RC25341 (WAT1212-124) December 20, 2012
Computer Science

IBM Research Report

Terminology Extraction and Deep Parsing

Arendse Bernth
Ossining, NY 10562
USA

Michael C. McCord
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 208
Yorktown Heights, NY 10598
USA

Carla Quinn
IBM Terminology
8200 Warden Avenue
Makham, Ontario L6G 1C7
Canada

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Terminology Extraction and Deep Parsing

Arendse Bernth
Independent Scholar
Ossining, NY 10562
USA
a_bernth@hotmail.com

Michael C. McCord
IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights NY 10598
USA
mcmccord@us.ibm.com

Carla Quinn
IBM Terminology
8200 Warden Avenue
Markham, Ontario L6G 1C7
Canada
cquinn@ca.ibm.com

Abstract

A terminological database is much more useful if it not only contains extracted terms, but also has informative context sentences associated with each term. Such examples enable the human user to understand the concepts denoted by the extracted terms, and this is valuable for document production and translation.

In this paper we describe an enhancement to IBM's terminology extraction tool, TermExt, for associating high-quality context sentences with each term, as well as the methodology behind this. One important ingredient is the use of full, deep parsing, and we explain how the ESG (English Slot Grammar) parser is used for both context sentence selection and the further TermExt enhancement of verb term extraction. Verb terms also play a role in selecting good context sentences for noun terms.

Key words: Terminology extraction, context sentences, verbal terms

1 Introduction

The importance of sufficient terminological resources has been stressed by researchers in many different fields, including document retrieval (Anick, 2001; Kozakov et al., 2004), information retrieval (Chien and Chen, 2001; Jing and Tzoukermann, 2001), summarization, abstracting (Oakes and Paice, 2001), and both human and machine translation (Castellví et al., 2001; Heid et al., 1996; Heid, 1999; Warburton, 2010).

A terminological database is much more useful if it not only contains extracted terms, but also has informative context sentences associated with each term. Such examples, if selected well, enable the human user to understand the concepts denoted by the extracted terms, and this is valuable for document production and translation (Feliu et al., 2004; Bernth et al., 2003). But much less attention has been given to this aspect of automatic term extraction, compared with term extraction itself. For example, Feliu et al. (2004) appears to describe a system where the contexts are added by hand by domain experts.

In this paper we describe an enhancement to IBM's terminology extraction tool, TermExt (Bernth et al., 2003; Warburton, 2010), for associating high-quality context sentences with each term, and we discuss the methodology behind this. One important ingredient is the use of full, deep parsing, and we explain how the ESG (English Slot Grammar) parser (McCord, 1980, 1982, 1993, 2010; McCord et al., 2012) is used for this. The latest TermExt extracts verb terms as well as noun terms, and the verb terms play a role in selecting good context sentences for noun terms.

IBM translates very large amounts of documents, and often several translators work in parallel on different documents related to one product. In order to preserve consistency and minimize post-editing, it is very helpful to provide the translators with pre-translated key product terms. This approach is quite common, and e.g. Heid (1999) describes a similar scenario.

TermExt outputs translatable terms that are deemed "important" for the translation process by virtue of their frequency, uniqueness, or translation difficulty (Warburton, 2010). The term candidates are either nominal terms, which can be single nouns or noun groups, or verbal terms, which can be single verbs or phrasal verbs. After a combination of automated and manual cleanup by trained terminologists, described in detail in Warburton (2010), the terms are sent to the translators ahead of time.

The ESG parser is used by TermExt for extracting terms as well as their associated context sentences. ESG is a *full* (or non-shallow) parser in the sense that it provides a complete, structured analysis for all parts of the sentence. ESG is a *deep* parser in the sense that it shows logical relations for many of the constituents: unwound passives, implicit arguments, common arguments in coordination, remote arguments in wh-questions, etc. The parse also shows normal surface structure of the sentence in the same tree. The use of full, deep parsing enables us, among many other things, to capture grammatically connected constituents such as the particle or prepositional phrase arguments of phrasal verbs, regardless of distance. At the same time, TermExt is highly robust and efficient (Warburton, 2010).

Addressing the need for formal structure of the output pointed out in Heid (1999), rather than presenting "raw" output, TermExt presents to the user the lemmatized form of each term candidate, along with the part of speech, frequency, and a list of context sentences, with options controllable by the user. The name of the input file where the context sentence was found is also included so that the terminologist can access the macro-context (Meyer, 2001), if necessary.

Term candidates that are already contained in specified terminology databases, or are non-recommended, or words unknown to TermExt (which are often neologisms or misspelled words), etc., are recognized and can be excluded in the output by user settings or be marked with special category markings identifying each of these cases. The degree of detail and the choice of output categories as well as use of various dictionaries to be used as exclusion dictionaries or term dictionaries are all controlled by the user.

The rest of this paper is organized as follows. Section 2 describes the role of full, deep parsing in terminology extraction. Section 3 describes the TermExt verb extraction, and Section 4 describes how TermExt identifies lexical, grammatical, and paralinguistic patterns indicative of informative context sentences. In Section 5 we present our results.

2 The Role of Full, Deep Parsing

While the traditional statistical technique of *frequency count* certainly plays a role in TermExt, deep linguistic processing plays the major role. Many people recognize that deeper linguistic processing benefits the term and example sentence extraction processes:

- **Improved accuracy:**
  - *Improved part-of-speech determination.* One of the conclusions of the study of terminology extraction tools presented in Castellví et al. (2001) is that "Most of the authors consider the POS disambiguation as one of the most important error sources." Parsing imposes more constraints on part-of-speech determination – see Heid (1999), Castellví et al. (2001) in the description of TERMINO, and Bernth et al. (2003).
  - *Improved recognition of boundaries.* Carl et al. (2004) states that the noise in the CLAT system could be considerably reduced by grammatical analysis because term recognition over phrase or sentence boundaries could be excluded.
  - *Improved general accuracy.* "We straightway [*sic*] dismissed the methods that are only statistical being incapable of satisfying the accuracy constraint." (Bourigault et al., 1996).
  - *Improved candidate contexts.* Heid et al. (1996) argues that full parsing is preferable, stating that "part-of-speech shapes do not constrain the candidate

contexts enough".

- **Improved functionality:**
  - *Recognizing non-contiguous modifiers.* A full parse allows the system to recognize non-contiguous modifiers, which is particularly relevant for extracting phrasal verbs as argued below in Section 3, and for determining good example sentences.
  - *Capturing linguistic variation.*
    In her seminal paper (Meyer, 2001), Ingrid Meyer mentions another difficulty that relates to the linguistic variation possible in real, running text. Specifically, she mentions the variation in position – left or right – of the term relative to a "defining" context. This is also a problem that is solved by a deep parse, since the parse reflects the grammatical relations between the various parts of the sentence, regardless of word order.
  - *Valency.* The analysis in Christensen (2002) demonstrates that recognizing actual valency patterns (the ones used) effectively identifies certain types of useful contexts.
  - *Different types of patterns.* Meyer (2001) distinguishes between lexical patterns and grammatical patterns as well as paralinguistic patterns. As we shall see, a parse is useful for both lexical and grammatical patterns.

Whereas Heid (1999) points out the desirability of identifying chunks and phrases, the terminology extraction system reported on actually only uses regular query expressions on input annotated with morphosyntactic agreement features. We take the idea of parsing further – improving accuracy by capturing non-contiguous modifiers and linguistic variation, and using valency patterns to identify useful contexts.

In spite of the wide acknowledgment of their usefulness, deep analysis and parsing are not used very often in the context of term extraction. One system that does use parsing is the term acquisition system developed at the Mayo Clinic and described in Harris et al. (2003) and Savova et al. (2003). However, this is an n-gram-centered system with a frequency filter that *additionally* uses Charniak's parser (Charniak, 2000). Verbs and verb phrases are extracted as well as noun phrases. The efficacy of hybrid methods has been confirmed (see e.g. Haller (2008)), and TermExt does indeed use both parsing and a frequency count. But the frequency counts are for the complete terms, replacing the focus on n-grams.

Where the Mayo Clinic system uses n-grams as the basic "handle" into the text string, and *then* applies parsing to determine whether an n-gram can be said to be a "complete syntactic node",[1] TermExt uses the ESG parse to delimit the phrases for consideration, and *then* uses the syntactic information provided by the parse to extract

[1] A *complete syntactic node* is defined as the complete word string dominated by a tree node as identified by the parser. These nodes are further limited to NPs, VPs and Ss, where the NPs and VPs must have from 1 to 5 words, and the Ss must have 4 or 5 words.

the term by e.g. stripping off irrelevant modifiers such as determiners, leaving the longest possible "n-gram" as a term candidate or looking at complements to determine phrasal verbs.

In this connection it is worth mentioning the different views of the noun term candidate for the Mayo Clinic system and TermExt. TermExt does not include the parts of NPs after the head noun in the extracted candidates, whereas the Mayo Clinic system does. Not many term extraction systems include post-modifiers, but for automatic ontology population, the ultimate goal of the Mayo Clinic system, post-modifiers could well make an important contribution by making the term more specific, and suggesting a place in the ontology. Precision, in the sense of including exactly the *useful* post-modifiers, may be a challenge worthy of further research.

In Savova et al. (2003) the point is made that parsing provides disambiguation. While we couldn't agree more with this point, it seems more straightforward to use the parsing as the first constraint and let the parse drive the extraction, the way TermExt does, since the n-grams are not really used for anything but counting the number of words in the term candidate. Additionally, since fewer n-grams would have to be considered, some increase in efficiency would be expected; parsing is done anyway.

The reason given in Harris et al. (2003) and Savova et al. (2003) for using node completeness as a criterion for termhood rather than looking at the node's internal structure is to decrease sensitivity to parsing errors. In other words, even if the parse is somewhat off within a given node, if an n-gram forms a complete node (*sans* stopwords) the n-gram is considered.

We agree that even very good parsers such as ESG or Charniak's are not perfect, but looking at node completeness to "neutralize" bad parses seems spurious, especially given the example described in Harris et al. (2003) used to justify this approach.[2] This seems like a "lucky" example in that the VP analysed by the parser happens to be a worthy NP term also. Parsing errors could as well show up in the complete node. Savova et al. (2003) does indeed mention a number of issues with the complete node criterion ranging from "distributed concepts", i.e. non-contiguous nodes – which caused them to make a cut-off at five content words for a multiword term – to modifiers that should not be considered part of the term, to short or fragmented sentences.

We would argue that using the more traditional approach of frequency is a better way to go. If a term candidate is reasonably frequent, it *will* appear in a variety of contexts, and presumably enough of these will be parsed decently, and the term be extracted. Relying on node completeness in order to increase recall may sacrifice precision. Of course, this is the usual trade-off between precision and recall, and one may be more important than the other for specific applications.

In contrast to Halskov and Barrière (2008),[3] who favor precision over recall, recall does indeed seem to be a concern for the Mayo Clinic system, which includes a recall-

[2] The phrase *requires max assist* is misparsed as *requires [that] Max assist* instead of the desired V+obj(n).

[3] This paper describes a system, WWW2Rel, that extracts semantic relation instances as a help for updating or expanding existing ontologies.

boosting frequency test applied to the n-grams. The Mayo frequency test differs from other frequency filters in that it is applied not to the *whole* term candidate but rather to each constituent word of the term; if just one word passes the test, the n-gram as a whole does. This obviously boosts recall for low-frequency multiword terms, and increasing recall is indeed the reason given for this design of the frequency filter. The fact that their system includes post-modifiers of the head word in the term may also necessitate this design of the frequency filter since the resulting variations in the term caused by different post-modifiers will cause several distinct, but related, terms, with individual, and hence lower, frequencies.

TermExt handles the issue of low-frequency terms by sorting the candidates in frequency order, and displaying the candidates down to any frequency threshold requested by the user, rather than completely discarding the low-frequency ones. This is completely in line with the analysis of Gillam et al. (2005), who were led to believe that collocations with lower frequencies can be of interest: occurrence of seven consecutive non-function words can be significant, regardless of frequency. Our choice also reflects the point made by e.g. Séguéla (2001) and Castellví et al. (2001) that termhood cannot be determined on syntactic or statistical grounds alone; review by a human is necessary. However, if the term candidate appears in a lexical pattern, it is a good indication of termhood.

In spite of using deep parsing, TermExt is both very fast and very robust. In fact, in contrast to the GlossEx system (Kozakov et al., 2004; Park et al., 2002), one of the few other systems to extract verbs, TermExt is routinely run on large collections of documents without problems. Terms may be extracted from a single file, or accumulated across a set of files, possibly very large – the largest attempted so far was 1/2 million files (Warburton, 2010). According to Kozakov et al. (2004), GlossEx suffers from a certain degree of lack of robustness, necessitating a *controller* that monitors the extraction process, and if necessary, interrupts it to save intermediate results to file and then restarts the extraction.

In addition to robustness, TermExt boasts support of over 100 file formats (Warburton, 2010).

Of course, as Séguéla (2001) says: the quality of a syntax-based system directly reflects the quality of the syntactic analysis. We are fortunate to have a very efficient and very high-quality parser, ESG, which disproves the generality of the statement made in Maynard et al. (2009) that full parsing is both extremely computationally expensive (speed) and inaccurate.

ESG is about 100 times faster than typical statistical parsers, and is more accurate (see study in McCord et al. (2012)). Additionally, you are more likely to get correct identification of the types of subphrases identified by shallow parsers, if you use a full parser because of its greater use of context. Even when ESG doesn't get a complete parse for a sentence, it creates a patched-together parse which will often have subphrases identified correctly.

3 Verbal Terms

Very few terminology extraction tools appear to extract verbal terms.

For TermExt, the issue became relevant due to the increased use of TermExt on user interface content for IBM. Many of the terms appearing in user interfaces are indeed verbal (Warburton, 2010).

But as pointed out by Harris et al. (2003) and Savova et al. (2003), it is also true in the domain of health care that many terms are verbal, and one might guess that many other fields could benefit from the automatic extraction of verbal terms, particularly given the tendency to "verb" nouns.

Verbs are also of interest due to the fact that they typically denote relations. In fact, WWW2Rel (Halskov and Barrière, 2008) *requires* the presence of a verb for extracting relations. Such verb-based relations play a role in our construction of informative context sentences, as we shall see.

3.1 Phrasal Verbs

In addition to extracting single verbs, TermExt extracts phrasal verbs. By "phrasal verbs" we mean verbs that have a PP (prepositional phrase) or particle complement. For example, the preposition "down" introduces a PP complement in *he rolled **down** the hill* but is used as a particle complement in *he rolled the window **down*** or *he rolled **down** the window.* Particles can be certain prepositions without objects, as in the example just given, or certain adverbs like "back" as in *roll the cover **back***. In the TermExt output, the particle uses are encoded in the part of speech as `vpt` and the prepositional uses as `vpp`.

The phrasal verbs are especially important to catch for translation purposes, as the meaning of e.g. "turn" is quite different in *he turned the corner* and *he turned the computer on*, and the proper translation depends on being able to distinguish these cases.

In a sentence, the verb part of a phrasal verb can be far separated from its preposition or adverb, and this makes it particularly desirable to be able to detect the non-contiguous modifiers. For example, the sentence

> *It is the meta-model **from** which new concrete instances will be **created**.*

in a document led to the creation of a `vpp` term entry for `create from`. And the sentence

> *The creation of an approximation to human speech by a computer **concatenating** basic speech parts **together**.*

led to a `vpt` term entry for `concatenate together`.

3.2 How Full, Deep Parses Help Detect Phrasal Verbs

Let us illustrate this for the first example of the preceding section. We show the ESG parse in Fig. 1.

The deep parse provides the connection between "create" and "from". The middle column of the parse tree shows the *word sense predications* for the sentence – for instance `be(2,1,4)` for "is". The first argument (`2` in this case) is like an event or entity argument for the word sense, and is typically the word number for the head word of that node. The following arguments are *deep* or *logical* arguments for the predication, and are specified by the entity numbers of the arguments. Now, note first that the entity number of the PP *from which* is `5`. And note that `5` is an argument of the predication `create1(12,u,9,u,5)`. (This is the argument filled by a "from"-PP for `create`.) So TermExt can see that we have a case of *create from* – even though the *from which* PP is not adjacent to *create*. It could be even further away if the extraposition of this "wh" phrase is taken further. So n-gram methods are not very suitable for recognizing such connections.

The GlossEx system (Kozakov et al., 2004; Park et al., 2002),[4] one of the few other systems to extract verbs, uses POS-tagging along with a shallow parser to identify NPs (up through the head nouns). While this certainly rules out some candidates for verbs, it is by no means clear that such an approach can handle non-contiguous *phrasal* verbs.

In fact, the highest-ranked 45 glossary items, out of a total of 9862, shown in Park et al. (2002), contain only 6 words that can conceivably be verbs,[5] and they are all single verbs.

In addition, as argued, their use of shallow parsing cannot constrain part-of-speech determination as much as a full parse.

A quite different approach to verbal term extraction is employed by the French CTBK system described in Condamines and Rebeyrolle (2001). This system uses a "generate and test" approach by applying derivational morphology to derive verbal candidate terms from the nominal candidate terms occurring in the documents; only if the resulting verbal term actually occurs in the document is it suggested to the user. This is an inventive work-around for a restriction posed by the shallow linguistic analysis they use,[6] which only identifies noun phrases.

Finally, the system described in Heid et al. (1996) mentions extraction of verbs, without giving any details except the general use of POS-tagging. In addition to extracting verbs, the system also extracts verb-noun (verb-object) collocations.

[4] For purposes of this paper, the distinction between terminology extraction and glossary extraction can safely be ignored.

[5] Part of speech is not stated.

[6] LEXTER; see Section 4.1.2

```
It is the meta-model from which new concrete instances will be created.
-----------------------------------------------------------------------
.------- subj(n)       it(1)               noun pron sg def perspron
o------- top           be(2,1,4)           verb vfin vpres sg vsubj
| .----- ndet          the1(3)             det sg def the ingdet
`-+----- pred(n)       meta-model(4)       noun propn sg notfnd
  | .--- comp(p)       from1(5,12,6)       prep wh nonlocp pobjp
  | | `- objprep(n)    which2(6,4,u)       noun pron sg pl sgpl wh
  | | .- nadj          new1(7,9,u)         adj erest adjnoun
  | | .- nadj          concrete1(8,9)      adj
  | .--- subj(n)       instance1(9,u)      noun cn pl evnt
  `-+--- nrel          will1(10,9,11)      verb vfin vpres pl wh vsubj
    `--- auxcomp(binf) be(11,9,12)         verb vinf
      `- pred(en)      create1(12,u,9,u,5) verb ven vpass
-----------------------------------------------------------------------
```

Figure 1: Non-contiguous modifiers

4 Context Sentences

There is no doubt that providing terminologists (and translators) with a high-quality context sentence adds considerable value to a term extraction system (Heid et al., 1996; Bernth et al., 2003; Warburton, 2010). A good example sentence enables the terminologist to do a proper conceptual analysis (Meyer, 2001), determining the meaning of the term as well as the scope of its domain.

In spite of this, little attention has been given to the automated extraction of good context sentences in the realm of automatic term extraction, but is rather found in the field of conceptual relation extraction or automatic ontology construction. Even though the work there is not directly aimed at *context* sentence extraction, many of the concerns are similar. Automatic identification of conceptional relations for use in a terminological database or an ontology cannot be fully automated (see e.g. Hamon and Nazarenko (2001)), but work has been done in this area. However, it is important to keep in mind that our objective is to identify contexts useful for translation purposes, as explained above, and not creating ontologies more or less automatically. Hence, there is a whole set of issues related to automatic ontology creation that we are not concerned with, such as determining the many ways *one* semantic relation can be expressed, mapping linguistic expressions to an ontology formalism, and placing the relation properly in the hierarchy. On the other hand, proper extraction of context sentences would most likely prove useful for creating ontologies. Indeed, in Sections 4.2.2 and 4.2.3 we shall see how certain constellations of known verbal or nominal terms and grammatical roles can also be used for extracting candidates for conceptual *relations*.

According to Meyer (2001), a good context sentence sheds light on either *attributes* of the concept or on *relations* which link a concept to other concepts. She calls such context sentences *Knowledge-Rich Contexts* or *KRCs*, and finds them expressed by certain types of *patterns* or *textual contexts* (Bourigault et al., 2001).

Meyer distinguishes three types of patterns: *Lexical* patterns involve certain lexical items such as “is part of”, “contains”, and “defined as”. *Grammatical* patterns are based on patterns of certain parts of speech, such as NOUN+VERB or ADJ+NOUN. *Paralinguistic* patterns form a catch-all category that includes other material such as punctuation and “various elements of the general structure of a text.”

Most work on extracting informative contexts has focused on identification of the lexical patterns and this is the focus of Section 4.1.

Whereas lexical patterns certainly are useful, they may be sparse (Condamines and Péry-Woodley, 2008): “The probability of occurrence of a [lexical] pattern, as well as its interpretation, are corpus-dependent.” Meyer (2001) says, “High-quality definitions are the exception rather than the rule in most of the corpora they [terminologists] work with.”

In the absence of lexical patterns, we still need to identify a useful context sentence. Since the grammatical patterns don’t depend on the occurrence of specific lexical items, this idea provides a good starting point. In Section 4.2 we’ll describe the use of a full parse to extract context sentences where term candidates play a grammatically significant role or co-occur with known terms in grammatically significant relations. This is an improved and expanded version of capturing the “aboutness” of the sentence that Meyer is aiming for with the grammatical patterns. Section 4.3 describes the use of some paralinguistic information to further assess the goodness of context sentences. Finally, in Section 4.5 we describe how to rank the contexts according to a “goodness score”. The higher the score, the more desirable the context is.

4.1 Lexical Patterns

Most work on extracting informative contexts or conceptual relations has focused on identification of *lexical patterns* – using the nomenclature of Meyer (2001). Whereas the overall idea is the same, viz. that certain patterns of lexical items are indicative of conceptual relations, the names are many. Ahmad and Fulford (1992) name them *Knowledge Probes*. Lyons (1981a, pages 292–295) in his discussion of hyponymy and meronymy uses the term *formulae*. Cruse (1986, page 13) uses *diagnostic frames* or *test frames*, and Winston et al. (1987), in their work on classifying meronomy simply says *frames*. Bowden et al. (1996) calls them *explicit relation markers*. Pearson (1998) names them *defining expositives* and states rules for these in Pearson (1996). Within the ontology effort on Ontology Design Patterns (ODPs),[7] the term *Lexico-Syntactic Pattern* (LSP) is used, and work is being done on identifying and cataloguing LSPs that correspond to ODPs for various languages. See e.g. de Cea et al. (2009) for a study of Spanish linguistic patterns. Another Spanish effort is the ECODE system (Martínez et al., 2008) that uses definitional verbal patterns to extract definitional contexts.

[7] See www.OntologyDesignPatterns.org.

The conceptual relations indicated by lexical patterns can be either *general* or *domain-specific* (Meyer, 2001; Séguéla, 2001). Domain-specific lexical patterns can be such as "shade of" in *Maroon is a shade of red* to indicate hyperonymy (Meyer, 2001) – this obviously hinges on proper word sense disambiguation. Similarly, in the medical domain, causality may be indicated by a variety of domain-specific lexical items such as "risk" or "complication", as in *The major short-term complications of CVS are pregnancy loss and diagnostic error* (Meyer, 2001).

General lexical patterns are typically of the form:

```
X is a Y
X is composed of Y
X occurs in Y
X is defined as Y
X is also known as Y
```

and other expressions of conceptual relations. More than one linguistic pattern can of course represent a certain conceptual relation.

A further distinction in the verb part of a pattern is made by Christensen (2004): Verbs (or verb groups), such as "made up of", "contain", and "include", that signal relations are called conjunctional verbs or relational verbs. Concept-related verbs are verbs like "define" and "characterize" that refer to the content side. Verbs like "call" and "denote" that refer to the expression side are called term-related verbs.

For TermExt, we have only considered so far lexical patterns that are general (not domain-specific). Further, for "defining contexts" (Meyer, 2001), contexts are provided regardless of whether the term appears in the role of *definiendum* or *definiens* since both have something potentially important to say about the term. Note, however, that rewarding the (surface) subject (see Section 4.2.2) will, for most patterns, automatically give preference to the *definiendum* over the *definiens*.

In order for the contexts to be maximally informative, we found it necessary to rule out contexts where either the *X* or the *Y* in the patterns are represented as a pronoun. In this we differ from the *relation-oriented*[8] approach of Christensen (2002), who sees being relation-oriented as an advantage since it will give results for terms that are represented by pronouns. For our purposes, this is not good since we really only want a *one*-sentence context and hence are not at all assured of getting the necessary antecedent that would disambiguate the pronoun. See also Grinsted (2000) for a similar view. TermExt does show the name of the file where the example sentence was found so that the user is able to refer to the macro context, but it seems better to try to avoid this when possible. Pronouns are also an issue for Halskov and Barrière (2008), who view pronouns as potential causes of noise when relying on lexical patterns to extract relation instances automatically from text.

[8] According to Meyer (2001), a relation-oriented approach provides results that show all term-sets that are linked by relation X, whereas a term-oriented approach shows the relations for a given term Y.

The ontology-building tool described in Carvalheira and Gomi (2007) appears to be one of the few ontological relation extraction systems employing pronoun resolution.

TermExt, like Meyer (2001) and Séguéla (2001), is *term-oriented*, even though TermExt actually can present relation-oriented results by changing user profile settings; more on this below in Section 4.6.

4.1.1 Lexical Patterns Used by TermExt

In this section, we list some of the lexical patterns used by TermExt. In the following table, each entry shows a verb (or verb group) "center" for the pattern, followed by one or more examples of a candidate term and a corresponding context segment obtained from that pattern.

- REPRESENT
  - *circuit*: From a more practical standpoint, the erlang is a measure of traffic intensity, where one erlang **represents** one circuit occupied for one hour.
- DEFINE
  - *Instance Composition*: The Instance Composition **defines** a relative path, called InstancePath, which is added to the GroupPath.
- INDICATE
  - *state table*: Information in an application profile **indicates** to the channel process what state table to load.
- SPECIFY
  - *subscriber type*: The system administrator **specifies** the subscriber type when creating the Message Center mailboxes.
- IS-A
  - *hot folder*: A hot folder **is a** directory that is associated with a logical destination.
  - *Certificate Authority*: The certificate is signed with a digital signature by the **Certificate Authority that is a** trustworthy authority.
  - *trustworthy authority*: The certificate is signed with a digital signature by the **Certificate Authority (CA), which is a** trustworthy authority.
- IS-THE-Y-THAT
  - *Application Server*: WebSphere Application Server 4.0 **is the** server software **that** hosts the Message Center Voice Interface Enterprise Application.
  - *framework operation*: A framework operation **is the** component **that** executes a business transaction.
  - *Connector component*: The AC Connector component **is the** framework component **that** provides compatibility with WSBC.
- IS-A-TYPE-OF
  - *media*: Media **are the types of** paper or forms on which jobs print.

- IS-A-KIND-OF
  - *component state*: Because a component state **is a kind of** Or state, all the local termination rules for Or states also apply to component states.
- AND-OTHER
  - *data field*: These tabs contain attributes **and other** data fields where you can set values.
- IS-CALLED
  - *extended editor*: These editors **are called** extended editors.
  - *Application Enabler component*: The tools are the Trace Facility, Server Monitor, Tivoli System Management Connection, and the functionality modules **are called** the Application Enabler components.
  - *central repository*: Support personnel can quickly make new transactions available to customers and employees by typing a few definitions into a central repository, which **is called** the Development Workbench.
  - Startup utility file: The Startup utility file **is called** Startup.xml and has the following sample structure:
- KNOWN-AS
  - *Flow Processor*: The Flow Processor, also **known as** the Automaton, provides an implementation of a state machine based on standard UML definitions, that helps a developer to define complex business operations as processes.
  - *Java Connector*: The Java Connector (also **known as** the Client/Server Mechanism) of the framework is based on the HTTP protocol, to which it adds the concept of a session between the client and server.
  - *Adobe Acrobat format*: The books are available in Portable Document Format (PDF) format, also **known as** Adobe Acrobat format.
  - *system*: Each system (**known as** a node) in the cluster is configured as either a client or a server.
- MADE-UP-OF
  - *composite graphic*: A composite graphic is **made up of** child graphics, which are positioned according to a composite layout by a Layout Manager.
  - *country code*: Country code is **made up of** characters between 'A'..'
- CONTAIN
  - *private workspace*: A user's private workspace only **contains** the current definitions being working on.
  - *function group*: A function group **contains** a set of functional interfaces for an AC.
  - *technical documentation*: The Information Center **contains** the technical documentation for WebSphere Adapters.
  - *central repository*: During the development process, a clear separation is established between the actual runtime application, and the definition of the processes to generate all these runtime files from the information **contained in** the central repository.
  - *secondary list*: After the help information is the secondary list that **contains** a list of tabs for that notebook.

- INCLUDE
    - *Lotus Connections*: Lotus Connections currently **includes** Profiles, Communities, Blogs, Social tagging and Activities.
    - *Enterprise Edition*: XL C Enterprise Edition for AIX **includes** the Mathematical Acceleration Subsystem (MASS).
    - *Rational Unified Process*: Rational Unified Process, **included** with Rational Method Composer, organizes projects in terms of disciplines and phases, each consisting of one or more iterations.
    - *Lotus Forms Turbo*: <strong>Lotus Forms Turbo</strong> is **included** in Lotus Forms and available as a stand-alone product to make it easy for non-technical users to quickly create, store and route eForms inside and outside the organization via e-mail and the web.
- REFER-TO
    - *adaptation*: Adaptation **refers to** the need to isolate interactions with other systems and to provide abstractions that insulate framework-based applications from the specifics of those other systems.
    - *data fetcher*: The software objects that retrieve external data are **referred to** as data fetchers.
    - *notification profile*: <ph style="bold">Objectname</ph> is the name of the destination or queue that this notification profile **refers to** (good example).
    - *system*: The system on which &liprodt; runs, specifically **referred to** as an IBM RS/6000.
- USED-FOR
    - *channel*: Two additional channels **are used for** synchronization, framing, and signalling.
    - *client*: An HTML client will generally **be used for** a home banking application built to use the framework.

4.1.2 The Importance of Grammatical Information in Identifying Lexical Patterns

Within the field of terminology, the traditional way (Bourigault et al., 2001; Meyer, 2001) of extracting lexical patterns is the so-called key-word-in-context (KWIC) approach, where you define a search window of a certain number of tokens around each term. This has the disadvantages of either producing too much noise if the window is too large, or too much silence, if the window is too small.

Attempts at improving the KWIC approach typically involve some kind of linguistic analysis, ranging from the very simple, like part-of-speech (POS) tagging, to shallow parsing, to full parsing.

For example, for German, the system described in Heid et al. (1996), which outputs two kinds of contexts, viz. sentences from a given language, and source and target pairs,[9] uses POS tagging. For French, the systems described in Condamines and Rebeyrolle (2001) and Hamon and Nazarenko (2001), respectively, use LEXTER (Bourigault et al., 1996; Bourigault, 1992), a shallow linguistic French analyzer dedicated to term extraction.

According to Bourigault (1992), LEXTER can get by with shallow linguistic analysis, in effect restricted to use of negative evidence given by morphological analysis – what cannot be a noun phrase is ruled out – and analysis of noun phrases for potential embedded terms, due to its limited goal of extracting nominal terms. However, with the expanded scope of their CTKB system to include extraction of conceptual relations from French text with the help of patterns and wanting to reduce overgeneration of terms, Condamines and Rebeyrolle (2001), found additional constraints necessary. Also Heid et al. (1996) states that for German, “part-of-speech shapes do not constrain the candidate contexts enough”.

The ontology-building tool described in Carvalheira and Gomi (2007) more directly utilizes syntactic information produced by NP and VP recognition, as well as named entity recognition and pronoun resolution, to extract semantic relations. The basic assumption is a strong correlation between the syntactic structure and the semantic relations among the entities that appear in a sentence. This is used to extract relations within noun phrases and among subject-object pairs. This is done in the GATE environment using JAPE grammars.

The analysis by Christensen (2002) does indeed demonstrate that deeper analysis is quite useful for determining the correct word sense of a verb, and shows how this word sense disambiguation effectively identifies KRCs for the *define* relation.[10] This analysis takes into consideration the variation in verb complements (valency patterns) (as well as weak semantic typing). The study is of Danish verbs, but Danish and English are similar enough that one could reasonably expect the results to carry over to English. In fact, Marshman and L’Homme (2006) describes work on using actantial structure[11] (i.e. the number of actants that relation verbs have, the order in which these actants appear, and the structures in which they participate, as well as semantic typing on the actants) to disambiguate English lexical knowledge patterns for cause-effect relations. Also the semantic relation instance extraction system, WWW2Rel, described in Halskov and Barrière (2008), encounters incorrect word sense as a source of noise for the cause-effect relation, mentioning that, in general text, *arise from* “almost always” indicates a cause-effect relation, but occurs as an indicator of locative relation in a medical text.

The word sense disambiguation system described in McCord (2004) also uses valency information as the basis for local context and contextual evidence, with frequency

[9] No details are given regarding how contexts are chosen or whether these are lexical or grammatical patterns.

[10] This study uses a character-string approach for searching.

[11] The authors prefer to use the terminology introduced by Tesnière (1959) and developed and generalized by Mel’čuk (see e.g. Mel’čuk (2004)) instead of valency patterns, but the overall point made is the same, viz. the value of using verb complement information for word sense disambiguation.

data feeding into a score that additionally uses the contextual evidence for senses. A somewhat similar approach to word sense disambiguation is described in Lin (1997); this also uses a broad-coverage parser and applies statistical measures to the arguments of words.

The report Christensen (2004), dealing with the same study as Christensen (2002), points out the importance of having more than the subject and object slots filled, in that verbs indicative of terms often have prepositional complements. The optional arguments which occur with prepositions can help to eliminate terminologically uninteresting patterns (noise) and ensure a more focused search. Hence the presence of prepositional complements helps distinguish between weak and strong knowledge patterns. A similar observation for English is made in Marshman and L'Homme (2006).

Both Marshman and L'Homme (2006) and Christensen (2004) recognize that not all word senses are suitable as terminological knowledge patterns, and each sense is dependent on a specific set of arguments (a specific valency pattern). Because Christensen does not have a (deep) parser, she needs to look at all the surface variations by hand to identify the useful valency patterns, which she then uses to constrain a character search in a corpus. Marshman and L'Homme (2006) makes a similar point in saying " ...there was an enormous variety in the surface realizations of actantial structures and actants for each of the patterns", and go on to say that automating this approach would, among other supporting technologies, require high-quality automatic parsing.

The common theme is that valency patterns are of great importance in understanding the lexical patterns. In Slot Grammar, valency information is strongly and flexibly captured by the central concepts of *slots* and *slot frames*. So the studies described in Christensen (2002) and Marshman and L'Homme (2006) provide a powerful argument for using full, deep parsing such as ESG provides. One could say that we share the philosophy of the importance of the valency patterns, but we differ in having the valency pattern supplied automatically by the ESG parse, and then we explore that to find a match with one of our knowledge patterns.

Systems that rely on a search window can have problems identifying knowledge patterns for sentences like (1), reported by Meyer (2001).

(1) a. This approach to composting is a viable method of dealing with animal carcasses.
  b. Vermicomposting, which is also known as worm composting (and means exactly what it says!), is an effective means of decomposing kitchen wastes when space is at a premium.

The referenced system uses the pattern "be* + ARTICLE" for extracting the *isa* relation and hence gets a false hit on "composting" in example (1a); "composting" in this sentence does not refer to a general concept but to a particular approach. This issue arises from the use of a search pattern within a given window to extract the relation. What we are really looking for is the subject of "be", which obviously is not "composting". Using a full parse, shown in Fig. 2, TermExt avoids this unwanted noise.

```
This approach to composting is a viable method of dealing with animal carcasses.
--------------------------------------------------------------------------
  .--------- ndet          this1(1)            det sg def
.-+--------- subj(n)       approach2(2,u,3)    noun cn sg act
| ‘--------- nobj(p)       to2(3,4)            prep pprefv motionp
|   ‘------- objprep(ing)  compost2(4,u,u)     verb ving
o----------- top           be(5,2,8)           verb vfin vpres sg vsubj
| .--------- ndet          a(6)                det sg indef
| .--------- nadj          viable1(7)          adj
‘-+--------- pred(n)       method1(8,u,9)      noun cn sg cognsa
  ‘--------- nobj(n)       of1(9,10)           prep pprefn nonlocp
    ‘------- objprep(n)    dealing1(10,u,u,11) noun cn sg evnt act
      ‘----- ncomp(p)      with1(11,13)        prep pprefv nonlocp
        | .- nnoun         animal1(12)         noun cn sg physobj
        ‘--- objprep(n)    carcass1(13)        noun cn pl physobj
--------------------------------------------------------------------------
```

Figure 2: ESG parse showing subject of "be" for *isa* relation

The problem with (1b) is determining the optimal window size. (Too broad tends to generate noise, whereas too narrow will miss a pattern.) In case (1b), the term "vermicomposting" is "quite a distance from the pattern *is an.*" (Meyer, 2001). However, full parsing enables us to capture the non-contiguous subject head, Meyer's concern, and correctly identify "vermicomposting" as the subject of "be". See the parse in Fig. 3. Our result is in direct opposition to the claim made by Maynard et al. (2009) that "... our patterns are defined at low levels of syntactic constituency, such as noun phrases, and by means of finite state transducers. Identifying and engineering on the basis of the linguistic building blocks that are relevant for each ontology editing task eliminates the need for a parser."

Another problem with KWIC, regular expression-based and shallow linguistic-based approaches that can be avoided by using a full parse is described in Maynard et al. (2009). The problematic pattern is the general pattern *NP have NP*, which when applied to *Writers have penguins based at the North Pole.* extracts the result "writers have penguins". This is again something that full parsing can fix, in that the two sentences have different structures, as shown below in the two parses that ESG produced without any adjustments. In the parse for *Writers have penguins based at the North Pole.*, shown in Fig. 4, the sense of "have" identified by ESG (`have3`) has three arguments: `have3(writers, penguins, based at the North Pole)`. Generally, `have3(x,y,z)` means `x` places, (or depicts, in this case) `y` in state `z`, which is the intended meaning here. So the parse does not mean "Writers possess penguins." However, in the sentence *Writers have penguins*, the parse of which is shown in Fig. 5, "have" is used in its more normal sense of possession or the like, as indicated by the

```
Vermicomposting, which is also known as worm composting (and means
exactly what it says!), is an effective means of decomposing kitchen
wastes when space is at a premium.
------------------------------------------------------------------------
.------------- subj(n)      vermicomposting1(1,u) noun cn sg
| | .--------- subj(n)      which2(2,u)           noun pron sg wh
| | .--------- lconj        be(3,2,5)             verb vfin vpres sg
| | | | .----- vadv         also1(4)              adv nounadv badadjmod
| | | `-+----- pred(en)     know2(5,u,2,6)        verb ven vpass
| | |   `----- comp(p)      as1(6,8)              prep nonlocp asprep
| | |     | .- nnoun        worm1(7)              noun cn sg h physobj
| | |     `--- objprep(n)   composting1(8,u)      noun cn sg
| `-+--------- nrel         and1(9)               verb vfin vpres sg wh vsubj
|   `--------- rconj        mean2(10,2,14)        verb vfin vpres sg
|     `------- vadv         exactly1(11)          adv ppadv
|     | .----- obj(n)       what2(12)             noun pron sg pl sgpl wh whnom
|     | .----- subj(n)      it(13)                noun pron sg def perspron
|     `------- obj(wh)      say1(14,13,12,u,u)    verb vfin vpres sg wh whnom vsubj
o------------- top          be(15,1,18)           verb vfin vpres sg vsubj
| .----------- ndet         a(16)                 det sg indef
| .----------- nadj         effective1(17,u)      adj
`-+----------- pred(n)      means1(18,19)         noun cn sg notnnoun act
| `----------- nobj(n)      of1(19,20)            prep pprefn nonlocp
|   `--------- objprep(ing) decompose1(20,u,22,u) verb ving
|     | .----- nnoun        kitchen1(21)          noun cn sg physobj artf strct
|     `------- obj(n)       waste2(22,u,u)        noun cn pl act massn sbst
`------------- vsubconj     when1(23,25)          subconj okadjsc oknounsc oknsubconj
  | .--------- subj(n)      space1(24,u)          noun cn sg abst location
  `-+--------- sccomp(bfin) be(25,24,26)          verb vfin vpres sg vsubj
    `--------- pred(lo)     at1(26,28)            prep pprefv staticp
      | .----- ndet         a(27)                 det sg indef
      `------- objprep(n)   premium1(28)          noun cn sg
------------------------------------------------------------------------
```

Figure 3: ESG parse showing *isa* relation in spite of great separation of term and “is”

```
Writers have penguins based at the North Pole.
-----------------------------------------------------------------------
.------- subj(n)    writer1(1,u)    noun cn pl h physobj ...
o------- top        have3(2,1,3,4) verb vfin vpres pl vsubj sta ...
`------- obj(n)     penguin1(3)     noun cn pl physobj anim unitph ...
`------- comp(en)   base1(4,u,3,5) verb ven vpass ri2 lcase ed
  `----- comp(p)    at1(5,8)        prep pprefv staticp ri1 lcase
    | .- ndet       the1(6)         det sg def the ingdet unitph lcase (def the)
    `--- objprep(n) North Pole1(8) noun propn sg location le3 (location)
-----------------------------------------------------------------------
```

Figure 4: "State" sense of "have"

```
Writers have penguins.
-----------------------------------------------------------------------
.- subj(n) writer1(1,u) noun cn pl h physobj ...
o- top     have2(2,1,3) verb vfin vpres pl vsubj ...
`- obj(n)  penguin1(3)  noun cn pl physobj anim  ...
-----------------------------------------------------------------------
```

Figure 5: "Possession" sense of "have"

sense `have2`. This is an excellent example of how the verbal complement pattern or slot frame can help determine the correct word sense as described in Section 4.1.2.

There is also the difficulty that Meyer (2001) mentions relating to the linguistic variation possible in real, running text. Specifically, she mentions the variation in position – left or right – of the term relative to a "defining" context. This type of problem is also reflected in the need for Christensen (2002) to analyze the different variations in surface patterns. The use of a deep parse solves this problem, since the parse reflects the grammatical relations between the various parts of the sentence, regardless of word order and surface variation. Another kind of linguistic variation, viz. morphological variation, is mentioned in Marshman (2004) as one of the bigger problems in connection with using regular expressions for extracting cause-effect relations.

Hence we think it is fair to claim that comprehensive morphosyntactic analysis such as provided by full, deep parsing, solves problems of word order and, generally speaking, reduces problems associated with the KWIC (and other shallow) approaches, by viewing context expressed as grammatical relations rather than as n-grams.

Some people take the exact opposite point of view. For example, the system for extracting conceptual hierarchies reported on in Gillam et al. (2005) uses an n-gram method, with a window of 5 words on each side after using a statistical POS-tagger and their hope is to completely avoid using POS information in order to more readily adapt their approach to other languages; however, as we have argued above, full, deep parsing does have significant contributions even for a language like English whose syntactic structure is particularly well suited to n-gram-based methods, and it is by no means clear to us that such an approach readily carries over to other languages with freer word order.

4.2 Grammatical Patterns

Meyer (2001) says of lexical patterns that "one has the sense that one will never get them all." Conversely, terms do not always occur neatly in context sentences exhibiting a lexical pattern.

These are good reasons not to restrict oneself to lexical patterns, but also to look for some more general properties of the term and its context since we still need to find the best context.

Meyer gets a start on this in her description of grammatical patterns as based on patterns of certain parts of speech, such as NOUN+VERB or ADJ+NOUN (Meyer, 2001).

However, with the help of full, deep parsing we can look for a much more constrained (and fuller) set of general characteristics (or *markers* as Condamines and Péry-Woodley (2008) and Marshman and L'Homme (2006) call them) indicative of desirable contexts. The presence of any of these characteristics can then be used for rewarding the context, adding to its score of desirability. Hence the score can be used to rank context sentences according to desirability even in the absence of specific patterns.

We shall consider both characteristics that apply on the segment level – see Section 4.2.1, and characteristics specific to nouns (Section 4.2.2) and verbs (Section 4.2.3), respectively.

Each user-configurable characteristic has a *characteristics code* that is used in the scoring profile as a means of referring to each characteristic. Section 4.5 describes the use of the scoring profile for ranking the context sentences.

4.2.1 Characteristics of the Segment

This section describes characteristics that apply on the level of the segment.

Segment Type A segment can either be a noun phrase or a sentence. Sentences are rewarded since they are more informative.

Incomplete Sentences Incomplete sentences are sentences that have a missing complement at the end, and they typically end in a colon and lead into a list.

Examples of incomplete sentences:

```
Dedicated persistent socket connections are intended to be:
Some of the algorithms used in cipher suites include:
To retrieve the queued output message on a dedicated persistent
socket, the client application must:
SSL protects information from:
The underlying message which triggers the other messages is:
In order to configure the system, you have to:
```

Incomplete sentences are undesirable and are given a score of 0, regardless of whatever else might apply. This causes these segments to be available if the input really does not provide any better contexts, but also allows these segments to be “pushed out” if something better is found. *This score is not user-configurable.*

4.2.2 Characteristics of Nominal Terms

This section describes characteristics that apply for terms that are nouns or noun groups.

Subject or Object The term is the (surface) subject or object of a finite verb:

```
main menu < n < 34
   < Upon entering administration mode,
     the main menu will expand to include links
     to the administration tasks.
client application < ex < n < 11
   < The client application is a Java applet
     that is downloaded on request from a web server.
```

Both subject and object are complements. However, subjects usually have more focus in the sentence (more “aboutness”) and are hence a better characteristic than objects, so the subject should be rewarded more.

Complement of a Non-Finite verb The term is a complement of a verb that is not finite, such as present and past participles, infinitives and imperatives:

```
e-mail message < n < 69 < To hear your e-mail messages,
                          say E-mail.

model object < n < 37
     < Note that deleting a model object deletes all
       the objects that are contained inside that object.
     < Obtain the model object that generated this message.
```

This situation is likely to be useful, but less informative than the case where the verb is finite.

Complement of a Known Term The term candidate is a complement of a verb that is already in the terminological database, i.e. the verb is a known term. This, as well as the complementary functionality described for verbs in Section 4.2.3, could also be used as a way of automatically deriving relations between terms for use in building up a knowledge base as described in Section 4.6. This idea is very similar to the approach to semantic relation extraction described in Halskov and Barrière (2008). The system WWW2Rel discovers new relation instances by using relation patterns with one argument instantiated and the other blank, corresponding to our known verbal term (the relation) and the uninstantiated argument (the new term candidate).

```
Application Server < n < 54
   < If you leave this box blank, WebSphere Application Server
      generates a name for you based on the Message Center
     Voice Interface Enterprise Application file.

Message Center Voice Interface Enterprise Application < n < 24
   < WebSphere Application Server 4.0 deploys the
     Message Center Voice Interface Enterprise Application
     into your voice interface application server.
```

4.2.3 Characteristics of Verbal Terms

This section describes characteristics that apply for verbal terms, be they single verbs or phrasal verbs.

The more complements co-occur with the verb, such as subject and object, the more information about the verb the context gives.

```
The process writes the result to a file.
```

is clearly more informative than

```
The results are written.
```

Hence we want to reward the presence of each of the two major complements: subject and object. As argued in Section 4.1, we consider pronouns uninformative and have decided to disregard contexts where the complement is a pronoun.

Given that the subject tells us more about the "aboutness" of the verb than objects, we think the presence of a subject should be rewarded more than the presence of an object.

Verb Has a Non-Pronominal Subject Based on the deep parse provided by ESG, we are able to recognize both the explicit subjects of finite verbs as well as many subjects implicit in nonfinite verbs.

```
back up < vpt < 4
   < The <TT>vvt_config</TT> script
      backs up the <TT>VVTdefaults</TT> file, so
      you can reapply the manual changes by using the
      information in the backup files.

concatenate together < vpt < 1
   < The creation of an approximation to human speech
     by a computer concatenating basic speech parts together.
```

Verb Has a Non-Pronominal Object

check off < vpt < 6
< Check off **the steps** when you have completed them.

hang up < vpt < 3
< A telephone line state, usually induced by hanging up
a receiver, in which the line is ready to receive a call.

Verb Is the Main Verb in the Sentence A verb that is the main verb in a sentence may have a more descriptive context than one that is "hidden" somewhere in a subordinate clause. The main verb can take several forms, a non-exhaustive list of which is given below.

The main verb can be finite:

bring up < vpt < 7
< This **brings up** the primary (first) list of help volumes.
< Selecting one of these tabs **brings up** a list of those
items that the system sets or that you can change.

The main verb can be an imperative:

fill in < vpt < 5
< **Fill in** the values for the italicized items exactly
as they appear in the entry you want to change.

A Complement is a Known Term The term has a subject or object (or prepositional object in the case of vpp verbs) that is already in the terminological database, i.e. a known term. This is quite in line with Meyer's view (Meyer, 2001) that a good context sentence sheds light on *relations* which link a concept to other concepts. It is also quite in line with Grindsted's view (Grinsted, 2000) that a way of reducing overgeneration for the lexical pattern *isa* is by using it only with a known term.

This, as well as the complementary functionality described for nouns in Section 4.2.2, could also be used as a way of automatically deriving relations between terms for use in building up a knowledge base as described in Section 4.6.

shut down < ex < vpt < 5
< **Channels** in an idle state are shut down immediately.

hang up < ex < vpt < 3
< A telephone line state, usually induced by
hanging up a **receiver**, in which the
line is ready to receive a call.

```
work with < vpp < 60
   < Validation also works with value types,
     which means Validable objects must validate their values.
```

4.3 Paralinguistic Patterns

Paralinguistic features most often (Lyons, 1981b) refer to the non-verbal aspects of *spoken* language, such as tone of voice and breathiness, as well as body language; for text-based communication, work has been done on techniques used to reflect these non-verbal aspects in a written context – change of font, capitalization, emoticons etc., see e.g. Hollingshead (2001); Lea and Spears (1992).

By extension, Meyer (2001) uses this term to denote a category of features that do not fit neatly into the lexical or grammatical patterns, such as punctuation and "various elements of the general structure of a text." In this paper we shall apply this broader notion of "paralanguage". In particular, it is worth noting that, for providing high-quality contexts for terms, the non-verbal aspects of communication of interest are quite different from the social and emotional information conveyed by the "classical" notion of paralanguage for written communication.

Below we will describe some paralinguistic patterns that we found useful. As opposed to the lexical and grammatical patterns, these "patterns" are perhaps better viewed as filters that rule out certain undesirable types of "contexts".

Meyer (2001) gives just four examples of paralinguistic patterns, two of which involve punctuation indicative of apposition. In contrast to Meyer, we consider punctuation as providing a *grammatical* function rather than a paralinguistic and shall not discuss those cases further here. Meyer also found "defining" questions, such as *What is compost?*, to be useful indicators of definitions or explanations immediately following; this is something we have not pursued, given our sentence-based approach.

The last example involves what she terms "dictionary defining KRCs." These have a rather formal strucure and typically involve typographical markup.

We find typographical markup immensely useful for ruling out certain uninformative contexts like the ones below that consists solely of an index entry or where the segment consists only of the term. Obviously, such a context is a very short context.

Index Tags Example of a segment consisting only of index tags:

```
<indexterm>programming models<indexterm>asynchronous output
</indexterm></indexterm>
<indexterm>dedicated persistent socket connections
</indexterm>programming
```

Contexts that Are the Same as the Term A context provided by a segment that consists only of the term is obviously not very informative.

Example of segments consisting only of the term:

```
Job Status < n < 4
   < <a href="User21.htm#print6a">Job Status </a>
Network Configuration Page < n < 4
   < <a href="User21.htm#print9">Network Configuration Page </a>
Printer Configuration Page < n < 4
   < <a href="User21.htm#print8">Printer Configuration Page</a>
```

Very short segments are usually not very useful, and segments like the ones described above are filtered out by giving them a non-configurable, bad score. Note however, that as Savova et al. (2003) points out, very short segments can actually be valid terms even though most often they just introduce unwanted noise. Also in connection with contexts, (IBM's) terminologists sometimes find them useful, and rather than totally disregarding such contexts, TermExt offers the possibility of directing the output to a special short-segment file instead of the regular output file, if desired by the user.

Sentence Length This is a balance between a sentence long enough to have a chance of saying something useful about the term candidate, and short enough not to risk burying the term-relevant information in less informative context. Based on our experience with the controlled-language checker EasyEnglishAnalyzer (EEA) (see e.g. Bernth (1997) and Bernth (1998)) we have chosen a sentence length between 8 and 25 words (inclusive) as a desirable length. Contexts falling within these limits are rewarded as specified in the user or default profile.

4.4 How Context Selection Works

In order to recognize the presence of a lexical, grammatical, or paralinguistic pattern for context sentence selection, TermExt explores the parse tree of any sentence in which the (previously determined) term candidate occurs, and compares the sentence with a catalog of predefined patterns or characteristics. This is done in a manner similar to that used in EEA for identifying forbidden or questionable constructions.

Namely, ESG expresses the parse tree in both a network and a phrase representation. Both representations offer convenient built-in exploration functions that work off a given node. These functions give access to features of the node, the slot filled by the node, the node's mother, as well as many other data.

Since TermExt is term-oriented, the "hook" into the parse is the node for the (given) term candidate, which is used as the starting point for exploring the parse according to the catalog of rules. For any rule that the parse fits, the "goodness" of the context is scored by rules using user-definable weights, as a means of ranking the context sentences. The ranking process is described in more detail below in Section 4.5.

4.5 Ranking Patterns

Term candidates often have many, many occurrences, which is a statistical aspect commonly put to good use by considering frequency counts.

However, this also means that there may be many different contexts! Above we have described a variety of possible context types; the issue now is how to select the best ones for presentation to the user. To that purpose we calculate a *score* for each context, which is then used for ranking. The user can specify how many context sentences to display, ranked in order of desirability; the default is five.

The semi-automatic ontology construction project reported on in Blomqvist (2008) also needs to decide on how to select the appropriate patterns from a pattern catalog. However, for Blomqvist, the purpose is to rank the fit of Ontology Design Patterns to the input in order to (semi-)automatically produce an ontology. This is done by string matching, and then the system uses measures that are relevant in the context of ontology construction such as concept and relation coverage, as well as density and proximity, to rank the matches found.

Rather than identifying the *ontological* pattern that best matches the input, *our* concern is to rank the contexts ("patterns") according to general informativeness as expressed by the syntactically based patterns and characteristics described above. Consequently, rather than using parameters that measure fit to an ontological structure, we mostly use syntactic parameters, and instead of using string matching we use exploration of the parse of the input sentence, a formal structure containing both the surface and deep structures, to supply the data needed to decide on the appropriate pattern. This gives us a great deal of flexibility in the irrelevant parts of the input and at the same time rather high confidence in identifying an actual pattern.

Like Blomqvist (2008), we found it useful to develop a score based on a linear combination of "parameters" (characteristics), but rather than giving them equal weights, we reward each characteristic individually.

The score is closely tied to the occurrence of the patterns and characteristics described above, the occurrences of which can be rewarded. The more of these occur in a given context, the more desirable the context is deemed to be. In other words, the score consists of the sum of these rewards, and the highest-scoring contexts can then be displayed to the user.

The rules have increasing specificity, and the cumulative effect of rewarding the success of each rule is that the score will reflect the highest level of desirability for that context.

For example, the pattern "TERM is a kind of" is good, but very specific. However, it is a more specific instance of "TERM is the subject of a finite verb".

So a simple example of a rule set could be:

```
if (TERM is the subject of a finite verb) reward 5;
if (TERM is a kind of) reward 10;
```

This will create a score of 5 for a sentence like

> The **business operation** is executed on the application server.

that only displays the first characteristic, but a score of 15 (5+10) for a context that displays both:

> The **root group** is considered to be a kind of Project.

Thus a higher score is a better score.

By having several components to the score, we allow each component to input to the total score, and the outcome is not as dependent on the success of any one component.

The example below shows a tracing of the total score as well as a breakdown into the triggered rules and their individual contributions to the overall score:

```
truststore file < ex < n < 3
   < A truststore file is a key database file (keystore) intended
     to contain public keys or certificates.
     Score: 247 [Rules: seglength 10; fullsent 20; issubj 7;
                 isa 100; contains 110]
```

The actual rewards for each pattern or characteristic can be specified by the user in a user profile, thus giving the user the possibility of adjusting the score according to the needs of the specific application or domain; in this we differ from Blomqvist who uses statistically calculated confidence values for each scoring aspect. In the absence of a user-specified profile, TermExt uses a default profile.

Here it is worthwhile mentioning a difference between how TermExt uses patterns and how Christensen (2002) uses them. Given a term, we use a pattern to rank its contexts, whereas she searches the context for a pattern in order to find a term.

4.6 How to use the profile for extracting relations

Using the profile to highly reward a specific pattern is a way of handling the need for extracting conceptual *relations* in addition to terms. If you increase the reward substantially (e.g. to 2000) for a given characteristic, that will cause this characteristic to figure prominently in the score and push any segment where this characteristic appears into the high, and hence "visible" (displayed), range of context sentences. This can be used to extract e.g. all sentences that contain a specific pattern, such as *isa*, indicative of hyperonomy/hyponymy. Conversely, by setting the reward for a certain characteristic to 0 (zero), the characteristic can be disabled completely.

5 Results and Impact

In this section we describe the practical impact that the enhancements to TermExt – especially the new context sentence selection – are having for IBM's docuument production and translation processes.

Every year IBM adds thousands of new terms to its information. We now find that if a term has been defined in the documentation, either partially or fully, TermExt picks up the definitional context which saves us time when it comes to defining these new terms. Our studies have shown that it takes on average 11 minutes for a writer to write a simple definition for a term (although it can take much longer). If we define 2200 new terms per year, providing contexts for even half of these terms saves over 200 hours of work. The quality benefits are more substantial as large numbers of new terms get accurate definitions that are used by our writers, translators, and customers.

As useful as context-rich context sentences are to writers, they are far more useful to translators. We estimate that out of all the the words we send to translation every year, 21 million words require attention in order for them to be translated accurately. Our translators tell us that it takes 10 minutes to translate a term. Providing translators with context sentences that help them determine the meaning of a term can substantially reduce the amount of time needed to translate the term, saving thousands of hours of translator time every year.

We deal with terms from many different industries and we frequently get terms that have a specialized sense in their domains that is different from their ordinary meanings. The translation for the specialized sense in the target languages is often different from that of the ordinary meaning. An example of this is "like" in Facebook, which can be translated in several different ways in many European languages. Providing meaningful, definitional context sentences can guide translators in selecting the correct translation for a specialized term, avoiding mistranslations that have to be fixed both in the original material and in the translation memories – an expensive process!

TermExt extracts verbs as well as nouns and this is very useful in providing terms for translators. Most technical terms are nouns, but there are some verbs with highly technical meanings that can be a challenge for translation, such as "enqueue", "cannibalize", or "prune". Many labels on software interfaces are verbs and it's important to translate these highly visible labels correctly and consistently.

Extracting verbs involving more than one word is a benefit for translators as well because these verbs present a challenge for translation. Providing translators with phrasal verbs such as "comment out" and "drill down", and multiword verbs such as "double click", "double tap", along with high-quality contexts, helps the translators to translate these verbs accurately and efficiently.

6 Conclusion

We have argued for the value of deep parsing in automatic term extraction, to improve accuracy in part-of-speech determination, to help identify non-contiguous modifiers such as those found with phrasal verbs, and not the least to help identify informative context sentences.

Our methodology for automatically identifying informative context sentences is built on exploration of deep parses to recognize lexical, grammatical, and paralinguistic patterns. The patterns are then scored for "goodness" by a simple, accumulative scoring algorithm so that the highest-ranked contexts can be presented to the terminologist. More than one context sentence can be presented to the terminologist in order to increase the information available to establish the meaning of a term. Poor contexts are filtered out by low scoring; only context sentences that meet a minimum score are extracted. Supplying only a limited number of contexts sentences, of a kind that terminologists agree on are informative, significantly reduces the workload of understanding the term compared with a KWIC approach where the terminologist is presented with all occurrences, informative as well as non-informative.

Additionally, a great deal of flexibility is built into the current systen in that the terminologist can configure various settings such as minimum score threshold and the score value of individual rules, allowing the advanced user to change settings in order to accommodate specific needs.

TermExt has proved itself by reducing time required for translation, resulting in savings as well as faster time-to-market for products, and by standardizing translations.

7 Acknowledgments

Kara Warburton had significant input to the research reported on in this paper when she was terminologist at IBM.

References

Khurshid Ahmad and Heather Fulford. Knowledge processing: 4. Semantic relations and their use in elaborating terminology. Technical report, University of Surrey, Guildford, Surrey, 1992. Computing Sciences Technical Report CS-92-07.

Peter G. Anick. The automatic construction of faceted terminological feedback for interactive document retrieval. In D. Borigault, C. Jacquemin, and M.C. L'Homme, editors, *Recent Advances in Computational Terminology*, pages 29–52. John Benjamins, 2001.

Arendse Bernth. EasyEnglish: A tool for improving document quality. In *Fifth Conference on Applied Natural Language Processing*, pages 159–165, Washington, DC, USA, 1997. Association for Computational Linguistics.

Arendse Bernth. EasyEnglish: Addressing structural ambiguity. In David Farwell, Laurie Gerber, and Eduard Hovy, editors, *Machine Translation and the Information Soup, Third Conference of the Association for Machine Translation in the Americas*, number 1529 in Lecture Notes in Artificial Intelligence, pages 164–173, Langhorne, PA, USA, 1998. Association for Machine Translation in the Americas, Springer-Verlag.

Arendse Bernth, Michael McCord, and Kara Warburton. Terminology extraction for global content management. *Terminology*, 9:1:51–69, 2003.

Eva Blomqvist. Pattern ranking for semi-automatic ontology construction. In *Proceedings of the 2008 ACM symposium on Applied Computing*, SAC '08, pages 2248–2255, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-753-7.

Didier Bourigault. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proceedings of COLING-92*, pages 977–981, Nantes, 1992.

Didier Bourigault, Isabelle Gonzalez-Mullier, and Cécile Gros. LEXTER, a natural language processing tool for terminology extraction. In Martin Gellerstam, Jerker Järborg, Sven-Göran Malmgren, Kerstin Norén, Lena Rogström, and Catarina Röjder Papmehl, editors, *Euralex '96 Proceedings, Part II. Papers submitted to the Seventh EURALEX International Congress on Lexicography in Göteborg, Sweden*, pages 771–779, Göteborg, Sweden, 1996. Göteborg University, Department of Swedish, 1996.

Didier Bourigault, Christian Jacquemin, and Marie-Claude L'Homme. Introduction. In D. Borigault, C. Jacquemin, and M.C. L'Homme, editors, *Recent Advances in Computational Terminology*, pages viii–xviii. John Benjamins, 2001.

Paul R. Bowden, Peter Halstead, and Tony G. Rose. Extracting conceptual knowledge from text using explicit relation markers. In Nigel Shadbolt, Kieron O'Hara, and Guus Schreiber, editors, *EKAW'96. Proceedings of the 9th European Knowledge Acquisition Workshop on Advances in Knowledge Acquisition*, volume 1076 of *Lecture Notes in Computer Science*, pages 147–162, London, UK, 1996. Springer-Verlag.

Michael Carl, Maryline Hernandez, Susanne Preuß, and Chantal Enguehard. English terminology in CLAT. In R. Costa, L. Weilgaard, R. Silva, and P. Auger, editors, *Workshop on Computational and Computer-assisted Terminology, LREC 2004, IV International Conference On Language Resources and Evaluation*, pages 10–13. European Language Resources Association, May 2004.

Luiz C.C. Carvalheira and Edson Satoshi Gomi. A method for semi-automatic extraction of ontologies based on texts. In Jean-Luc Hainaut, Elke A. Rundensteiner, Markus Kirchberg, Michela Bertolotto, Mathias Brochhausen, Yi-Ping Phoebe Chen, Samira Si-Said Cerfi, Marin Doerr, Hyoil Han, Sven Hartmann, Jeffrey Parsons, Geert Pols, Colette Roland, Juan Trujillo, Eric Yu, and Esteban Zimányi, editors, *Advances in Conceptual Modeling Foundations and Applications. Proceedings ER 2007 Workshops CMLSA FPUML ONISW QoIS RIGiMSeCoGIS*, number 4802 in Lecture Notes in Computer Science, pages 150–159, Auckland, New Zealand, November 2007. Springer.

M. Teresa Cabré Castellví, Rosa Estopà Bagot, and Jordi Vivaldi Palatresi. Automatic term detection: A review of current systems. In D. Borigault, C. Jacquemin, and M.C. L'Homme, editors, *Recent Advances in Computational Terminology*, pages 53–87. John Benjamins, 2001.

Eugene Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 132–139, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

Lee-Feng Chien and Chun-Liang Chen. Incremental extraction of domain-specific terms from online text resources. In D. Borigault, C. Jacquemin, and M.C. L'Homme, editors, *Recent Advances in Computational Terminology*, pages 90–109. John Benjamins, 2001.

Lise Lotte Weilgaard Christensen. Danish verbs as knowledge probes in corpus-based terminology works. *LSP and Professional Communication*, 2(2):77–93, October 2002.

Lise Lotte Weilgaard Christensen. Valency patterns of Danish verbs as terminological knowledge patterns. In R. Costa, L. Weilgaard, R. Silva, and P. Auger, editors, *Workshop on Computational and Computer-assisted Terminology, LREC 2004, IV International Conference On Language Resources and Evaluation*, pages 20–23, Lissabon, Portugal, May 2004. European Language Resources Association.

Anne Condamines and Marie-Paule Péry-Woodley. Linguistic markers of lexical and textual relations in technical documents. In Denis Alamargot, Patrice Terrier, and Jean-Marie Cellier, editors, *Written documents in the workplace*, volume 21 of *Studies in Writing*, pages 3–16. Elsevier Science, 2008.

Anne Condamines and Jossette Rebeyrolle. Searching for and identifying conceptual relationships via a corpus-based approach to a terminological knowledge base (ctkb). In D. Borigault, C. Jacquemin, and M.C. L'Homme, editors, *Recent Advances in Computational Terminology*, pages 127–148. John Benjamins, 2001.

D. Alan Cruse. *Lexical Semantics*. Cambridge University Press, 1986.

Guadalupe Aguado de Cea, Inmaculada lvarez de Mon, and Elena Montiel-Ponsoda. From linguistic patterns to ontology structures. In *8th International Conference on Terminology and Artificial Intelligence*, 2009. URL `http://www.irit.fr/TIA09/thekey/articles/montiel-aguado-alvarezdemon.pdf`.

Judit Feliu, John Jairo Giraldo, Vanesa Vidal, Jorge Vivaldi, and M. Teresa Cabré. The GENOMA-KB project: a concept based term enlargement system. In R. Costa, L. Weilgaard, R. Silva, and P. Auger, editors, *Workshop on Computational and*

Computer-assisted Terminology, LREC 2004, IV International Conference On Language Resources and Evaluation, pages 32–35. European Language Resources Association, May 2004.

Lee Gillam, Mariam Tariq, and Khurshid Ahmad. Terminology and the construction of ontology. *Terminology*, 11:1:55–81, 2005.

Annelise Grinsted. "Knowledge probes" og eksempler. På jagt efter definitioner og begrebsrelationer i et korpus inden for området "entrepreneurship". In Anita Nuopponen, Bertha Toft, and Johan Myking, editors, *I terminologins tjänst. Festskrift for Heribert Picht på 60-årsdagen, Vaasan Yliopiston Julkaisuja 13, Proceedings of the University of Vaasa, Report 13*, pages 36–51. Vaasan Yliopisto, University of Vaasa, 2000.

Johann Haller. AUTOTERM: Term candidate extraction for technical documentation (Spanish/German). *Revista Tradumàtica. "Terminologia i Traducció'*, 6, 2008. URL `http://www.fti.uab.cat/tradumatica/revista/num6/articles/04/04art.htm`.

Jakob Halskov and Caroline Barrière. Web-based extraction of semantic relation instances for terminology work. *Terminology*, 14:1:20–44, 2008.

Thierry Hamon and Adeline Nazarenko. Detection of synonym links between terms: Experiment and results. In D. Borigault, C. Jacquemin, and M.C. L'Homme, editors, *Recent Advances in Computational Terminology*, pages 185–208. John Benjamins, 2001.

Marcelline R. Harris, Guergana K. Savova, Thomas M. Johnson, and Christopher G. Chute. A term extraction tool for expanding content in the domain of functioning, disability, and health: proof of concept. *Journal of Biomedical Informatics*, 36: 250–259, 2003.

Ulrich Heid. Extracting terminologically relevant collocations from German technical texts. In Peter Sandrini, editor, *Proceedings of the TKE '99 International Conference on Terminology and Knowledge Engineering*, pages 241–255, Innsbruck, 1999. TermNet-Verlag.

Ulrich Heid, Susanne Jauß, Katja Krüger, and Andrea Hofmann. Term extraction with standard tools for corpus exploration - experience from German. In Christian Galinski and Klaus-Dirk Schmitz, editors, *Proceedings of the TKE '96 International Conference on Terminology and Knowledge Engineering*, pages 139–150, Vienna, 1996. Index-Verlag.

Andrea B. Hollingshead. Communication technologies, the internet, and group research. In Michael A. Hogg and Scott Tindale, editors, *Blackwell Handbook of Social Psychology: Group Processes*, page 564. Blackwell Publishers, 2001.

Hongyan Jing and Evelyne Tzoukermann. Determining semantic equivalence of terms in information retrieval. In D. Borigault, C. Jacquemin, and M.C. L'Homme, editors, *Recent Advances in Computational Terminology*, pages 246–260. John Benjamins, 2001.

Lev Kozakov, Youngja Park, Tong-Haing Fin, Youssef Drissi, Yurdaer Doganata, and Thomas A. Cofino. Glossary extraction and utilization in the information search and delivery system for IBM Technical Support. *IBM Systems Journal*, 43:3:546–563, 2004.

Martin Lea and Russell Spears. Paralanguage and social perception in computer-mediated communication. *Journal of Organizational Computing and Electronic Commerce*, 2:3:321–341, 1992.

Dekang Lin. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 64–71, Madrid, Spain, 1997. Morgan Kaufmann Publishers / ACL 1997.

John Lyons. *Semantics, 1.* Cambridge University Press, 1981a.

John Lyons. *Semantics, 1.* Cambridge University Press, 1981b.

Elizabeth Marshman. The cause-effect relation in a French-language biopharmaceuticals corpus: Some lexical knowledge patterns. In R. Costa, L. Weilgaard, R. Silva, and P. Auger, editors, *Workshop on Computational and Computer-assisted Terminology, LREC 2004, IV International Conference On Language Resources and Evaluation*, pages 40–43. European Language Resources Association, May 2004.

Elizabeth Marshman and Marie-Claude L'Homme. Disambiguating lexical markers of cause and effect using actantial structures and actant classes. In Heribert Picht, editor, *Modern Approaches to Terminological Theories and Applications*, volume 36 of *Linguistic Insights*, pages 261–285. Peter Lang, Bern, 2006.

Rodrigo Alarcón Martínez, Gerardo Sierra Martínez, and Carme Bach Martorell. ECODE: A pattern based approach for definitional knowledge extraction. In Elisenda Bernal Gallén and Janet DeCesaris Ward, editors, *Proceedings of the XIII EURALEX International Congress*, pages 923–928, Barcelona, 2008.

Diana Maynard, Yaoyong Li, and Wim Peters. NLP techniques for term extraction and ontology population. In *Proceedings of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 107–127, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press. ISBN 978-1-58603-818-2. URL http://dl.acm.org/citation.cfm?id=1563823.1563834.

Diana Maynard, Adam Funk, and Wim Peters. Using lexico-syntactic ontology design patterns for ontology creation and population. In Eva Blomqvist, Kurt Sandkuhl, Francois Scharffe, and Vojtech Svatek, editors, *Proceedings of the Workshop on Ontology Patterns (WOP 2009), collocated with the 8th International Semantic Web Conference (ISWC-2009)*, volume 516, pages 39–52, Washington D.C., USA, October 2009. CEUR-WS.org.

M. C. McCord, J. William Murdock, and Branimir K. Boguraev. Deep Parsing in Watson. *IBM Journal of Research and Development*, 56(3/4):3:1–3:15, 2012.

Michael C. McCord. Slot Grammars. *Computational Linguistics*, 6:31–43, 1980.

Michael C. McCord. Using slots and modifiers in logic grammars for natural language. *Artificial Intelligence*, 18:327–367, 1982.

Michael C. McCord. Heuristics for broad-coverage natural language parsing. In *Proceedings of the ARPA Human Language Technology Workshop*, pages 127–132. Morgan-Kaufmann, 1993.

Michael C. McCord. Word sense disambiguation in a Slot Grammar framework. Technical report, IBM T. J. Watson Research Center, 2004. RC 23397.

Michael C. McCord. Using Slot Grammar. Technical report, IBM T. J. Watson Research Center, 2010. RC23978REVISED.

Igor Mel'čuk. Actants in semantics and syntax i: actants in semantics. *Linguistics*, 42:1:1–66, 2004.

Ingrid Meyer. Extracting knowledge-rich contexts for terminography. In D. Borigault, C. Jacquemin, and M.C. L'Homme, editors, *Recent Advances in Computational Terminology*, pages 297–302. John Benjamins, 2001.

Michael P. Oakes and Chris. D. Paice. Term extraction for automatic abstracting. In D. Borigault, C. Jacquemin, and M.C. L'Homme, editors, *Recent Advances in Computational Terminology*, pages 353–355. John Benjamins, 2001.

Youngja Park, Roy J. Byrd, and Branimir K. Boguraev. Automatic glossary extraction: Beyond terminology identification. In *Proceedings of the Nineteenth International Conference on Computational Linguistics, COLING2002*, pages 1–7, Taipei, Taiwan, August 2002. Howard International House and Academia Sinica.

Jennifer Pearson. The expression of definitions in specialized texts: A corpus-based analysis. In Martin Gellerstam, Jerker Järborg, Sven-Göran Malmgren, Kerstin Norén, Lena Rogström, and Catarina Röjder Papmehl, editors, *Euralex '96 Proceedings, Part II. Papers submitted to the Seventh EURALEX International Congress on Lexicography in Göteborg, Sweden*, pages 817–824, Göteborg, Sweden, 1996. Göteborg University, Department of Swedish, 1996.

Jennifer Pearson. *Terms in Context.* John Benjamins, 1998.

Guergana K. Savova, Marcelline R. Harris, Thomas M. Johnson, Serguei V. Pakhomov, and Christopher G. Chute. A data-driven approach for extracting "the most specific term" for ontology development. In *Proceedings of AMIA Annual Symposium*, pages 579–583, 2003.

Patrick Séguéla. *Construction de modèles de connaissances par analyse linguistique de relations lexicales dans les documents techniques.* Doctoral thesis, UNIVERSITÉ TOULOUSE III, Ecole Doctorale Informatique et Télécommunications, Toulouse, France, March 2001.

Lucien Tesnière. *Éléments de syntaxe structurale.* Klincksieck, Paris, 1959.

Kara Warburton. Extracting, evaluating, and preparing terminology for large-scale translation jobs. In *Proceedings of Workshop on Methods for the automatic acquisition of Language Resources and their evaluation methods*, pages 1–6, La Valleta, Malta, May 2010. LREC.

Morton E. Winston, Roger Chaffin, and Douglas Herrmann. A taxonomy of part-whole relations. *Cognitive Science*, 11:417–444, 1987.

Ziqi Zhang, José Iria, Christopher Brewster, and Fabio Ciravegna. A comparative evaluation of term recognition algorithms. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odjik, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC08)*, pages 2108–2113, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).