Complex structuring of term variants for Question Answering

James Dowdall, Fabio Rinaldi
Institute of Computational Linguistics
University of Zürich
{dowdall,rinaldi}@cl.unizh.ch

Fidelia Ibekwe-SanJuan
ERSICO
University of Lyon3
ibekwe@univ-lyon3.fr

Eric SanJuan
LITA EA3097
University of Metz
eric.sanjuan@iut.univ-metz.fr

Abstract

Question Answering provides a method of locating precise answers to specific questions but in technical domains the amount of Multi-Word Terms complicates this task.

This paper outlines the Question Answering task in such a domain and explores two ways of detecting relations between Multi-Word Terms. The first targets specific semantic relations, the second uses a clustering algorithm, but they are both based on the idea of syntactic variation. The paper demonstrates how the combination of these two methodologies provide sophisticated access to technical domains.

1 Introduction

Nominal compounds are inherently ambiguous on both the syntactic and semantic fronts. Whilst the number of syntactic possibilities increase exponentially with word length (Isabelle, 1984), semantic interpretation is at best contextually dependent and in the worst cases determined by extra-linguistic (pragmatic) factors.[1] Technical documentation is an attractive domain in which to explore nominal compounds for two reasons. First, they present an abundance of compounds, secondly they restrict semantic interpretation by excluding compounds with opaque (extra-linguistic) interpretation. The result is multi-word terms (MWT) which are both compositional, their formation is a function of their constituent elements (Kageura, 2002) and endocentric, the compound is a hyponym of its head (Barker and Szpakowicz, 1998).

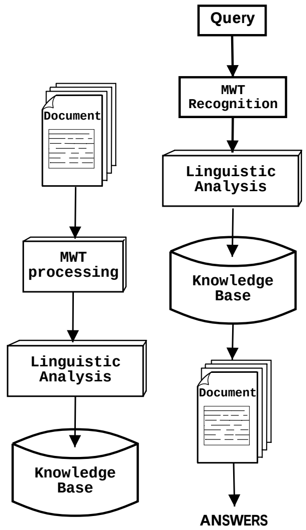
[1] For example, "`apple juice place`" (Levi, 1979)

This paper addresses the issue of structuring the Multi-Word Terms (MWTs) for Question Answering (**QA**) in technical domains. The central problem is that unfamiliarity with MWTs that characterize such domains creates an effective barrier against users finding answers.

Section 2 outlines the domain of focus, the MWT extraction method and examples characteristic MWTs. Section 3 explores the QA task in technical domains by describing the **ExtrAns** system, and how it structures the MWTs for the task. Section 4 presents **TermWatch** which identifies syntactic variants and uses a hierarchical clustering algorithm to build classes of term variants. The common ground between these two approaches is in the use of syntactic variants to structure the terminology as a whole. Section 5 explores how the resulting structures can be used in the QA task. After surveying some related work in Section 6 the paper ends by drawing conclusions on the approaches presented.

2 MWT Extraction

Before the MWTs can be structured, the terms need to be extracted from a corpus of texts. This stage was performed using the INTEX linguistic parser (Silberztein, 1993). INTEX is a finite state transducer parser. The corpus used in the present study concerns scientific publications on the bread-making process. It was made available by the French Institute of Scientific Information (INIST). Without going into much detail regarding the candidate term extraction rules, the approach adopted can be summarized as selective NLP followed by shallow parsing, much in the same way as (Evans et al., 1992). We defined morpho-syntactic properties of complex nominal syntagms written as finite state automata, implemented in the INTEX linguistic toolbox. INTEX is equipped with linguistic resources to perform



(a) Offline (b) Online

Figure 1: Schematic: ExtrAns Processing Stages

an overall morpho-syntactic analysis on the texts. The NP automata are applied in an iterative way on the corpus until we reach a satisfactory medium-grained noun phrase splitting. Our concern was to extract more or less complex terms as they appeared in the text corpus and not atomic NP extraction. The rationale was to conserve the associations between terms as the scientists (authors) made them during their write-up. Examples of candidate terms extracted at this stage are: "*hydrophilic powdered lecithin, traditional sour dough starter cultures, development of traditional bread flavour*". More details on the NP splitting rules can be found in (Ibekwe-SanJuan, 2001). Manual validation by a domain expert produced 3651 MWTs.

3 ExtrAns

Question Answering systems attempt to extract small snippets of text in response to a natural language query. Briefly, ExtrAns achieves this in two distinct stages:

Off-line the entire document collection is subjected to linguistic analysis, which produces a full syntactic parse for each sentence. After some intermediate steps, such as anaphora resolution and disambiguation, the syntactic parse is translated into a semantic representation designed to capture the core meaning of the sentences. These representations are stored in a Knowledge Base.

On-line user queries are subjected to the same linguistic analysis. The resulting semantic representation of the query is 'matched' against the knowledge base. These 'matches' can be identified in their original document location, so users can contextualize these potential answers. Interest in the specifics of this process should be directed toward (Rinaldi et al., 2002) (Dowdall et al., 2002).

In dealing with technical domains we have identified two major obstacles for a QA system which can be summarize as the Parsing Problem and the Paraphrase Problem.

The **Parsing Problem** consists in the increased difficulty of parsing text in a technical domain due to domain-specific sublanguage. Various types of MWT characterize these domains, in particular referring to specific concepts like tools, parts or procedures. These multi word expressions might include lexical items which are either unknown to a generic lexicon (e.g. "*acetic acid*") or have a specific meaning unique to this domain. Abbreviations and acronyms are another common source of incosistencies. In such cases the parser might either fail to identify the compound as a phrase and consequently fail to parse the sentence including such items. Alternatively the parser might attempt to 'guess' their lexical category (in the set of open class categories), leading to an exponential growth of the number of possible syntactic parses. Not only the internal structure of the compound can be multi-way ambiguous, even the boundaries of the compounds might be difficult to detect and the parsers might try odd combinations of the tokens belonging to the compounds with neighbouring tokens.

The **Paraphrase Problem** resides in the imperfect knowledge of users of the systems, who cannot be expected to be completely familiar with the domain terminology. Even experienced users, who know the domain very well, might not remember the exact wording of a MWT and use a paraphrase to refer to the underlying domain concept. Besides even in the documents themselves, unless the editors have been forced to use some strict terminology control system, various paraphrases of the same compound will appear, and they need to be identified as co-referent. However, it is not enough to identify all paraphrases within the manual, novel paraphases might be created by the users each time they query the system.

The task of QA in technical domains is to identify: 'what' needs to be known about 'which' multi-word term. Then to extract sentences that provide the answer. How to find the 'what' is dependant on the approach. ExtrAns uses linguistic processing which results in a semantic representation. However, in the TREC domain of newswire, considerable success has been achieved by statistical measures and even pattern matching. Here, these distinctions are unimportant.

What is of concern is in how to meet the two competing search needs of answering specific questions and navigating through a domain of specialized, unfamiliar MWTs.

Designed specifically for technical domains, ExtrAns involves strategies for exploiting the abundant MWTs that these domains hold. The approach utilizes WordNet to gather the MWTs into synonymy sets based on variation rules. The terminology is also related through an hyponymy hierarchy.

Synonymy between MWTs is either strict, or detected through WordNet. Strictly synonymous MWTs coreference a single object/concept. This link is a result of morpho-syntactic variation taking "chemical improver action" and producing the anitsymmetrical term "action of chemical improver". The process simply involves inverting the Head and introducing modifiers with a preposition.

WordNet synonymy, on the other hand, comes in three types of symmetrical variation depending on which tokens from two MWTs can be found in the same synset:

- WordNet Head substitution, ("*bread ingestion*" and "*bread consumption*")

- WordNet modifier substitution ("*quantity of yeast*" and "amount of yeast")

- WordNet Modifier and head substitution ("*key ingredient*" and "*functional component*").

However, synonymy identified through WordNet is defined by WordNet. As a general lexical database not designed for specilized domains it represents common synonymy between words. The resulting links created between multi-word terms translates into concepts non-specialists cannot easily distinguish. These links produced 1277 synsets the vast majority of which contain two MWTs.

Hyponymy The MWTs are organized into a lexical hyponymy (is_a) hierarchy that exploits their endocentricity (Barker and Szpakowicz, 1998). The hyponymy relation is identified through two types of rules, Left Expansions which further modifies "*dough stickiness*" to be "*intense*" producing "*intense dough stickiness*". Here the original head-modifier relations of the hypernym are unaltered in the hyponym. However, with Insertion rules these relations are changed in the potential hyponym. For example, whatever is going on in "*wheat dough stickiness*", inserting the word "*surface*" to produce "*wheat dough surface stickiness*" has altered the original head-modifier relations. So a generic/specific relation is less certain. For the moment such links are permitted.

This process allows multiple parents for a given term. So "*wheat dough surface stickiness*" is also a hyponym of "*surface stickiness*" through a left-expansion rule. An example of this kind of hierarchy can be seen in figure 2.

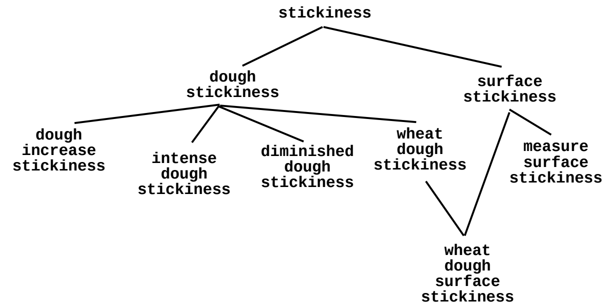


Figure 2: Hyponymy Hierarchy

These two structures are exploited in the search process during 'matching' of queries against answers. The strengths they bring and the limitations imposed are explored in Section 5 after description of an alternative approach to term variant structuring.

4 The TermWatch system

TermWatch (Ibekwe-SanJuan and SanJuan, 2003) clusters term variants into classes, thus producing a three-level structuring of terms: term, connected component and class levels. It integrates a visual interface developed with the *Aisee* graphic visualization to enable the user explore the classes and browse through the links between terms. Earlier stages of this work were presented in (Ibekwe-SanJuan, 1998).

The system comprises of two major modules: a syntactic variant identifier and a clustering module whose results are loaded onto the *Aisee* visualization tool.[2]

4.1 Variants identifier module

Automatic term variant identification has been extensively explored in (Jacquemin, 2001). In the sections below, we will recall briefly the definitions of the variation types we identify and give examples each type.

[2]http://www.aisee.com/

Expansions are subdivided along the grammatical axis: those that affect the modifier words in a term and those that affect the head word. **Modifier expansions** (L-Exp) describes two elementary operations: left-expansion (L-Exp) and Insertion (Ins). They both denote the addition at the leftmost position (L-Exp) or inside a term (Insertion or Ins) of new modifier elements. For instance, "*gas holding property of dough*" is a left-expansion of "*gas holding property*" because by transformation to a nominal compound structure, we obtain "*dough gas holding property*". Likewise, "*bread dough quality characteristics*" is an insertion variant (Ins) of "*bread characteristics*". **Head expansions** (R-Exp) describes the addition of one or more nominals in the head position of a term, thus shifting the former headword to a modifier position. Thus "*frozen sweet dough baking*" is a R-Exp of "*frozen sweet dough*". A combination of the two expansion types yield left-right expansion (LR-Exp) in that it describes addition of words both in the modifier and head positions. For example, the relation between "*nonstarch polysaccharide*" and "*functional property of rye nonstarch polysaccharide*" (*"rye nonstarch polysaccharide functional property"*). These relations are constrained in that the added or inserted words have to be contiguous, otherwise, we may not have the expected semantic relations. Only nominal elements are considered (nouns, adjectives).

Substitutions are also defined along the grammatical axis to yield two sub-types : modifier and head substitution. **Modifier substitution** (M-Sub) describes the replacing of one modifier word in term t_1 by another word in term t_2. Thus "*bread dough leavening*" is a modifier substitution (M-Sub) of "*composite dough leavening*". **Head substitution** (H-Sub) relates terms which share the same modifiers but different heads : "*effect of xanthan gum*" and "*addition of xanthan gum*". These relations are equally constrained in that they can only link terms of equal length where one and only one item is different, thus guaranteeing the interpretability of the relations. Substitutions, since they denote non-directional relations between terms of equal length, engender symmetrical relations between terms on the formal level: $t_1 \sigma t_2$. Their transitive closure creates classes of terms. For instance, a set of terms related by modifier substitution (M-Sub) seem to point to a class of "properties/attributes" shared by a same concept (the head word) as in "*bread texture, endosperm texture, good texture*" for binary terms and "*sour corn bread, sour dough bread, sour maize bread*" for ternary terms. In this last case, the changing properties seem to point to the possible specializations ("*sour-*") of the concept ("*bread*"). Head substitution on the other hand gathers together sets of terms that share the same "properties" (the modifier words), thus creating a class of "concepts". For instance, the set of term variants "*frozen dough baking, frozen dough characteristics, frozen dough products*". The common attribute is "*frozen dough*", shared by this class of concepts "*products, characteristics, baking*". (Ibekwe-SanJuan, 1998) already put forward the idea of these semantic relations and (Jacquemin, 1995) reported similar conceptual relations for his insertion and coordination variants.

4.2 Variant Clustering Module

The second module of *TermWatch* is a hierarchical clustering algorithm, CPCL (Classification by Preferential Clustered Link), which clusters terms based on the variations described above. The six elementary variation relations are represented as a di-graph. Clustering is a two-stage process. First the algorithm builds connected components using a subset of the variation relations, usually the modifier relations (L-Exp, Ins, M-Sub), these are the COMP relations. The transitive closure COMP* of COMP partitions the whole set of terms into components. These connected components are sub-graphs of term variants that share the same headword. At the second stage, the connected components are clustered into classes using the head relations (R-Exp, LR-Exp, H-sub), this subset of relations is called CLAS. At this stage, components whose terms are in one of the CLAS relations are grouped basing on an edge differentiation coefficient computed thus:

$$d_{ij} = \sum_{R \in \text{CLAS}} \frac{n_R(i,j)}{|R|}$$

where CLAS is the set of binary head relations (Exp_D, Exp_GD, Sub_C), and $n_R(i,j)$ is the number of variants of type R between components i and j. This coefficient is higher when terms of two components share many CLAS relations of a rare type in the corpus. Components with the highest d_{ij} are clustered first. The CPCL algorithm can be iterated several times to suit the user's requirement or until it converges. This means that the user is free to either set the number of iterations or leave the algorithm to do all the iterations until convergence. The user only has to specify which set of variations s/he wants to play the COMP and the CLAS role. In theory, this distinction is already made in the system but the user can change it. On the linguistic

Component 1	component 2
bromate measurement	*dough stickiness*
dough stickiness measurement	diminished dough stickiness
dough surface stickiness measurement	dough increase stickiness
stickiness measurement	intense dough stickiness
	measure surface stickiness
	soft red winter wheat lines dough stickiness
	surface stickiness
	wheat dough stickiness
	wheat dough surface stickiness

Table 1: Example of a class built by TermWatch.

level, a class contains at least two connected components, each comprising of sets of term variants around the same head word. Class here should be understood in a formal way: it corresponds to groupings of connected components resulting from a hierarchical clustering algorithm. They are not strictly defined semantically. Although, we find semantically related terms within these classes, the exact semantic relations involved between pairs of terms are not explicitly tagged. So on the semantic level, a class here comprises subsets of term variants reflecting, "class_of" relations (engendered by substitutions) and "hypernym/hyponym" relations (engendered by modifier expansions). For instance, Table 1 displays the term variants found in one class.

This class was built around two components, one structured around the concept of "*stickiness measurement*" (most frequent repeated segment) and the other around the concept of "*dough stickiness*". We can observe the COMP relations between term variants inside each component. The variants that initiated this class formation are in italics (the ones sharing CLAS relations).

The *TermWatch* programs have been implemented in the AWK language and can run on a Unix or Windows system. The system is computationally tractable and processing time is quite acceptable for real-life applications. For instance, it took 40 seconds on a normal PC running Windows to process a graph of 3651 term variants and to load the results onto the *Aisee* graphic interface. 33 classes of variable sizes were produced at the 3rd iteration of the clustering algorithm. The smallest class had 4 terms and the biggest 218 terms! So class size depends very much on the number and types of variation relations present in the initial graph.[3]

5 Combining the two systems

The two outlined methodologies use the existence of syntactic variation between multi-word terms to structure the terminology as a whole. However, each approach reflects a different aspect of this structure.

The ExtrAns approach is designed to identify explicit relations between terms. The results are (relatively) small synsets and a hierarchy of types. For TermWatch, the organizing principle results in larger classes of terms built around different head words related by syntactic substitution or expansion. Whilst, not specifically targeting semantic relations the classes do exhibit related terms. Some of these relations are definable within the classes. For example, the class presented in Table 1 contains all of the hyponyms of "*stickiness*" identified in ExtrAns (figure 2), but the relations are not rendered explicit in the class. Also the class contains other terms not involved in a specific hyponymy relation.

The utility of the classes is in capturing more "fuzzy" relations between terms whilst avoiding the problems of trying to define the relation. For example, how can the relation between t_1: "frozen sweet dough" and t_2: "frozen sweet dough baking" be defined ? The most obvious candidate is a **part_whole** relation but this is defendable only on a formal level: i.e. t_1 is a subset of t_2, but does that make t_1 really a part of t_2 in any semantic sense? In other words, is "frozen sweet dough" really a part of "frozen sweet dough baking"?

The *TermWatch* system does not grapple with this issue. The interest of these classes for the QA task is that they exhibit these fuzzy relations. These represent wider categories of terms to be used for specific search types. For example, when looking for general information on "*frozen sweet dough*" a user may well be interested in "*baking*" it, but when extracting specific information on the same term the relation is inappropriate. *TermWatch* was designed originally for scientific and technological watch (STW).

[3] *TermWatch* was initially designed as a scientific and technology watch system, hence the choices made in syntactic term variant definitions, the clustering algorithm and visualization mechanisms are tightly related to this application. A WWW interface is currently under construction to facilitate the return to the source texts through hyperlinks.

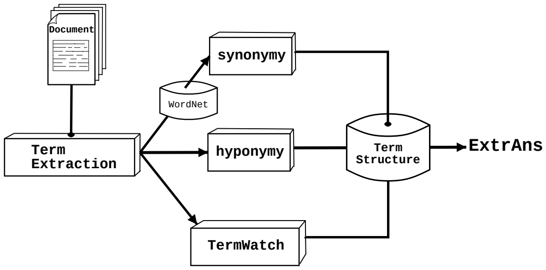


Figure 3: Using the structures in ExtrAns

In this type of application, the expert is less interested in strict semantic relations between terms in a taxonomy but more in capturing the association of research topics in his/her field. So such “fuzzy” relations become all important.

Currently *ExtrAns* uses the synsets and hyponymy hierarchy during the ‘matching’ of queries against documents. However, when this fails to locate anything the process is finished without providing users with any information or any further access into the domain. What is required is to “relax” the definition of semantic relation, or facilitate domain investigation through visualization of the terminology.

The combination of the two methodologies (depicted in figure 3) results in a terminology structured along four levels of granularity. This structure represents MWTs that are: Strictly synonymous, WordNet related, Hierarchy of types and Clustered by Class.

These levels can be effectively exploited in locating answers. First, extract potential answers that involve strictly synonymous MWTs. Second, look for potential answers with WordNet related MWTs. Third, try hypernyms/hyponyms of the search MWT. Finally, allow the user to browse the classes of MWTs to identify which are of interest in answer to the question.

TermWatch allows a user-friendly navigation of the clustering results. Classes are mapped out as nodes connected by edges whose length denote the distance between them. The longer the length, the farther the classes are from one another and thus the lower their edge coefficient (d_{ij}). The **Aisee** interface offers standard navigation functions which allow users to unfold a class into its components and then into the terms they contain. It thus reflects the three-level structuring effected by the *TermWatch* modules.

Figure 4 gives the graphic representation of results obtained on the corpus. Note that only classes linked to others are shown in this figure. Classes are labeled automatically by the most active term. The layout points out central or core classes, here classes (32, 22) which can represent the dominant terminology, and by extension, core research topics in the field. This layout also brings out interesting configurations like complete graphs and linear graphs. **Complete graphs.** The four classes labeled by the terms “*dough behaviour*” (32), “*wheat flour bread*” (29), “*wheat bran*” (6) and “*dough improver*” (20) form a complete graph. They are all linked by symmetrical head substitution relations. We found in these classes term variants like “*wheat flour dough*” (class 32); “*wheat flour bread*” (class 29), “*wheat flour supplementation, wheat flour blend, wheat flour fractionation*” (class 6), and finally “*wheat flour composition*” (class 20). This complete graph is thus structured around the two modifier elements “*wheat flour*” which can reflect a property shared by the concepts of these four classes. **Linear graphs.** The anti-symmetrical relations engendered by insertions and expansions generate linear graphs, i.e., chains of relatively long vertices starting from a central class to the border of the graph. The visualization tool naturally aligns the elements of these linear graphs, thus highlighting them. For instance, the linear graph formed by the three classes “*dough behaviour*” (32), “*frozen dough baking*” (10), “*dough procedure*” (21) is structured around the set of variants: “*frozen sweet dough* (32) → “*frozen sweet dough baking* (10) ← “*frozen dough baking”* (10). The last term “*frozen dough baking*” establishes a strong variation relation with terms in the third class (21) in which we found the modifiers “*frozen dough*” associated to three different head words: “*characteristic, method, product*”.

Given that the syntactic variations which helped group terms give off semantic links, and given our restricted definitions of variation relation (see 4.1), a user seeking information can be offered these class’s contents at this stage in order to see loosely related terms semantically which a terminological resource (thesaurus) or WordNet may not have identified. For instance, in the class shown in Table 1, many of the terms may not have been related by any semantic relation in WordNet (*bromate measurement* and *dough stickiness*) because none of the head or the modifier words are in any synsets. The clustering algorithm, brings these terms in one class because “*bromate measurement*” is a modifier substitution of “*stickiness measurement*” which is why they are in the same component. Both tell us something about “*measurement* (or rather about measurable objects). On the other hand, “*dough surface stickiness measurement*”, in the same component, is a left expansion of “*stickiness measurement*”. The

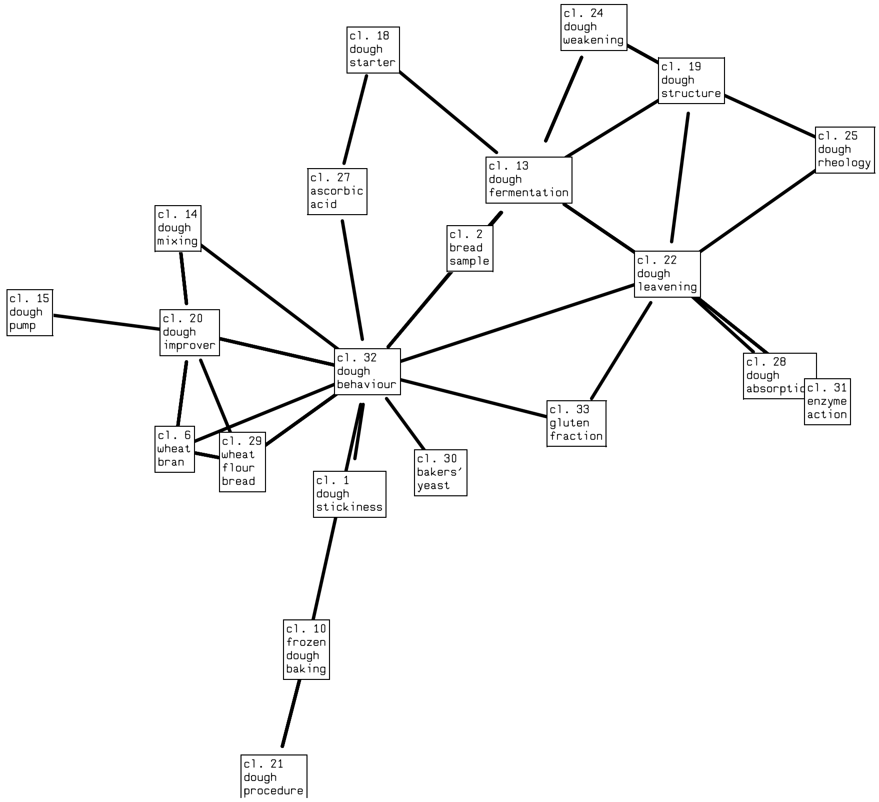


Figure 4: Navigating the clusters of MWTs

two could point to a 'hypernym/hyponym' relation. Thus, from link to link, these terms are connected to terms of the second component owing to the one anti-symmetrical link between "*dough surface stickiness measurement*" and "*surface stickiness*".

From this kind of investigation, a user can choose the MWTs of interest. This set then becomes the basis of a second round of answering specific questions. In this way the system can provide high precision access to answers, whilst facilitating navigation through a domain of unfamiliar MWTs.

6 Related Work

The importance of multi-word expressions (MWE) in various natural language tasks such as automatic indexing, machine translation, information retrieval/extraction and technology watch need no longer be proved.

The Multi-word Expression Project aims at studying the properties of a wide range of expressions including collocations, metaphors and terminology. The motivation is in explicitly defining the characteristics of such phrases. The results of the project will suggest efficient strategies for overcoming the problems MWEs cause for NLP applications (Sag et al., 2002)

Much work has been dedicated to the process of nominal compounding (Levi, 1979) and the semantic interpretation of nominal compounds (Downing, 1977) (Finin, 1980). Other works have addressed the specific problem of extracting nominal multi-word expressions for IR applications (Evans et al., 1992) (Smeaton and Sheridan, 1992) (Smadja, 1993) or of representing them semantically in order to enhance IR systems (Popowich et al., 1992) (Gay and Croft, 1990).

Many systems are dedicated towards structuring terminology for ontology building or terminology knowledge base construction (Aussenac-Gilles et al., 2003). These approaches use the corpus to identify linguistic markers which in turn point to certain semantic relations between terms (hypernym/hyponym, synonyms, meronyms). The approaches we describe are different in that relations are gained through syntactic variations between the terms.

Active research by the computational terminology community (Jacquemin, 2001) (Bourigault et al., 2001) (Pearson, 1998) has highlighted the importance of discourse as a means of capturing the essence of terms, hence as a good basis for structuring them. Jacquemin's extensive study has also highlighted the fact that terms are given to variations in discourse, so any endeavor to capture the relations between terminological units should integrate the variation paradigm.

7 Conclusions

Defining and identifying semantic relations between terms is problematic but can be utilized as part of the QA process. However, clustering MWTs based on syntactic variation uncovers classes of terms which reflect more "fuzzy" semantic relations. These are ideally suited to enabling navigation through the domain identifying terms to be used in the Question Answering process, offering sophisticated access to a domain. The resulting term structure can be utilized as a computational thesaurus or incorporated as part of a larger domain ontology.

References

N. Aussenac-Gilles, B. Biebow, and S. Szulman. 2003. D'une méthode a un guide pratique de modélisation de connaissances a partir de textes. In *Proc. of the 5th Conference on Terminologie et Intelligence Artificielle*, Strasbourg, March 31 - April 1.

K. Barker and S. Szpakowicz. 1998. Semi-Automatic Recognition of Noun Modifier Relationships. In *Proc. of COLING-ACL98*, Montreal, Quebec, Canada, August 10-14.

D. Bourigault, C. Jacquemin, and M-C. L'Homme, editors. 2001. *Recent Advances in Computational Terminology*, volume 2. John Benjamins.

J. Dowdall, M. Hess, N. Kahusk, K. Kaljurand, M. Koit, F. Rinaldi, and K. Vider. 2002. Technical Terminology as a Critical Resource. In *Proc. of LREC-02*, Las Palmas, 29 – 31 May.

P. Downing. 1977. On the creation and use of english compound nouns. *Language*, (53):810 – 842.

D.A Evans, R.G. Lefferts, G. Grefenstette, S.K. Handerson, W.R. Hersh, and A.A.Archbold. 1992. CLARIT TREC design, experiments and results. Technical report, Carnegie Mellon University.

T. Finin. 1980. The semantic interpretation of nominal compounds. In *Proceedings "Artificial Intelligence*, pages 310 – 312. Stanford.

L.S. Gay and W.B. Croft. 1990. Interpreting nominal compounds for information retrieval. *Information Processing and Management*, 26(1):21 – 38.

F. Ibekwe-SanJuan and E. SanJuan. 2003. From term variants to research topics. *Journal of Knowledge Organization (ISKO), special issue on Human Language Technology*, 29(3/4).

F. Ibekwe-SanJuan. 1998. Terminological variation, a means of identifying research topics from texts. In *Proc. of Joint ACL-COLING'98*, pages 564 – 570, Québec, 10-14 August.

F. Ibekwe-SanJuan. 2001. Extraction terminologique avec intex. In *Proc.of the 4th Annual INTEX Workshop*, Bordeaux, 10-11 June.

P. Isabelle. 1984. Another look at nominal compounds. In *Proc. of the 10th International Conference on Computational Linguistics (COLING '84)*, pages 509–516, Stanford, USA.

C. Jacquemin. 1995. A symbolic and surgical acquisition of terms through variation. In *Proc. of IJCAI95*, Montréal.

C. Jacquemin. 2001. *Spotting and discovering terms through Natural Language Processing.* MIT Press.

K. Kageura. 2002. *The dynamics of Terminology: A descriptive theory of term formation and terminological growth.* John Benjamins, Amsterdam.

J. N. Levi. 1979. *The syntax and semantics of complex nominals.* Academic press, New York.

J. Pearson. 1998. *Terms in Context.* John Benjamins, Amsterdam.

F. Popowich, P. Mcfetridge, D. Fass, and G. Hall. 1992. Processing complex noun phrases in a natural language interface to a statistical database. In *Proceedings COLING'92*, pages 46 – 51, Nantes, August 23 – 28.

F. Rinaldi, M. Hess, D. Molla, R. Schwitter, J. Dowdall, G. Schneider, and R. Fournier. 2002. Answer Extraction in Technical Domains. In *Proc. of CICLing 2002*, Mexico City, February.

I. A. Sag, T. Baldwin, F. Bond, A. Copestake, and D. Flickinger. 2002. Multiword Expressions: a Pain in the Neck for NLP. In *Proc. of CICLing 2002*, Mexico City, February.

M. Silberztein. 1993. *Dictionnaires Electroniques et Analyse Lexicale du Francais - Le Systeme INTEX.* Masson, Paris.

F. Smadja. 1993. Retrieving collocations from text: Xtract. *Computational Linguistics*, (19):143 – 177.

A. F. Smeaton and P. Sheridan. 1992. The application of morpho-syntactic language processing to effective phrase-matching. *Information Processing and Management*, 28(3):349 – 369.