Reducing Information Variation in Text

Agata Savary[1] and Christian Jacquemin[2]

[1] LADL, IGM, Université Marne-la-Vallée, 5, bd Descartes, Champs-sur-Marne
77454 Marne-la-Vallée, France xsavary@free.fr
[2] LIMSI-CNRS, BP 133, 91403 Orsay, France jacquemin@limsi.fr,
WWW home page: http://www.limsi.fr/Individu/jacquemi/

Abstract. We discuss the nature and the scope of linguistic (morphological, syntactic and semantic) variation of terms and its impact on two information retrieval tasks: term acquisition and automatic indexing. A review of natural language processing techniques existing in these two areas is done, along with an in-depth presentation of FASTR, a corpus processor for the recognition, normalization, and acquisition of multiword terms.

1 Introduction

Because of the recent dramatic increase in the number of electronic documents, efficient retrieval of information from texts is a crucial issue. Terminological variation is one of the major obstacles for this task. Consider for example an automatic searching for documents that are relevant to a given subject. One may indicate keywords to be searched for, but the relevant documents may not match them precisely. For instance, while looking for texts concerning the *genetic disease* it is necessary to consider that all of the following variants are valid instances of the query: *genetic diseases* (inflectional variant), *disease is genetic* (syntactic variant), *hereditary disease* (semantic variant), *genetically determined forms of disease* (morphological variant), etc. Conversely, not every co-occurrence of the constituent words of a given term is relevant to this term, e.g. *genetic risk factors for coronary artery disease* is not a correct variant of *genetic disease*. Therefore, a straightforward matching of documents against keywords will result either in misses of relevant responses, if the matching is performed in a rigid way (i.e. by fixed phrases), or in an excess of irrelevant responses, if it is performed loosely (i.e. by a bag of words).

We will show how a compromise may be reached between rigid and loose keyword matching through natural language processing (NLP). We present a survey of existing term extraction tools, with a particular concern for their ability to handle term variation. One of them, FASTR[1] (Fast Term Recognizer), is a shallow parser dedicated to the recognition, normalization and acquisition of compound terms. For a given set of documents and an initial set of controlled

[1] FASTR can be downloaded from *http://www.limsi.fr/Individu/jacquemi/FASTR/* and freely used for research projects by noncommercial and academic institutions.

terms FASTR produces a set of linguistic links between text sequences and initial terms. The exceptional accuracy of the resulting term spotting supports the argument that access to full text documents does not require a complete understanding of their content.

FASTR is a unique combination of several NLP techniques: lexical analysis through large terminological and lexical resources, shallow parsing, novel transformational unification-based techniques, and optimization techniques for fast processing of large corpora. The design of FASTR was based on detailed observation of numerous types of term variations in French and English documents from various specialized domains. The in-depth description of most aspects of the present study is described in [1].

2 Term Variation

Contrary to the traditional view of terms as being fixed labels for well-defined concepts of a given technical sublanguage ([2]), terms are linguistic objects prone to orthographic, syntactic and denotational fluidity, from both diachronic ([3]) and synchronic ([4]) point of view. Therefore ignoring term variability in an automatic information-processing system may result in inability to relate conceptually close but linguistically different occurrences.

For the aim of *term normalization*, i.e. grouping together occurrences of the same multi-word term, we admit the following definition of term variation:

Definition 1. *A morphological, syntactic, or semantic variation is a transformation of a controlled multi-word term that satisfies the following three conditions:*

1. *All "content" words (i.e. words other than prepositions, determiners, etc.)*[2] *of the controlled term are preserved by the transformation or transformed into any of the 3 types of variants listed in point 2. For instance in French,* moniteur temps réel *(real time monitor) is a variant of* Moniteur en temps réel *(lit. monitor (Noun) in real time), which we will mark* moniteur temps réel → Moniteur en temps réel[3]*, but* cell recognition *is not a variant of* Neural cell recognition.
2. *Content words of the variant may be graphically modified, and morphologically or semantically related to those of the controlled term:*
   - *Variations that involve graphic variants of content words or omissions and insertions of word delimiters are called* graphic variations *(e.g.* behavioural model → Behavioral model, lookup → Look-up*)*[4].

[2] Sometimes, prepositional or adverbial particles may belong to a term's content words, as in *on-line process, parlimentary by-election*, etc. If these words are omitted we cannot consider that a variant is valid, (e.g. *elections to the parliament* is not a variant of *parlimentary by-election*,) but cases of this kind are difficult to detect in our model.

[3] By convention, the first word of controlled terms is written with a capitalized letter.

[4] Graphic variations are not accounted for in FASTR.

- *Variations that involve a morphological relationship of inflectional or derivational morphology are called* morphological variations *(e.g.* students union → Student union, image converter → Image conversion*).*
- *Variations that involve a semantic relationship are called* semantic variations *(e.g.* speech development → Language development*).*

3. *Words may be inserted or deleted and the order of words (or of their graphic, morphological or semantic variants) may be modified but the dependency relations existing between content words of the original term must be preserved in the variant (e.g.* genetic risk factors for coronary artery disease *is not a variant of* Genetic disease *because the syntactic dependency between* genetic *and* disease *is lost). Variations that involve such word insertions/deletions or word order modifications are called* syntactic variations *(e.g.* processing of cardiac image → Image processing*).*

   Condition 3 doesn't exclude variants obtained through left or right extensions of a controlled term, e.g. *arterial blood pressure fluctuations, pressure fluctuation diagram*, and *abnormal fluctuations in blood pressure* may all be considered correct variants of *Pressure fluctuation.* In the case of FASTR system, however, only the variants which satisfy the following additional condition are taken into consideration:

4. *The leftmost and the rightmost constituents of a variant must be content words of the original term (thus, the 3 above variants of* Pressure fluctuation *are not dealt with). In particular, the variant should not contain the original term.*

Two reasons account for the extra limitation given in condition 4. On the one hand, the identification of the correct frontiers of variants obtained through left or right extensions of a term cannot be performed reliably in our model. On the other hand, the identification of such frontiers is not our main goal since we essentially wish to be able to point at all text sequences relevant to a given controlled term[5]. Thus, in the example above all text sequences containing e.g. *abnormal fluctuations in blood pressure* will necessarily be identified if only we manage to recognize *fluctuations in blood pressure* as a variant of *Pressure fluctuation.*

Different types of variations may occur together in a variant, for example *disease is familial* and *transmissible neurogenerative diseases* are both syntactic and semantic variants of *Genetic disease.* They are called syntactico-semantic variants.

Variation, as defined above by points 1-3, is a crucial characteristic of terms in corpora. Variants represent approximately one third of the term occurrences in an English scientific corpus ([5]). This high percentage is due to the fact that terms in corpora are supposed to satisfy the communication criterion of *appropriateness* ([2, p. 106]), i.e. the compromise between two concurrent needs: *precision* (fulfilled by adding modifiers to a term if it is ambiguous in a given context) and *economy* (fulfilled by reusing exiting terms in new combinations to

[5] The recognition of left and right frontiers of extended terms may though be one of the aims of some term acquisition tools presented in Section 4.

name new concepts, and by using short variants of terms if their full forms can be deduced from the context). For that reason terms in corpora often deviate from their canonical forms found in term banks. Exhaustive listing and in-depth analysis of term variants being often unrealistic, an economical computational treatment of term variation is necessary to help overcome the gap between terms in corpora and in thesauri.

3 Term Extraction

Identification of terms in textual corpora, called term extraction[6], has several applications: automatic indexing, corpus-based terminology, computer assisted translation, machine translation, etc. Especially the two first of them are concerned in this study. Both are closely related and sometimes tools developed for automatic indexing are used for corpus-based terminology or vice versa, but the essential difference is in the fact that in the former application terms are the means of investigation while in the latter they are its purpose.

Both applications divide into two distinct subfields depending on whether initial terminological knowledge is available or not. The purpose of automatic indexing is to assign to documents terms capable of representing the content of these documents ([6]). If this task is performed with reference to a controlled vocabulary it is called *controlled indexing*, in the opposite case it is called *free indexing*. Likewise, the corpus-based terminology, whose purpose is to create terminological thesauri on the basis of term occurrences in corpora, can be either *thesaurus enrichment* if prior terminological knowledge is used, or *term acquisition* otherwise.

Table 1. Subdomains of term extraction

	Indexing	**Corpus-based terminology**
With initial data	Controlled indexing	Thesaurus enrichment
Without initial data	Free indexing	Term acquisition

Depending on whether we work with single-word terms or multi-word terms, the central issues in the design of a term extraction system are very different:

- Single-word terms are generally polysemous and call for word-sense disambiguation and context analysis.
- Multi-word terms are far less polysemous than single-word terms, but since they have a phrase structure, they are prone to variations. Their identification calls for morpho-syntactic analysers or statistical measures.

[6] The notion of "term extraction" is sometimes confused with "term acquisition", which we define below.

In this study we concentrate on multi-word terms. The next section contains a survey of existing concepts and techniques for multi-word term acquisition and for automatic multi-word indexing, called *phrase indexing*. In the following sections we describe FASTR, a term processor whose scope falls into the first line of Table 1.

4 State of the Art in Automatic Term Extraction

Different terminological, morphological and semantic resources, linguistic methods and computational tools contribute to the automatic extraction of terminology nowadays.

Controlled terms are contained in thesauri in which they are often organized in a hierarchical structure and accompanied by various kinds of information. For instance, AGROVOC, a multilingual thesaurus for agronomy ([7]), has a hierarchical structure. An entry, as the French one in Table 2, consists of a term, its linguistic variants and synonyms (marked *ep*), its generic terms (TG1, TG2,...), its specific terms (TS1), its associated terms (ta), and its equivalents in other languages.

Table 2. A French entry from AGROVOC

Code	Text	Gloss
	IMMUNISATION	[descriptor]
	(immunisation spécifique d'antigène)	[note of usage]
ep	*immunisation active*	[nondescriptor (synonym)]
ep	*immunisation croisée*	—
ep	*sensibilisation immune*	—
TG1	*immunostimulation*	[generic term (level +1)]
TG2	*immunothérapie*	[— (level +2)]
TG3	*thérapeutique*	[— (level +3)]
TG4	*contrôle de la maladie*	[— (level +4)]
TS1	*vaccination*	[specific term (level -1)]
ta	*antigène*	[associated term]
ta	*réponse immunitaire*	—
ta	*résistance aux maladies*	—
ta	*résistance induite*	—
En	*immunization*	[English equivalent]
Es	*immunización*	[Spanish equivalent]

The UMLS (Unified Medical Language System) is a hierarchical meta-thesaurus aimed at unifying medical terminological data from different sources ([8]).

It allows to assign several kinds of information to terms: lexical variants (e.g. *Atrial fibrillation* and *Atrial fibrillations*), synonyms (e.g. *Atrial fibrillation* and *Auricular fibrillation*), generic/specific relations (e.g. *Atrial fibrillation* **is_a** *Arrhythmia*), concept attributes (e.g. the semantic type *pathologic function* is an attribute of the concept *Atrial fibrillation*), etc. In addition, UMLS contains a semantic network (see Figure 1 for an extract), a map of information sources, and a lexicon which serves the attached NLP module for generating graphical and morphological variants of terms.

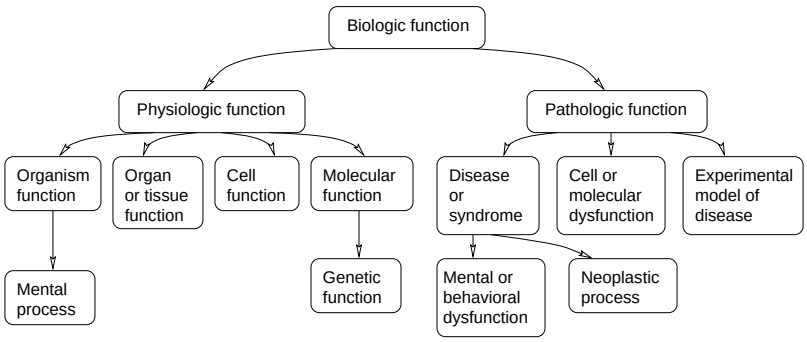

Fig. 1. Hierarchy of *biological functions* in the UMLS.

4.1 Linguistic Levels of Term Analysis

According to the nature of variation (see definition in Section 2), the normalization of terms is to be done at various linguistic levels: orthographic, morphological, syntactic and semantic.

At the orthographic level, variations may occur either within single words (*flavor* - *flavour*, *boolean* - *Boolean*), or at boundaries between adjacent words (*bolthole* - *bolt-hole*). In both cases, formal methods dealing with string similarities ([9]) may be useful ([10, p. 29], [11, p. 153-156]) but they may give many spurious conflations (e.g. *flavor* - *favor*). Systematic description of orthographic variants ([12], [13]) is more reliable but labor intensive. Some word separators, such as blanks and hyphens, may be arbitrarily interchanged (e.g. *air(-)conditioning*) but conflating them is not a good strategy in every case, as they may sometimes be disambiguating features. For example, in nominal terms containing an adverbial particle, like *a by-product* or *a take-in*, the hyphen is obligatory and allows to distinguish them from prepositional or verbal phrases. The recognition of orthographic variants of terms is closely connected to the problem of their orthographic correction. In some applications, an error tolerant morphological analysis (as in [14]) may be crucial for the quality of term extraction.

At the morphological—inflectional and derivational—level a word[7] is analysed as a root (a word that cannot be morphologically decomposed), as an inflectional form of a root, or as a word constructed from a root (through suffixing, prefixing, or compounding). Reducing morphological variation means attaching each word to its root (e.g. *beautification* → *beauty*), or to its stem, i.e. a string that is common to morphologically related words (e.g. *denied*→*deni-*). To this purpose, one of three approaches may be adopted: rule-based, dictionary-based, or hybrid.

In the rule-based approach the inflectional and derivational morphology of words is described with respect to their endings and/or to their prefixes. Rules are conceived through observation of a sufficiently big number of words with common characteristics, and then they are generalized to all words that share these characteristics. The main advantage is robustness and size: each word that belongs to a described paradigm can be analysed, and the size of the set of rules may be kept relatively small. The disadvantage is the risk of an erroneous analysis if an exception to a rule has been overseen. The main rule-based approach is stemming. Two significant stemming algorithms for English are the Lovins stemmer ([15]) and the Porter stemmer ([16]). Both are based on rules for suffix stripping (e.g. *absorption* → *absorpt-*) and stem normalization (e.g. *absorpt-* → *absorb-*) that may be applied to words under specified conditions. Stemming of morphologically richer languages like French raises particular issues such as the combination of rules with exception lists ([17]).

Alternatively, the inflectional and morphological analysis can be based on a dictionary, in which the construction of inflectional and/or derivational forms is explicitly described for each root, like in the word-based approach ([18]), in the concatenative approach ([19]), or in the DELAS system ([20]). An important dictionary-based approach to computational morphology is the two-level approach ([21]), which has been applied to numerous languages and stands out as particularly well adapted to morphologically complex, e.g. agglutinative, languages. The advantage of dictionary-based systems is their reliability and extension facility (new words can be added easily). The disadvantages are the high cost and the lack of robustness (unknown words cannot be analysed otherwise than by an additional rule-based guesser). To solve this last problem, dictionary-based and rule-based approaches may be combined into hybrid systems ([22]).

The inflectional and morphological analyses of compounds raise particular problems discussed in the following section.

After inflectional and morphological analyses words may remain ambiguous with respect to their part-of-speech and inflectional features. This ambiguity may be dealt with in corpus through increasingly available grammatical taggers, such as [23] and [24] for English, or [25] and [26] for French.

[7] Identification of word and sentence boundaries, called tokenizing, is a preliminary task to higher levels of treatment. It must deal with non trivial problems such as variable status of punctuation marks (spaces, hyphens, apostrophes, dots, etc.) that may either belong to items (as the dots in *a.m.*) or be separators between items (as a full stop at the end of a sentence).

At the syntactic level the analysis of multi-word terms aims at determining different syntactic structures of a given term that preserve its conceptual content. A number of linguistic studies have been carried out in areas closely related to this problem. Their main idea is that some groups of words, called idioms or compounds, have limited syntactic flexibility with comparison to free constructions. For example, in French, *une chambre est froide* (a room is cold) is not an acceptable variant of the complex word *Chambre froide* (a refrigerated place used for cold storage), while *une chambre est aérée* (a room is ventilated) is a variant of the free noun compound *chambre aérée* (ventilated room).

This phenomenon has been analysed in an introspective approach ([27]) inspired by the notion of lexicon-grammar ([28]). In this approach, a set of generic transformations is first established for each syntactic structure. Then, for each compound of the given structure, and for each transformation, a linguist gives her/his judgement of acceptability which is reported in a boolean table, such as Table 3. This method has been applied to a relatively high coverage of standard French compounds, but it is less appropriate for large-scale terminology description for two reasons: it is very labor-intensive due to the high number of terms and corresponding variants, and it requires not only a linguistic competence but also a deep technical knowledge in the particular domain of description.

Table 3. Examples of acceptabilities for Noun-Adjective compounds: predicativity (e.g. *la nuit est blanche* $\rightarrow$ *Nuit blanche*), nominalization (e.g. *l'historicité de ce fait* $\rightarrow$ *Fait historique*), selective restriction (e.g. *accent aigu* $\rightarrow$ *Accent grave*).

Compound	Predicativity	Nominalization	Selection restriction on adjective
Accent grave (grave accent)	−	−	+
Cinéma muet (silent films)	+	−	−
Fait historique (historic event)	+	+	−
Nuit blanche (a sleepless night)	+	−	+

An alternative approach by Barkema is based on a large-scale corpus investigation ([29]). It consists in building a *flexibility profile* of idioms by determining the list and the number of forms (including external or internal modifiers, and coordinations) that each idiom takes in a reference corpus.

At the semantic level variations of terms can be detected through identifying semantically related words. This can be based on terminological resources like thesauri which contain semantic classes or semantic relations, as those described above. A method of conceptual indexing is proposed in [30] and [31]. It relies on the *kind-of* relationship described in a knowledge representation system. A controlled term, such as *automobile steam cleaning*, is extended to a set of conceptually related terms containing *automobile cleaning*, *automobile upholstery*

cleaning, *automobile washing*, and *car washing*. In [32] non-transitive synonymy relations between single words are used in order to recognize semantic variants among candidate terms produced by an automatic term extractor.

4.2 Morphological Analysis of Compounds

The conflation of inflectional variants of compounds is often handled by stemming or lemmatizing of their component words, as in WordNet[8] thesaurus: *morphy* module) or in automatic term acquisition systems, e.g. in ACABIT. This method may give erroneous results in some cases.

First, a compound's base form may be in plural. For instance, *bits and pieces* doesn't have the singular form, while its lemmatizing performed on a single word basis yields the incorrect form **bit and piece*[9]. One may agree that such a form be an "abstract" reference form, as it is the case in stemming of single words (e.g. *deni* for *denied*). However, this is inconvenient for two reasons: it makes any human postfiltering tedious, and it may result in spurious conflations with free forms, as in *think a bit and piece the jigsaw together.*

Second, some compounds, such as *cross-roads*, have their singular and plural form identical. Here again a simplistic lemmatizing produces an incorrect form **cross-road.* In the general case, the lemma of a compound may contain words that are not lemmas themselves. For instance, the singular of *customs duties* is *customs duty* instead of **custom duty.*

Third, a single word may carry an inflection or derivation mark (or both) of a compound although this word has no inflection/derivation as an individual lexical item. For instance, in *court martials, good-for-nothings, stand-bys, take-aways, cure-alls, forget-me-nots, has-beens, johny-come-latelies*, etc. the underlined words cannot be lemmatized by a standard dictionary-based morphological analyser. Similarly, in *up-to-dateness, captain-generalcy,* and *ivory-towerist*, the underlined words are no valid derivations of the corresponding single words *date*, *general*, and *tower*. Such cases may be handled by a stemmer though.

The lemmatizing of compounds in French is even more complex due to the gender inflection of nouns and adjectives, as well as to the important number of compounds with a non-standard nominal construction, e.g. *un porte-avions* (aircraft carrier), *une deux-chevaux* (a car with a two horsepower engine).

Such non-standard cases of compounds' inflected forms may be lemmatized in a reliable way only if they have been explicitly described. This can be done either in a static approach, in which all inflected forms of controlled compounds are generated beforehand, or in a dynamic approach, in which the relevant inflected form of a controlled compound is calculated on demand during the runtime of the morphological analyser. In [33, pp. 98-100] it is argued that the static approach is preferable for the morphological analysis of compounds. In [11, pp. 48-101] a detailed analysis of inflection irregularities of compounds in English, French

[8] WordNet is available from *http://www.cogsci.princeton.edu.*

[9] The * symbol preceding a text sequence indicates that it is an invalid occurrence of a term in this context.

and Polish is shown, together with a formalism and an algorithm for automatic generation of their inflected forms.

4.3 Computational Techniques

Computational techniques most frequently used in NLP tools for information extraction are: finite-state machines, context-free and unification-based grammars, and statistical measures ([34]).

Many aspects of a natural language can be seen as formal languages described by regular expressions. For instance, in [35] the following part-of-speech pattern is used to extract well formed noun phrases in English: ((A | N)$^+$| (A | N)* (N P) (A | N)*) N. The symbols A, N and P stand for adjective, noun and preposition respectively, alignment of symbols stands for concatenation, "|" stands for union, "+" for one or more occurrences of a symbol, and "*" for zero or more occurrences. Thus, the possible patterns matching this regular expression include AN, NN, AAN, ANN, NAN, NNN, NPN, ... Regular expressions are equivalent to finite-state automata, i.e. for every regular expression there is a unique minimum deterministic finite-state automaton defining the same language, and vice versa. Finite-state transducers are more complex tools than automata because of their two-way functioning based on an input alphabet and an output alphabet. They are applied to many areas of natural language processing: phonology ([36], [37]), morphology ([21]), part-of-speech tagging ([24]), and parsing ([38]). In the field of information extraction, a transducer cascade is an efficient technique. A cascade is a set of transducers that are applied to a text one after another. Each transducer parses the text and performs some transformations on it. The resulting transformed text becomes the input for the following transducer. Three of the systems using this technique are Cass ([39]), FASTUS ([40]) and INTEX ([41]).

One of the reasons why finite-state automata and transducers are widely used in NLP in their classical and extended ([42]) versions is their time and space efficiency obtained by determinisation (sequentialisation) and minimisation ([43], [44], [45], [46]). These two properties can be characterized as follows. For each non-deterministic finite-state automaton there exists a minimal deterministic finite-state automaton recognizing the same language ([47], [48]). In the general case, due to the determinization process, the number of states of the resulting automaton may theoretically increase exponentially, but for some subclasses of finite-state automata the worst-case space complexity of determinization is far lower ([49]). The problem of minimisation and determinization of finite-state transducers is more complex than that of finite-state automata. A transducer may be interpreted as a simple automaton whose alphabet contains couples of input and output symbols. Then, the minimisation algorithms designed for automata may also be applied to transducers. However, a word lookup in such a transducer may not be deterministic. A transducer which is deterministic with respect to its input alphabet is called a sequential transducer. Not all transducers can be sequentialized, but their sequetiability is decidable ([46]).

Unification-based grammars, that stem from context-free grammars, represent languages that are more complex than the regular expressions, in particular they may describe arbitrarily distant dependencies and deep recursive structures. Context-free grammars are composed of rewriting rules with a unique nonterminal as the left component, and with concatenation of nonterminals and terminals as the right-hand component. For example, the following context-free grammar describes some well-formed English noun phrases, like *bone marrow*, *normal bone marrow*, *blood and bone marrow, blood and bone marrow cell*, etc.

$$\langle \text{NP} \rangle \rightarrow \langle \text{PreMod} \rangle \ \langle \text{Noun} \rangle \tag{1}$$
$$\langle \text{PreMod} \rangle \rightarrow \langle \text{PreMod} \rangle \ \textit{and} \ \langle \text{PreMod} \rangle \tag{2}$$
$$\langle \text{PreMod} \rangle \rightarrow \langle \text{PreMod} \rangle \ \langle \text{PreMod} \rangle \tag{3}$$
$$\langle \text{PreMod} \rangle \rightarrow \langle \text{Adj} \rangle \mid \langle \text{Noun} \rangle \tag{4}$$
$$\langle \text{Adj} \rangle \rightarrow \textit{normal} \tag{5}$$
$$\langle \text{Noun} \rangle \rightarrow \textit{blood} \mid \textit{bone} \mid \textit{cell} \mid \textit{marrow} \tag{6}$$

The language generated by such a grammar is the set of all sequences of terminal symbols (words) obtained by derivations starting from a particular nonterminal called the start symbol (here $\langle \text{NP} \rangle$). A derivation can be displayed either as a bracketing of the resulting sequence, or as a derivation tree in which leaves are terminals, interior symbols are non-terminals, and each interior node with its daughters corresponds to a rewriting rule. A sequence is ambiguous if it has more than one parse tree (bracketing) in the same grammar. For example, *blood and bone marrow cell* is ambiguous in the above grammar as its two possible bracketings (in simplified notation) are: ((*blood and* (*bone marrow*)) *cell*) and (((*blood and bone*) *marrow*) *cell*).

The classical context-free grammars have been extended to more complex models in which the mechanism of *unification* ([50]) allows for an efficient description of some dependencies between words, e.g. agreement rules. Such extended models contain the Lexical Functional Grammar (LFG, [51]), the Generalized Phrase Structure Grammar (GPSG, [52]), the Tree Adjoining Grammar (TAG, [53]), and the Head-Driven Phrase Structure Grammar (HPSG, [54]). As far as the computational complexity is concerned, these models are equivalent to various classes of formal grammars: HPSG has the complexity of a Turing machine, LFG of a context-sensible grammar, GPSG of a context-free grammar, and TAG of a "slightly context-sensible" grammar.

Two classical approaches to parsing standard and extended context-free languages are the top-down and the bottom-up approach, in which the derivation is done by applying the rewriting rules, respectively, from left to right or from right to left. Both approaches may encounter serious efficiency problems ([55]) due to non-determinism in the derivation process. Various optimisation techniques ([56]), e.g. the left-corner parsing, may speed-up the parsing, but the gain seems not to be satisfactory enough for large-corpus applications such as information retrieval. Therefore, most of the term extraction systems presented

below rely on less precise but faster NLP techniques, like tagging or shallow parsing or a combination of both.

Statistical techniques are very frequently used for term extraction. They rely on the hypothesis that words building a multi-word term tend to co-occur more frequently than if they were independent. The simplest statistical observations about two words w_1 and w_2 are: the frequency of their co-occurrence within a window of a given size, and the frequencies of their isolated occurrences, which form the following contingency table ([57, chap 4.3.2]):

	w_2	$w' \neq w_2$
w_1	$a = f(w_1, w_2)$	$b = \sum_{w' \neq w_2} f(w_1, w')$
$w \neq w_1$	$c = \sum_{w \neq w_1} f(w, w_2)$	$d = \sum_{w \neq w_1, w' \neq w_2} f(w, w')$

An information-theoretic measure called *Mutual Information* ([58, ch.2]), based on the above table, is widely used in computational linguistics (e.g. [59], [60], and [61]):

$$MI(X,Y) = \log_2 \frac{P(w_1, w_2)}{P(w_1)\ P(w_2)} = \log_2 \frac{a}{(a+b)(a+c)} \tag{7}$$

Other statistical measures used to compute word associations or document similarities in information retrieval are: Dice coefficient ([62]), Jaccard coefficient ([63]), Cosine coefficient ([64]), log-likehood ratio ([65]), etc. The statistical methods of term extraction have two important drawbacks. The first one is the risk of conflating co-occurrences of words which represent different concepts, like *horse race* et *race horse* ([57, p.113]). The second drawback is the difficulty of dealing with terms that occur rarely in corpora. Such terms are very numerous and in some applications must not be ignored. With [66], in a 280,000 word corpus from the domain of computer science submitted to a pattern matching term extractor, the term hapax legomena (i.e. terms occurring only once in a given corpus) are twice as many as other terms. To remedy such and other problems statistical measures may be accompanied by linguistic methods to build hybrid systems ([67]).

4.4 Evaluation of Term Extraction

Whatever technique is used to extract terms from corpora it is necessary to evaluate the outcome of term extractors. The goal of automatic term extraction, being to retrieve all occurrences of terms and their variants and only them, is never fully attained in practice. This results in a certain level of *noise* (wrongly retrieved occurrences), and a certain level of *silence* (correct but unretrieved occurrences). The two main measures of quality of term extraction systems, the *precision* and the *recall*, are taken from the evaluation in information retrieval

([68]). Precision P is the proportion of relevant occurrences within all retrieved occurrences, and recall R is the proportion of retrieved occurrences within all relevant occurrences. A complementary measure is *fallout* (F), the proportion of retrieved occurrences within all irrelevant occurrences. In the following formulas I stands for the set of all occurrences, I_E for the set of retrieved occurrences, and I_R for the set of relevant occurrences.

$$P = \frac{|I_E \cap I_R|}{|I_E|} \tag{8}$$

$$R = \frac{|I_E \cap I_R|}{|I_R|} \tag{9}$$

$$F = \frac{|I_E \setminus I_R|}{|I \setminus I_R|} \tag{10}$$

All three measures have values within 0 and 1. The bigger are precision and recall, and the smaller is fallout, the higher the quality of term extraction. However, precision is a decreasing function of recall, in the sense that tuning a system toward a higher precision results in lower recall and vice versa. Hence, a trade-off is necessary between these two factors, according to the needs of the particular application.

The actual difficulty of the evaluation of term extraction lies in the determination of the set of relevant occurrences (I_R) of terms in a given corpus. In the classical approach, this set is supposed to be a correct and exhaustive reference list, elaborated a priori by an expert after a manual or semi-manual analysis of the corpus. Unfortunately, the definition of a relevant term occurrence is far from being clear because it depends on the particular application of term extraction. For instance, a term like *disease is genetic* may be relevant in the context of automatic indexing because it may be a good descriptor of a document's content, but it may not be relevant in the same document for thesaurus enrichment because of its non-compound structure. In other applications, like computer aided translation, term relevance may have as complex conditions as: text origin (due to a company's internal terminology), date (some old terms may be out-of-date), translation contract (a reference list of terms and their translations may be an integral part of this contact), etc. ([69]).

As [70] put it, the terminology of a given technical domain is not a set of predefined labels attributed to rigid concepts, that need only be discovered. The terminology should be *textual* rather than metalinguistic, in the sense that it should be constructed individually for each new application by a terminologist accompanied by a technical expert of the given domain. In conclusion, an objective evaluation of a term extraction system is only possible with respect to a particular application.

The evaluation of FASTR system presented below is less complex than in the general case (at least as far as the variant recognition is concerned) because a list of valid terms is an input provided by the user. Thus, an extracted occurrence is considered as relevant if and only if it is one of the formally defined variants of a controlled, i.e. a proiri relevant, term.

In the following two sections we present a survey of existing tools for term acquisition and automatic phrase indexing (see also [71] and [72]).

4.5 Term Acquisition

All automatic term acquisition tools presented below take a raw or a tagged corpus as input, and provide, as output, a list of candidate terms, possibly enhanced with conceptual links. We deliberately exclude tools for term management from our consideration, since the term extraction is only a small subcomponent of such tools.

ACABIT ([57] and [73]) is a term acquisition tool based on a hybrid - linguistic and statistical - approach. The corpus is first tagged with part-of-speech categories and morphological features, and analysed by a finite-state shallow parser which extracts various noun phrase patterns, lemmatizes them and normalizes them by attaching variants to canonical binary forms such as N N, A N or N P N. For example, the following sequences: *permanent failure, permanent failures, permanent physical failure*, and *failure is permanent*, are all recognized as variants of the canonical binary term candidate *permanent failure*. Such conflated binary term occurrences are then submitted to different statistical filters (frequency, log-likehood, mutual information, etc.) which are evaluated with respect to their ability to separate valid term candidates from non-terminological candidates. The recognition of variants is based on syntactic transformations:

- coordination of binary terms: *assemblage de paquets* (packet assembly) + *désassemblage de paquets* (packet disassembly) → *assemblage/désassemblage de paquets* (packet assembly/disassembly),
- overcomposition of binary terms: *réseau à satellites* (satellite network) + *réseau de transit* (transit network) → *réseau de transit à satellites* (satellite transit network),
- insertion of a modifier in a binary term: *liaison par satellites* (satellite link) → *liaisons multiples par satellites* (multiple satellite links),
- shifting of a modifier from epithet to attribute position: *permanent failure* → *failure is permanent*.

ANA ([10]) is a fully statistical termer which uses a raw untagged corpus with no linguistic analysis (therefore it is language independent). Its input is a bootstrap containing some controlled terms that are used to discover new terms. A new term candidate may be either a frequent co-occurrence of bootstrap terms, or a frequent co-occurrence of a bootstrap term and of any other word (not belonging to a stop-list), possibly combined by a function word (preposition or determiner). Term discovery is incremental: newly discovered terms are included in the bootstrap, and the process is repeated until no new terms can be found. Different morphological forms of single words are conflated through approximate string matching based on string edit distance ([74]).

LEXTER ([75], [76], and [77]) performs term acquisition in French through shallow parsing, without any statistical measure, which allows for a high recall

of extraction of both frequent terms and single-occurrence terms (hapax legomena). The corpus is tagged, lemmatized and bracketed through noun phrase frontier detection (e.g. a past participle followed by any preposition except *de* builds a right frontier). The resulting chunks—called maximal noun phrases—are then decomposed into binary term candidates (e.g. *rejet d'air froid* [cool air exhaust] → *rejet d'air* [air exhaust] + *air froid* [cool air]). Then an endogenous disambiguation process retains only those ambiguous candidates which are encountered anywhere else in the corpus in a nonambiguous situation. Finally, a terminological network is created by grouping candidate terms sharing the same head (*vanne motorisée* [powered valve], *vanne d'isolement d'enceinte* [zone insulation valve],...) or the same extension (*vanne manuelle* [manual valve], *commande manuelle* [manual control], *lignage manuel* [manual lining],...).

TERMINO ([78] and [79]) is another linguistic approach based on a partial parser. Rule-based morphological analysis and lemmatizing are performed on the corpus. A partial parser allows for word disambiguation and extraction of noun phrase nuclei (e.g. in the sequence *un traitement de texte très performant* [a very efficient word processor] only *traitement de texte* [word processor] is extracted). A term recognizer filters the output of the parser in order to discover nested structures (e.g. *système de gestion de bases de données* [database management system] → *gestion de bases de données* [database management], *base de données* [database]), and ambiguous or nonambiguous expansions (e.g. the participle *intégré* [integrated] in *logiciel intégré* [integrated software] may either be attached to the head noun *logiciel* [software] or not). Obtained term candidates are filtered and ordered according to some heuristics like stop-lists and endogenous disambiguating process similar to the one in LEXTER. Finally, a term base management module allows for visualization and classification of term candidates.

TERMS ([35]) is based on matching the following regular expression pattern in an untagged corpus: ((A | N)$^+$| (A | N)* (N P) (A | N)*) N. Part-of-speech ambiguities are handled by a simple noncontextual preference selection and a stop-list. All extracted sequences appearing only once are rejected, all others are retained. TERMS was the inspiration for a term extractor for Japanese, JBrat ([80, sec. 3.2]).

Xtract ([81]) is intended for extraction of collocations (e.g. *heavy smoker*, *agree to*, *to hit a record*) which are a wider class of word co-occurrences than terms. Xtract is a hybrid system which, contrary to ACABIT, applies statistical filters before linguistic ones. It is presumed that components of a collocation appear together more often than expected by chance, therefore term discovery starts with statistical observation of word couples. For each couple, the co-occurrences within a 5-word window are summarized by a histogram that describes the possible relative positions of the two words. Then the first statistical filter retains only frequent pairs of words, and the second retains only those that co-occur most often in the same relative position. The whole procedure is reiterated to expand binary collocation candidates into n-ary ones. The resulting sequences are analysed by a parser and only syntactically correct expressions

are retained. The Chinese version of Xtract ([80, sec. 3.1]) must deal, additionally, with the lack of word delimiters which calls for a sophisticated preliminary tokenizing task.

Table 4 presents a comparative summary of all mentioned term extraction systems.

Table 4. Comparative features of term acquisition tools

		ACABIT	*ANA*	*LEXTER*	*TERMINO*	*TERMS*	*XTract*
1	Tagging	×		×			×
2	Morphological analysis				×	×	
3	Stemming		×				
4	Syntactic patterns	×		×		×	×
5	Grammar				×		
6	Statistical filtering	×	×			×	×
7	Text simplification		×				
8	Incrementality		×				
9	Language	Fr/En/Mal[10]	∀	Fr	Fr	En	En

The term acquisition tools presented above are concerned with term variation to a variable extent.

ACABIT's approach allows for conflating inflected forms by lemmatizing, and provides a good coverage of syntactic variants (coordinations, overcompositions, modifier insertions, attributive structures).

ANA attaches some inflected or derived forms to their roots through approximate string matching, which may conflate some orthographic or morphological term variants (e.g. *behavioral model - behavioural model, compiler option - compile options*), but approximate string matching is usually not enough to detect syntactic or semantic variants (except some "accidentally" morphologically close synonymes like *multiple output - multiple outlet*, etc.).

LEXTER includes lemmatizing, and treats some term overcompositions by decomposition rules, but no morphological or semantic variants are taken into account.

[9] Malagasy.

The lack of consideration of term variants, such as coordinations (*programmation locale ou subrégionale*) or acronyms (*émetteur AM*), is one of the main reasons of the limited precision of TERMINO, according to [82].

In TERMS, the matching pattern based on observation of terms in specialized dictionaries does not take into account that dictionaries contain only normalized forms of terms, while term occurrences in a corpus are prone to variation.

In Xtract, variation is only partially dealt with. On the one hand, collocation variants rarely exceed the 5-word window, therefore they contribute to the collocation's frequency examined by the first, statistical, filter. On the other hand, the collocations whose structure is not fixed are most prone to be rejected by the second, histogram-based, filter. Moreover, the linguistic analysis doesn't allow to conflate different forms having the same root or the same meaning (*rise - rose*, *rise - go up*, etc.).

Let's have a closer look to why variant recognition should be useful for term acquisition. The first reason is a statistical one. If any kind of frequency count is used to filter term candidates, as e.g. in ACABIT, ANA or Xtract, the results are always more accurate if term variants are conflated. For instance, if the following sequences: *permanent failure, permanent failures, permanent physical failure*, and *failure is permanent* are seen as different term candidates, their respective frequencies are much lower, and so their chance to be retained is smaller, than if they are conflated into one candidate. Besides, some terms may never appear in the corpus in their base forms, but may nevertheless be acquired from occurrences of their variants, as in the case of e.g. ACABIT. The second reason is ergonomic: if different variants of a term candidate are extracted but not conflated, the resulting list is longer and its validation more tedious. The third reason is the recall and the precision of the acquisition. If a sequence is recognized as a variant of a controlled term, or of a highly ranked term candidate, this sequence (or its subgroup) may itself be a relevant new term candidate. For example, if *permanent failure* is supposed to be a relevant term, then it is probably also the case for *permanent physical failure* and *permanent failure detection.* Speculations of this type prove to be justified in systems like ANA, Xtract and FASTR. They are also with relation to the problems referred to as *nested collocations*, i.e. collocations that are included inside each other ([83] and [84]), and *interrupted collocations*, i.e. collocations joined together to build lager collocations ([85] and [86]). The fruitful association of term acquisition and variant conflation is illustrated through a combination of FASTR and LEXTER in [87]. Candidate terms produced by LEXTER are clustered by FASTR and placed into a relational database with an expert interface for human term validation.

Due to the coming up of new NLP general purpose tools like parallel corpora alignment (cf the evaluation project Arcade, [88]), term acquisition gains a bilingual dimension useful for the construction of translation aid tools. The usual procedure is to perform alignment of sentences in two parallel corpora one of which is the translation of the other. Then a monolingual term acquisition takes place in either corpus, and finally the extracted terms are aligned. In this kind of approach the acquisition phase is usually not the central issue, and is

performed either through relatively simple regular expression matching ([89], [90] and [91]) or statistical measures ([59]). The interesting issue concerning the term acquisition for bilingual thesauri, as opposed to monolingual acquisition, is that an equivalent for a term in one language does not necessarily have a terminological status in the other language (that supports also our discussion in Section 4.4 on the status of relevant terms). Therefore, the extraction needs to be performed loosely enough in order to be able to provide such non terminological associations.

4.6 Phrase Indexing

The purpose of automatic indexing is to assign content descriptors to documents in order to fulfil the three following purposes ([92] and [6]):

1. Locate items within a document that deal with a given topic.
2. Build hypertext links that connect documents with similar content.
3. Assist information retrieval by predicting the relevance of individual documents with respect to a query.

Automatic indexers are generally parts of larger applications for information access. Extracted content descriptors (called *terms*, even if they are not always genuine terms), may be either single words or multi-word units. In the former case, the basic indexing techniques consist of: text simplification (based on stemming and on a stop-list of high frequency words), selection of best indices (usually based on a frequency criterion), and ranking indices according to their relevance for information retrieval. In our study, we are more interested in *phrase indexing*, i.e. indexing through multi-word units, which consists of two stages ([93, sec. 1.4]): *phrase identification* and *phrase normalization.* The latter is used to group phrase indexes with different forms and similar meanings, in the same way that term variants are conflated for term acquisition. At present, we proceed to the presentation of a survey of some existing phrase indexing tools.

CLARIT ([94]) contains three large-scale NLP modules: a lexicon-based morphological analyser, a disambiguation module based on a probabilistic grammar, and a context-free parser for identifying noun phrases (NPs). The NPs extracted by the last module are submitted to a filtering and matching module which enriches them with statistical scores. The system is able to perform free indexing as well as controlled indexing. In the latter case extracted NP indexes are matched with initial terms in such way that partial overlaps are tolerated. Two further studies, [95] and [96], concentrate on enhancing *CLARIT's* output in that the structure of extracted NPs is submitted to a statistically driven disambiguation.

The *Constituent Object Parser* (COP, [97], [98], and [99]) also relies on large grammatical and lexical data. First, it filters documents through keywords contained in the query, and then parses the query and the sentences in the selected documents to produce binary trees expressing dependency relations. Such trees are simpler and more easily obtained than fine-grained syntactic structures. A

tree matching algorithm allows to rank each document according to the number of dependencies it shares with the query. No explicit phrase indexes are produced.

COPSY ([100] and [101]) also performs the extraction of dependency relations but, contrary to *COP*, does it for noun phrases rather than for entire sentences. After prefiltering of documents through query keywords, and a stemming procedure based on a suffix tree and exception lists, noun phrases are isolated through detection of noun phrase delimiters (verbs, punctuations, etc.). Then, the dependency structure of these noun phrases is calculated by applying syntactic rules that make the head/modifier relations explicit. Finally, the dependency trees of the noun phrases in the query and of those in documents are matched (possibly partially).

Fagan's syntactic phrase indexer[11] ([93]) relies on the output of a general-purpose syntactic parser, PLNLP ([103]). The parser produces parse trees, each of which is composed of a head word, premodifiers and/or postmodifiers. Then, the trees are recursively reduced by encoding rules and stop-lists of semantically empty premodifiers and heads (*the, four, also, abundant, ability, procedure*, etc.) to create binary phrase indexes. The categories of phrases treated in this way are:

- general noun phrases: *the efficiency of these four sorting algorithms* → *algorithm efficiency* + *sorting algorithm*,
- conjoined noun phrases: *the philosophy, design and implementation of an experimental interface* → *interface philosophy* + *interface design* + *interface implementation* + *experimental interface*,
- adjective phrases: *a system for encoding, automatically matching, and automatically drawing chemical structures* → *automatically matching*+ *automatically drawing* + *structure encoding* + *structure matching* + *structure drawing* + *chemical structure*,
- verb phrases: *the machine coding these chemical structures* → *machine coding* + *structure coding* + *chemical structure*,
- phrases with semantically empty heads (belonging to a stop-list): *an automated document clustering procedure* → *an automated document clustering* → *automated clustering* + *document clustering.*

The most serious problem caused in this system by the generative phrase parsing is the lack of structural disambiguation which results in incorrect descriptors due to wrongly identified dependency relations (e.g. *they design software for browsing interfaces* → **interface browsing* + *browsing software*). Nevertheless, Fagan's work paved the way for exploitation of large coverage parsers in information retrieval.

FASIT ([104]) is based on text simplification techniques. In the process of morphological analysis words are looked up in several small exception dictionaries (i.e. lists of frequent words, of semantically empty words, of domain-dependent words, etc.), and if they are not found, they are submitted to a

[11] Fagan also proposes a statistical phrase indexing method which we will not present here as it is a generalization of the approach presented in [102].

suffix-based stemming. Then, multiply tagged words are disambiguated through contextual rules (some ambiguities are replaced by single multi-category tags such as *Adjective|Noun*). Finally, a set of 161 syntactic patterns is applied to the tagged document in order to extract single-word or multi-word indexes. Synonymous indexes are conflated into canonical forms through deletion of function words, stemming and word sorting (*library catalogs, library cataloging, catalogs of library* $\rightarrow$ *catalog librar*).

IRENA ([105]) and [106] uses NLP techniques for phrase recognition and term normalization. First, a tagger, a shallow parser and a syntactic normalizer allow to extract noun and verb phrases and represent them in a canonical form containing the head and the list of modifiers (*air pollution, pollution of the air* $\rightarrow$ [*air, pollution*]). Then, each inflected word is reduced to its lemma by the Porter stemmer enhanced with an exception dictionary. Complex phrases are decomposed into binary dependencies. Finally, the normalization of semantic variants is done through discovery of synonymy and hyperonymy relations between heads or modifiers, based on semantic resources such as WordNet. This pipeline of processing modules is applied both to the query and to documents. Documents are ranked according to which types of query term variants they contain, and how distant the components of each variant are in their text occurrences.

NPtool [107] is a finite-state noun phrase parser not meant for the term extraction as such but adapted for this task by [108]. A raw corpus is first processed by a two-level morphological analyser, followed by a morphological and syntactic disambiguation module, based on the *Constraint Grammar* [109], which may leave some ambiguities unresolved. Then, two parallel noun phrase parsers are run upon the tagged corpus: an NP-friendly parser, and an NP-hostile parser. They retain, respectively, the longest and the shortest possible sequences of tags that may constitute a correct noun phrase. For instance, if the sequence *cylinder head* may be interpreted either as a pair of nouns or as a noun followed by a verb, the NP-friendly parser will favor the first interpretation and extract *cylinder head*, while the NP-hostile parser will only retain *cylinder*. The final NP candidates are obtained by the intersection of the outputs of both parsers.

In the phrase matcher by Sheridan and Smeaton ([110] and [111]), the query and the corpus are first analyzed by a two-level morphological analyser accompanied by local disambiguation rules. Dependencies between heads and arguments of phrases are marked by syntactic tags. Then, phrases with their morphological and syntactic tags are transformed into partially ambiguous binary trees. No explicit indexes are produced. A tree matching algorithm is used instead for pairing texts and queries.

The variant generator by Sparck Jones and Tait ([112] and [113]) uses a conventional syntactic analyser ([114]) in order to transform the query into a parse tree, in which nodes are words accompanied by syntactic and semantic labels (noun, verb, determiner, etc. and man, thing, kind, etc.), and branches are labelled with thematic roles (agent, object, recipient, etc.). Then, candidate terms are extracted from the query's parse tree, and rich sets of their possible syntactic variants are generated. For example, a sub-tree corresponding to the

term *circuit detail* yields, in particular, the following variants: *the details about the circuits, detail about the circuits, details about a circuit, the detail of circuits.* The variants are converted into boolean queries which are applied to documents.

SPIRIT system ([115]) results from a large project for exploitation of NLP tools in information retrieval. Its inputs are both the document and the query. They are first processed by a morphological analyzer, whose two main particularities are: linking of words to their derivational families (e.g. in French, *taxer* [to tax], *taxation* [taxation], *taxable* [taxable]), and recognition of semantically opaque frozen expressions (e.g. *afin de* [in order to]). Then, a contextual disambiguation of syntactic tags takes place. Finally, the compound recognition module, based on Debili's parser ([116]), splits sentences into maximal-length verbal and nominal chunks, and recognizes, possibly ambiguous, head-modifier dependencies within the chunks. The dependency ambiguities are resolved through a corpus-based learning, similar to the one existing in *LEXTER* and in *CLARIT*. The retained dependencies are directly transformed into binary indexes, enhanced with frequency-based weights. The distance between a query and a document is a function of the number of common indexes, their weights, and the nature of the syntactic dependencies holding between these indexes.

TTP ([117], [118], [119], and [120]) is a fast and robust parser allowing to recover from ill-formed or too complex inputs, and from structures not covered by the grammar. Its first stage is part-of-speech tagging which associates each word with a unique syntactic category. A dictionary-based morphological analysis conflates inflected words with their lemmas and verb nominalizations with the corresponding verbs (e.g. *implementation* $\rightarrow$ *implement*). Then, parsing based on wide-coverage *Linguistic String Grammar* ([121]) produces a normalized representation of each sentence, where both head-modifier and predicate-argument relations are explicitly described. Finally, a termer extracts binary head-modifier terms and organizes them into similarity classes. For example, the sentence:

> *The former Soviet president has been a local hero aver since a Russian tank invaded Wisconsin.*

yields the following set of binary indexes: *president soviet*, *president former*, *hero local*, *tank russian*, *tank invade*, *invade wisconsin.*

Table 5 presents a comparative summary of the automatic phrase indexing tools presented here. Their three main differentiating characteristics seem to be: (a) the depth of the morphological analysis, whether restricted to inflectional morphology or extended to derivational morphology, (b) the nature of structural description, whether text fragments (chunks), interword links (dependencies), or traditional phrase structures, (c) the possible concern for term variations.

As far as the third differentiating aspect is concerned, the following remarks can be made with respect to the phrase indexing systems presented above.

In *CLARIT*, different inflectional forms of single words are conflated with their roots, but as far as multi-word units are concerned, variation is not treated deeply. The extracted NP indexes are matched against controlled terms only on the adjacent subsequence basis, i.e. a noun phrase ABCD is decomposed into ten adjacent substrings: A, B, C, D, AB, BC, CD, ABC, BCD, ABCD, that are

Table 5. Comparative features of phrase indexing tools

		CLARIT	*COP*	*COPSY*	Fagan	*FASIT*	*IRENA*
1	Morphological analysis	×	?		×		
2	Stemming		?	×		×	×
3	P-o-s disambiguation	Probabilities	?			Rules	
4	Chunks	×					
5	Dependency relations		×	×			×
6	Phrase structures	×	×		×		×
7	Structural disambiguation	×					
8	Variant generation						
9	Variant conflation	×		×	×	×	×
10	Language	En	En	En	En	En	En

		NPtool	Sheridan and Smeaton	Sparck Jones and Tait	*SPIRIT*	*TTP*
1	Morphological analysis	×	×	?	×	×
2	Stemming			?		
3	P-o-s disambiguation	Rules	Rules	?	Rules	Probabilities
4	Chunks				×	
5	Dependency relations	×	×		×	
6	Phrase structures		×	×		×
7	Structural disambiguation				×	×
8	Variant generation			×		
9	Variant conflation		×		×	
10	Language	En	En	En	Fr	En

compared to substrings derived from controlled terms. Thus only few syntactic transformations are accounted for, except (partially) overcompositions.

COP allows to decompose coordination term variants due to a fine-grained grammar for conjunctions. Its first part, the *equal-grammar*, allows to analyse coordinations of syntactically similar constituents (*the robber with the gun and the cop with the dog*). Its second part, the *unequal-grammar*, describes coordinations of constituents that need to be complemented if appearing alone (*he is a cop and good at it*).

In *COPSY* the stemming of single words as well as the conversion of noun phrases into dependency trees allow to normalize some cases of syntactic variation, but its simplistic model of dependencies in noun phrases may produce erroneous conflations, like *transport in containers, transport of containers,* and *transport from containers.*

Fagan's indexer covers a large variety of syntactic transformations that allow to combine binary terms into complex variants, including nominal, adjectival and verbal phrases.

FASIT performs index grouping through nonlinguistic techniques (function word deletion, stemming, word reordering) which allows to account for some syntactic and morphological variants, but also results in incorrect conflations like of *school library* and *library school.*

IRENA recognizes three families of variations: syntactic, inflectional and lexicosemantic, the first of which is treated most deeply through syntactic normalization and unnesting of complex phrases into binary dependencies.

In *NPtool*, whose prime interest is the noun phrase extraction and not particularly the identification of terms, the issue of terminological variation is not raised. However, the term extractor by ([108]) based on *NPtool* performs a noun phrase normalization in that the extracted NPs of different syntactic structures are transformed into a canonical "germanic" form in which the head noun is preceded by all its modifiers. The normalization procedure works by placing a postmodifying prepositional phrase between the modified head and its nominal premodifiers except possessive nouns, and the premodifying adjectives and possessive nouns. In NP's with multiple heads, this procedure is performed recursively. For instance the following NP: *exact form of the correct theory of quantum* gravity is transformed into *exact correct quantum gravity theory form.*

In Sheridan and Smeaton phrase matcher the morphosyntactic term variation is accounted for in the tree matching algorithm, even if the description of acceptable variants is not explicit enough. Query phrases and document phrases are matched with respect to nodes and dependencies existing in their binary trees, as well as with respect to textual sequences separating the matched words. For example, the query phrase *classification systems* and the textual phrase *the development of a classification schema using library system theory* don't match because of the verb *using* present in the residual structure.

The approach of Sparck Jones and Tait is opposite to other variant recognition tools. Instead of normalizing extracted phrases into canonical forms, they treat query terms as base forms which they expand into possible syntactic vari-

ants resulting from inflection, addition of determiners, and production of syntactic synonyms. The generated variants are searched for in documents on a string matching basis.

Two modules of *SPIRIT* benefit from the conflation of term variants. First, in the process of parsing, the ambiguities of head-modifier dependencies are resolved due to a corpus-based learning module which accounts for syntactic, morphological and semantic variation. For example, if a nonambiguous dependency is discovered in a sequence like *affichage mural* (wall posting), it is extended to the derivational families of the constituent words. Thus, the dependencies present in the sequences like *affichage sur les murs* (posting on the walls) and *afficher sur les murs* (post on the walls) are also considered as nonambiguous. Second, the binary dependecies are represented in an normalized form: they are ordered couples of words abstracted from their textual realization in the corpus. Therefore, the distance measure between queries and documents based on such binary dependencies, allows for syntactic and morphological variant conflation.

TTP system encompasses normalization of inflected forms of terms, as well as of some morphological variants (those involving verb nominalizations). Moreover, the full exploitation of head-modifier and predicate-argument dependencies existing in a whole sentence permits the discovery of a wide range of syntactic variations, although this issue is not explicitly illustrated in the reference papers.

In conclusion, some general remarks that can be made on the tendencies existing in term acquisition and phrase indexing tools:

- The benefit of phrase indexing with respect to single-word indexing is a subject of debate in the information retrieval community. [93] suggests that phrase indexing does not outperform classical single-word indexing. In [122] it is shown that retrieval efficiency decreases when phrases are used as indexing terms unless the query is precise. [123] presents a study of statistical and syntactic phrase indexing based, respectively, on co-occurrences and on tag patterns. Neither approach proved to do significantly better than single-word indexing, except in determining the relative ranks of low-ranked documents.
- Most acquisition and indexing systems concentrate on the extraction of noun phrases, although verbal and adjectival phrases may be equally informative. The few approaches in which this argument is at least partially taken into account are: Constituent Object Parser, Fagan's indexer, IRENA, Sheridan and Smeaton's matcher, and TTP.
- Many of the presented methods rely on decomposing complex phrases into binary dependencies (ACABIT, LEXTER, IRENA, TTP, etc.), and propose resulting binary phrases as term candidates or document descriptors. On the one hand, this technique must cope with the problem of phrase structure ambiguities that may result in incorrect decompositions (e.g. *dynamic information processing* $\rightarrow$ **dynamic information* + *information processing*). On the other hand, sets of binary substructures may be far less informative than the original complex structures. In [66] an evaluation experiment of term acquisition is presented concerning ACABIT in particular. Some binary term candidates extracted by ACABIT as basic terms, such as *history*

table, significant bit, number generator, appear only within strongly terminological complex terms, like *absolute address history table* (*AAHT*), *most (least) significant bit,* and *random number generator* (*RNG*). Such binary substructures seem to be far less adequate term candidates than the corresponding complex terms.

In the following sections we present the outline of FASTR, a shallow parser dedicated to the recognition, normalization and acquisition of compound terms.

5 Variant Conflation in FASTR

FASTR is a natural language processor essentially meant for controlled indexing. It is implemented in a unification-based framework, inspired by *PATR-II* [124] and *OLMES* [125]. However, the efficiency problems typical for general unification-based grammars are not encountered in FASTR because it is a *shallow* parser: few or no recursive dependencies between terms need to be described.

The input of FASTR is a corpus and an initial set of controlled complex terms that are analyzed morphologically and transformed into syntactic rules. The output is a set of linguistic links between text sequences and initial terms. In order to attain this goal, FASTR relies on three levels of description:

1. A word level in which single words are accompanied by morphological and semantic features and links.
2. A terminological level in which terms are represented by syntactic structures.
3. A metaterminological level in which variations are implemented by local rules that transform term structures into variant structures.

For example, the following rule indicates that *compensation* is a noun derived from the canonical verb root *compens* through appending of the suffix *-ation*. Its inflectional number is 1, which means that the two suffixes corresponding to the singular and the plural form are the empty string and *-s*, respectively.

Word '*compensation*' :
⟨cat⟩ ≐'*N*'; ⟨*inflection*⟩ ≐1; ⟨root cat⟩ ≐'*V*'; ⟨root lemma⟩ ≐'*compens*;
⟨history⟩ ≐'*ation*'.

Another rule states that *genetic* is and adjective with inflection number 1 and 6 synonyms.

Word '*genetic*' :
⟨cat⟩ ≐'*A*'; ⟨*inflection*⟩ ≐1;
⟨syn⟩ ≐('*familial*',*A*) | ('*genetical*',*A*) | ('*genic*',*A*) | ('*hereditary*',*A*) | ('*inherited*',*A*) | ('*transmitted*',*A*).

The data for the inflectional and derivational morphology of English words come, respectively, from the Tree-Tagger (http://www.ims.uni-stuttgart.de), and the CELEX base (over 52,000 English lemmas corresponding to more than

160,000 word forms, http://www.kun.nl). The semantic data for English is obtained from WordNet. Synonym sets (*synsets*) of *WordNet* thesaurus (95,000 simple words and compounds, http://www.cogsci.princeton.edu) are compiled in such a way that each word is mapped to the union of all synsets containing this word.

Controlled complex terms which are given as input to the system are first automatically recycled into syntactic rules by a morphological analyser and a generic noun phrase grammar. For instance, the rule below results from the input term *Umbilical artery* which contains an adjective with lemma *umbilical* and inflectional number *1* (no gradation), as well as a noun with canonical lemma *arter* and inflectional number *2* (suffix *-y* for singular and suffix *-ies* for plural). The rule is linked to the lexical item *arter*, called the *lexical anchor* (notion adopted from the LTAG formalism, [126]).

Rule $\mathbf{N_1 \rightarrow A_2\ N_3}$:
$\langle N_1$ lexicalization$\rangle \doteq N_3$; $\langle A_3$ lemma$\rangle \doteq$'*umbilical*'; $\langle A_3$ inflection$\rangle \doteq 1$;
$\langle N_3$ lemma$\rangle \doteq$'*arter*'; $\langle N_2$ inflection$\rangle \doteq 2$;
$\langle N_1$ agreement$\rangle \doteq \langle N_3$ agreement$\rangle$.

In such a rule it is possible to express dependencies between lexical items which are either at the same level of analysis (here: A_2 and N_3) or at two neighbouring levels (here: N_1 and A_2, or N_1 and N_3). For example, the last constraint indicates that agreement features (here: the number of nouns) are propagated from the head noun to the whole complex term. However, in FASTR's formalism the syntactic term rules may also be embedded one within another when nested term structures are to be described. For instance, the following rule describes the complex term *Measure of* [*arterial pressure*]:

Rule $\mathbf{N_1 \rightarrow N_2\ P_3\ (N_4 \rightarrow A_5\ N_6)}$:
$\langle N_1$ lexicalization$\rangle \doteq N_2$; $\langle N_2$ lemma$\rangle \doteq$'*measure*'; $\langle N_3$ inflection$\rangle \doteq 1$;
$\langle P_3$ lemma$\rangle \doteq$'*of*'; $\langle A_5$ lemma$\rangle \doteq$'*arterial*'; $\langle A_5$ inflection$\rangle \doteq 1$;
$\langle N_6$ lemma$\rangle \doteq$'*pressure*'; $\langle N_6$ inflection$\rangle \doteq 1$;
$\langle N_1$ agreement$\rangle \doteq \langle N_2$ agreement$\rangle$;
$\langle N_4$ agreement$\rangle \doteq \langle N_6$ agreement$\rangle$.

This mechanism gives the formalism the descriptive power similar to the one in LTAGs: the dependencies between distant nodes of a lexical entry may be expressed. This property is called *extended domain of locality*. In the rule above, the feature agreement is described between adjacent nodes only: N_1 and N_2, as well as N_4 and N_6, but other dependencies could also be expressed between non neighbouring nodes, like N_6 and N_2, or N_6 and N_1, etc.

The description of terminological variation in FASTR stems from the Harissian notion of *transformation* [127]: term variants result from base terms through application of relevant linguistic transformations. The transformations are represented by metarules (concept introduced into a number of formalisms, such as GPSG [52] and FB-LTAG [128], in order to reduce the size of the grammar). For instance, the following metarule:

Metarule Coor($N_1 \rightarrow A_2\ N_3$) $\equiv N_1 \rightarrow A_2\ C_4\ A_5\ N_3$: .

when unified with the rule for *Umbilical artery* introduced previously, produces the following new rule which matches coordination variants such as *umbilical or carotoid artery*:

Rule $N_1 \rightarrow A_2\ C_4\ A_5\ N_3$:
$\langle N_1$ lexicalization$\rangle \doteq N_3$; $\langle A_2$ lemma$\rangle \doteq$'*umbilical*'; $\langle A_2$ inflection$\rangle \doteq 1$;
$\langle N_3$ lemma$\rangle \doteq$'*arter*'; $\langle N_3$ inflection$\rangle \doteq 2$;
$\langle N_1$ agreement$\rangle \doteq \langle N_3$ agreement$\rangle$

The descriptive power of metarules is enhanced by regular expressions and constraints. For example, the following metarule covers any variants of Noun-Noun terms, e.g. of *Tumor cell*, in which the noun modifier N_2 can be coordinated with another modifier containing up to 3 words, as in *tumor or nontumorous hepatic cells*. The constraint on the number of N_2 allows to filter out incorrect variants such as *...(but failed to lyse) tumors or K562 cells*, which result most often from the fact that frontiers between syntagms cannot be detected reliably without a full syntactic analysis of a sentence.

Metarule Coor($N_1 \rightarrow N_2\ N_3$) $\equiv N_1 \rightarrow N_2\ \langle C\ \langle A \mid N \mid A_{pp}\rangle^{1-3}\rangle\ N_3$:
$\langle N_2$ agreement number$\rangle \neq$'*plural*'.

Another role of metarules' constraints is to describe morphological and semantic term variants[12]. The rule below states that an Adjective-Noun ($A_4\ N_3$) sequence is a valid variant of a Noun-Noun term if the head nouns (N_3) are equal and if the adjective (A_4) belongs to the same morphological family (i.e. has the same root) as the modifier noun (N_2). For instance, *enzymatic activity* is a valid variant of *Enzyme activity* extracted by using the following metarule:

Metarule NountoAdj($N_1 \rightarrow N_2\ N_3$) $\equiv N_1 \rightarrow A_4\ N_3$:
$\langle N_2$ root$\rangle \doteq \langle A_4$ root$\rangle$.

Similarly, the following metarule allows for the replacement of the modifier adjective (A_2) by another adjective (A_4) which has the same semantic value (e.g. which belongs to the same set of synonyms [*synset*] in *WordNet*), as in *hard lens* $\rightarrow$ *Rigid lens*.

Metarule SemArg($N_1 \rightarrow A_2\ N_3$) $\equiv N_1 \rightarrow A_4\ N_3$:
$\langle A_2$ syn$\rangle \doteq \langle A_4$ syn$\rangle$.

During the construction of FASTR's metagrammar, various types of variants have been deeply studied for both binary and n-ary terms. For each suggested metarule its context-free skeleton was first applied to a training corpus. Then, the extracted sequences were analyzed manually and constraints were added

[12] By the same means, the description of some graphic variations could be obtained if FASTR were coupled with an adequate database of single words' graphic variants.

gradually to the metarule in order to filter out as many spurious occurrences as possible, without eliminating correct variants.

As a result, the following syntactic variations can be extracted by FASTR's 113 metarules[13]:

- Permutation of a nominal modifier: *effect of light* → *Light effect.*
- Modification (or a substitution) by an additional modifier: *blood mononuclear cell* → *Blood cell*, (*red blood cell* is obtained by an insertion out of the controlled terms range and is not considered as a valid variant, according to condition 4 of the definition of variation in Section 2).
- Coordination of heads or arguments: *axillary artery and vein* → *Axillary vein*, *intercostal and bronchial arteries* → *Intercostal arteries*, *central venous or oesophageal pressure* → *Central venous pressure.*

Some of the 113 syntactic metarules describe compositions of these elementary variations[14]. For instance, *expression of lymphokine gene* may be recognized as a variant of *Gene expression* through a permutation (*expression of gene*), followed by a modification (*expression of lymphokine gene*). Such sequence of transformations is implemented by the following metarule:

Metarule $\mathrm{Perm}(\mathrm{X}_1 \rightarrow \mathrm{X}_2\ \mathrm{X}_3) = \mathrm{X}_3\ \mathrm{P}_4\ \mathrm{X}_5\ \mathrm{X}_2$

As far as morphological transformations of words are concerned, inflections and derivations are treated differently by FASTR. Inflections are not really considered as variants since both the rules' constituents and the corpus words are lemmatized by the part-of-speech tagger so the conflation of inflected forms is instantaneous. Conversely, affixing is seen as a genuine variation and needs to be described by metarules. Variations that involve morphological transformations of words are most often morphosyntactic variations, and only rarely pure morphological variations. The reason is that morphologically transformed words usually change their category which deeply modifies the syntactic structure of the whole term. This fact makes it necessary for FASTR to cover nonnominal phrase structures: verbal, adjectival, and adverbial phrases. Since these structures are of a great syntactic diversity, their description calls for an extensive use of regular expressions. The types of morphosyntactic variants retained in FASTR are:

- Noun to adjective variants: *disease of the abdomen* → *Abdominal disease*, *sparse data* → *Data sparseness*, *error tolerant* → *Error tolerance.*
- Noun to verb variants: *consolidate those loans* → *Loan consolidation*, *estimating gestational age* → *Age estimation.*

[13] In particular, elision variants, such as *Kerr effect* → *Kerr magnetooptical effect*, have been studied but not retained for the design of FASTR due to their poor extraction results (precision value as low as 34%).

[14] Alternatively, compositions of variations could be implemented through successive application of metarules corresponding to simple variations. This solution has been discarded due to its higher computational cost.

- Adjective to adverb variants: *observed simultaneously* → *Simultaneous observation*, *simultaneously obtained measurements* → *Simultaneous measurement.*
- Noun to noun and adjective to adjective variants: *air ventilation filter* → *Air filtration*, *analytic methods* → *Analytical method.*

As soon as the input terms have been recycled into a grammar of syntactic rules, and the metarules describing adequate syntactic, morphological and semantic variations have been experimentally tuned, such lexicalized grammar can be applied to a tagged corpus. For each sentence in the corpus the set of active grammar rules (term rules) is determined by verifying if the sentence contains all of a rule's lexical items or their morphologically or semantically related words (method inspired by LTAGs [129]). The parsing of the sentence takes place in two steps. In the first step, the input string is matched on the bottom-up basis against the context-free skeleton of each active grammar rule and its *lemma* features, while all other feature equations are ignored. Only if this first matching step succeeds the remaining features are unified. If the unification succeeds the parsed sequence is reported as a valid term occurrence together with a link to the corresponding term. If the unification fails all metarules relevant to the given rule are activated to generate new rules which in turn are applied to the sentence.

This two-step parsing mechanism—context-free parsing and unification—allows to speed-up the parsing considerably because the unification rarely fails when the context-free parsing step succeeds. The parsing efficiency grows even more due to the *lexicalization of the grammar* ([130]), i.e. the distribution of rules' lexical anchors in such a way that each single word is the anchor of possibly smallest number of rules. The parsing speed is a function of the size of the terminological and transformational data. When indexing a medical corpus with a list of about 72,000 terms and a set of 115 metarules, on a Pentium (300 Mhz), 32 Mbytes main memory running Linux, the average speed is 25,000 words/min. With a 10 times smaller list of terms, the parsing speed is 6 times higher.

6 Term Enrichment in FASTR

Apart from FASTR's primary goal, i.e. controlled indexing, its variant recognition results may also have a secondary utility: term enrichment. Most variants involve more than one term in their construction. At least one of these terms must appear in the controlled vocabulary so that the variant can be recognized by FASTR. Other terms that contribute to the variant may be absent from the controlled vocabulary but they may be spotted relatively easily due to the fact that they are associated with a controlled term. For example, having recognized *uterine and carotid artery* as a variant of *Uterine artery* we may suppose that *carotid artery* is itself a correct although unlisted term.

The case of coordinated binary terms is simple to decompose: given a binary term and its coordinated variant, there is only one possible binary candidate,

such as *carotid artery* in the above example. Where ternary terms are coordinated there are two possible candidates, only one of which must be selected. For instance, *inflammatory and erosive joint disease* which is a variant of *Inflammatory joint disease* may yield either a binary candidate *erosive joint* or a ternary one, *erosive joint disease*. An experimental study ([1, pp. 244-246]) shows that the latter choice, i.e. the choice of the longest candidate, is most often the correct one. If the two candidates are of equal length the one which is a continuous substring of the variation is preferable.

The case of a substitution variation is similar to the one of coordination. For instance, *regional cerebral blood flow* which is the variant of *Regional blood flow* yields three candidates: *cerebral blood*, *cerebral flow*, and *cerebral blood flow*. Here again the latter, i.e. the longest, candidate is expected to be the best one.

The analysis of the more complex cases: compositions of substitutions, compositions of coordinations, and compositions of permutations and substitutions is presented in [1, pp. 246-248].

Once the term candidates have been obtained from the extracted variants, a postprocessing is necessary to produce terminologically valid candidates. For instance, the variant *performance of an expert system* $\rightarrow$ *System performance* yields the candidate *an expert system* in which the leading determiner needs to be cut off in order to obtain the standard nominal compound structure *expert system*. In addition, candidate terms which already belong to the controlled vocabulary need to be discarded. The retained candidates are ranked according to an association ratio, as well as to a symbolic criterion which advantages the candidates obtained from several variants of different types. For instance, the candidate *bile duct* is produced both from a coordination variant (*pancreatic and bile duct* $\rightarrow$ *Pancreatic duct*), and from a modification variant (*hepatic bile duct* $\rightarrow$ *Hepatic duct*), which suggests that it is probably a secure candidate term.

The term enrichment is implemented in FASTR as an incremental process. Acquired candidates may be considered as controlled terms, and given as input to a new step of variant recognition and term enrichment. This processing chain may be reiterated until no new terms are found. This allows to increase the number of acquired terms, as well as to build a conceptual network of terms and candidates. The latter task is realised due to the supposition that terms acquired from coordination variants share a common hypernym with the original term, while those acquired from substitution variants are more specific than the original term.

The term enrichment by FASTR is an original and, relatively to other term acquisition methods, a very reliable way of discovering new terms but has a limited scope because only terms involved in terminological variants can be acquired. The number of such terms is not very high. In the evaluation experiment described in the following section a corpus of 120,000 words and the initial vocabulary of 6,621 terms yielded only 165 new correct term candidates.

Another method of term enrichment using FASTR's output is described in [1, pp. 228-240]. Due to a statistical calculus the extracted substitution variants

are filtered in order to retain only those that have the biggest chance to be correct new terms, while the use of statistical measures allows to disambiguate their structure.

7 Results

The quality of term extraction by FASTR has been evaluated in terms of precision (P), recall (R), and precision of fallout (P_F), which is a complementary measure to fallout (P_F = 1 - F) and has the advantage that the better are the results the higher is its value, while it is the opposite for fallout. Two different evaluation experiments have been performed for different types of variations. Table 6 presents their summary results.

Table 6. Precision, recall and fallout of the term extraction by FASTR

	Precision P	Recall R	Prec. Fallout P_F
Syntactic variants	94.5%	71%	96.5%
Morphosyntactic variants	46% 74% 80%	58% 70% 58%	73% 76.5% 91%
Semantic variants	91% 78%	not evaluated	not evaluated
	55% 29%		
Term acquisition	79%		

The evaluation of syntactic and morphosyntactic metarules has been done with a 120.000 word corpus from the metallurgy domain, a list of 6,621 terms in the same domain, and the morphological data from CELEX database containing 160,000 word forms.

While tuning the syntactic metarules, the precision of extraction has been preferred to recall, which is reflected by the experimental results: 94.5% vs. 71%. The syntactic variants proved to be pervasive: they constitute as much as 28% of all term occurrences. The most frequent type of syntactic variation is modification of a binary term.

The qualitative evaluation of the morphosyntactic metarules depends on two factors: whether prefixed words are included in the morphological families (since they most often result in incorrect variants, e.g. *surface interaction* is not a variant of *Surface reaction*), and whether prefixed hyper/hyponyms and antonyms are considered as valid variants (e.g. *magnetic/electromagnetic*, *dioxide/monoxide*). Each cell of the second line of Table 6 is divided into three parts: the first two parts contain the results in the case when both nonprefixed and prefixed variants are taken into account. In the first part antonyms/hypernyms/hyponyms are considered as incorrect, and in the second part as correct. The third part describes only nonprefixed cases.

The precision of the semantic term extraction has been evaluated for French [5] with a 1.2 million word corpus from the agricultural domain, and two types

of semantic data: links from the AGROVOC thesaurus ([7]) and links from Microsoft Word97 thesaurus. The results for both thesauri are given, respectively, in the first and in the second part of the third line in Table 6. The precision is calculated separately for pure semantic variants (91% and 78%) and for hybrid, i.e. morphosyntactico-semantic variants (55% and 29%).

The evaluation of term enrichment by FASTR was done with the same data as for syntactic variant recognition. The total precision obtained is of 79%. The recall and the fallout were not calculated because the set of all correct terms in a corpus is very difficult to determine.

8 Conclusion

This chapter has presented several tools for term extraction: term acquisition for corpus-based thesaurus construction and term recognition for machine-aided indexing. Most of these studies show a concern for term variant conflation. The weakest approaches correspond to "bags of stems", the most elaborated ones correspond to conceptual analysis and paraphrase detection. FASTR lies somewhere in the middle by combining large scale shallow parsing and systematic variant generation. It offers a reasonable and however efficient means for recognizing and grouping term variants. The kind of conflation performed by FASTR is positively evaluated in a framework of machine-aided indexing [131].

The application has been designed to support extensions to other languages. Currently, French ([132], [133], and [134]), Spanish and Catalan ([135]), German, and Japanese ([136]) have been considered in addition to the English language. But the scope of variant recognition is wider than core automatic indexing. In Section 6 variant deconstruction and variant ranking both serve the purpose of term enrichment. In [1, chap. 8] it is suggested how variant recognition can also be useful in cross-lingual information retrieval, document filtering in Web search, or corpus-based morphological acquisition. More elaborated approaches to linguistic normalization such as paraphrase detection can have applications in document summarization, information extraction, and open domain question answering. Since these fields tend to be very active at present, other developements in variation reduction should be expected in the near future.

References

Jacquemin, C.: Spotting and Discovering Terms through NLP. MIT Press, Cambridge, MA (2001)

Sager, J.C.: A Practical Course in Terminology Processing. John Benjamins, Amsterdam (1990)

Guilbert, L.: La formation du vocabulaire de l'aviation. Larousse, Paris (1965)

Frenkel, K.A.: The human genome project and informatics. Communications of the ACM **34** (1991) 41–51

Jacquemin, C.: Syntagmatic and paradigmatic representations of term variation. In: Proceedings, 37th Annual Meeting of the Association for Computational Linguistics (ACL'99), University of Maryland, CollegePark, ACL (1999) 341–48

Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
AGROVOC: AGROVOC - Multilingual Agricultural Thesaurus. Food and Agricultural Organization of the United Nations. (1995) http://www.fao.org-/catalog/Book/products/v9669-e.htm.
UMLS: Unified Medical Language System, UMLS Knowledge Source. National Library of Medicine. Sixth experimental edn. (1995) http://www.nlm.nih.gov-/research/umls/UMLSDOC.HTML.
Hall, P.A., Dowling, G.R.: Approximate string matching. Computing Surveys **12** (1980) 381–402
Enguehard, C., Pantera, L.: Automatic natural acquisition of a terminology. Journal of Quantitative Linguistics **2** (1995) 27–32
Savary, A.: Recensement et description des mots composés — méthodes et applications. Thèse de doctorat, Université de Marne-la-Vallée, Noisy-le-Grand, France (2000)
Mathieu-Colas, M.: Orthographe et informatique: Établissement d'un dictionnaire électronique des variantes graphiques. Langue Française **87** (1990) 104–11
Monceaux, A.: Le dictionnaire des mots simples anglais: mots nouveaux et variantes orthographiques. Sèrie Informes IGM 95-15, Institut Gaspard Monge, Université de Marne-la-Vallée, Noisy-le-Grand, France (1995)
Oflazer, K.: Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. Computational Linguistics **22** (1996) 73–89
Lovins, J.B.: Development of a stemming algorithm. Translation and Computational Linguistics **11** (1968) 22–31
Porter, M.F.: An algorithm for suffix stripping. Program **14** (1980) 130–37
Dal, G., Hathout, N., Namer, F.: Construire un lexique dérivationnel: Théorie et réalisations. In: Proceedings, Conférence de Traitement Automatique du Langage Naturel (TALN'99), Cargèse, ATALA, Paris (1999) 115–24
Byrd, R.J., Klavans, J.L., Aronoff, M., Anshen, F.: Computer methods for morphological analysis. In: Proceedings, 24th Annual Meeting of the Association for Computational Linguistics (ACL'86), New York, ACL (1986) 120–27
Tzoukermann, E., Liberman, M.: A finite-state processor for Spanish. In: Proceedings, 13th International Conference on Computational Linguistics (COLING'90), Helsinki, ACL (1990)
Courtois, B.: Un système de dictionnaires électroniques pour les mots simples du français. Langue Française **87** (1990)
Koskenniemi, K.: Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production. PhD dissertation, University of Helsinki, Helsinki (1983)
Clemenceau, D.: Finite-state morphology: Inflections and derivations in a single framework using dictionaries and rules. In Roche, E., Schabes, Y., eds.: Finite-State Language Processing. MIT Press, Cambridge, MA (1997) 383–406
Brill, E.: A simple rule-based part of speech tagger. In: Proceedings, Third Conference on Applied Natural Language Processing (ANLP'92), Trento, ACL (1992) 152–55
Roche, E., Schabes, Y.: Deterministic part-of-speech tagging with finite-state transducers. In Roche, E., Schabes, Y., eds.: Finite-State Language Processing. MIT Press, Cambridge (1997) 205–40
Chanod, J.P., Tapanainen, P.: Statistical and constraint-based taggers for french. Technical report, Xerox Research Centre Europe, Grenoble, France (1994)
Laporte, E., Monceaux, A.: Elimination of lexical ambiguities by grammars: the *ELAG* system. Linguisticae Investigationes **22** (1998) John Benjamins Publishing Company.

Gross, G.: Degré de figement des noms composés. Langages **90** (1988) 57–72
Gross, M.: Grammaire transformationnelle du français, 2: Syntaxe du nom. Systématique de la langue française. Cantilène, Paris (1986)
Barkema, H.: Determining the syntactic flexibility of idioms. In Fries, U., Tottie, G., Schneider, P., eds.: Creating and using English language corpora. Rodopi, Amsterdam (1994) 39–52
Ambroziak, J., Woods, W.A.: Natural language technology in precision content retrieval. In: Proceedings, Natural Language Processing and Industrial Applications (NLP+IA'98), Moncton, New Brunswick, University of Moncton (1998)
Woods, W.A.: Conceptual indexing : A better way to organize knowledge. Technical Report SMLI TR-97-61, Sun Microsystems Laboratories, Mountain View (1997)
Hamon, T., Nazarenko, A., Gros, C.: A step towards the detection of semantic variants of terms in technical documents. In: Proceedings, 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98), Montreal, ACL (1998) 498–504
Silberztein, M.: Dictionnaires électroniques et analyse automatique de textes: Le système INTEX. Masson, Paris (1993)
Charniak, E.: Statistical Language Learning. A Bradford Book. MIT Press, Cambridge (1993)
Justeson, J.S., Katz, S.M.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering **1** (1995) 9–27
Kaplan, R., Kay, M.: Regular models of phonological rule systems. Computational Linguistics **20** (1994)
Laporte, E.: Rational transductions for phonetic conversion and phonology. In Roche, E., Schabes, Y., eds.: Finite-State Language Processing. MIT Press, Cambridge (1997)
Roche, E.: Parsing with finite state transducers. In Roche, E., Schabes, Y., eds.: Finite-State Language Processing. MIT Press, Cambridge, MA (1997)
Abney, S.: Partial parsing via finite-state cascade. In: Proceedings, Workshop on Robust Parsing, 8th European Summer Schol in Logic, Language and Information, Prague, Czech Republic (1996) 8–15
Hobbs, J.R., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., Tyson, M.: FASTUS: A cascaded finite-state transducer for extracting information from natural-language text. In Roche, E., Schabes, Y., eds.: Finite-State Language Processing. MIT Press, Cambridge (1997) 383–406
Friburger, N., Maurel, D.: Finite-state transducer cascade to extract proper nouns in texts. In: Proceedings, 6th Conference on Implementations and Applications of Automata, Pretoria, South Africa (2001) 97–106
Kornai, A.: Extended Finite State Models of Language. Cambridge University Press, Cambridge, UK (1999)
Watson, B.: Taxonomies and Toolkits of Regular Language Algorithms. PhD. Thesis, University of Technology, Eindhoven, the Netherlands (1995)
Daciuk, J., Mihov, S., Watson, B., Watson, R.: Incremental construction of minimal acyclic finite state automata. Computational Linguistics **26** (2000) 3–16
Mohri, M.: Compact representations by finite-state transducers. In: Proceedings, 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94), Las Cruces, NM, ACL (1994) 204–08
Gaál, T.: Is this finite-state transducer sequentiable? In: Proceedings, 6th Conference on Implementations and Applications of Automata, Pretoria, South Africa (2001) 107–115

Hopcroft, J.E.: An $n \log n$ algorithm for minimizing the states of in a finite automaton. In Kohavi, Z., Paz, A., eds.: The Theory of Machines and Computations. Academic Press, New York (1971) 189–96

Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison-Wesley, Reading (1979)

Melishar, B., Skryja, J.: On the size of deterministic finite automata. In: Proceedings, 6th Conference on Implementations and Applications of Automata, Pretoria, South Africa (2001) 203–216

Abeillé, A.: Les nouvelles syntaxes. Grammaires d'unification et analyse du français. Armand Colin, Paris (1993)

Bresnan, J., ed.: The Mental Representation of Grammatical Relations. MIT Press, Cambridge, MA (1992)

Gazdar, G., Klein, E., Pullum, G.K., Sag, I.A.: Generalized Phrase Structure Grammar. Harvard University Press, Cambridge (1985)

Joshi, A.K.: An introduction to Tree Adjoining Grammars. In Manaster-Ramer, A., ed.: Mathematics of Language. John Benjamins, Amsterdam (1987) 87–115

Pollard, C., Sag, I.A.: Information-Based Syntax and Semantics. Volume 1 : Fundamentals. CSLI Lecture Notes vol. 13. Chicago University Press, Chicago (1987)

Abeillé, A.: Grammaires et analyseurs syntaxiques. In Pierrel, J.M., ed.: Ingénierie des langues. Hermes Sciences, Paris (2000)

Kay, M.: Algorithm schemata and data structures in syntactic processing. In: Proceedings, Nobel Symposium on Text Processing, Gotheborg, Danemark (1980) 35–70 reprint in Grosz, B., Sparck Jones, K., Webber, B. (eds.) *Readings in Natural Language Processing*, Morgan Kaufman.

Daille, B.: Approche mixte pour l'extraction de terminologie : Statistique lexicale et filtres linguistiques. Thèse en informatique fondamentale, Université de Paris 7, Paris (1994)

Fano, R.M.: Transmission of Information: A Statistical Theory of Communications. MIT Press, Cambridge (1961)

Smadja, F., McKeown, K.R., Hatzivassiloglou, V.: Translating collocations for bilingual lexicons: A statistical approach. Computational Linguistics **22** (1996) 1–38

Church, K.W., Hanks, P.: Word association norms, Mutual Information and lexicography. Computational Linguistics **16** (1990) 22–29

Brown, P.L., Della Pietra, V.J., deSouza, P.V., Lai, J.C., Mercer, R.L.: Class-based n-gram models of natural language. Computational Linguistics **18** (1992) 467–79

Dice, L.R.: Measures of the amount of ecologic association between species. Journal of Ecology **26** (1945) 297–302

Tanimoto, T.T.: An elementary mathematical theory of classification. Technical report, IBM (1958)

Salton, G., Lesk, M.E.: Computer evaluation og indexing and text processing. Journal of the Association for Computational Machinery **15** (1968) 8–36

Dunning, T.: Accurate methods for the statistics of surprise and coincidence. Computational Linguistics **19** (1993) 61–74

Savary, A.: Etude comparativee de deux outils d'acquisition de termes complexes. In: Proceedings, Conference Terminologie et Intelligence Artificielle (TIA-2001), INIST-CNRS, Nancy (2001)

Klavans, J.L., Resnik, P., eds.: The Balancing Act: Combining Symbolic and Statistical Approaches to Language. MIT Press, Cambridge (1996)

Salton, G.: Automatic Text Processing : The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, Reading (1989)

Gouadec, D., ed.: Terminologie et Phraséologie pour Traduire - Le concordancier du Traducteur. La Maison du Dictionnaire, Paris (1997)

Bourigault, D., Slodzian, M.: Pour une terminologie textuelle. Terminologies Nouvelles **19** (1999)

Habert, B., Jacquemin, C.: Noms composés, termes, dénominations complexes : Problématiques linguistiques et traitements automatiques. Traitement automatique des langues **34** (1993) 5–42

Cabré Castellví, M.T., Estopà Bagot, R., Vivaldi Palatresi, J.: Automatic term detection: A review of current systems. In Bourigault, D., Jacquemin, C., L'Homme, M.C., eds.: Recent Advances in Computational Terminology. John Benjamins, Amsterdam (2001)

Daille, B.: Study and implementation of combined techniques for automatic extraction of terminology. In Klavans, J.L., Resnik, P., eds.: The Balancing Act: Combining Symbolic and Statistical Approaches to Language. MIT Press, Cambridge (1996) 49–66

Wagner, R.A., Fisher, M.J.: The string-to-string correction problem. Journal of the Association for Computational Machinery **21** (1974) 168–73

Bourigault, D.: An endogeneous corpus-based method for structural noun phrase disambiguation. In: Proceedings, Sixth Conference of the European Chapter of the Association for Computational Linguistics (EACL'93), Utrecht, ACL (1993) 81–86

Bourigault, D.: LEXTER un Logiciel d'EXtraction de TERminologie. Application à l'extraction des connaissances à partir de textes. Thèse en mathématiques, informatique appliquée aux sciences de l'homme, École des Hautes Études en Sciences Sociales, Paris (1994)

Bourigault, D.: LEXTER, a Natural Language tool for terminology extraction. In: Proceedings, Seventh EURALEX International Congress, Göteborg, EURALEX (1996) 771–79

David, S., Plante, P.: De la nécessité d'une approche morpho-syntaxique dans l'analyse de textes. Intelligence Artificielle et Sciences Cognitives au Québec **3** (1990) 140–54

David, S., Plante, P.: Le progiciel TERMINO : de la nécessité d'une analyse morphosyntaxique pour le dépouillement terminologique des textes. In: Colloque International sur les Industries de la Langue: Perspectives des Années 1990, Montréal, Office de la Langue Fran caise et Société des Traducteurs du Quebec (1990) 71–88

Fung, P.: Using Word Signature Features for Terminology Translation from Large Corpora. PhD dissertation, Graduate School of Arts and Science, Columbia University, New York (1997)

Smadja, F.: Xtract : An overview. Computer and the Humanities **26** (1993) 399–413

Lauriston, A.: Automatic recognition of complex terms: Problems and the TERMINO solution. Terminology **1** (1994) 147–70

Chen, K.H., Chen, H.H.: Extracting noun phrases from large-scale texts: A hybrid approach and its automatic evaluation. In: Proceedings, 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94), Las Cruces, NM, ACL (1994) 234–41

Frantzi, K.T., Ananiadou, S.: Retrieving collocations by co-occurrences and word order constraints. In: Proceedings, 16th International Conference on Computational Linguistics (COLING'96), Copenhagen, ACL (1996) 41–46

Ikehara, S., Shirai, S., Uchino, H.: A statistical method for extracting uninterrupted and interrupted collocations from very large corpora. In: Proceedings, 16th International Conference on Computational Linguistics (COLING'96), Copenhagen, ACL (1996) 574–79

Shimohata, S., Sugio, T., Nagata, J.: Retrieving collocations by co-occurrences and word order constraints. In: Proceedings, 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL'97), Madrid, ACL (1997) 476–81

Bourigault, D., Jacquemin, C.: Term extraction + term clustering: An integrated platform for computer-aided terminology. In: Proceedings, Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL'99), Bergen, ACL (1999) 15–22

Véronis, J., Langlais, P.: Evaluation of parallel text alignement systems: Arcade. In Véronis, J., ed.: Parallel Text Processing. Kluwer Academic Publisher, Dordrecht (2000)

Van der Eijk, P.: Automating the acquisition of bilingual terminology. In: Proceedings, Sixth Conference of the European Chapter of the Association for Computational Linguistics (EACL'93), Utrecht, ACL (1993) 113–19

Dagan, I., Church, K.W.: *Termight*: Identifying and translating technical terminology. In: Proceedings, Fourth Conference on Applied Natural Language Processing (ANLP'94), Stuttgart, ACL (1994) 34–40

Gaussier, É.: Flow network models for word alignment and terminology extraction from bilingual corpora. In: Proceedings, 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98), Montreal, ACL (1998) 444–50

Keen, E.M.: On the generation and searching of entries is printed subject indexes. Journal of Documentation **33** (1977) 15–45

Fagan, J.L.: Automatic phrase indexing for document retrieval: An examination of syntactic and non-syntactic methods. In: Proceedings, Tenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'87), ACM (1987) 91–101

Evans, D.A., Ginther-Webster, K., Hart, M., Lefferts, R.G., Monarch, I.A.: Automatic indexing using selective NLP and first-order thesauri. In: Proceedings, Intelligent Multimedia Information Retrieval Systems and Management (RIAO'91), Barcelona, CID, Paris (1991) 624–43

Evans, D.A., Zhai, C.: Noun-phrase analysis in unrestricted text for information retrieval. In: Proceedings, 34th Annual Meeting of the Association for Computational Linguistics (ACL'96), Santa Cruz, ACL (1996) 17–24

Zhai, C.: Fast statistical parsing of noun phrases for document indexing. In: Proceedings, Fifth Conference on Applied Natural Language Processing (ANLP'97), Washington, DC, ACL (1997) 312–19

Metzler, D.P., Haas, S.W.: The Constituent Object Parser: Syntactic structure matching for Information Retrieval. ACM Transactions on Information Systems **7** (1989) 292–316

Metzler, D.P., Haas, S.W., Cosic, C.L., Wheeler, L.H.: Constituent Object Parsing for Information Retrieval and similar text processing problems. Journal of the American Society for Information Science **40** (1989) 398–423

Metzler, D.P., Haas, S.W., Cosic, C.L., Weise, C.A.: Conjunction ellipsis, and other discontinuous constituents in the Constituent Object Parser. Information Processing and Management **26** (1990) 53–71

Schwarz, C.: Content-based text handling. Information Processing and Management **26** (1989) 219–26

Schwarz, C.: Automatic syntactic analysis of free text. Journal of the American Society for Information Science **41** (1990) 408–17

Salton, G., Yang, C.S., Yu, C.T.: A theory of term importance in automatic text analysis. Journal of the American Society for Information Science **26** (1975) 33–44

Heidorn, G.E.: Augmented phrase structure grammars. In Schank, R., Nash-Webber, B.L., eds.: Theoretical Issues in Natural Language Processing: An Interdisciplinary Workshop in Computational Linguistics, Psychology, Linguistics, and Artificial Intelligence. Lawrence Erlbaum Associates, Hillsdale, NJ (1975) 10–13

Dillon, M., Gray, A.S.: FASIT : A fully automatic syntactically based indexing system. Journal of the American Society for Information Science **34** (1983) 99–108

Arampatzis, A.T., Koster, C.H.A., Tsoris, T.: IRENA: Information retrieval engine based on natural language analysis. In: Proceedings, Intelligent Multimedia Information Retrieval Systems and Management (RIAO'97), Montreal, CID, Paris (1997) 159–75

Arampatzis, A.T., Tsoris, T., Koster, C.H.A., van der Weide, T.P.: Phrase-based information retrieval. Information Processing and Management **34** (1998) 693–707

Voutilainen, A.: *NPtool*, A detector of English noun phrases. In: Proceedings, Workshop on Very Large Corpora : Academic and Industrial Perspectives, Columbus, Ohio, ACL (1993) 48–57

Arppe, .A.: Term extraction from unrestricted text. http://www.lingsoft.fi/doc/nptool/term-extraction.html (1995)

Karlsson, F., Voutilainen, A., Heikkilä, J., Anttila, A., eds.: Constraint Grammar A Language-Independent System for Parsing Unrestricted Text. Mouton de Gruyter, Berlin (1995)

Smeaton, A.F., Sheridan, P.: Using morpho-syntactic language analysis in phrase matching. In: Proceedings, Intelligent Multimedia Information Retrieval Systems and Management (RIAO'91), Barcelona, CID, Paris (1991) 415–29

Sheridan, P., Smeaton, A.F.: The application of morpho-syntactic language processing to effective phrase matching. Information Processing and Management **28** (1992) 349–69

Sparck Jones, K., Tait, J.I.: Linguistically motivated descriptive term selection. In: Proceedings, Tenth International Conference on Computational Linguistics (COLING'84), Stanford, ACL (1984) 287–90

Sparck Jones, K., Tait, J.I.: Automatic search term variant generation. Journal of Documentation **40** (1984) 50–66

Boguraev, B.K., Sparck Jones, K.: A natural language front end to databases with evaluative feedback. In Boguraev, B.K., Sparck Jones, K., eds.: New Applications of Databases. Academic Press, London (1984)

Andreewsky, A., Debili, F., Fluhr, C.: Computational learning of semantic lexical relations for the generation and automatic analysis of content. In: Proceedings, IFIP Congress, Toronto, IFIP (1977) 667–73

Debili, F.: Analyse syntaxico-sémantique fondée sur une acquisition automatique de relations lexicales-sémantiques. Thèse de doctorat d'état en sciences informatiques, University of Paris 11, Orsay (1982)

Strzalkowski, T., Vauthey, B.: Information retrieval using robust natural language processing. In: Proceedings, 20th Annual Meeting of the Association for Computational Linguistics (ACL'92), Newark, DE, ACL (1992) 104–11

Strzalkowski, T.: Robust text processing in automatic information retrieval. In: Proceedings, Fourth Conference on Applied Natural Language Processing (ANLP'94), Stuttgart, ACL (1994) 168–73

Strzalkowski, T.: Natural language information retrieval. Information Processing and Management **31** (1995) 397–417

Strzalkowski, T., Scheyen, P.G.N.: Evaluation of the Tagged Text Parser. In Bunt, H., Tomita, M., eds.: Recent Advances in Parsing Technology. Kluwer Academic Publisher, Boston (1996) 201–20

Sager, N.: Natural Language Information Processing : A Computer Grammar of English and Its Applications. Addison-Wesley, Reading (1981)

Gonzalo, J., Peñas, A., Verdejo, F.: Lexical ambiguity and information retrieval revisited. In: Proceedings, Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC'99), University of Maryland, CollegePark, ACL (1999) 195–203

Mitra, M., Buckley, C., Singhal, A., Cardie, C.: An analysis of statistical and syntactic phrases. In: Proceedings, Intelligent Multimedia Information Retrieval Systems and Management (RIAO'97), Montreal, CID, Paris (1997) 200–14

Shieber, S.M.: An Introduction to Unification-Based Approaches to Grammar. CSLI Lecture Notes vol. 4. Chicago University Press, Chicago (1986)

Habert, B.: *OLMES* : a versatile and extensible parser in CLOS. In: Proceedings, Fourth International Conference on Technology of Object-Oriented Languages and Systems (TOOLS'91), Paris, Prentice-Hall, Englewood Cliffs, NJ (1991) 149–60

Schabes, Y., Abeillé, A., Joshi, A.: Parsing strategies with 'lexicalized' grammars. In: Proceedings, 12th International Conference on Computational Linguistics (COLING'88), Budapest, ACL (1988) 578–83

Harris, Z.S.: Mathematical Structure of Language. John Wiley, New York (1968)

Srinivas, B., Egedi, D., Doran, C., Becker, T.: Lexicalization and grammar development. In: Proceedings, KONVENS'94, Vienna, ÖGAI (1994) 310–19

Schabes, Y., Joshi, A.K.: Parsing with Lexicalized Tree Adjoining Grammar. In Tomita, M., ed.: Current Issues in Parsing Technologies. Kluwer Academic Publisher, Boston (1990)

Jacquemin, C.: Optimizing the computational lexicalization of large grammars. In: Proceedings, 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94), Las Cruces, NM, ACL (1994) 196–203

Jacquemin, C., Daille, B., Royauté, J., Polanco, X.: In vitro evaluation of a program for machine-aided indexing. Information Processing and Management (2001) forthcoming.

Jacquemin, C., Klavans, J.L., Tzoukermann, E.: Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In: Proceedings, 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (ACL-EACL'97), Madrid., ACL (1997) 24–31

Klavans, J.L., Jacquemin, C., Tzoukermann, E.: A natural language approach to multiword term conflation. In: DELOS Workshop on Cross-Language Information retireval, ETHZ, Zurich, Switzerland, ERCIM: European Consortium for Informatics and Mathematics (1997)

Jacquemin, C., Tzoukermann, E.: NLP for term variant extraction: A synergy of morphology, lexicon, and syntax. In Strzalkowski, T., ed.: Natural Language Information Retrieval. Kluwer Academic Publisher, Boston (1999) 25–74

Vivaldi Palatresi, J.: Extracción de candidatos a término mediante combinación de estrategias heterogéneas. Tesi doctoral, Universitat Politécnica de Catalunya, Barcelona, Spain (2001)

Yoshikane, F., Tsuji, K., Kageura, K., Jacquemin, C.: Detecting japanese term variation in textual corpus. In: Proceedings, Fourth International Workshop on Information Retrieval with Asian Languages (IRAL'99), Academia Sinica, Taipei, Taiwan (1998) 97–108

Index des Auteurs