with modifiers. Whenever the head alone is not precise enough, it is modified by one or more adjectives, nouns or preposition phrases. Similarly the Verb Phrase describes a situation or process by relating a main verb to a number of NPs and other phrases (like a small semantical network).

Our own experiments (Arampatzis, to be published) have shown that:

1. using only NP heads and main verbs as terms, the error rate of classification is somewhat worse than when using all keywords. We concluded that the modifiers are essential to achieve precision.
2. using complete NP's and VP's as terms, the error rate of classification is also worse, and learning is

very slow. We concluded that the recall is too low, because the probability of a phrase reoccurring literally is too low.

In order to achieve precision, we use complete NP's and VP's as terms. However, in order to retain recall we introduce a number of normalizations and a matching technique, as described in the following sections.

1.3 The DORO project

A problem faced by a growing number of companies in the Information Age is the routing of human-readable documents which are in electronic form. Such documents can either be received in the form of Electronic messages, or they can be the result of OCR-processing of paper mail and documents. How can it be assured that they arrive at the proper destination within the company with the smallest possible time delay?

For properly addressed mail this is just a matter of logistics, but for generically addressed mail and for documents without specific addressing information, this involves at present some form of human intervention: reading the document, interpreting its contents, and forwarding it to the person or department that should deal with it. Manual mail routing is error prone and time consuming. How can it be automated?

The goal of the DOCUMENT ROUTING project DORO (Esprit 22716) is to develop a system performing the automatic routing of human-readable documents in electronic form on the basis of a superficial analysis of the contents of the documents and knowledge of the characteristics of the possible destinees. An overview of the system is shown in Fig. 1. Incoming document images are first separated into forms (which are sent directly to the Work Flow system) and other (typewritten or printed) documents, in particular letters. These documents are OCR converted, classified by the Linguistic Classification system LCS, Workflow Relevant Data is extracted and the result passed to the Work Flow system. Documents found to be misclassified are returned through the Work Flow system, and used to adjust the classifiers.

The main characteristics of the IR approach taken in DORO are:

- rather than with (short) queries, the system deals with *profiles* of documents and document classes
- profiles consist of linguistically extracted *phrases* rather than keywords
- both *noun phrases* and *verb phrases* are extracted
- *syntactic normalization* is introduced to enhance precision and recall
- *frames* are produced from the phrases by *syntax-directed transduction* techniques

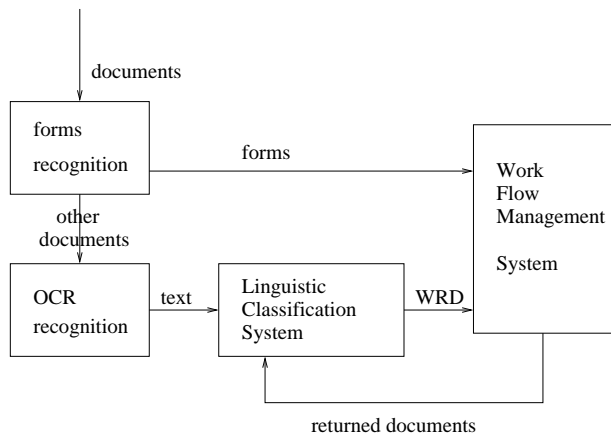


Figure 1: Overview of the DORO system

- *fuzzy matching* on the basis of semantical relations is used to enhance Recall
- *automatic learning* techniques are used for document classification.

The document routing system is intended to bridge the gap between the OCR scanning of documents and the Work Flow system of the company, enabling the fully automatic input and routing of free-format texts. More in general, it can be used to direct newsitems to interested readers (“narrowcasting”).

The DORO Consortium is composed of: industrial partners from the OCR and Document Flow areas, providing up to date technology in those areas and serving as integrators; academic partners providing the relevant technologies and resources in Natural Language Processing and Information Retrieval; and Users who face a growing document routing problem in their business.

Versions of the system are being developed for three Community languages (Dutch, Spanish and Greek) and three applications (a major Dutch insurance company, the largest Spanish department store and the Athens stock exchange).

1.4 Linguistic resources

In the course of the DORO project, grammars for three natural languages have been developed:

- A grammar of Dutch (a further development of the AMAZON grammar), in collaboration between the Computer Science Department of the University of Nijmegen and the Department of Language and Speech.
- A grammar of Spanish, developed by the group of Guillermo Rojo and Paula Santalla at the University of Santiago de Compostela
- A grammar of Modern Greek, developed by the group of Dora Noussia at the Computer Technology Institute in Patras.

These grammars have been written in the AGFL formalism (“Affix Grammars over a Finite Lattice”, see [Koster, 1992]), from which the AGFL system (see <http://www.cs.kun.nl/agfl/>) generates parsers. Furthermore, heavy use is made of the transduction mechanism of the AGFL system which allows the description of *compositional transductions* in the grammar: a translation is (recursively) defined for each rule, as a sequence of symbols composed out of the transductions of its elements and inserted symbols (e.g. markers).

For each language, an extensive general lexicon has been compiled, based for as far as possible on existing lexical material. Furthermore, for each language and application in the DORO project a domain lexicon has been developed, providing morphological, syntactical and semantical information about typical domain words and collocations.

2 Syntactic normalization

Natural languages allow the expression of one same thought or concept in very many different ways. In fact, we avoid literal repetitions of phrases, making use of *linguistic variation* for stylistic reasons.

By *linguistic normalisation* we mean the process of undoing linguistic variation, mapping different but equivalent formulations onto one same representative formulation. Linguistic variation comprizes

- *morphological variation*, traditionally normalized by stemming
- *syntactic variation*, e.g. the alternative use of phrases like *air pollution*, *polluted air* and *pollution of the air*
- *semantic variation*, by choosing related words (noxious gases) or periphrasis (dust-laden clouds)
- *anaphora*, (the pollution in a context relating to air), which is outside the linguistic scope of this project.

The goal of syntactic normalization is, for as far as this is possible without “deep” semantic analysis, to map syntactically different but semantically equivalent phrases onto one same frame.

In this section we briefly describe the syntactic normalization performed by the DORO system. A more comprehensive discussion of syntactic normalization including numerous examples can be found in [Arampatzis et al., 1998].

2.1 The noun phrase

Semantically, the noun phrase is a reference to or a description of a thing or a fact – a natural sub-

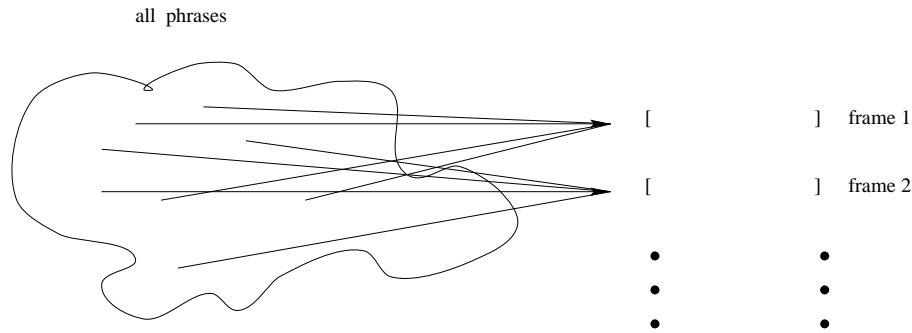


Figure 2: Syntactic normalisation

ject for search in a retrieval-based information system. Noun phrases are therefore an accepted term representation in a number of IR systems, and the basic idea of normalization can be found in e.g. [Strzalkowski et al. 1997] where it is called the conflation of noun phrases.

2.1.1 The structure of the Noun Phrase

A noun phrase is roughly that which can occur as the subject or object of a sentence, for instance:

- software engineering conference
- the air pollution in industrial activity zones
- a threshold, which should not be exceeded

Syntactically, a noun phrase may consist of at least the following elements:

1. it starts with zero or more *determiners*: an article (a or the), a quantifier (e.g. all or many), a number, a demonstrative or possessive pronoun (this, my), or certain combinations of these
2. it contains as its *head* a noun, a name or a personal pronoun; in the examples the heads are the nouns conference, pollution and threshold
3. the head may be preceded by some *pre-modifiers*: an adjective like industrial, or a noun, like air in air pollution or even another NP, like software engineering
4. the head may be followed by some *post-modifiers*: a post-adjective like general in director general (very rare in English), a *preposition phrase* (PP) like in industrial activity zones, or a *relative clause* like which should not be exceeded.

Notice that a noun phrase may recursively contain another, or even a (nearly) complete sentence, like in the last example. Therefore, an NP may be arbitrarily

complex. Furthermore, we have ignored the *coordination* between NP's or between their components by means of conjunctors like and and or.

2.1.2 Normalizing the Noun Phrase

Syntactic normalization of noun phrases is obviously dependent on the language to which it is applied. The following normalizing transformations come to mind:

- Elimination of determiners (articles, quantors and numbers).
- Elimination of case distinctions (very important for inflected languages like Greek).
- English (in particular its american variant) has an unnerving tendency to use nouns as premodifiers to avoid a postmodifier with a preposition (in particular of). For example, the phrases

air pollution
pollution of the air

have the same meaning and therefore one should be mapped on the other.

- Unnesting of embedded NP's, examples in section 3.4.
- Other transformations should allow a sensible treatment of coordinated elements. We shall not go into further detail.

All in all, the noun phrase presents only few opportunities for normalizing transformations.

2.2 The Verb Phrase

The verb phrase is the largest constituent of the (simple) sentence. It comprizes a verb form together with its complements. Semantically, a verb phrase can be seen as the description of a fact or event. It can describe something dynamic, in contrast to the noun

phrase which describes something static. Verb phrases should be terms for retrieval in their own right.

Furthermore, it is hard to find the boundaries of NP's without determining the VP structure:

- two NP's may follow one another

In software engineering conferences abound.

- in English many verbforms may be (mis)taken for a noun, or vice versa

Time flies like an arrow.

That is why Nounphrase Pickers can never be precise enough.

The robust and complete syntax analysis performed in the DORO system provides us with quite precise NP's and VP's (apart from the well-known problems with ambiguity and attachment, which require further progress in NLP).

2.2.1 The structure of the Verb Phrase

The *verb phrase* (*VP*) is particularly rich in linguistic variation, but the underlying structure of the VP is quite simple and elegant: it consists of a *verbal clause* which is an inflected form of some verb (the *kernel*) composed with other (auxiliary) verb forms and/or clitics (possibly even distributed over the sentence) accompanied by one or more *complement*, each fulfilling a certain role: the *subject* and (possibly) *object*, *indirect object*, *preposition complement*, etc. Complements can be realized by an NP (in the wide sense, including personal names, personal pronouns etc.) or by a PP. Some examples:

I	subject
would have liked to hit	verbal clause
the man	object
with a hammer.	prep. complement (instrumental)
my husband	subject
does not understand	verbal clause
me.	object
I	subject
owe	verbal clause
you	indirect object
twenty dollars.	object

2.2.2 Normalizing the verb phrase

For the verb phrase, many more normalizing transformations can be found which preserve meaning, or rather: which have a normalising effect and do not lose information which is obviously relevant for retrieval purposes.

1. elimination of irrelevant elements, like adverbs.

2. *depassivation*

Bringing passive VP's into active form

3. elimination of time and modality

The verbal clause of the verb phrase can be reduced to the infinitive of the kernel (with clitics). It may make sense to retain an indication of certain modalities (present or future, wish or fact).

4. *word-order normalization*

Eliminating topicalization and free word order by bringing the complements of each VP in some standard order, like

kernel subject [object] [indirect object] ...

5. *unnesting* of the complements, reducing each NP complement to its head plus a separate NP frame.

The result of all these transformations is a *VP frame*, an abstraction from the verb phrase suitable for retrieval purposes, with possibly a number of accompanying NP frames.

3 Phrases, frames and terms

The process of extracting terms from a document text consists of a number of steps, as shown very schematically in figure 3.

The *syntax analysis* transduces the text to a sequence of parse trees for the individual sentences, containing the phrases as subtrees. It uses a *parser* obtained from an AGFL grammar, a general lexicon and an (application specific) domain lexicon. It produces parses in an SGML format. The *transducer*, which is also described by an AGFL grammar, performs the syntactic normalization and transduces each phrase into a (possibly nested) *frame*. A frame is an abstraction from a phrase, as described in this section. The *unnester* (described in section 3.4) unfolds the nested frames to obtain the terms describing the document. Finally, the *matcher* (4) performs a limited form of semantical normalization and computes the similarity between terms (frames) occurring in the document and those occurring in the class profiles.

3.1 Frames

How should we represent phrases in order to use them as terms? A large number of possibilities offer themselves.

On the one hand a phrase can be represented as a parse tree. This provides certainly too much detail, most of which is redundant or irrelevant for retrieval. At the other extreme, it can be represented as a sequence or even set of words. This gives too little detail,

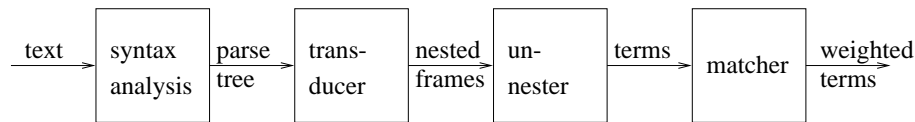


Figure 3: From documents to terms

essential information about the meaning is not represented. For example, it is no longer visible what noun is the head of the noun phrase. In fact we can take all kind of intermediate forms.

The actual production of frames is performed by a separate transducer (described by another grammar), which takes as input the trees produced by the parser (coded in SGML). This separates the issues of language description and frame production.

The question which parts of a phrase to express in the form of frames and how is left entirely to the grammar writer. The system merely provides a general notational and implementational framework. Experimentation will have to decide what is the best choice. In this section we will outline the general approach.

3.2 Headedness

According to the principle of headedness, any phrase has a single word as *head* (usually a noun or personal noun in the case of an NP, the main verb in case of a VP). The notion of head is common to most linguistic theories and by itself relatively language independent.

Phrases with the same head are related. The head gives the central concept of the phrase, and the other elements (modifiers in the NP, complements in the VP) serve to make it more precise. Conversely, the head may be used as an abstraction of the phrase (loosing precision but gaining recall).

3.3 Binary frames

Rather than introducing general tree structures as terms, we shall use *frames*, defined as pairs [head, modifier], where the head is a nonempty string (a lemmatized word) and the modifier is an (arbitrarily long) string of lemmatized words, which is empty in the case of a bare head. This last case may also be denoted by [head].

The head also serves as an index into a list of frames with frequencies etc., organised into groups with the same head like

```

engineering 1026
  , of software 77
  , reverse 102
  , software 842
  , ...
  
```

Note that the frequency of the head includes that of its modified occurrences.

The result of the transduction of a document is a sequence of terms, according to the following syntax:

```

terms: term; terms, term.
term: frame; symbol.
frame : "[", term, ",", terms "]"
       "[" , term, "]"
  
```

More about frames in the following section. Notice that frames may be nested (a frame may contain one or more others):

```

he visited a conference on software engineering
may be transduced to
[visit, [conference, [engineering, software]]]
  
```

3.4 Unnesting

In order to raise recall, we follow the strategy of *unnesting* all nested frames: a composed term like [c, [b, a]] will be decomposed into two phrases [b, a] and [c, b], using b as an abstraction for [b,a] in the second one. More in general, $\alpha[h, mods]\beta$ is replaced by $\alpha h \beta$ while introducing a separate term [h, mods]. Unnesting is applied recursively to the head and modifiers of a frame. In this way, a term containing frames is rewritten into a collection of terms without nesting.

```

As an example, the nested frame
[visit, [conference, [engineering, software]]]
is unnested to
  
```

```

{[visit, conference],
 [conference, engineering],
 [engineering, software]}
  
```

Another example: the famous phrase
the Hillary Clinton Health Bill proposal
resulting in a nested term

```

[[[proposal, [bill, health]], [clinton, hillary]]]
  
```

will be unnested to the collection of terms

```

{[bill, health],
 [clinton, hillary],
 [proposal, bill],
 [proposal, clinton]}
  
```

which does not include [bill, clinton].

The result is that from one phrase a (multi)set of frames without nesting is produced as terms, rather

than a single nested one. This strategy raises the recall and loses some precision. Of course the unnesting makes it all the more important that the parser should be able to deduce the right dependency structure in complicated phrases.

4 Matching

The matching process comprises the following tasks:

1. syntactic matching (partial matching of composed terms), and
2. semantical matching (exploiting synonymy and hypernymy relations).

In our matching strategy we exploit certain properties of the classification algorithms.

4.1 Relevant properties of the classification algorithms

The classification algorithms used in the project [Ragas and Koster, 1998] have the property that they compute a *linear classifier* in the sense that the similarity between a document profile and a class profile (each of which is given as a (multi)set of terms), is calculated as the sum of the weights given to those terms that are present. They perform best when faced with a low number of terms with high predictive value.

It is well known that, as the number of documents grows, the number of different keywords occurring in them grows approximately with the square root of the number of documents. The number of phrases however grows even faster than the number of words. This has negative consequences for the classification algorithms, negating the positive effect of having more precise terms:

- a large number of terms with hard to tract dependencies between them
- a low probability of any particular term reappearing literally in another document.

4.2 The matching strategy

The conventional approach to syntactic matching of phrases is to define some similarity function between phrases, based e.g. on subtree overlap. This function should then be used to weigh the phrases in the classification algorithms.

We have studied this approach, but found it hard to define a reasonable measure. Instead, we have introduced unnesting to obtain from a composed phrase only those terms which are considered relevant from a linguistic point of view. This may be more efficient and is certainly more tractable than subtree matching. Moreover, the experience collected with the classification algorithms has made clear that applying syntactic matching to composed terms and all their subterms is not a good strategy: One would have to combine in an ad-hoc fashion fractional contributions from very many terms. It is better to rely on the property of the algorithms, firmly founded in probability theory, of being able to make their own judgement as to the significance of individual terms. And how should one count the frequencies of partially matched terms?

Once a term has occurred an appropriate number of times in documents belonging to specific classes, the judgement of the classification algorithm can be trusted. The problem is how to deal with terms that have not occurred before, or have not yet achieved significance. It is in this “fringe” that similarity to other terms becomes significant.

Our strategy will be to apply semantical matching only to terms that are *insignificant* according to the classification algorithm, trying to find the most significant related term. This closest term is then used by the classification, but the original term is used by the learning algorithm. Thus, given enough documents the term may become significant later. A more precise formulation of this (literally) marginal strategy will be found in section 4.5.

4.3 Term significance

Intuitively, a term is *significant* if its presence or absence makes a difference in the outcome of the classification process. Due to the linear character of our classification algorithms, significance is a gradual scale: some terms will be more significant than others. What terms are significant, and to what degree, depends to a certain extent on the classification algorithm used, but in all cases a term that has not occurred in a training document will have no significance, and a term that occurred only once or in only one document will have very little. In fact, the proper treatment of insignificant terms presents a problem for most classification algorithms. It is preferable to have a small number of highly significant terms rather than a large number which are of doubtful significance.

In the Machine Learning community, where many

classification algorithms originate, a number of techniques have been invented to perform “feature selection”. In applying such techniques to text classification [Yang and Pederson, 1997] it was found that the number of features (in our case terms) could be reduced by a factor of ten without impairing classification accuracy, or even slightly improving it.

Our own experiments with two dynamic stoplist heuristics (Winnnow steps and a frequency heuristic) show similar results [Ragas and Koster, 1998]. Applying three winnow steps to the SBC algorithm raised in our main experiment the overall precision from 80% to 84%, while reducing the number of terms by a factor eight.

After training a classifier, it may be much reduced in length by eliminating all insignificant terms from it. This allows the classification in the production phase to be more economical in time and space. It is only during training that classifiers over the full set of terms are needed. The process of searching a classifier for semantical matching also is speeded up by using shorter classifiers.

4.4 Lexico-semantical relations

For the DORO project we have decided to adopt (Euro)WordNet [Vossen et al., 1998] as the general source of lexico-semantical data for semantical matching, using only synonymy and hyper/hyponymy (other relations are available, but their value is unclear). In WordNet, a word with its synonyms is collected into a *synset*. The notion of synonymy in principle means *substitutional equivalence in a certain context*, and synsets are distinguished according to context. Besides words, a synset may also contain a reference to a hypernymous word, indicated by @, and a reference to a hyponym, indicated by ~; from this word, again its synset can be taken. Thus, the WordNet datastructure can be characterized by three operations:

$$\begin{aligned} \text{synset} &: \text{word} \rightarrow \{ \text{word} \} \\ \text{hyper} &: \text{word} \rightarrow \text{word} \\ \text{hypo} &: \text{word} \rightarrow \text{word} \end{aligned}$$

The domain lexica developed within the project are structured similarly.

4.5 Semantical matching of terms

Our term matching strategy for terms which are not significant according to the class profiles is that of *selective expansion*.

By the *expansion* of a symbol we shall mean the symbol together with its synonyms and its *direct* hyper- and hyponyms.

$$\begin{aligned} \text{expansion} &: \text{symbol} \rightarrow \{ \text{symbol} \} \\ \text{expansion}(s) &= \end{aligned}$$

$$\begin{aligned} &\{s\} \cup \text{synset}(s) \\ &\cup \text{synset}(\text{hyper}(s)) \cup \text{synset}(\text{hypo}(s)) \end{aligned}$$

As an example, the (simplified) synset for tree

$$\{ \text{tree}, \sim\text{plant}, \text{conifer} \}$$

through the hyponym pointer ~plant points to the synset

$$\{ \text{plant}, \text{flora}, \text{organism}, @\text{tree} \}$$

(note the corresponding hypernym pointer to tree for plant). The expansion of tree gives the extended set

$$\{ \text{tree}, \text{plant}, \text{flora}, \text{organism}, \text{conifer} \}$$

By the expansion of a frame $[x, y]$ we mean the set

$$\begin{aligned} \text{expansion} &: \text{term} \rightarrow \text{term} \\ \text{expansion}([x, y]) &= \\ &\{ [t, u] \mid t \in \text{expansion}(x), u \in \text{expansion}(y) \} \end{aligned}$$

The semantical matching of a profile (a set of terms) consists in the replacement of all insignificant terms (symbols and frames) by the most significant term in their expansion.

$$\begin{aligned} \text{match} &: \text{symbol} \rightarrow \text{symbol} \\ \text{match}(\text{symbol}) &= \\ &\text{IF } \text{significant}(\text{symbol}) \\ &\text{THEN } \text{symbol} \\ &\text{ELSE SOME } s \in \text{expansion}(\text{symbol}) \mid \\ &\quad \text{significance}(s) \text{ is maximal} \\ &\text{FI} \\ \text{match} &: \text{term} \rightarrow \text{term} \\ \text{match}([x, y]) &= \\ &\text{IF } \text{significant}([x, y]) \\ &\text{THEN } [x, y] \\ &\text{ELSE SOME } t \in \text{expansion}([x, y]) \mid \\ &\quad \text{significance}(t) \text{ is maximal} \\ &\text{FI} \end{aligned}$$

The terms obtained by expansion should be weighted by some suitable constant λ (e.g. $\lambda = .9$ for synonyms and $\lambda = .5$ for hyper/hyponyms). Insignificant terms having no significant expansion are dropped.

The semantical matching technique described here is related to the traditional technique of “massive expansion” of queries, but with the difference that we do not extend the query with *all* nyms, but rather replace insignificant terms by their single most significant nym.

5 Conclusion

We have described a number of techniques, developed in the DORO system, for increasing the precision and

recall in the matching of terms in term-based learning systems, and the rationale behind them. The design decisions taken are as wellfounded as present evidence permits. There is no question that further research might come up with very different solutions, but the timescale of the project does not permit more speculative research. The time pressure of the project also explains why at the time of writing we cannot provide experimental results: implementing the system is hard enough, and the time for experimentation, evaluation and tuning comes later.

We have described the rationale behind our design decisions, leading to a conceptually simple and rather general system that leaves to its integrators a number of areas for adaptation to the particular language and application:

1. the grammar, with the choice of the constructs described and the choice of transduction to terms
2. the lexico-semantic relations expressed in the domain lexicon
3. certain parameters, like the value of λ .

Further experimentation will be necessary to achieve an optimal linguistic support to classification. For such experiments, the DORO system provides a good framework, both during and after the project.

References

- [Arampatzis et al., 1998] A.T. Arampatzis, T. Tsoiris, C.H.A. Koster and T.P. van der Weide (1988), Phrase-based Information Retrieval. *Information Processing & Management*, 34(6), pp. 693-707.
- [Cohen and Singer, 1996] W.W. Cohen and Y. Singer (1996), Context-sensitive methods for text categorization. In: Proceedings of SIGIR '96, pp. 307-315.
- [Dagan et al., 1997] I. Dagan, Y. Karov and D. Roth (1997), Mistake-Driven Learning in text Categorization. In: *Proceedings of the Second Conference on Empirical Methods in NLP*, pp. 55-63.
- [Evans et al., 1993] D.A. Evans, R.G. Lefferts, G. Grefenstette, S.H. Handerson, W.R. Hersch and A.A. Archbold (1993), CLARIT TREC design, experiments and results. In: TREC-1 proceedings, pp. 251-286.
- [Koster, 1992] Cornelis H.A. Koster (1992), Affix Grammars for Natural Languages. In H. Alblas and B. Melichar, editors, *Attribute Grammars, Applications and Systems*, volume 545 of *Springer Lecture Notes in Computer Science*, pp. 469-484.
- [Porter, 1980] M.F. Porter (1980), An algorithm for suffix stripping. *Program*, 14, pp. 130-137.
- [Ragas and Koster, 1998] H. Ragas and C.H.A. Koster (1998), Four classification algorithms compared on a Dutch corpus. In: Proceedings SIGIR 98, pp. 369-370.
- [van Rijsbergen, 1979] C.J. van Rijsbergen (1979), *Information Retrieval*. 1979, London, Butterworths.
- [Rocchio, 1971] J.J. Rocchio (1971). Relevance feedback information retrieval. In: Salton, G. (ed.) *The smart retrieval system—experiments in automatic document processing* p. 313-323. Prentice-Hall, Englewood Cliffs, NJ.
- [Smeaton, 1997] A.F. Smeaton (1997), Using NLP and NLP resources for Information Retrieval Tasks. In: T. Strzalkowski (Ed.), *Natural Language Information Retrieval*, Kluwer Academic Publishers.
- [Sparck Jones, 1998] K. Sparck Jones (1998), Information retrieval: how far will *really* simple methods take you? in: Proceedings TWTL 14, Twente University, the Netherlands, pp. 71-78.
- [Strzalkowski et al. 1997] T. Strzalkowski, F. Lin and J.P. Carballo (1997), Natural Language Information Retrieval: TREC-6 Report. To appear.
- [Vossen et al., 1998] P. Vossen (Ed.) (1998), EuroWordNet A Multilingual Database with Lexical Semantic Networks. Kluwer Academic publishers.
- [Yang and Pederson, 1997] Yiming Yang and Jan Pederson (1997), Feature selection in statistical learning of text categorization. In: ICML 97, pp. 412-420.