Semantic Clustering and Disambiguation by Means of Word Usage Cues from Corpora

M. Victoria Arranz*
Centre for Computational Linguistics
Department of Language Engineering, UMIST
P.O. Box 88, Manchester M60 1QD, Reino Unido
E-mail: victoria@ccl.umist.ac.uk

Abstract

This paper focuses on the general problem of the *lexical bottleneck* and, in particular, on the issues of semantic clustering and disambiguation by means of word usage cues obtained from sublanguage-specific corpora. Our approach combines the use of numerical techniques with some symbolic modules. Our numerical tool *Dynamic Context Matching* is supported by three symbolic modules and a numerical one, which help considerably to reduce any remaining ambiguity. Furthermore, the development of a Unix-specific ontological knowledge hierarchy is also detailed. This ontology consists in a series of function categories, which reflect the different meaning aspects for each word as well as the relationships that can be established among these aspects and among the words holding them. Therefore, this hierarchy can be seen both as a semantic knowledge repository where all the semantic information extracted from the corpus is allocated, as well as an evaluation standard for our module, given that it contains all the information required to evaluate the clusters automatically acquired by the system.

1 Introduction

As already known, word sense disambiguation (WSD) is a major problem in NLP and is partly responsible for the more general problem of the *lexical bottleneck*. As the Consortium for Lexical Research (CLR, 1991) points out, the performance of many natural language processing systems is limited by the fact that their lexicons are too small. The information contained within the lexical items is essential for a series of tasks, thus making it necessary to ensure that this information is as accurate and complete as possible. This is the reason why the CLR is founded, so as to share any relevant information which might help us deal with the *lexical bottleneck*, such as lexical data, tools and even the results obtained from individual researches. This paper presents the work carried out in one of those researches, which develops a module for the discovery of semantic clusters and the disambiguation of their components by means of sublanguage-specific corpora.

Furthermore, the sublanguage-specific ontological knowledge hierarchy built is also described. This ontology[1] allocates all the disambiguated lexico-semantic information extracted by our process as well as serves as evaluation standard for our module.

2 Lexical Knowledge Acquisition

Some of the lexical gaps that can take place within the lexicon are, for instance, *missing words, compound words, word senses, collocations, metaphors, etc.* (Zernik, 1991). As expected, they might affect the performance of the different NLP applications in a variety of ways, but in general they all have a detrimental effect on them. As a consequence, lexical knowledge acquisition has become a necessary step to try and boost any system's performance and, thus, has become the concern of a large number of research studies currently taking place ((Velardi and Pazienza, 1989), (Calzolari and Bindi, 1990), (Church and Hanks, 1990), (Basili et al., 1994) and (Vanderwende, 1994)).

The approaches followed by such researches can belong to two broad categories (Boguraev and Pustejovsky, 1996): those that generate handcrafted lexical knowledge bases, and those that carry out an automatic extraction of knowledge from on-line resources. Despite the fact that the former approach is still rather popular, due to its low startup costs in terms of both the resources and tools required, it is the automatic approach that has received a constantly increasing support over the past few years, with the application of new analysis techniques, such as those based on probabilistic theories. Be-

* Research sponsored by the *Departamento de Educación, Universidades e Investigación* of the Basque Government, Spain.

[1]The term *ontology* is used here as a synonym of ontological knowledge hierarchy/classification.

low follows the description of our approach to the semi-automatic extraction of semantic clusters and their disambiguation by means of a combination of both numerical and symbolic methods.

3 *Semantic Clustering and Disambiguation*

Clustering belongs to a higher level of analysis called *multivariate analysis*. In general, cluster analysis groups words into classes that reflect similarity or dissimilarity in some pre-established property. In the current work, this technique is applied to associate words which present similarity in their semantic activity, i.e, in their usages, and such phenomenon is manifested in their contexts. By means of a similarity measure called *Dynamic Context Matching* (Arranz et al., 1995), our clustering procedure computes the similarity between the contexts in which these words occur and then groups them accordingly, thus indicating whether the words under study are semantically related or not in those usages.

Regarding word sense disambiguation, approaching the study of meaning from a general language point of view can become a daunting task in cases where, for instance, a system faces a wide semantic choice without the right help to handle it. Unfortunately, this general language approach can easily increase further the already difficult job of studying words' meaning, which has encouraged us to pursue a sublanguage-based methodology. As already known, the lexical and semantic restrictions encountered in any sublanguage (Lehrberger, 1986) prove the finiteness of these specific languages in a variety of issues, such as the lexicon. Given that a sublanguage attempts to cover the language needed to describe a more limited conceptual area, a smaller number of words is required to build its lexicon. Likewise, sublanguage usage also imposes other types of semantic restriccions, for example, on the classes of subjects and objects that certain sublanguage-specific verbs can take. All these different issues on semantic restrictions contribute noticeably towards a joint reduction in lexical ambiguity.

4 *Approaches to Semantic Clustering and Disambiguation*

Numerous word clustering techniques have emerged in the last few years which are based on lexical distributions, such as (Grefenstette, 1992) and (Grishman and Sterling, 1994). Among these approaches, a majority shares the concern of tackling the data sparseness problem. In particular for stochastic methods, low-frequency occurrences represent a serious hurdle for probability estimation. In contrast with most of these methods, our clustering procedure uses relatively-small corpora (100,000 words approximately), which prevents us from applying purely statistical techniques and thus helps us to avoid the aforementioned data sparseness problem, as well as the opacity caused by such techniques.

WSD methods using existing lexical knowledge sources have received considerable attention in the past few years. According to the type of knowledge source employed, these techniques can be divided into those using **MRDs** (Cowie et al., 1992); those using the broad coverage semantic taxonomy **WordNet** (Resnik, 1995) and (Rigau et al., 1997), and those using the **Roget's Thesaurus** (Yarowsky, 1992). However, our disambiguation procedure focuses on the study of sublanguages and the semantic information contained in such specific languages cannot be extracted from broad-coverage general language resources. This is specially the case if the sublanguage under study is as technical as ours is: the Unix on-line manual. Therefore, we have adopted a corpus-based approach, since the only place to search for such type of knowledge is where it actually takes place, i.e., in a corpus.

5 *Combining Numerical and Symbolic Methods*

Semantic clustering and disambiguation are performed by the joint effort of our main discovery tool *Dynamic Context Matching* together with a series of problem-solving submodules. Figure 1 provides an overview of these modules within our entire knowledge acquisition process. Basically, *Dynamic Context Matching* is a context matching technique which can be applied, for instance, to the disambiguation of polysemous terms by comparing their surrounding contexts. This comparison is carried out by looking at the matches between the individual words within each context and then calculating the different possibilities of total match values. Among all the different match values for each pair of contexts, the maximum subset is extracted, which represents the similarity value for those environments[2]. All the representative matching values acquired for the different contexts are stored in a *correlation matrix*, which the clustering program uses in order to build the semantic clusters for the different existing word senses. As it can be observed, word sense is studied here in terms of word usage. A word's different meanings, or senses, are established according to the different uses of the word in the different contexts,

[2]For a more detailed explanation on how the matching is performed, refer to (Arranz et al., 1995).

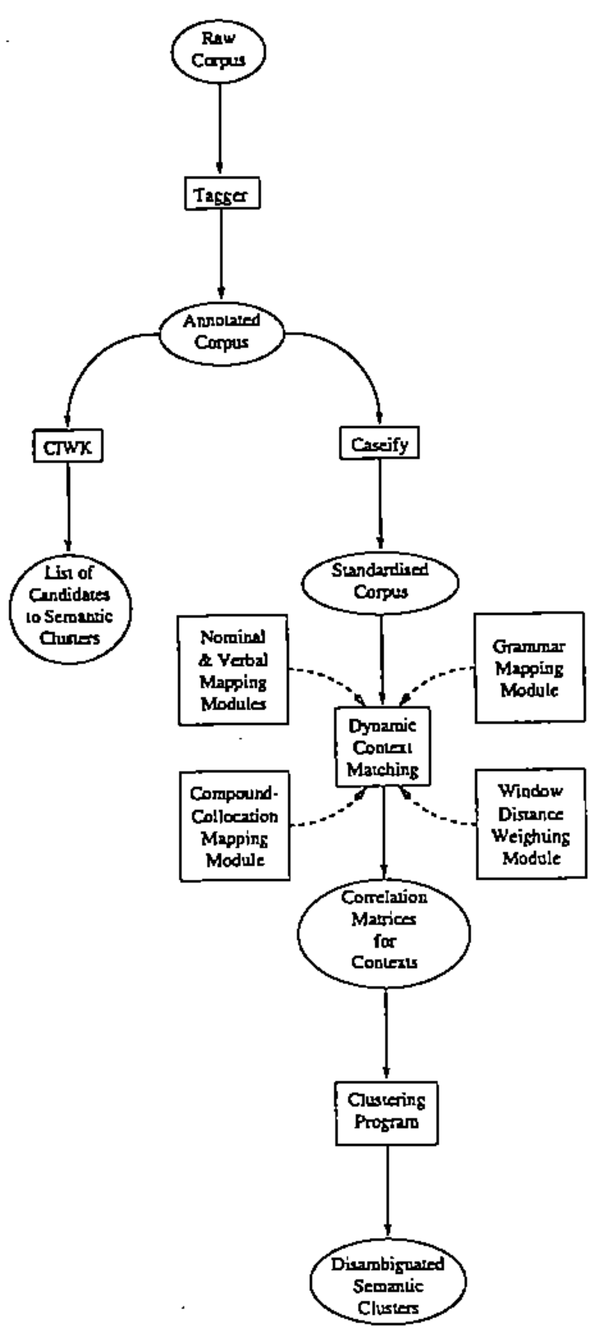

Figure 1: Knowledge Acquisition Framework

which allows us to explore a very dynamic and real-data-based notion of meaning. For more details on this approach to the study of meaning, cf. section 6.

Further on our semantic clustering and disambiguation module, despite the fact that our similarity technique represents a much more flexible approach than previously explored techniques (Arranz et al., 1995) and that it offers promising results, it also encounters certain ambiguity problems that have required the implementation of some problem-solving submodules. These new submodules interact in an easy-to-use

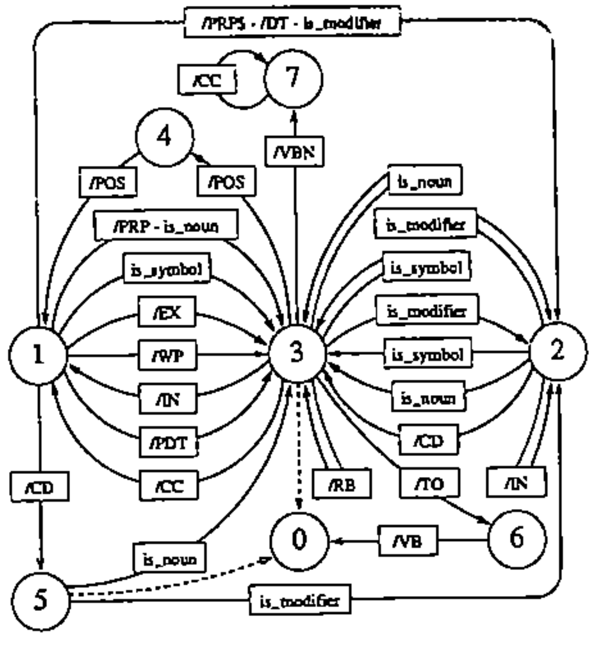

Figure 2: Network for NP Recognition

manner with the main tool *Dynamic Context Matching* and represent a very effective means to handle those ambiguity problems. Firstly, structural changes or differences existing in the corpus are an important cause of ambiguity given that *Dynamic Matching* has no means to match, for example, active and passive voice constructions. This problem is further increased by the fact that cross matching (i.e., matchings across the keywords, between elements on different-side environments) are not allowed by *Dynamic Context Matching*'s principle of linear ordering constraint. In order to solve these problems, a **grammar mapping module** has been developed, which prepares the contexts for matching by restoring their canonical order. This allows us to recover all those constituents, such as NP-subjects or -objects that have been structurally moved and have thus caused a loss of information. This mapping module is based on a deterministic finite state automaton, as it can be seen in its transition network for NP recognition shown in figure 2.

Secondly, and also related to *Dynamic Matching*'s linear ordering constraint is the problem suffered by certain compound structures. Certain matches and, consequently clusters, are lost due to the change of order within compound structures. To solve this problem, the **compound-collocation mapping module** has been developed, which locates possible candidates to this problem and establishes the order alterations to be permitted within the compound structures before matching takes place. Both our **grammar** and **compound-collocation mapping modules** prove to be very efficient in helping us to recover semantically related items, given that our Unix corpus comprises a very high number of both passive voice structures and compound/collocational elements that need to be matched.

Thirdly, another module developed to interact with *Dynamic Context Matching* is the **morphological mapping module**, which consists in both a nominal and a verbal module that deal with differences in number for nouns (such as, for example, *algorithm/NN* - *algorithms/NNS*[3]), and in verbal forms (such as *add/VB* - *added/VBN*), respectively. It is by means of these modules that we manage to recover a great number of matches that are lost for some slight variation in their morphological endings.

Finally, the last problem-solving module developed is the **window distance weighting module**. This is in charge of helping to avoid certain mis-clusterings by means of prioritising important contextual cues within the environments. Given that there is a strong tendency to find the most relevant terms very close to the keywords under study, this module calculates the importance of the words within a window by looking at their position with relation to the keyword. That is, a full weight is attributed to those matches immediately preceding or succeeding the keyword, given a full-value window size, while the rest of the matches are measured with a decaying weight, which decreases with distance from the keyword.

As it can be observed, a series of symbolic modules (**grammar**, **compound-collocation** and **morphological mapping modules**) and a numerical one (**window distance weighting module**) have been incorporated to the general framework of our knowledge acquisition process, which support successfully *Dynamic Matching* to resolve some remaining ambiguity problems.

6 *Unix-Specific Ontology*

6.1 Word Meaning and Lexical Semantics

As introduced in the previous section, the study of meaning and words' senses follows a different approach in the present work. Some of the principles behind this approach are related to those behind Pustejovsky's notion of *Qualia Structure* (Pustejovsky, 1991). Close observation of the textual data during the clustering and disambiguation processes has shown that a word is not an isolated element defined by an already fixed primitive. On the contrary, a lexical item represents a relational component, which acquires its meaning from its particular usage in the context, i.e., by means of a combination of factors which result from its relation with the other elements in the sentence. Therefore, the key issue to discover a word's meaning lies in the study of the relationships which exist between that word and the rest of the context.

Furthermore, the study of the Unix sublanguage has led us to believe that *usage* is a crucial point in order to explore lexical meaning. In particular when studying such a restricted language, a word may alter its meaning so drastically that it will only make sense if considered in its particular *usage*. Thus, it can be seen that there is a merge between the notion of *usage* and that of *meaning* in our conception of the lexicon. Given that in our extraction of semantic information we are concerned with the generative nature of the elements under study, our interest lies in the extraction of semantic clusters and disambiguation in terms of all the meaningful relations that exist between the elements in the context. It is with respect to certain aspects of those relations that the semantic properties of the words should be explored.

Moreover, and as a consequence of this relational study of the words, meaning becomes a multi-dimensional issue, since it is constructed out of the combination of the different contributions within the context. For example, when examining the semantics of a noun *N*, its meaning will be a compendium of the semantic information obtained from the relations with its surrounding elements in context, such as verbs, other nouns, or any other semantically related argument. It is through all this very specific information that our tools will determine whether noun *N1* is related to *N2* and if so, how closely related they are.

Bearing all this in mind, a conceptual hierarchy has been developed for Unix. Moreover, this ontology has been individually applied to 20 cases of semantic clusters and their respective hierarchies have also been created (see following sections).

6.2 Hierarchy Structure

Both the general and individual semantic hierarchies are developed manually, following the lists of contexts used by *Dynamic Matching*. This implies that relations are established in terms of local contexts, in an attempt to reflect the associations between the cluster keywords and their surrounding semantic patterns. In fact, it is during the clustering process that it is observed how semantic groups are created according to the types of specific *roles* that the keywords relate to, which makes the clustering and disambiguation processes very sublanguage-specific. That is, keywords are considered se-

[3]The corpus has been tagged a priori with Brill's tagger (Brill, 1993).

mantically related if they share any of the surrounding *roles*, or implicit syntactic functions.

Taking this into consideration, and also given that our ontology is aim-oriented and focuses on that knowledge which is relevant for both semantic clustering and disambiguation, the Unix ontology compilation process concentrates on capturing an abstract wide-coverage semantic knowledge hierarchy. However, depending on the grammatical categories of the cluster keywords under study, the ontology can be structured in three different ways. Given that meaning is approached based upon the ideas of *word usage* and *multi-dimensionality of meaning*, the type of semantic information required to study a noun will differ from that required for a verb. As a consequence, our Unix ontology can be viewed from three different perspectives which relate to the three types of content words that can be studied: nouns, verbs and adjectives. Figure 3 shows the abstract hierarchy developed for nominal terms, where the following four semantico-functional classes are used to allocate the semantic knowledge encountered in the neighbouring contexts:

1. *Objects*: In general, it refers to the surrounding nominal elements that can collocate with a noun keyword, creating thus a potential compound construction.

2. *Actions*: As it can be expected, nouns collocate mainly either with other nouns, to form compound-like structures like the *objects* above, or with verbs, with which they can establish three types of relationship. Depending on this relationship, actions can thus be classified as *performed*, *undergone* or *complemented*.

3. *Action Arguments*: This category holds those PPs which do not contain any keyword and simply function as some sort of indirect action receivers that usually follow the direct object within the sentence.

4. *Modifying Elements*: This contains those adjectives that modify a keyword directly.

7 Customized Hierarchies

Customized hierarchies are the applied hierarchies which represent the semantic information contained in a particular cluster case. They are of a clear and easy-to-access nature and become very accurate and practical knowledge repositories. At present, these hierarchies have been built both for storage and evaluation purposes, given the need to allocate the semantic knowledge acquired from the corpus, as well as to evaluate the acquisition procedure. Further on evaluation, it will just be added that the sublanguage-specific nature of the corpus prevents us from using any gold standard available because these are usually supported by wide-coverage on-line resources such as MRDs or thesauri (Grefenstette, 1993). Consequently, the customized hierarchies developed become our evaluation standards, performing quantitative as well as qualitative evaluations.

In regard to the meaning storage nature of the hierarchies, figure 4[4] shows an example of one such customized hierarchy, that for the semantic cluster *BEGIN/NNP - END/NNP*, which takes place in the following contexts within the corpus:

0 : expression/NN ,/, or/CC a/DT boolean/JJ combination/NN of/IN these/NNP ./. the/DT special/JJ pattern/NN -BEGIN/NNP- may/MD be/VB used/VBN to/TO capture/VB control/NN before/IN the/DT first/JJ input/NN line/NN is/VBZ

1 : control/NN before/IN the/DT first/JJ input/NN line/NN is/VBZ read/VBN ,/, in/IN which/WDT case/NN -BEGIN/NNP- must/MD be/VB the/DT first/JJ pattern/NN ./. the/DT special/JJ pattern/NN END/NNP may/MD be/VB

2 : which/WDT case/NN BEGIN/NNP must/MD be/VB the/DT first/JJ pattern/NN ./. the/DT special/JJ pattern/NN -END/NNP- may/MD be/VB used/VBN to/TO capture/VB control/NN after/IN the/DT last/JJ input/NN line/NN is/VBZ

3 : control/NN after/IN the/DT last/JJ input/NN line/NN is/VBZ read/VBN ,/, in/IN which/WDT case/NN -END/NNP- must/MD be/VB the/DT last/JJ pattern/NN ./. A/DT single/JJ character/NN c/NN may/MD be/VB

4 : may/MD be/VB used/VBN to/TO separate/VB the/DT fields/NNS by/IN starting/VBG the/DT program/NN with/IN -BEGIN/NNP- /(FS/NNP =/SYM "/" c/NN "/" /) or/CC by/IN using/VBG the/DT -fc/NN

5 : here/RB is/VBZ a/DT menu/NN file/NN that/WP demonstrates/VBZ some/NNP of/IN these/DT features/NNS :/: -END/NNP- MENU/NNP END/NNP the/DT sunview/NN program/NN

[4] The dollar sign stands for the place where the keyword occurs within the context and the three dots for the existence of some noise in between the elements. When used next to an *action* category element (with a verb), the dollar sign refers to the entire nominal structure comprising the keyword, i.e., the noun keyword and its surrounding *modifying elements* and *objects*, given that its function is merely that of indicating the approximate location of the keyword in the context.

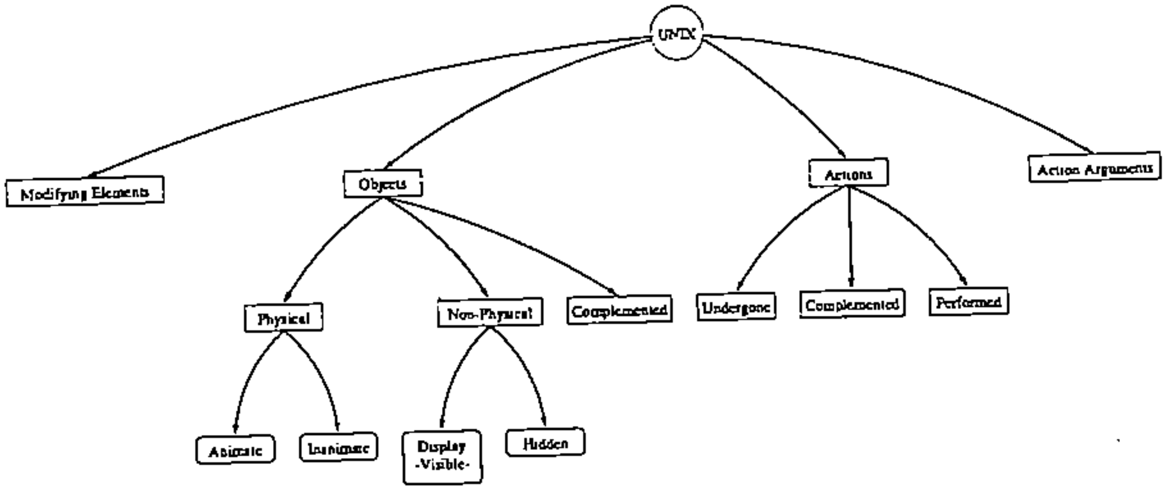

Figure 3: Unix Ontology for Nouns

runs/VBZ on/IN either/IN a/DT monochrome/JJ or/CC color/JJ

6 : a/DT menu/NN file/NN that/WP demonstrates/VBZ some/NNP of/IN these/DT features/NNS :/: END/NNP MENU/NNP -END/NNP- the/DT sunview/NN program/NN runs/VBZ on/IN either/IN a/DT monochrome/JJ or/CC color/JJ screen/NN ./.

As it can be seen, the common boxes represent the meaning aspects shared by the two terms. In this particular case, both words can be characterised by being a *non-physical object* pattern, which can be modified by different types of *modifying elements*. In addition, *BEGIN/NNP* and *END/NNP* are also defined by the fact that they can undergo the same type of *action*, that of use, which requires to have special pattern $ as its accompanying elements within the corpus. The types of requirements and relationships established between the different meaning components are indicated by means of the arrows. Further on the cluster under study, it can be observed that both words also present their own aspects of usage, i.e., meaning aspects that they do not share with each other. For example, *BEGIN/NNP* can also be defined as an element which *performs an action*, that of starting a program, while *END/NNP* can be defined as the *non-physical object* feature which can *undergo an action* (demonstrate) that differs from that it can undergo as *object* pattern, according to what the textual data indicates.

8 Results

As already mentioned above, we have established our own evaluation standards, which perform quantitative as well as qualitative evaluations. Regarding the former type, we look at both the number of correctly captured blocks of meaning and the number of

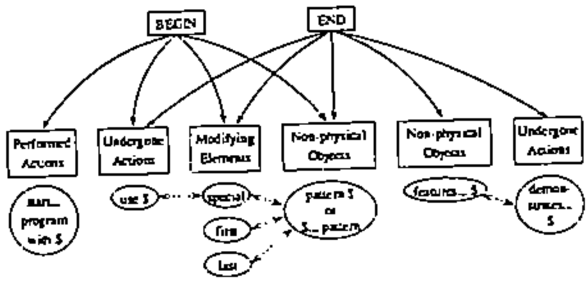

Figure 4: Hierarchy for Test Case 1

fully correct clusters captured. This allows us to establish whether our clustering and disambiguation module can successfully extract the different blocks of meaning, and if so, also determine the accuracy and noise-free nature of those blocks. As it can be expected, the results from both calculations differ and while those looking at the blocks of meaning captured regardless of noise obtain very high accuracy (up to 100 % in many cases), those considering only noise-free clusters obtain more modest values. A summary of the clustering and disambiguation results obtained from the application of *Dynamic Matching* to our test set of 20 semantic clusters can be seen in table 1.

This quantitative approach has also performed the evaluation of the different parameters used during the acquisition process and it can be concluded that the best three parameters are those in which the *grammar mapping module* is used. Furthermore, out of these three, the two reaching the highest scores are those which apply the *window distance weighting module*. Therefore, in spite of the remaining noise in some of the clusters, *Dynamic Context Matching* succeeds in capturing the blocks of meaning for each cluster, and the interactive modules implemented an improvement over the use of the ...larity measure.

...ding the latter approach, that of ...ing a qualitative evaluation of the re... detailed error analysis is carried out ... cluster within the test set and its dif... parameters, examining both the num... ...d type of errors encountered. Once this evaluation confirms that the best ... are obtained by means of *Dynamic ... Matching* interacting with the im...nted modules described in section 5. ...dition, this evaluation procedure also ... some insight into the reasons behind ...oise remaining in some clusters. One ... reason is the multi-dimensional nature ...eaning itself. Since the meaning of a ...d is built based upon the combination of different usages and these usages are the ... establishing semantic relationships with ...r words, some mis-clusterings, or rather ...leading clusterings, can easily take place ...ring the process. This is the case, for ex...ple, if a word w1 is semantically related to ...ords w2 and w3 and despite the fact that ...here is a closer link between w1 and w2, ...uster (w1, w3) is selected because of some misleading contextual cue which happens to obtain a higher value in the final calculation.

Conclusions

This paper provides some insight into a problem of current concern in NLP, the *lexical bottleneck*, by presenting an approach to semantic clustering and disambiguation from small sublanguage-specific corpora. The advantages of this approach over other existing methods, such as purely stochastic techniques, have been outlined. A further advantage that remains to be added is that with the exception of the optional symbolic modules, our clustering and disambiguation process requires no pre-established knowledge, making it highly portable cross-linguistically, as well as in applications to other sublanguages.

Regarding the Unix ontology developed, it represents a simple accurate semantic knowledge classification which can allocate all the Unix information extracted. When this abstract hierarchy is applied to store the information for the particular clusters, customized hierarchies are built, which become semantic knowledge representation templates for the individual cases. Each such hierarchy contains all the semantic information related to a particular cluster, i.e., all the different usages and relationships that those terms can undergo. It should also be added that given that one of the aims of developing these Unix hierarchies is the evaluation of the knowledge acquisition process itself, they are developed manually for the time being. However, work on their automation is currently being considered.

In relation to the results obtained, our semantic clustering and disambiguation procedure succeeds in locating the main blocks of meaning for each of the studied cases. Despite the fact that a number of clusters present some remaining noise, ambiguity is considerably reduced with the help of the problem-solving submodules implemented. Regarding the noise, though, we intend to explore further the possible causes and solutions.

The two final issues for future work are also related to the semantic ontologies developed for Unix. The first one would imply the application of the KA process to a Unix corpus in Spanish in order to do a comparative study of the domain in both languages. The second issue deals with the possible application of such ontological representations to a conceptual knowledge graph. This would allow the user to have fast and easy access to the information and to direct the search for information according to the point of interest.

References

M. Victoria Arranz, Ian Radford, Sofia Ananiadou, and Jun'ichi Tsujii. 1995. Tools for sublanguage-based semantic knowledge acquisition from corpora. In *XI Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN'95)*, Bilbao, Spain.

Roberto Basili, M. Teresa Pazienza, and Paola Velardi. 1994. A "not-so-shallow" parser for collocational analysis. In *15th International Conference on Computational Linguistics (COLING'94)*, Kyoto.

Branimir Boguraev and James Pustejovsky. 1996. Issues in text-based lexicon acquisition. In Branimir Boguraev and James Pustejovsky, editors, *Corpus Processing for Lexical Acquisition*. MIT Press, Cambridge, MA.

Eric Brill. 1993. *A Corpus-Based Approach to Language Learning*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania, USA.

Nicoletta Calzolari and Remo Bindi. 1990. Acquisition of lexical information from a large textual italian corpus. In *13th International Conference on Computational Linguistics (COLING'90)*.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information and lexicography. *Computational Linguistics*, 16(1):22–29.

CLR. 1991. The Consortium for Lexical Research. In *DARPA Speech and Natural Language Workshop*, Pacific Grove, California.

Case	# of Contexts	% of Correctly Captured Blocks of Meaning by Best Parameter	% of Fully Correct Clusters Captured by Best Parameter
1	7	100	100
2	24	100	60
3	8	100	100
4	22	100	100
5	32	100	25
6	19	100	50
7	4	100	100
8	40	100	50
9	4	100	0
10	77	52	8
11	13	100	0
12	47	100	40
13	39	100	60
14	27	100	0
15	33	88	44
16	28	100	0
17	2	100	100
18	14	100	50
19	10	100	100
20	23	100	100

Table 1: Clustering and Disambiguation Results

Jim Cowie, Joe Guthrie, and Louise Guthrie. 1992. Lexical disambiguation using simulated annealing. In *DARPA Speech and Natural Language Workshop*, Harriman, New York.

Gregory Grefenstette. 1992. Finding semantic similarity in raw text: the deese antonyms. In *AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, Cambridge, Massachusetts.

Gregory Grefenstette. 1993. Evaluation techniques for automatic semantic extraction: Comparing syntactic and window based approaches. In *ACL SIGLEX Workshop on Acquisition of Lexical Knowledge from Text.*

Ralph Grishman and John Sterling. 1994. Generalizing automatically generated selectional patterns. In *15th International Conference on Computational Linguistics (COLING'94)*, Kyoto.

John Lehrberger. 1986. Sublanguage analysis. In Ralph Grishman and Richard Kittredge, editors, *Analysing Language in Restricted Domains: Sublanguage Description and Processing*. Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey.

James Pustejovsky. 1991. The generative lexicon. *Computational Linguistics*, 17(4):409–441.

Philip Resnik. 1995. Disambiguating noun groupings with respect to wordnet senses. In *Third Workshop on Very Large Corpora*, Cambridge, Massachusetts.

German Rigau, Jordi Atserias, and Eneko Agirre. 1997. Combining unsupervised lexical knowledge methods for word sense disambiguation. In *35th Annual Meeting of the Association for Computational Linguistics (ACL'97) and 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL'97)*, Madrid.

Lucy Vanderwende. 1994. Algorithm for automatic interpretation of noun sequences. In *15th International Conference on Computational Linguistics (COLING'94)*, Kyoto.

Paola Velardi and M. Teresa Pazienza. 1989. Computer aided interpretation of lexical cooccurences. In *27th Annual Meeting of the Association for Computational Linguistics.*

David Yarowsky. 1992. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *14th International Conference on Computational Linguistics (COLING'92)*, Nantes.

Uri Zernik, editor, 1991. *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, chapter Introduction. Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey.