Clustering Syntactic Positions with Similar Semantic Requirements

Pablo Gamallo*
CITI
Alexandre Agustini†
CITI
Gabriel P. Lopes‡
CITI

This paper describes an unsupervised strategy to acquire syntactico-semantic requirements of nouns, verbs, and adjectives from partially parsed text corpora. The linguistic notion of requirement underlying this strategy is based on two specific assumptions. First, it is assumed that two words in a dependency are mutually required. This phenomenon is called here "co-requirement". Second, it is also claimed that the set of words occurring in similar positions defines extensionally the requirements associated to these positions. The main aim of the learning strategy presented in this paper is to identify clusters of similar positions by identifying the words that define their requirements extensionally. This strategy allows us to learn the syntactic and semantic requirements of words in different positions. This information is used to solve attachment ambiguities. Results of this particular task are evaluated at the end of the paper. Extensive experimentation was performed on Portuguese text corpora.

1 Introduction

Word forms, as atoms, cannot arbitrarily combine with each other. They form new composities by both imposing and satisfying certain requirements. A word uses a linguistic

* FCT, Universidade Nova de Lisboa, Portugal, gamallo@fct.unl.pt
† FCT, Universidade Nova de Lisboa, Portugal, aagustini@fct.unl.pt
‡ FCT, Universidade Nova de Lisboa, Portugal, gpl@di.fct.unl.pt

requirement (constraint or preference) in order to restrict the type of words with which it can combine in a particular position. The requirement of a given word is characterised by at least two different objects: the position occupied by the words that can be combined with the given word, and the condition that those words must satisfy in order to be in that position. For a word w and a specific description of a location loc, the pair $< loc, w >$ represents a "position" with regard to w. In addition, condition *cond* represents the set of linguistic properties that words must satisfy in order to be in position $< loc, w >$. So, a linguistic requirement of w can be represented as the pair:

$$<< loc, w >, cond > \tag{1}$$

Consider, for instance, position $< of\_right, ratification >$, where *of_right* is a location described as *"being to the right of preposition* of*"*. This position represents the argument slot *'ratification of [_]'*. Consider also that *cond* stands for the specific property *"being a nominal phrase (*np*) whose head denotes a legal document"* (abbreviated by *doc*), then the pair $<< of\_right, ratification >, doc >$ means that the particular position *'ratification of [_]'* selects for nouns denoting legal documents. In other words, *ratification* requires nominal arguments denoting legal documents to appear after preposition *of*. Suppose that there exist some words such as *law, treaty, constitution*, etc. that are nouns denoting legal documents. Then, it follows that they fill the condition imposed by *ratification* in the *of_right* location. An expression like *the ratification of the treaty* is then well-formed because *treaty* satisfies the required condition.

Let us look now more carefully at several linguistic issues we consider to be important to characterise the notion of linguistic requirement: extensionality/intensionality, soft/hard requirements, the scope of a condition, syntactic/semantic requirements, and co-requirements.

A condition can be defined either intensionally or extensionally. For example, the two specific properties *"being the head of a np"* and *"being a legal document"* are used to define intensionally the condition imposed by position $< of\_right, ratification >$. However, it is also possible to define it extensionally by enumerating all those words that actually possess such properties: e.g., *law, treaty, constitution*, etc.

Moreover, the process of satisfying a condition can be defined as a binary action producing a Boolean (yes/no) value. From this point of view, a word either satisfies or does not satisfy the condition imposed by another word in a specific location. This is a *hard* requirement. By contrast, the satisfaction process can also be viewed as a *soft* requirement, where some words are "preferred" without completely excluding other possibilities. In (Beale, Niremburg, and Viegas, 1998), hard requirements are named "constraints", whereas the term "preferences" is employed for soft requirements. In the following, we will use one of these two terms only if it is necessary to distinguish between hard or soft requirements. Otherwise, "requirement" will be taken as the default term.

Let's describe now what we call the *scope* of a condition. A position imposes a specific condition on the words that can appear in that position. Yet, a specific condition is not generally imposed by only one position, but by a large set of them. If a condition were only bound to a particular position, every combination of words would be a non-compositional idiomatic expression. So, speakers could not combine words easily and new composite expressions would be difficult to learn. The scope of a condition embraces the positions that use it to restrict word combination. For instance, the condition imposed by *'ratification of [_]'* seems to be the same as the one imposed by verb *ratify* on the words appearing at its right: $< right, ratify >$ (*'to ratify [_]'*). In addition, these positions also share the same conditions as *'to approve [_]'*, *'to sign [_]'*, or *'signatories to [_]'*. Each of these similar positions is within the scope of a specific condition, namely, *"being a np whose head denotes a legal document"*. In this paper, we assume that every linguistic condition is associated to a set of similar positions. This set represents the scope of the condition. The larger the set of similar positions, the larger the condition scope, and more general the property used to characterise the condition.

We distinguish syntactic and semantic requirements. A syntactic requirement is characterised by both a position and a morpho-syntactic condition. For instance, requirement $<< of\_right, ratification >, np >$ consists of a position, $< of\_right, ratification >$, which selects for a nominal phrase. Note that the different syntactic requirements of a word can serve to identify the set of subcategorisation frames of that word. Note also that, in some cases, a particular position presupposes a particular morpho-syntactic condition. In our example, position $< of\_right, ratification >$ only requires a *np*. So, we can use this position as a shorter form of the syntactic requirement $<< of\_right, ratification > , np >$. We call "syntactic position" a position that presupposes a specific morpho-syntactic condition. On the other hand, a semantic requirement (also known as "selection restriction") is characterised by both a position and a semantic condition, which presupposes a syntactic one. So $<< of\_right, ratification >, doc >$ means that position $< of\_right, ratification >$ selects for the head of a *np* denoting a legal document. Condition *doc* presupposes then a *np*. Identifying a particular semantic requirement entails the identification of the underlying syntactic one.

The final linguistic issue to be introduced is the phenomenon called "co-requirements". It will be assumed that each syntactic dependency between two words (which are the heads of two phrases) is composed by two complementary requirements. For instance, it seems that two different requirements underlie expression *ratification of the treaty*: $< of\_right, ratification >$ (*'ratification of [_]'*) requires to be filled by words like *treaty*, while $< of\_left, treaty >$ (*'[_] of the treaty'*) requires to appear with words such as *ratification*.

The main objective of this paper is to describe an unsupervised method to learn syntactic and semantic requirements from large text corpora. For instance, our method discovers that word *secretary* is associated with several syntactic positions (i.e., positions with morpho-syntactic conditions), such as *'secretary of [_]'*, *'[_] of the secretary'*, *'[_] to the secretary'*, *'[_] with the secretary'*, etc. The set of syntactic positions defined by a word can be used to characterise a set of subcategorisation frames. The precise characterisation of these frames will remain, however, beyond the scope of this article. In addition, for each syntactic position, we will assess the specific semantic condition a word needs to fill in order to appear in that position. Another important objective of the paper will be to use the semantic requirements to capture contextually relevant semantic similarities between words. In particular, we will assume that two words filling the same semantic requirement share the same contextual word sense. Consequently, learning semantic requirements also leads us to induce word senses. Suppose that word *organisation* fills the condition imposed by *'secretary of [_]'*. In this syntactic context, the word denotes a social institution and not a temporal process nor an abstract setup.

To achieve our objectives, we follow a particular clustering strategy. Syntactic positions (and not words) are compared according to their word distribution. Similar syntactic positions are put in more clusters following some constraints that will be defined later. Each cluster of positions represents a semantic condition. The features of each cluster are the words that can fill the common condition imposed by those positions: they are the fillers. They are used to extensionally define the particular condition they can fill. That is, a condition will be defined by identifying those words likely to appear in positions considered as similar. Given that a condition is extensionally defined by the words that are able to fill it, our method describes the process of satisfying a condition as a Boolean constraint (yes/no), and not as a probabilistic preference. The similar positions defining a cluster are within the *scope* of a particular semantic condition. The association between each position of the cluster with that condition characterises the semantic requirement of a word. This learning strategy does not require handcrafted external resources such as WordNet-like thesaurus nor Reader-Machine Dictionary.

The information captured by this strategy is useful for two different NLP disambiguation tasks: selecting contextual senses of words (word sense disambiguation), and solving structural ambiguity (attachment resolution). This paper is focused on the latter application.

In sum, the main contribution of our work is the large amount of linguistic information we learn for each lexical word. Given a word, we acquire, at least, three types of information: i) an unordered set of syntactic positions, which is a first approximation to define the set of subcategorisation frames of the given word, ii) the semantic requirements the word imposes on its arguments, and iii) the different contextual senses of the word. By contrast, related work only focuses on one or two aspects of this linguistic information. Another contribution is the use of co-requirements to characterise the arguments of a word.

To conclude the introduction, let's outline the organisation of the article. In next section, (2), we situate our approach with regard to related work on acquisition of linguistic requirements. Later, in sections 3 and 4, we describe in detail the main linguistic assumptions underlying our approach. Special attention will be paid to both the relativised view on word sense (i.e., contextual sense) and co-requirements. Then, section 5 depicts a general overview of our strategy. Two particular aspects of this strategy will be analysed next, in sections 6 and 7. More precisely, these sections describe both how syntactic positions are extracted (6) and how they are clustered in larger classes (7). Finally, in section (8), we evaluate the results by measuring their performance in a particular NLP task: syntactic attachment resolution.

2 Statistics-Based Methods for Learning Linguistic Requirements

During the last years, various stochastic approaches to linguistic requirements acquisition have been proposed (Basili, Pazienza, and Velardi, 1992; Hindle and Rooth, 1993; Sekine et al., 1992; Grishman and Sterling, 1994; Framis, 1995; Dagan, Marcus, and Markovitch, 1995; Resnik, 1997; Dagan, Lee, and Pereira, 1998; Marques, Lopes, and Coelho, 2000; Ciaramita and Johnson, 2000). In general, they follow comparable learning strategies, despite significant differences observed. In this section, we will present first the common strategy followed by these approaches, and then, we will focus on their differences. Special attention will be paid to lexical methods. At the end, we will situate our strategy with regard to the related work.

2.1 A Common Strategy

The main design of the strategy for automatically learning requirements is to compute the association degree between argument positions and their respective linguistic conditions. For this purpose, the first task is to count the frequency of occurring $<< loc, w >, cond >$ in a large corpus:

$$F(<< loc, w >, cond >) \tag{2}$$

where F counts the frequency of co-occurring $< loc, w >$ with *cond*. Then, this frequency is used to compute the conditional probability of *cond* given position $< loc, w >$:

$$P(cond| < loc, w >) \tag{3}$$

This probability is then used to measure the strength of statistical association between $< loc, w >$ and *cond*. Association measures such as Mutual Information or Log-likelihood are used for measuring the degree of (in)dependence between these two linguistic objects.

Intuitively, a high value of the association measure is the evidence for the existence of a true requirement (i.e., a type of linguistic dependence).

The stochastic association values obtained by such a strategy turn out to be useful for NLP disambiguation tasks such as attachment resolution in probabilistic parsing and sense disambiguation.

2.2 Specific Aspects of the Common Strategy

Despite the apparent methodological unanimity, approaches to learning requirements propose different definitions for the following objects: association measure, position $< loc, w >$, and linguistic condition *cond*. Many approaches only differ in the way in which the association measure is defined. Yet, such differences will not be discussed in this paper.

As regards position $< loc, w >$, we distinguish, at least, among three different definitions. First, it can be considered as a mere word sequence (Dagan, Marcus, and Markovitch, 1995): for instance, $< right, w >$, where *right* means *"being to the right of"*. Second, a position can also be defined in terms of co-occurrence within a fixed window (Dagan, Lee, and Pereira, 1998; Marques, Lopes, and Coelho, 2000). Finally, it can be identified as the Head or the Dependent role within a binary grammatical relationship such as subject, direct object, modifier, etc (Sekine et al., 1992; Grishman and Sterling, 1994; Framis, 1995) . In section 4, we will pay special attention to the grammatical characterisation of *loc*.

As far as *cond* is concerned, various types of information are used to define a linguistic condition: syntactic, semantic, and lexical information. The approaches to learning requirements are easily distinguished by how they define *cond*. Table 1 displays three different ways for encoding the condition imposed by verb *approve* to the nominal *the law* in the expression *to approve the law*.

syntactic level	$<< right, approve >, \mathbf{np} >$
semantic level	$<< right, approve >, \mathbf{doc} >$
lexical level	$<< right, approve >, \mathbf{law} >$

Table 1
Various levels of encoding linguistic conditions

Requirement conditions of the pairs in Table 1 represent three descriptive levels for the linguistic information underlying the nominal expression *"the law"*, when it appears to the right of the verb *approve*.[1]. The properties *np*, *doc*, and *law* are situated at different levels of abstraction. The morpho-syntactic tag *np* conveys more abstract information than the semantic tag *doc* (document), which , in turn, is more general than the lemma *law*. Some conditions can be inferred from other conditions. For instance, *doc* is only used to tag nouns, which are the heads of nominal phrases. So, the semantic tag *doc* entails the syntactic requirement *np*. Likewise, the lemma *law* is only associated to nouns. It entails then a *np*.

Some approaches describe linguistic conditions only at the syntactic level (Hindle and Rooth, 1993; Marques, Lopes, and Coelho, 2000). They count the frequency of pairs like $<< right, approve >, np >$, in order to calculate the probability of an *np* occurring given $< right, approve >$. This probability is then used to compute the degree of association between *approve* and a *np* located to the right. This association value may be useful in different linguistic tasks. For instance, it may serve to solve structural ambiguities (Hindle and Rooth, 1993), or to build a subcategorisation lexicon (Marques, Lopes, and Coelho, 2000). Most approaches to learning syntactic requirements assume that syntactic properties can be identified by means of some specific morphological "cues" appearing in the corpus. For instance, the article *a* following a verb is a clear evidence for a *np* appearing at the *right* of the verb; the preposition *of* following a verb is an evidence

1 In case of Portuguese, for intransitive verbs the occurrence of a *np* to the right of the verb does not mean that the verb is transitive. In fact, this is the standard position of the subject for intransitive verbs.

for an *of_right* complement; the conjunction *that* after a verb introduces a *that_clause*, etc. Morphological cues are used to easily identify syntactic requirements. This technique allows working directly on raw text. Let us note that these techniques do not allow the acquisition of complete "subcategorisation frames" (Brent, 1991; Manning, 1993). They are able to acquire that, for instance, *approve* subcategorises a *np* on two locations: both *right* and *of_right* locations (e.g., "to approve the laws", "to approve of the decision"). So, they associate that verb with two syntactic arguments. However, they are not able to learn that the two arguments are incompatible and must belong to two different subcategorisation frames of the verb. We will return to this issue in subsection 8.1.

In other approaches to requirement learning, linguistic conditions are defined in semantic terms by means of specific tags (Basili, Pazienza, and Velardi, 1992; Resnik, 1997; Framis, 1995). In order to calculate the degree of association between tag *doc* and position $< right, approve >$, these approaches count the frequency of pairs like $<< right, approve >, doc >$ through out the corpus. If the association value is higher than other related cases, then one might learn that verb *approve* requires nominal phrases denoting *doc* entities to appear at the right.

According to other learning approaches, the linguistic conditions used to characterise requirements may be situated at the lexical level (Dagan, Lee, and Pereira, 1998; Dagan, Marcus, and Markovitch, 1995; Grishman and Sterling, 1994; Sekine et al., 1992). A pair like $<< right, approve >, law >$ matches those expressions containing a form of lemma *law* (e.g., `law, laws, Law, Laws`, ...) appearing to the right of the verb *approve* (to be more precise, to the right of any form of lemma *approve*). The frequency of this pair in the corpus serves to compute the degree of association between *law* and verb *approve* at the *right*. In these approaches, then, conditions are learnt from lexical co-occurrences. From now on, when it will not be necessary to distinguish between lemmas and word forms, we will use the term "word" for both objects.

To compare the three types of approaches in a more accurate way, let's analyse their behavior regarding different quantitative aspects: (i) the continuum between supervised and unsupervised learning, (ii) the continuum between knowledge-poor and knowledge-rich methodology, and (iii) the continuum between general and specific information acquisition.

2.2.1 Supervised/Unsupervised Learning The first continuum ranges over the degree of human supervision that is needed to annotate the training corpus. Among the works cited above, (Basili, Pazienza, and Velardi, 1992) has the highest degree of supervision. This semantic approach requires hand-tagging text nouns using a fixed set of semantic labels. The other approaches are close to total unsupervision, since they do not require a training corpus to be annotated by hand. However, some degree of human supervision could be involved in building automatic tools (e.g., a neural tagger in (Marques, Lopes, and Coelho, 2000)) or linguistic external sources (e.g., WordNet in (Resnik, 1997; Framis, 1995; Ciaramita and Johnson, 2000)), which are used to annotate the corpus.

2.2.2 Knowledge-rich/Knowledge-poor Methods The second continuum refers to the notions introduced by G. Grefenstette (Grefenstette, 1994). He distinguishes the learning methods according to the quantity of linguistic knowledge they need. The most knowledge-rich approaches need a handcrafted thesaurus (WordNet) to semantically annotate nouns of the training corpus (Resnik, 1997; Framis, 1995; Ciaramita and Johnson, 2000). At the opposite, the most knowledge-poor methods are introduced in (Dagan, Marcus, and Markovitch, 1995; Dagan, Lee, and Pereira, 1998), which merely need to identify lemmas in the corpus.

2.2.3 General/Specific Conditions As regards the general/specific continuum, "syntactic methods", i.e., approaches to learn syntactic requirements, are the learning methods that use the most general linguistic information. At the opposite, we find the "lexical methods", i.e., those strategies situated at the lexical level. Methods using tags like *doc*, *human*, *institution*, ... are situated at an intermediate level, and are known as "semantic methods". One of the most difficult theoretical problems is to choose the appropriate generalisation level for learning requirement information.

The syntactic level seems not to be appropriate if requirements are used to solve structural ambiguity. Concerning the parsing task, syntactic information is not always enough to produce a single parse. Consider the following analyses:

$$[_{vp}\text{cut} \quad [_{np}\text{the potato}] \quad [_{pp}\text{with a knife}]] \tag{4}$$

$$[_{vp}\text{cut} \quad [_{np}\text{the potato} \quad [_{pp}\text{with a hole}]]] \tag{5}$$

In order to decide the correct analysis, either (4) or (5), we must be helped by our world knowledge concerning cutting actions, use of knifes, and the potatoes properties. In general, we know that knifes are used for cutting, and potatoes are objects likely to have holes. So, the parser is able to propose a correct analysis only if the lexicon is provided with, not only syntactic requirements, but also with information on semantico-pragmatic requirements (i.e., with selectional restrictions). Selection restrictions are typically used to capture facts about the world, which are generally, but not necessarily, true (Androutsopoulos and Dale, 2000). So, the main goal of semantic and lexical methods is precisely the acquisition of selection restrictions.

As has been said before, semantic methods use handcrafted sources of linguistic knowledge such as WordNet. There are several disadvantages associated with these knowledge-rich approaches: manually created thesauri contain many words either having rare senses, or missing domain-specific meanings. In sum, the semantic information level provided by handcrafted thesauri is either too specific or too general, and it is usually incomplete. It seems not to be appropriate for most NLP tasks (Grefenstette, 1994). By contrast, lexical methods are able to acquire information at the level of detail required by the corpus domain. They are domain dependent approaches. However, they are very sensitive to the problem of data sparseness.

2.3 Lexical Methods and the Sparseness Problem

Most word co-occurrences (for instance the co-occurrence of *agreement* with *approve* at location *right*) have very small probabilities of occurring in the training corpus. Note that if they were not observed in the corpus, they would have identical probabilities (i.e., probability 0) to incorrect co-occurrences such as *cow* appearing to the right of *approve*. This is what is known as the sparseness problem. To solve this problem, many lexical methods estimate the probabilities of unobserved pairs by taking into account word similarity. Suppose that the pair $<< right, approve >, agreement >$ is not observed in the training corpus. To obtain an appropriate association measure between *agreement* and $< right, approve >$, the degree of association between $< right, approve >$ and each word most similar to *agreement* is computed. The total association value for the specific lexical co-occurrence is the average of these association values.

Information on word similarity is used to generalise the pairs appearing in the corpus, and to smooth their co-occurrence probabilities. That is, very specific requirements described at the lexical level can be generalised by means of word similarity information. For instance, the following pair:

$$<< right, approve >, MOST\_SIM(agreement) > \tag{6}$$

associates the information $MOST\_SIM(agreement)$ to the position $< right, approve >$, where $MOST\_SIM(agreement)$ represents the most similar words to *agreement*: e.g., *law, treaty, accordance, conformity*, etc. Word similarity allows to smooth (generalise) the probabilities computed at the lexical level. Similar words minimise the sparseness problem to a certain extent. Lexical methods provided with similarity-based generalisations are (Sekine et al., 1992; Grishman and Sterling, 1994) and (Dagan, Lee, and Pereira, 1998). Later, in subsection 8.3.4, we will use a lexical method with similarity-based generalisation to solve syntactic attachments. The results obtained using this method will be explicitly compared to those obtained by our clustering strategy.

The methodology for automatically measuring word similarity is often based on Harris' distributional hypothesis on word meaning (Harris, 1985). According to this assumption, words occurring in similar syntactic contexts (i.e., in similar syntactic positions) are semantically similar. A simple way of implementing this hypothesis is to compute the similarity between words by comparing the *whole* information concerning their context distribution. (Allegrini, Montemagni, and Pirrelli, 2003) calls this strategy the "absolute view" on word similarity. The absolute view leads to characterise word similarity as an intransitive relation (Dagan, Lee, and Pereira, 1998). Let us see the expressions (7-10) below. They show that even if *treaty* is similar to *agreement*, and *agreement* is similar to *conformity*, it does not mean that *treaty* is similar to *conformity*.

to approve the agreement/treaty (7)

to ratify the agreement/treaty (8)

we are in agreement/conformity with your proposal (9)

my signature indicates my agreement/conformity to the rules (10)

Intransitivity makes this type of word similarity to be not very efficient to identify contextual word senses. For instance, it does not help to foresee that *agreement* is similar to *treaty* in quite a different way as it is similar to *conformity*. Expressions (7) and (8) introduce the linguistic contexts where *agreement* denotes a document containing legal information. This word is considered to be semantically similar to *treaty* with regard to the contexts introduced by verbs *approve* and *ratify*. By contrast, (9) and (10) introduce different linguistic contexts. There, *agreement* conveys a different sense: the verbal act of agreeing. In these contexts, it becomes similar to *conformity*. Word similarity methods based on the absolute view seem not to be able to distinguish such contextual meanings. This weakness may perturb the smoothing process defined above. As *conformity* and *accordance* are part of the most similar words to *agreement*, they are involved in the process of computing the degree of association between this word and $< right, approve >$. Yet, this is counterintuitive since they are not semantically required by the verb in such a particular position.

2.4 General Properties of our Method

The objective of this article is to propose a new strategy to learn linguistic requirements. This strategy will be designed to overcome the main drawbacks underlying the different approaches introduced above. Our method will be characterised as follows:

- The information it acquires will be described at a semantically *appropriate* level of generalisation.

- It will be defined as a knowledge-poor and unsupervised strategy.

As regards the first characteristic, we will consider that the method is semantically appropriate only if the acquired requirements are useful to solve disambiguation problems such as those illustrated above by parses (4) and (5). So, our acquisition method will be focused on more specific information than the one contained in syntactic requirements. Given a word, our aim is to learn not only the syntactic positions in which that word appears, but also the semantico-pragmatic constraints (i.e., what is broadly called *selection restrictions*) associated with each syntactic requirement. Selection restrictions will be extracted from position-word co-occurrences. We thus follow a lexical method. However, selection restrictions will be defined in accordance with a theory of word sense that is not based on the absolute view on word similarity. We will use a more relativised viewpoint on word senses. In sum, we follow a strategy slightly different from that described in subsection 2.3. In the next section, we will describe our basic assumptions on word sense and word similarity.

Concerning the second characteristic (i.e., knowledge-poor and unsupervised strategy), our method does not rely on external structured sources of lexical information (e.g., WordNet), nor on a training corpus built and corrected by hand. Unlike the semantic methods outlined above (in 2.2), we attempt to reduce human intervention to a minimum.

3 The Foundations of our Learning Strategy

In this section, we outline the basic assumptions underlying our learning strategy. This strategy relies on a particular definition of semantic condition (subsections 3.1 and 3.2), a relativised view on word similarity (3.3), as well as a specific viewpoint on word sense (3.4).

3.1 Extensional Definition

Given a requirement $<< loc, w >, cond >$, we define a semantic condition, *cond*, as the set of words that can occur in position $< loc, w >$. This means that linguistic conditions are defined extensionally. For instance, consider again position $< right, approve >$ and one of its possible conditions, namely *doc*, which, as has been seen, means *"being a noun denoting a legal document"*. This condition is extensionally defined by enumerating the set of words likely to occur with both $< right, approve >$ and their *similar* positions. Identifying such a word set is not a trivial task. This set is not a closed, fixed, and predefined list of nouns. Rather, it turns out to be a set open to a great variety of extensions, since it can be modified as time, domain, or speaker change. The aim of our method is to learn, for each argument position, the open set (or sets) of words it requires. Each word set represents, in extensional terms, a specific linguistic condition. For this purpose, we opt for the following learning strategy.

The condition imposed by an argument position is represented by the set of words actually appearing in this position in the training corpus. For instance, let's suppose that $< right, approve >$ occurs with 4 different words: *law, agreement, convention* and *oil* (to simplify the explanation, frequencies are not taken into account). For the present, we only know that these words are mere candidates to satisfy the condition imposed by that position. In order to actually know whether the candidate fillers satisfy or not such a condition, we select the most *similar* positions to $< right, approve >$. So, we get clusters of similar positions imposing the same condition. Consider, for instance, the following cluster:

$$\begin{aligned} &\{< right, approve >, < right, ratify >, < to\_right, signatories >, \\ &< by\_left, becertified >, < of\_right, ratification >\} \end{aligned} \tag{11}$$

which is made of positions sharing features such as:

$$\textit{law, agreement, article, treaty, convention, document} \tag{12}$$

So, cluster features in (12) are the words that may fill the specific condition imposed by the similar positions in (11). These words can be viewed as fillers satisfying the intensional property: "being a noun denoting a legal document". Note that (12) contains some words (e.g.,*article* or *treaty*) that do not actually occur with position $< right, approve >$ in the corpus. However, as these words actually occur with most of the positions that are similar to $< right, approve >$, we may assume that they satisfy the condition of this particular position. This is the technique we use to generalize (smooth) occurrences of position-word pairs that are not observed in the training corpus. Details of our method of clustering will be given later in section 7.2. Notice also that the set of fillers does not contain the word *oil*. This word does not belong to the shared features because it does not occur with none of the positions similar to $< right, approve >$. This is the method we use to identify and remove wrong associations between a position and a word. It will be explained in section 7.1.

In sum, positions are considered to be similar because they impose the same conditions (i.e., they share the same selection restrictions). As has been said earlier, similar positions are within the *scope* of one common requirement. The set of similar positions in (11) represents the scope of condition (12). The fillers are those words that characterise the extension of such a condition.

3.2 Hard Requirements

We assume that the process of condition satisfaction may be defined as a Boolean function and not as a probabilistic one. The association value between, for instance, word *treaty* and position $< right, approve >$ is either yes or no. Our method merely attempts to learn whether or not there is a true association between them. If the association is actually true, then we learn that the word satisfies the condition. Hard requirements can be easily used to constrain the grammar of a symbolic parser. In particular, we use them to improve the parser described in (Rocio, de la Clergerie, and Lopes, 2001). Although linguistic constraints are defined in Boolean terms, they are open to potential changes. Clusters and their features are supposed to be modified and extended as the training corpus grows and is progressively annotated with more trustful syntactic information. Moreover, a new domain-specific corpus can lead us, not only to create new clusters, but also to tune old ones. From this viewpoint, Boolean constraints cannot be considered as necessary and sufficient conditions. They evolve progressively.

3.3 Relativised Word Similarity

Our learning strategy relies on a specific assumption on word similarity. We are interested in computing similarity between words with regard to a set of similar positions. So, we have to compute first similarity between positions. As it has been said before, similar positions impose the same linguistic condition. Hence, they are likely to be filled by the same set of words. Statistically, this means that they have *similar* word distribution. A definition of this similarity will be given later in subsection 7.1. Unlike the absolute view stated above, we are not interested in measuring similarity between words on the basis of the distribution of all their corpus-based positions (their whole context distribution). Our aim is to compute first the similarity between positions via their word distribution. Positions are in fact less ambiguous than words. Then, we consider two words to be similar if they occur with at least a pair of similar positions. This way of using similar positions allows capturing all possible dimensions of similarity of a given word. This is close to the "relativised view" on word similarity by (Allegrini, Montemagni, and Pirrelli, 2003).

3.4 Contextual Hypothesis on Word Sense

Behind this account of similarity, there is a particular view of word sense, which is not far from (Schütze, 1998):

> *Contextual Hypothesis for Word Sense*: A set of similar positions defines a particular type of context. A word occurring with positions of the same type keeps the same sense. The sense of a word changes if the word appears in a different type of context.

For instance, *agreement* refers to a legal document when it satisfies the requirement of similar positions such as *'to approve [_]'* or *'ratification of [_]'* By contrast, this word denotes a verbal act when it appears in positions such as *'in [_] with your proposal'* or *'[_] to the rules'*.

According to this hypothesis, identifying word senses relies on identifying sets of similar positions (i.e., types of contexts). The noun *book*, for instance, can denote at least three different contextual senses provided it appears in three context types: e.g., physical actions (carrying the book, putting it on the table, etc.), symbolic processes (writing or reading books), economical transactions (selling or buying books). This notion of word sense is dependent on the ability to grasp classes of contexts, i.e., the ability to learn clusters of similar positions. The more accurate is this ability the more precise are the senses identified in a particular corpus. This means that the set of senses associated to a word cannot be predefined by an external lexical resource like WordNet or any machine readable dictionary. Word senses are dynamically learned as the text is processed and positions are organised in semantically homogenous clusters. Each cluster of similar positions (or context type) represents a particular word sense. From this viewpoint, the set of contextual senses of a word represents its whole meaning. Such a notion of word meaning is in accordance with the encyclopaedic hypothesis on lexical meaning within the Cognitive Grammar's framework (Langacker, 1991). According to this hypothesis, every word or lexical unit is associated with a *continuum* of encyclopaedic knowledge (the word meaning). The use of the word in a particular context makes a partial aspect of this continuum more salient (a specific word sense).

Within a particular corpus, we assume that the *meaning* of a word is defined by the context types that organise the different positions of the word. In other words, a word's meaning is described by identifying the types of requirements the word fulfills. In the next section, we will explore the notions of requirement and syntactic position.

4 Syntactic Positions and Co-Requirements

This section discusses the general properties of syntactic positions and their role for extracting linguistic requirements. Syntactic positions are defined here as internal elements of binary dependencies. Two aspects of dependencies will be retained: the head-dependent distinction and the predicate-argument structure. Special attention will be paid to co-requirements.

4.1 Head-Dependent distinction

The "head-dependent" pattern takes over the process of transferring morpho-syntactic features to higher grammatical levels. A composite expression inherits the features of the head word. There are two different locations (or grammatical roles) within a binary dependency: both the *Head* and the *Dependent*. Consider the binary dependencies shown in the first column of Table 2, which represent the expressions *to ratify the law* and *long dinner*. The grammatical relations between the two words are expressed by both *robj*, which stands for the nominal object appearing to the right of the verb[2], and *mod*(ifier), which stands for the noun-adjective dependency. The word indexed by "$^{\downarrow}$" (*down* location) plays the role of Head, whereas the word indexed by "$^{\uparrow}$" (*up* location) plays the role of Dependent. Since a binary dependency is constituted by two grammatical locations, we can split the dependency in two complementary syntactic positions.

Dependencies	Contexts
$(robj;\, ratify^{\downarrow},\, law^{\uparrow})$	$< robj\_down,\, ratify >$ $< robj\_up,\, law >$
$(mod;\, dinner^{\downarrow},\, long^{\uparrow})$	$< mod\_down,\, dinner >$ $< mod\_up,\, long >$

Table 2
Two binary dependencies and their positions

Each pair of position in the second column of Table 2 was extracted from a bi-

2 In Portuguese, a right object (without governing preposition) can be elaborated as either a direct object or a subject under specific conditions

nary dependency. We will see below that the two positions extracted from a dependency are associated with specific semantic conditions. Hence, they will be used to characterise syntactico-semantic requirements. In our work, the different types of binary relations from which we extract all positions are the following: *lobj*, *robj*, *iobj_prepname*, *aobj_prepname*, *prepname*, and *mod*. Relation *lobj* designates the nominal object appearing to the left of the verb; *robj* represents the nominal object appearing to the right of the verb; *iobj_prepname* introduces a nominal after a verb and a preposition; *aobj_prepname* represents a nominal after an adjective and a preposition; *prepname* corresponds to a nominal following a noun and a preposition; *mod* refers to the adjective modification of nouns. Note that each relation not only conveys two argument positions, but also specific morpho-syntactic conditions. *robj*, for instance, means that there is a *np* to the right of a *vp*. So, $< robj\_down, ratify >$ contains the same information as the syntactic requirement: $<< robj\_down, ratify >, np >$, while $< robj\_up, law >$ is equivalent to $<< robj\_up, law >, vp >$.

4.2 Predicate-Argument structure

Besides the head-dependent pattern, binary dependencies are also organised as predicative structures: **Predicate**(Argument). While the former pattern drives the process of inheriting morpho-syntactic information throughout grammatical levels, the latter is directly related to semantic requirements. This subsection starts by introducing the standard account concerning the role of the **Predicate**(Argument) structure in the process of imposing linguistic requirements. Then, we will make new assumptions on what we consider to be a more accurate notion of requirement information. This notion will be modeled by means of what we call “co-requirements”. Co-requirements will be used later, in sections 6 and 7, to elaborate our learning method.

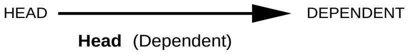

Figure 1
Complement structure.

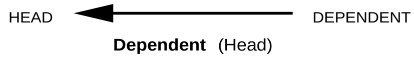

Figure 2
Modifier structure.

4.2.1 Simple Requirements It is broadly assumed that a binary syntactic dependency is constituted by both the word that imposes linguistic constraints (the predicate) and the word that must fill such constraints (its argument). In a syntactic dependency, each word is considered to play a fixed role. The argument is perceived as the word specifying or modifying the syntactic-semantic constraints imposed by the predicate, while the predicate is viewed as the word that is specified or modified by the former. Notice that a "predicate" is not defined here in model-theoretical terms. We inherit the intuitive definition assumed in the linguistic tradition on Dependency Grammar (Hudson, 2003). According to this tradition, a predicate is the semantic representation of one of the two words taking part in a binary dependency. More precisely, it is the representation of the word (either head or dependent) that actually imposes semantic requirements on the other word.

In standard linguistic approaches, **Predicate**(Argument) structure is the semantic counterpart of the "head-dependent" pattern. The former relates to the latter in the following way. Typically, the dependent word playing the role of Argument is conceived as the *complement* or *object* of the head (see Figure 1). By contrast, when it plays a more active role behaving more like a Predicate, it is viewed as a *modifier* or the *adjunct* of the head (Figure 2). In other words, the dependent of a head-dependent structure will be described either as a passive complement if it satisfies the linguistic requirements of the head (Argument role), or as an active modifier when it requires itself a specific type of head (Predicative role).

The most typical case of a head being a predicate is when a verb is the head within a direct object dependency. The noun is viewed here as a complement, i.e., as a dependent expression filling the conditions imposed by the verbal predicate. The most typical cases of a dependent taken as a predicate is the standard use of an adjective or an adverb. In this case, it is the adjective (or adverb) that imposes the selection restrictions on the noun (or verb), which is the head of the dependency.

By contrast, in case of dependencies such as prepositional relations it is not possible to distinguish a complement from a modifier, unless we have access to the specific semantico-pragmatic information conveyed by words. However, there are many cases in which the borderline between complement and modifier is not clear. In these cases, even semantic-pragmatic knowledge is not enough to decide for one particular predicative structure. For instance, consider the expression *to play with a doll*. Which is the word that can be taken as the predicate? and the argument?

Linguists have made a considerable effort to discriminate between complements and modifiers (= adjuncts). The complement/modifier distinction is probably one of the most unclear issues in linguistics (Langacker, 1991). No linguistic proposal is able to distinguish in absolute terms complements from external adjuncts, e.g., is *with a doll* an internal complement or an adverbial modifier of *play*? In other words, is position $< iobj\_with\_down, play >$ that requires as argument the noun *doll* (complement construction)? or, on the contrary, is position $< iobj\_with\_up, doll >$ that requires as argument verb *play* (modifier structure)? There are no reliable evidences to choose between the two possible requirement structures. Most linguistic proposals may be situated in one of the two following general trends: (i) Some linguists interpolate finer distinctions between the two extremes (Pustejovsky, 1995). So, between true or basic complements and completely optional adjuncts it is possible to find default complements, shadow complements, and so on, which share properties of both complements and adjuncts. (ii) A most radical view is to consider the complement/modifier opposition as a *continuum* in

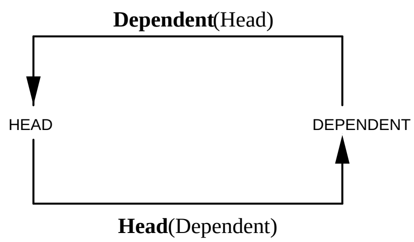

Figure 3
Dependency with co-requirements.

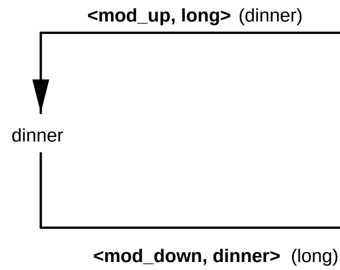

Figure 4
Co-requirements in *long dinner*.

which it is not easy to fix borderlines between what is entirely optional and obligatory (Langacker, 1991).

The idea of a continuum entails that complements and modifiers cannot be defined in absolute terms. All binary dependencies always contain a certain degree of both complementarisation and modification. That is, given a dependency, the head requires the dependent (complementarisation), and conversely, the dependent requires the head (modification). We will assume in this paper that such co-requirements underlie any binary dependency.

4.2.2 Co-requirements

Recent linguistic research assumes that two words related by a syntactic dependency are mutually constrained (Pustejovsky, 1995; Gamallo, 2003). Both words impose linguistic requirements on each other. It does not exist a single pre-fixed "predicate-argument" pattern. Each related word is at the same time both a predicate and an argument. We call such a phenomenon *co-requirement* structure.

Consider again the expression *potato with a hole*. It does not seem obvious whether *hole* is either the complement or the modifier of *potato* within the *with* dependency. If it is considered as the complement, then it is the head *potato* that should be provided with the appropriate requirements. Otherwise, it should be the modifier *hole* the word imposing specific requirements on the head. Following recent research, we claim, however, that such a radical opposition is not useful to describe linguistic requirements. It will be assumed here that two syntactically related expressions presuppose two complementary requirements. In other words, every binary dependency is constituted by two compatible predicate-argument structures.

On the one hand, the noun *potato* requires words denoting parts or properties of potatoes to appear in the *with_down* location. The noun *hole* satisfies such a requirement. On the other hand, the noun *hole* is also provided with selective requirements in the *with_up* location. Indeed, in this location, it requires nouns denoting material objects that can have holes. The noun *potato* fulfils such a condition. Note that the expressions *cut with a knife* and *play with a doll* could also be considered borderline cases.

Co-requirements are not only useful for modeling borderline cases. We believe that they are also pertinent for typical complement structures (e.g., the direct object relation between verbs and nouns), as well as for typical modifier constructions (i.e., adjective-noun and verb-adverb dependencies). In *long dinner*, for instance, the noun seems to behave as a predicate constraining the adjective to denote a temporal dimension (and not a spatial one). So, not only the adjective disambiguates the noun, but also the latter disambiguates the former.

Therefore, according to the assumption on co-requirements, two syntactically dependent expressions are no longer interpreted as a standard pair "predicate-argument", where the predicate is the active function imposing semantic conditions on a passive argument, which matches such conditions. On the contrary, each word of a binary dependency is perceived simultaneously as a predicate and an argument. That is, each word both imposes semantic requirements and matches semantic requirements in return. Figure 3 depicts a standard syntactic dependency between two words, the Head and the Modifier, with two **Predicate**(Argument) structures. Figure 4 illustrates the two specific **Predicate**(Argument) structures extracted from the modifier relation between

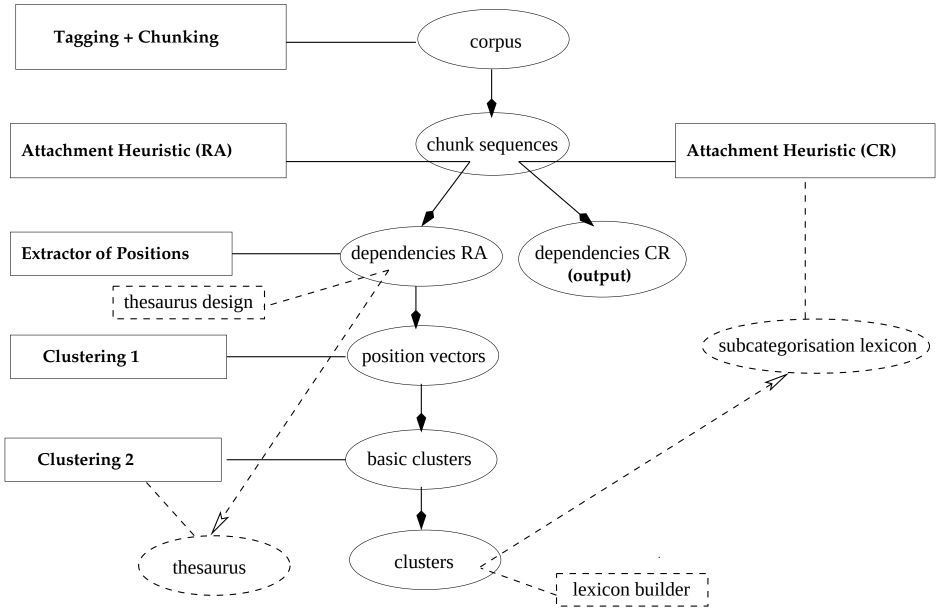

Figure 5
System modules

noun *dinner* (the head) and adjective *long* (the dependent).

The learning strategy described in the remainder of the paper takes advantage of the co-requirement structure.

5 System Overview

To evaluate the hypotheses presented above, a software package was developed to support the automatic acquisition of syntactic and semantic requirements. The system is constituted by 6 main processes, which are displayed as rectangles with solid lines in Figure 5. They are organised as a linear sequence of data transformations. In Figure 5, solid ellipses are used to display the data transformed by these processes. Two local subprocesses (dotted rectangles) build extra data (dotted ellipses), in order to constrain some of the main transformation processes. In the remainder of this section, we merely outline the overall functionalities of these processes. Then, in the next sections, we will describe them in detail.

Tagging and Chunking: Raw text is tagged (Marques and Lopes, 2001) and then analysed in chunks using some potentialities of the shallow parser introduced in (Rocio, de la Clergerie, and Lopes, 2001). This parser is implemented using tabulation capabilities of the DyALog system (de la Clergerie, 2002). The output is a sequence of basic chunks. For instance, the sentence *The President sent the document to the Minister* is analysed as a sequence of four basic chunks: *np*, *vp*, *np* and *pp*. These chunks neither contain dependencies nor recursivity.

Attachment Heuristic RA: An attachment heuristic based on Right Association (RA) is applied to chunk sequences in order to put together pairs of chunks. The head words of two related chunks form a *syntactic dependency*. Section 6.1 describes some properties of the dependencies extracted using the RA strategy.

Extractor of Position Vectors: Dependencies are used to extract *syntactic positions*, which are internally characterised as vectors of word frequencies. This process will be described in section 6.2.

Clustering-1: Position vectors are compared with each other using a specific similarity measure. Pairs of similar positions are put in *basic clusters*. A basic cluster is constituted by two similar positions whose features are the words they share. Section 7.1 describes this process.

Clustering-2: Basic clusters are successively aggregated using a conceptual clustering methodology to induce more general classes of positions. A corpus-based thesaurus, which has been built on the basis of the extracted dependencies, is used to constraint cluster aggregation. We present this process (together with the subprocess *thesaurus design*) in section 7.2.

Attachment Heuristic CR: Finally, the resulting clusters are used to parse again the chunks and propose new dependencies (section 8). This is achieved in two steps. First, a module called *lexicon builder* organises the information underlying the learnt clusters and builds a lexicon with syntactico-semantic co-requirements (see subsection 8.1). Then, the grammar underlying the parser is provided with a specific attachment heuristic (called CR), which uses co-requirement information from the lexicon. This heuristic allows the parser to propose a new set of dependencies (section 8.2). We evaluate the resulting dependencies in section 8.3.

The system was performed on two different Portuguese text corpora: *P.G.R.* [3] and *E.C.*[4] Experiments and some results are given in section 7.3.

6 Extracting Dependencies and Positions

In this section, we describe two modules of the method: the heuristic of attachment RA and the elaboration of position vectors. These modules involve the extraction of candidate binary dependencies and syntactic positions.

6.1 Attachment Heuristic RA

Attachment heuristic RA takes as input parses constituted by sequences of chunks. It uses the Right Association strategy. That is, a new chunk is preferentially attached to the preceding chunk. The two head words of two attached chunks form a possible binary dependency. Consider the expression:

`...a lei citada em o anterior parecer...`(*the law cited in the previous opinion*) (13)

3 P.G.R.(*Portuguese General Attorney Opinions*) is constituted by case-law documents.
4 E.C. (*European Commission*) contains documents on different aspects (legislation in force, social policy, environment, etc.) of the European Commission. This corpus is in: http://europa.eu.int/eur-lex/pt/index.html.

Binary Dependencies	Syntactic Positions
$(lobj;\ \ citar : vpp^{\downarrow},\ \ lei^{\uparrow})$ $(be\ cited,\ law)$	$< lobj\_down,\ \ citar : vpp >$ $< lobj\_up,\ \ lei >$
$(iobj\_em;\ \ citar : vpp^{\downarrow},\ \ parecer^{\uparrow})$ $(be\ cited\ in,\ report)$	$< iobj\_em\_down,\ \ citar : vpp >$ $< iobj\_em\_up,\ \ parecer >$
$(mod;\ parecer^{\downarrow},\ anterior : pre^{\uparrow})$ $(opinion,\ previous)$	$< mod\_down,\ parecer >$ $< mod\_up,\ anterior : pre >$

Table 3
Binary dependencies and syntactic positions extracted from expression (13).

The RA heuristic allows us to identify three candidate dependencies, which are illustrated at the left column of Table 3. These dependencies are not considered to be true attachments, but only potential candidates. Later, the parser will be provided with the learnt requirements stored in the lexicon, in order to propose new dependencies, which will be the output the parsing strategy. Note that *lobj* denotes a nominal object appearing to the left of the verb. This cannot be identified with the subject grammatical function. The order of verbal objects is not the main feature to identify the subject and direct object functions in Portuguese (and in most Latin languages). The position of verb complements is quite free in these languages. We consider then that grammatical functions are semantic-dependent objects, since we need semantic-pragmatic knowledge to identify them.

In addition, we also provide some dependencies with specific morpho-syntactic information. For instance, verb `citar` (*to cite*) is annotated using the past participle *vpp* tag. This morpho-syntactic information is relevant to define the semantic requirements of dependencies. As we will see later, only semantic information will enable us to consider the dependency underlying `a lei citada` (*the law that was cited*) as being semantically similar to the one underlying `citar a lei` (*to cite the law*). These dependencies are not merged in one single relation by merely using morpho-syntactic rules. Such rules posit some important problems: first, they require specific knowledge on particular languages; second, they introduce a great amount of noise. In our approach, these two dependencies will be merged in one cluster only if our learning process provides us with semantic evidence. In fact, one of the objectives of our learning method is to use information on semantic requirements for identifying morpho-syntactic alternations of dependencies: e.g., `citada pelo ministro` / `o ministro citou` (*cited by the Minister / the Minister cited*) ; `mencionar a lei` / `mencionou-se a lei` (*to mention the law / the law was mentioned*); `ratificar a lei` / `ratificação da lei` (*to ratify the law / ratification of the law*). If two morpho-syntactic alternations are considered to share the same semantic requirements, then they will automatically occur in the same cluster. This strategy allows us to reduce language-dependent knowledge to a minimum.

It is also worth noticing that tag *pre* in Table 3 is used to annotate adjectives in the left position with regard to the modified noun (i.e., in the *mod* relation). We distinguish three different adjective relations: the left modifier, the right modifier, and the prepositional object. It is assumed here that these three dependencies can stress different semantic aspects of one adjective. For instance, our strategy led us to learn that `anterior` (*previous*) is semantically similar to `primeiro` (*first*) and `seguinte` (*following*) when it takes the role of left modifier. However, when the adjective is to the right of a noun and is followed by a prepositional object (`anterior a` – *previous to*), it is clustered together with `inferior` (*inferior*) and `igual` (*equal*), which also appear within prepositional dependencies: *equal to, inferior to.*

Since the Right Association strategy is knowledge-poor, the attachments it proposes give rise to a great amount of odd syntactic dependencies (25%), including those caused by POS tagging errors. The overall precision of the tagger is 96.2%. Yet, considering only those tags we will use in the learning strategy (i.e., nouns, adjectives, articles, verbs, etc.), the precision is close to 90%. To overcome such a noisy input, we need a reliable learning method.

< *iobj_em_down, citar vpp* > :	`(nota 53) (parecer 7) (conclusão 3) (informação 2) (regulamento 1) (artigo 1)` *`(apoio 1) (sentido 1)`*
(*be cited in [_]*)	*note, report, conclusion, information, regulation, article,* ***support, sense***
< *iobj_em_up, parecer* >	`(afirmar:vpp 9) (defender:vpp 7) (citar:vpp 7) (analisar:vpp 7) (escrever 3) (reafirmar:vpp 2) (esclarecer:vpp 1) (notar:vpp 1) (publicar:vpp 1) (concluir:vpp 1) (assinalar:vpp 1)`
(*[_] in the report*)	*be affirmed, be defended, be cited, be analysed, write, be affirmed-again, be clarified, be noted, be published, be concluded, be pointed out*

Table 4
Two position vectors

6.2 Position Vectors

Given that each dependency contains two complementary grammatical locations (head and dependent), we split a dependency into two syntactic positions: the position associated with the head (or down) location and the one associated with the dependent (or up) location. The positions extracted from expression (13) are illustrated at the right column of Table 3. Following the assumption on co-requirements, each position must be provided with a particular linguistic requirement.

We represent each syntactic position as a feature vector. Each feature corresponds to a word occurring in the position. The value of the feature is the frequency of the word in that position. A position is thus defined by means of its word distribution. As has been said before, those words appearing in a position can be used to represent, in extensional terms, a first approximation to the semantic condition the position requires (i.e., its selection restrictions). Clustering will enable us to enlarge the scope of each condition. In Table 4 , we illustrate the word distribution of the two complementary positions underlying `citada no parecer` (*be cited in the report*).

Notice that those words occurring once in a position are also considered as features. This allows us to minimise the "sparseness" problem. Linguistic corpora are sparse in the sense that most co-occurrences occur few times in a given corpus. So, if co-occurrences with lower frequencies were not used by the learning strategy, pertinent linguistic information would be missing and coverage would remain low. In order to minimise missing information and coverage reduction, we keep infrequent words in position vectors.

Nevertheless, the fact of taking into account infrequent co-occurrences increases noise and, then, may disturb the learning task. There are several noise sources: words missing from the dictionary, words incorrectly tagged, wrong attachments, etc. The position shown in the first line of Table 4 occurs with at least two words that are not syntactically required: `apoio` (*support*) and `sentido` (*sense*).[5] Note that these words have frequency 1 in that position. Keeping requirements with frequency 1 enables us to keep other correct words, such as `artigo` (*article*) and `regulamento` (*regulation*), which also occur only once. The next step of our method (clustering 1) will focus on the automatic removal of the odd features introduced in position vectors.

7 Clustering of Positions

Positions that share similar features are put together into clusters. Clustering is divided in two different agglomerative processes: Clustering 1 and Clustering 2.

7.1 Clustering 1

This process builds pairs of similar positions called "basic clusters". A basic cluster is the result of merging two positions considered as being similar. The features associated to a basic cluster are only those words appearing in both similar positions. This allows us to filter out odd features from clusters. Features defining a basic cluster are, then, the most reliable fillers of the semantic condition imposed by the two similar positions. Those words that are not required by both positions are removed from the cluster. The algorithm of this process is the following:

Similarity: We calculate the "similarity" between each pair of positions. To do it, we measure the distance between their word distributions (see below the details of

5 Word `sentido` (*sense*) appears in that position, not as a verb complement, but as a member of the preposition locution `no sentido de` (*in the sense that*), which is attached to the whole sentence.

this measure).

Selection: Then, for each position, we select N (where $N = 20$) most similar ones.

Aggregation: Then, given a position and the list of N most similar positions, we merge the position with each member of the list. So, given a position, we create N aggregations.

Filtering: Finally, for each aggregation of two similar positions, we select the intersection of their features; that is, the features of a basic cluster are those words that appear in both positions.

As a result, we obtain a set of basic clusters, each of them augmented by reliable features. The aim is to automatically filter out noisy features from each pair of similar syntactic positions. Many incorrectly tagged words are removed at the filtering step.

Let's take an example. Consider the position shown in the first row of Table 4, that is: $< iobj\_em\_down,\ citar : vpp >$. According to our similarity measure, its word distribution is similar to that of the following positions[6]:

$$\begin{array}{rl} < iobj\_em\_down,\ mencionar : vpp > & < iobj\_em\_down,\ cite > \\ < iobj\_em\_down,\ assinalar : vpp > & < de\_down,\ leitura > \\ < iobj\_em\_down,\ referir : vpp > & < iobj\_em\_down,\ referenciar : vpp > \ldots \end{array} \quad (14)$$

Then, $< iobj\_em\_down,\ citar : vpp >$ is merged with each one of the above positions. Note that it is similar to the position associated with the active form: `citar`. Finally, each pair of similar positions (i.e., each basic cluster) is defined by their common words.

6 English translation of (14): *be mentioned in [_], cite in [_], be pointed out in [_], reading of [_], be refered in [_], be made reference in [_]* .

For instance, take the basic cluster shown below in (15):

$$\{< iobj\_em\_down,\ citar : vpp > + < iobj\_em\_down,\ mencionar : vpp >\} \;=$$

`nota conclusão informação artigo`

$$(\textit{note, conclusion, information, article}) \qquad (15)$$

Looking at those words appearing as prepositional objects of both *'cited in [_]'* and *'mentioned in [_]'*, one can see that they are semantically homogeneous. Filtered features do not include any more odd words such as *support* and *sense* (see Table 4). Indeed, the fact of selecting the words shared by two similar positions relies on the Contextual Hypothesis stated above in subsection 3.4, as well as on the following corpus-based observation: those words that incorrectly appear in a particular position are not likely to occur in similar positions.

The result of merging two similar positions by intersecting their features allows associating a semantic condition with two positions. In (15), a single set of words is associated with the two positions, since they have in common the same semantic condition (or selection restrictions). However, the scope of the condition is still too narrow: it merely embraces two positions. In order to extend the scope of semantic conditions, we will cluster them using a less restrictive clustering process. It will allow us to build more general classes of words and positions.

Before explaining the final process (clustering 2), let us describe the measure used to calculate similarity between syntactic positions. We used a particular weighted version of the Lin coefficient (Lin, 1998). Our version, however, does not use the "pointwise mutual information" to characterise the weight on position-word pairs. As (Manning and Schütze, 1999) argued, this seems not to be a good measure of the strength of association between a word and a local position. When the similarity between two positions is computed, it assigns higher scores to rare attributes (i.e., words in our case) of compared objects (positions). By contrast, this measure is not sensitive to the fact that frequent pairs can have a strong association. In order to undertake this problem, we used a weight very similar to that proposed in (Grefenstette, 1994). Consequently, we used, on the one hand, the general structure of the Lin coefficient, and on the other, the weight proposed by Grefenstette.

Words are weighted considering their dispersion (global weight) and their conditional probability given a position (local weight). The weight *Assoc* measuring the degree of association between word w in a position p is computed by equation (16):

$$Assoc(p, w) = \log_2(P_{MLE}(w|p)) * \log_2(disp(w)) \tag{16}$$

On the one hand, the conditional probability P_{MLE} is estimated by using the *maximum likelihood estimate (MLE)*, which is calculated in 17:

$$P_{MLE}(w|p) = \frac{f(p, w)}{F(p)} \tag{17}$$

where $f(p, w)$ represents the frequency of word w appearing in position p, and $F(p)$ is defined, for a particular position, as the total sum of its word frequencies: $\sum_i f(p, w_i)$. On the other hand, word dispersion, *disp*, is defined as the following mean:

$$disp(w) = \frac{F(w)}{number\, of\, positions\, for\, w} \tag{18}$$

where $F(w)$ is defined as the total sum of position frequencies of w: $\sum_i f(p_i, w)$. The higher values of (18) are assigned to those words that are not dispersed, that is, to those words frequently appearing in few positions. It measures the ability of a word to be semantically selective with regard to its positions. So, Lin similarity measure LIN

Input	Set of basic clusters organised by number of features.
Output	A list of larger clusters representing classes of semantic conditions.
Step 1	*Pre-restrictions on candidates to be clustered* For each *obj*, select those objects that: have the same number of features than *obj* AND share at least 80% of features
Step 2	*Similarity restrictions* From candidates extracted in step 1, take those objects that: either share all features with *obj* OR the different features are related by a thesaurus
Step 3	*Merging objects and their features* *obj* is merged with all objects filling the conditions stated in steps 1 and 2. The new object has the following properties: it is constituted by the union of the features defining the merged objects it is put together with objects having the same number of features
Iteration	Repeat steps 1, 2, and 3, increasing the number of features, until no cluster fills the restrictions.

Table 5
algorithm of clustering 2

between two positions is computed using equation (19):

$$sim(p_1,p_2) = \frac{\sum\limits_{\{w:\exists(p_1,w),\exists(p_2,w)\}}(Assoc(p_1,w)+Assoc(p_2,w))}{\sum\limits_{\{w:\exists(p_1,w)\}}Assoc(p_1,w)+\sum\limits_{\{w:\exists(p_2,w)\}}Assoc(p_2,w)} \tag{19}$$

At the numerator of (19), the condition of the summation indicate that each word w must be found with both positions p_1 and p_2. At the denominator, w varies over all words found in p_1 and p_2.

7.2 Clustering 2

Basic clusters are the input objects of the second process of clustering. We use an agglomerative (bottom-up) clustering for aggregating basic clusters into larger ones. The clustering algorithm is described in Table 5. According to this algorithm, two objects are clustered if they satisfy the following restrictions: i) they have the same number of features (i.e., words); ii) they share more than 80% common features; iii) the features that are different must be thesaurically related to, at least, one of the common features. In order to provide words with thesaurical relations, we automatically build a thesaurus

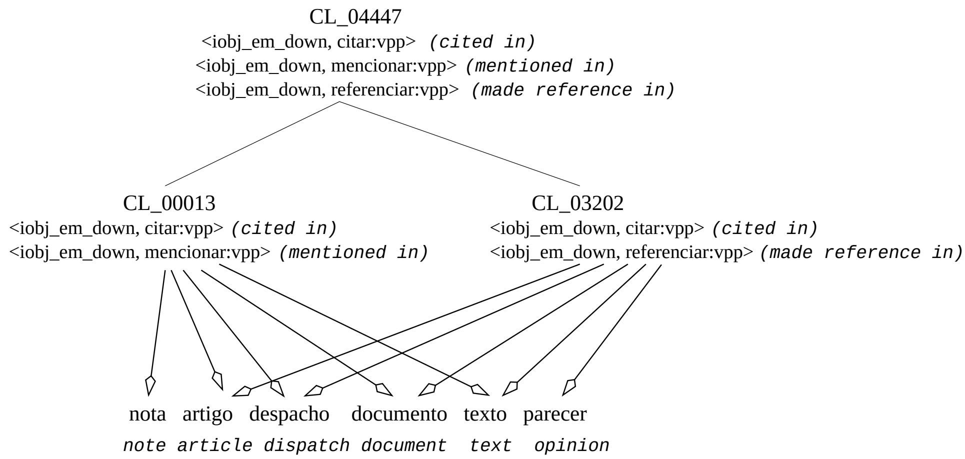

Figure 6
Clustering 2

of similar words. Details of the thesaurus design will be given in subsection 7.5 below.

Figure 6 shows how two basic clusters are merged into one more general class of positions. For two basic clusters such as CL00013, which contains the features *note, article, dispatch, document, text*, and CL03202, whose features are *article, dispatch, document, text, opinion*, we obtain the more general cluster CL04447, which is constituted by all the different positions and words of their basic components. Note that the two basic clusters are different with regard to two features: *note* and *opinion.* According to our clustering restrictions, the two clusters can be merged if each different feature (i.e., *note* and *opinion*) is thesaurically related to, at least, one of the common features: *article, dispatch, document, text.* A word is thesaurically related to another one if it belongs to the list of most similar words, a list that was automatically generated and entered in our thesaurus. The thesaurus is, then, used to control and constrain the construction of abstract classes of positions. In addition, the larger class, CL04447, allows us to induce collocation data that does not appear in the corpus. For instance, we induce that word `parecer` (*opinion*) may appear in position $< iobj\_em, mencionar : vpp >$ (*mentioned in [_]*). Similarly, we also learn that word `nota` (*note*) can occur with $< iobj\_em, referenciar : vpp >$ (*made reference in [_]*).

7.3 Tests and Results

We tested our learning strategy over two training corpora: *P.G.R.* and *E.C.*[7] Data concerning the information extracted from these two corpora is presented in Table 6.

	Corpus P.G.R.	**Corpus E.C.**
Word Occurrences	6,643,579	3,110,397
Binary Dependencies	966,689	487,916
Syntactic Positions	178,522	113,847
Basic Clusters	370,853	166,886
Clusters (Clustering 2)	16,274	10,537

Table 6
Corpus data

The clusters generated by clustering 2 are used to build a lexicon of words with syntactic and semantic requirements. Each corpus has its own lexicon. Later, in subsection 8.1, we will describe how this information is stored in the lexicon entries.

Learnt clusters represent linguistic requirements that cannot be reduced to a smaller set of general syntactico-semantic roles, such as Agent, Patient, Theme, Instrument and so on. On the other hand, they cannot be associated with word-specific roles like, for instance, Reader, Eater, Singer, etc. The level of elaboration of these clusters is ranged between the very abstract and the very specific lexical level. They are situated, in fact, at the domain-specific level, which is considered the more appropriate to be used in computational tasks (Gildea and Jurafsky, 2002). However, given the too restrictive constraints of the algorithm, the clustering method also overgenerates redundant clusters. In future work, we will attempt to reduce redundancy using clustering algorithms based on concept lattices (L.Kovacs and Baranyi, 2002)

In order to evaluate the linguistic relevance of these clusters, we will check in section 8 if they are useful in a parsing task. The degree of efficiency in such a task (parsing) may serve as a reliable evaluation for measuring the soundness of the learning strategy.

7 Some results can be consulted at `http://terra.di.fct.unl.pt/~agustini/restr_web`.

7.4 Related Clustering Methods

There are other approaches to acquisition of word senses by clustering words according to context-sensitive information. Similarly to our work, these approaches assume the following. On the one hand, a word can appear in different clusters (soft clustering). On the other hand, each cluster represents a particular sense distinction of the words that are elements of it. Different clustering methods can be distinguished.

First, some methods compare the similarity between pairs of syntactic positions (and not pairs of words) in order to generate clusters of syntactic positions, whose features are sets of words (Allegrini, Montemagni, and Pirrelli, 2003; Faure and Nédellec, 1998; Reinberger and Daelemans, 2003). Similarly to our approach, they follow both the relative view on word similarity and the assumption on contextual word sense, which have been introduced above, in subsections 3.3 and 3.4. However, these methods differ from ours in several aspects. (Reinberger and Daelemans, 2003) does not use any kind of filtering process. So, given a cluster of positions, the set of its features is basically defined as the union of their co-occurring words. This method turns out to be not appropriate when extracted co-occurrences are noisy. The cooperative system Asium presented in (Faure and Nédellec, 1998) filters out incorrect words from clusters of positions. However, unlike our work, this task is not automatic. It requires manual removal of those words that have been incorrectly tagged or analysed. Similarly to our approach, (Allegrini, Montemagni, and Pirrelli, 2000) developed an automatic procedure to remove odd words from clusters. It consists in defining a first clustering step where positions are aggregated in basic clusters, which are called "Substitutability Islands". As in clustering 1 (subsection 7.1), each basic cluster only selects those words occurring in all positions of the cluster. However, (Allegrini, Montemagni, and Pirrelli, 2000) define a second clustering step containing significant differences with regard to our clustering 2. Given a position p, they define a list of basic clusters containing p. This list is ranked and then used as the input of a clustering strategy that only aggregates basic clusters belonging to that list. So, a cluster containing p cannot be aggregated to a cluster that does not contain p. This is a very strong constraint. It reduces significatively the ability of the system to make generalisations.

Second, other methods discover word senses by clustering words according to their whole distributional similarity (Pantel and Lin, 2002; Lin and Pantel, 2001). These methods follow then both the "absolute view" on word similarity and the Harris' distributional hypothesis, which we have introduced in subsection 2.3 above. However, in order to make the absolute view more relative, a collection of small and tight clusters (called "committees") is proposed in a first step. These tight clusters are supposed to represent different word senses. Then, in a second step, each word is assigned to its most similar committees.

Finally, (Pantel and Lin, 2000) is a hybrid method based on the two basic views on semantic similarity: both absolute and relative views. Given a word w occurring in position p, which can be any pair of type $< verb, function >$ or $< noun, preposition >$, the system generates classes of contextually similar words. A contextual class is the result of intersecting the words occurring in p and the similar words to w. The definition of a contextual class contains the two views on word similarity. On the one hand, the words occurring in p are called the "cohorts" of w. The cohorts are similar to w only with regard to position p (relativised view). On the other hand, a corpus-based thesaurus is used to select similar words to w with regard to its whole position distribution (absolute view). Note that a contextual class is not far from what we call "basic cluster". In a second step, contextual classes are used to compute attachment association scores. The aim of the method is not to discover word senses (like in the methods outlined above), but to solve syntactic ambiguities. No clustering strategy is proposed to generate more general contextual senses.

Our system could also be considered as a hybrid method, since besides the contextual hypothesis and the relative view, we also take into account the absolute view on word similarity to design a corpus-based thesaurus.

7.5 Automatic Thesaurus Construction

Clustering 2 uses a thesaurus of similar words to avoid undesirable aggregations. To design a corpus-based thesaurus, we follow the absolute view on word similarity: similarity between two words is computed by comparing their whole context distribution. Our thesaurus was not specifically designed to be involved in the clustering process. It was firstly designed with the aim of measuring the discriminative capabilities of syntactic positions defined on the basis of co-requirements (Gamallo et al., 2001). In particular, we checked whether co-required positions are semantically more selective than those used by Grefenstette in (Grefenstette, 1994), which were defined in terms of simple requirements. Experimental tests showed that co-requirements permit a finer-grained characterisation of "meaningful" syntactic positions.

To compute word similarity, we used the weighted version of the binary Jaccard measure defined in (Grefenstette, 1994). The weighted Jaccard similarity WJ between two words, w_1 and w_2 is computed by:

$$WJ(w_1, w_2) = \frac{\sum_i \min(Assoc(w_1, p_i), Assoc(w_2, p_i))}{\sum_i \max(Assoc(w_1, p_i), Assoc(w_2, p_i))} \tag{20}$$

In (20), the weight $Assoc$ is the result of multiplying a local and a global weights, whose definitions are analogous to those given in formulas (17) and (18). The major difference is that, in (20), positions are taken as attributes and words as objects.

We designed a particular thesaurus for each training corpus. As regards PGR corpus, we have obtained $42,362$ entries: $20,760$ nouns, $16,272$ verbs, and $15,330$ adjectives. For each entry w, the thesaurus provides a list containing the 20 words most similar to w. This is the list that was later used in the clustering process.

8 Application and Evaluation

The acquired classes will be used to solve attachment ambiguities. For this purpose, first, a lexicon is designed by using the linguistic information contained in the learnt clusters. Then, a particular heuristic uses this information to propose correct attachments. Some experiences are performed on two text corpora. The results are evaluated in subsection 8.3.

8.1 Design of a Lexicon with Co-Requirements

The learning method provides a lexicon with syntactic and semantic information. A word entry is divided in two types of information (see Table 7). SUBCAT is the repository of syntactic and semantic requirements. SENSE contains the different word sets to which the entry belongs. Each word set corresponds to a particular sense distinction. However, only the SUBCAT information will be used here for the purpose of attachment resolution. Table 7 shows an excerpt of entry `secretário` (*secretary*). This entry is associated with a SUBCAT repository with six requirements and a SENSE repository containing two word senses.

Word `secretário` requires two nominal and four verbal arguments. Concerning the nominal positions, we learn that *secretary* selects for nouns such as *post* or *rank* in the *de_up* location, whereas it requires a class of nouns denoting institutions or functions in the *de_down* location. Concerning the verbal positions, we also learn that *secretary* requires various verb classes in different verbal positions: two classes in location *iobj_a_up*, one class in *iobj_por_up*, and one more in *lobj_up*.

A syntactic pattern of subcategorisation arguments underlies the organisation of the

SUBCAT repository in Table 7. Such a pattern can be represented as follows:

$$(X_v\, a_{prep}\, \alpha_n)_{vp} \vee (Y_v\, por_{prep}\, \alpha_n)_{vp} \vee (Z_n\, de_{prep}\, \alpha_n)_{np} \vee (\alpha_n\, de_{prep}\, W_n)_{np} \vee (\alpha_n\, U_v)_{vp} \tag{21}$$

Where $X, Y, Z, \ldots$ stand for variables of subcategorised words, while α is the subcategoriser. If α is the lexicon entry `secretário`, then the instantiation of its argument positions is determined by the semantic information stored in this entry. For example, according to Table 7, noun `cargo` instantiates Z, while verb `pertencer` instantiates X. Symbol $\vee$ stands for boolean disjonction. We consider that, at least in Portuguese, all word arguments are optional. Even the subject of a verb may be omitted. Note however that the syntactic pattern in (21) does not allow to distinguish whether arguments are compatible or not. For instance, it is not able to predict that $(Y_v\, por_{prep}\, \alpha_n)_{vp}$ and $(\alpha_n\, U_v)_{vp}$ are argument positions that cannot appear in the same sentence. Moreover, there are no restrictions on the linear order of arguments. As we do not learn this type of syntactic information, the pattern depicted in (21) can be merely viewed as a set of potential arguments of a word. So, our method does not allow capturing, for each word, a set of entirely organised subcategorisation frames.

Notice that it is the co-requirement structure that allows us to acquire a great number of requirement positions that are not usual in most standard approaches. Five positions of *secretary* do not require standard dependent complements, but different types of heads. This is a significant novelty of our approach. Consider the positions that impose non-standard requirements (i.e., non-standard predicates). According to the standard definition of predicate given in section 4.2.1 (simple requirement definition), only locations *robj_down*, *lobj_down*, and *mod_up* give rise to positions with requirements[8]. By contrast, positions defined by the complementary locations (*robj_up*, *lobj_up*, *mod_down*) are con-

8 Positions with prepositions are not taken into account in this analysis because they are ambiguous.

secretário (*secretary*)

SUBCAT

- $< de\_up, \text{secretário} >$ (*[_] of secretary*) =
`cargo, carreira, categoria, competência, escalão, estatuto, funȥão, remuneracão, trabalho, vencimento`
(post, career, category, qualification, rank, status, function, remuneration, job, salary)
- $< de\_down, \text{secretário} >$ (*secretary of [_]*) =
`administração, assembleia, autoridade, conselho, direcção, empresa, entidade, estado, governo, instituto, juiz, ministro, ministério, presidente, serviço, tribunal órgão`
(administration, assembly, authority, council direction, company, entity, state, government , institute, judge, minister, ministery, president, service, tribunal organ)
- $< iobj\_a\_up, \text{secretário} >$ (*[_] to the secretary*) =
`aludir, aplicar:refl, atender, atribuir, concernir, corresponder, determinar, presidir, recorrer, referir:refl, respeitar`
(allude, apply, attend, assign, concern, correspond, determine, resort, refer, relate)
- $< iobj\_a\_up, \text{secretário} >$ (*[_] to the secretary*) =
`caber, competir, conceder:vpp, conferir, confiar:vpp, dirigir:vpp, incumbir, pertencer`
(concern, be incumbent, be conceded, confer, be trusted, be sent, be incumbent, belong)
- $< iobj\_por\_up, \text{secretário} >$ (*[_] by the secretary*) =
`assinar:vpp, conceder:vpp, conferir:vpp, homologar:vpp, louvar:vpp, subscrito`
(be signed, be conceded, be confered, be homologated, be complimented, subscribe)
- $< lobj\_up, \text{secretário} >$ (*the secretary [_])*) =
`definir, estabelecer, fazer, fixar, indicar, prever, referir`
(define, establish, make, fix, indicate, foresee, refer)

SENSE

- `administração, assembleia, autoridade, chefe, comandante, comissão, conselho, director, direcção, entidade, estado, funcionário, gabinete, governador, governo, instituto, juiz, membro, ministro, ministério, presidente, provedor, secretaria, secretário, senhor, serviço, tribunal, órgão`
(administration, assembly, authority, chief, commander, commission, council, director, direction, entity, state, official, cabinet, governor, government, institute, judge, member, minister, ministery, president, purveyor, secretary, secretary, mister, service, tribunal, organ)
- `primeiro-ministro, autoridade, entidade, estado, membro, ministro, ministério, presidente, secretário`
(prime minister, authority, entity, state, member, minister, ministery, president, secretary)

Table 7
Excerpt of entry `secretário` (*secretary*). It belongs to the *PGR* corpus.

sidered as mere complements of verbs, or objects modified by adjectives. So, they cannot impose any requirement and thereby they are not semantically defined as predicates. In opposition to this viewpoint, our system learns more classes of requirements imposed by positions considered as non-standard predicates $(5, 192)$ than requirements imposed by positions considered as standard predicates (4.600). These experimental results seem to prove that non-standard predicates correspond to positions with requirements. In sum, we may infer that binary dependencies are structured by co-requirements.

Consider now the SENSE repository in Table 7. It contains two word sets which should represent two senses of `secretário`. Unfortunately, our clustering algorithm generates some redundancy. In this case, the two clusters should have been merged into only one, since they seem to refer to the same concept. Cluster redundancy is the major problem of our learning strategy.

8.2 Attachment Heuristic CR

The syntactic and semantic requirements provided by the lexical entries are used to improve a parser and the DCG grammar it is based on. The description of the parser remains beyond the scope of this article; it has been described in (Rocio, de la Clergerie, and Lopes, 2001). Details of a symbolic DCG grammar with information on linguistic co-requirements can be found in (Gamallo, Agustini, and Lopes, 2003). In this paper, we only outline how the grammar uses this information to solve syntactic attachments. Co-requirements are at the centre of attachment resolution. They are used to characterise a particular heuristic on syntactic attachment. This heuristic, called CR, is supposed to be more precise than RA. It states that two chunks are syntactically and semantically attached only if one of these two conditions is verified: either the *dependent* is semantically required by the *head*, or the *head* is semantically required by the *dependent*. Take the expression:

$$\texttt{...compete a o secretário ...}\ (\textit{is incumbent on the secretary}) \qquad (22)$$

This expression will be analysed as a $vp - pp$ construction if only if, at least, one of the two following requirements is satisfied:

down requirement: context $< iobj\_a\_down, competir >$ (*be-incumbent on [_]*) requires a class of nouns to which `secretário` (*secretary*) belongs;

up requirement: context $< iobj\_a\_up, secretário >$ (*[_] on secretary*) requires a class of verbs to which `competir` (*be-incumbent*) belongs.

Co-requirements are viewed here as constraints on the syntactic rules of a symbolic grammar. Attachments are then solved by using Boolean, and not purely probabilistic, constraints. According to the lexical information illustrated in Table 7, expression (22) can be analysed as a $vp{-}pp$ construction because, at least, the *up* requirement is satisfied. Note that, even if we had no information on the verb requirements, the attachment would be allowed since the noun requirements in the dependent (*up*) location were learnt. So, we learnt that noun `secretário` has as argument the verb `competir` in location $< iobj\_a\_up >$. As we will see in the evaluation procedure, co-requirements are also used to solve long-distance attachments.

8.3 Evaluating Performance of Attachment Resolution

We evaluated the performance of CR, i.e. the attachment heuristic based on Boolean co-requirements. The general aim of this evaluation is to check whether the linguistic requirements we have learnt are adequate to be used in a parsing task. The degree of efficiency in such a task may serve as a reliable evaluation for measuring the soundness of our learning strategy.

8.3.1 Test Data Test data is constituted by sequences of basic phrases (i.e., chunks). The phrase sequences selected belong to three types: $vp - np - pp$, $vp - pp - pp$, and $np - pp - pp$. They were randomly selected from two different (and already chunked) test corpora: a group of 633 sequences was selected from *EC* corpus and another group of 633 again was selected from *PGR*. Each group of 633 sequences was constrained to have three equal partitions: 211 $vp - np - pp$ sequences, 211 $vp - pp - pp$ sequences, and 211 $np - pp - pp$ sequences. The test corpus from which each group was selected was previously separated from the training corpus. So the sequences used for the test were excluded from the learning process. Then, the annotators (the co-authors) manually propose the correct attachments for each phrase sequence, using the full linguistic context. Some specific instructions were given to the annotators for the most controversial cases. An excerpt of these instructions are the following: i) if a *pp* seems to be a modifier of the verb, then it is attached to the *vp*; ii) if a *pp* is a modifier of the sentence, no attachment is proposed; iii) if a *np* following a *vp* is either the direct object or the subject of the verb, then the *np* is attached to the *vp*; iv) if a *pp* seems to be attached to two phrases, two attachments are proposed (we keep the ambiguity); v) if a phrase contains a word that was not correctly tagged, no attachment is proposed. Note that verbal modifiers and verbal complements are treated in the same way (see subsection 4.2.2 above). Moreover, we consider that a *robj* (i.e., a *np* following a *vp*) can be instantiated by two different functions: both a direct object and a subject (subsection 6.1 above).

Most works on attachment resolution use as test data only phrase sequences of type $vp - np - pp$ (Sekine et al., 1992; Hindle and Rooth, 1993; Ratnaparkhi, Reymar, and Roukos, 1994; Collins and Brooks, 1995; Li and Abe, 1998; Niemann, 1998; Grishman and Sterling, 1994). These approaches consider that each sequence selected for evaluation can be syntactically ambiguous in two ways. For instance, the sequence of chunks:

$np - pp - pp$	`[`$_{np}$ `o artigo relativo] [`$_{pp}$ `a o decreto] [`$_{pp}$ `de a lei]` (*the article referring to the decree-law*)
$vp - pp - pp$	`[`$_{vp}$ `publicou] [`$_{pp}$ `em os estatutos anexos] [`$_{pp}$ `a o citado diploma]` (*published in the statutes appended to the referred diploma*)
$vp - np - pp$	`[`$_{vp}$ `tem] [`$_{np}$ `acesso] [`$_{pp}$ `em a medida]` (*has access in so far as*)

Table 8
Different types of syntactic sequences and various types of syntactic ambiguities

$$[_{VP}\, cut]\, [_{NP}\, the\, potato]\, [_{PP}\, with\, a\, knife] \tag{23}$$

can be elaborated either by the parse:

$$[_{VP}\, cut\, [_{NP}\, the\, potato\, [_{PP}\, with\, a\, knife]]] \tag{24}$$

which represents a syntactic configuration based on proximity (*phrase2* is attached to *phrase1* and *phrase3* is attached to *phrase2*), or by:

$$[_{VP}\, cut\, [_{NP}\, the\, potato]\, [_{PP}\, with\, a\, knife]] \tag{25}$$

which is here the correct configuration. It contains both a contiguous and a long distance attachment: *phrase2* is attached to *phrase1* and *phrase3* is attached to *phrase1*.

We consider, however, that the process of attachment resolution can be generalised to other syntactic sequences and ambiguity configurations. On the one hand, we evaluated, not only one, by three types of phrase sequences: $vp - np - pp$, $vp - pp - pp$, and $pp - pp - pp$. On the other hand, these sequences cannot be reduced to only two syntactic configurations (two parses). They can be syntactically ambiguous in different ways. These ambiguities are introduced by adjective arguments and sentence adjuncts (see Table 8).

Table 8 shows phrase sequences that cannot be analysed by means of the two standard configurations underlying parses (24) and (25). None of the sequences in that table matches the two standard configurations. For instance, `a o decreto` (*to the decree*), which is the *phrase2* of the first example, is not attached to the head of *phrase1* but to the adjective `relativo` (*referring*). Similarly, in the second expression, `a o citado diploma` (*to the referred diploma*) is attached to the adjective `anexos` (*appended*) and not to the head of *phrase2*. Subcategorisation of adjectives introduces a new type of structural ambiguity, which makes attachment decisions more difficult to be taken. Finally, in the third sequence, `em a medida` (*in so far as*) is the beginning of an adverbial sentence, so it is not attached to one of the individual phrases but to the whole previous sentence. In sum, solving structural ambiguity cannot be reduced to a binary choice between the two configurations depicted above in (24) and (25). We return to this matter below.

Another important property of test data is that it contains incorrectly tagged words. We do not remove these cases since they can give us significant information about how (in)dependent of noisy data is our learning method.

8.3.2 The Evaluation Protocol Each sequence selected from the test corpus contains three phrases and two candidate attachments. So, given a test expression, two different attachment decisions will be evaluated:

Decision A: is *phrase2* attached to *phrase1* or is not attached at all?

Decision B: is *phrase3* attached to *phrase2*, to *phrase1*, or is not attached at all?

As we selected $633 * 2$ test expressions, and each expression implicitly contains two attachment decisions, the total number of decisions that we evaluated was $2,532$. By contrast, in most related approaches, test expressions are ambiguous in only two senses: *phrase3* is attached to either *phrase2* or *phrase1*. They do not consider the attachment between *phrase2* and *phrase1*. So, in these approaches, Decision A is not taken into account. Moreover, they do not evaluate those cases where *phrase3* is not attached to *phrase2* nor *phrase1*. In sum, only one decision per expression is evaluated, namely the decision concerning the PP-attachment. This type of evaluation, however, is not appropriate to measure the capability of the system to identify the no-standard structural ambiguities described above (subsection 8.3.1). For instance, we expect that the system does not propose the *pp* `ao diploma` *(to the [referred] diploma)* to be attached to the previous *np*, headed by `estatutos` in the second example of Table 8. The correct decision is to propose no attachment between the *pp* (*phrase3*) and none of the two previous phrases taking part in the sequence $vp - pp - pp$. The attachment is actually with a word, namely adjective `anexo`, which is not a direct constituent of the abstract sequence $vp - pp - pp$.

Another important aspect of the evaluation protocol is that CR overgenerates attachments. There are several cases in which the three phrases of a sequence are semantically related. In those cases, CR often proposes three attachments even if only two of them are syntactically allowed. For instance, take the $np - pp - pp$ sequence :

$$[_{np}\texttt{a remuneração}] \;\; [_{pp}\texttt{de o cargo}] \;\; [_{pp}\texttt{de secretário}] \tag{26}$$

$$(\textit{the salary concerning the post of secretary}) \tag{27}$$

which would be correctly analysed by using the same configuration as in parse (24) above, i.e.:

$$[_{np}\texttt{a remuneração} \;\; [_{pp}\texttt{de o cargo} \;\; [_{pp}\texttt{de secretário}]]] \tag{28}$$

Note that there exists a strong semantic relationship between *phrase3* (`de secretário`) and *phrase1* (`a remuneração`), even if they are not syntactically attached in (28). Taking into account the semantic requirements stocked in the lexicon (see Table 7), CR is induced to propose, besides the two correct attachments, a long distance dependency, which seems not to be syntactically correct in this particular case. We call this phenomenon "attachment overgeneration". When a sequence contains two semantically related phrases that are not actually syntactically dependent, CR overgenerates an additional attachment. Attachment overgeneration was found in $\approx 15\%$ expressions selected from the test corpus. In order to overcome this problem, we use a default rule based on Right Association. The default rule removes the long distance attachment and only proposes the two contiguous ones. This simple rule has an accuracy of more than 90% with regard to the 15% sequences containing overgeneration.

From a semantic viewpoint, attachment overgeneration seems not to be a real problem. The semantic interpretation of sequence (26) needs to account for all conceptual relations underlying the sequence. So, the semantic requirements linked `secretário` to `remuneração` (even if they are not syntactically dependent) are useful to build a semantic representation of the sequence.

8.3.3 Baseline (RA) Concerning the ability to propose correct syntactic attachments, we made a comparison between CR and a baseline strategy. As a baseline, we used the attachments proposed by Right Association (RA). For each sequence of the test data, RA always proposes the configuration underlying parses (28) and (24), that is: *phrase2* is attached to *phrase1*, and *phrase3* is attached to *phrase2*.

8.3.4 Similarity-Based Lexical Association We also compared CR to a very different learning strategy: the similarity-based lexical method (Sekine et al., 1992; Grishman and Sterling, 1994; Dagan, Marcus, and Markovitch, 1995; Dagan, Lee, and Pereira, 1998). This strategy is described in subsection 2.3 above. We simulated here a particular version of this strategy. First, we used the Log-Likelihood ratio as association score between pairs of syntactic positions and words. We restricted the lexical association procedure to suggest attachments only in cases where the absolute value of the ratio was greater than an empirically set threshold (> 3.00). Then, in order to generalise from unobserved pairs, a list of similar words were used to compute non-zero association scores. For this purpose, the thesaurus described in subsection 7.5 above turned out to be useful.

According to (Dagan, Marcus, and Markovitch, 1995), the similarity-based lexical association LA_{sim} between position p and word w is obtained by computing the average of likelihood ratios between p and the k most similar words to w:

$$LA_{sim}(p, w) = \frac{\sum_{i=0}^{k} LA(p, w_i)}{NZ} \tag{29}$$

where $LA(p, w_i)$ is the likelihood ratio between p and one of the k most similar words to w. NZ represents the number of non-zero values among $LA(p, w_1), LA(p, w_2) \ldots LA(p, w_k)$.

Co-requirements are also considered. Given dependency $(robj;$ `ratificar`$^{\downarrow}$, `lei`$^{\uparrow})$ (*ratify the law*), we compute the two following lexical associations:

$$LA_{sim}(< robj\_down, \, ratificar >, \texttt{lei})$$
$$LA_{sim}(< robj\_up, \, lei > \texttt{ratificar}) \tag{30}$$

The scores of these two associations are taken into account in the evaluation procedure. In particular, the sum of both scores (if each of them is greater than the threshold) will be used to make a decision on the attachment between a *np* headed by `lei` and a *vp* headed by `ratificar`.

8.3.5 Precision and Recall The evaluation of each attachment decision taken by the system can be:

- true positive (*tp*): the system proposes a correct attachment;
- true negative (*tn*): the system proposes correctly that there is no attachment;

- false positive (*fp*): the system proposes an incorrect attachment;

- false negative (*fn*): the system proposes incorrectly that there is no attachment.

The evaluation test measures the ability of the system to make true decisions. We call both *tp* and *tn* "true decisions" (*td*). As far as our strategy and the similarity-based approach are concerned, a false negative (*fn*) is interpreted as the situation in which the system has not enough subcategorisation information to make a decision. By contrast, the baseline always proposes an attachment.

Taking into account these variables, *precision* is defined as the number of true decisions suggested by the system divided by the number of total suggestions. That is:

$$precision = \frac{td}{td + fp} \tag{31}$$

Recall is computed as the number of true decisions suggested by the system divided by all the decisions that have been taken (i.e., the total number of ambiguities).

$$recall = \frac{td}{td + fp + fn} \tag{32}$$

In order to clearly understand the evaluation procedure, see Table 9. It displays the different attachment decisions taken on the following test sequence:

$[_{vp}$ `assistir` $[_{pp}$ `por o representante` $[_{pp}$ `de o Estado-Membro]]]`

(*assisted by the delegate of the Member-State*) (33)

The two correct attachments in 33, proposed by the human annotator, are compared against the attachment decisions proposed by the three methods at stake: heuristic with Boolean co-requirements (CR), Similarity-Based Lexical Association (LA_{sim}), and Right

CR	Decision A: $< iobj\_por\_D, assistir >$ requires `representante` : YES $< iobj\_por\_H, representante >$ requires `assistir` : YES Result : *tp* Decision B: $< iobj\_por\_D, assistir >$ requires `Estado-Membro` : NO $< iobj\_por\_H, Estado-Membro >$ requires `assistir` : NO $< de\_D, representante >$ requires `Estado-Membro` : YES $< de\_H, Estado-Membro >$ requires `representante` : YES Result: *tp*
LA_{sim}	Decision A: $LA_{sim}(< iobj\_por\_D, assistir >, \texttt{representante})$: 0 $LA_{sim}(< iobj\_por\_H, representante >, \texttt{assistir})$: 0 Result : *np* Decision B: $LA_{sim}(< iobj\_por\_D, assistir >, \texttt{Estado-Membro})$: 0 $LA_{sim}(< iobj\_por\_H, Estado-Membro >, \texttt{assistir})$: 0 $LA_{sim}(< de\_D, representante >, \texttt{Estado-Membro})$: 136.70 $LA_{sim}(< de\_H, Estado-Membro >, \texttt{representante})$: 176.38 Result: *tp*
RA	Decision A: [$_{vp}$ assistir [$_{pp}$ por o representate]] : YES Result: *tp* Decision B: [$_{pp}$ por o representate [$_{pp}$ de os Estados-Membros]] : YES Result: *tp*

Table 9
Evaluation of a test sequence.

Association (RA), which is the baseline. Table 9 assesses the two different decisions (A and B) taken by each method. Note that both CR and LA_{sim} take advantage of co-requirements. Indeed, each decision is taken after having considered two types of subcategorisation information: the requirements the dependent word must satisfy and the requirements that the head word must satisfy.

Decision A concerns the first candidate attachment, that is the dependency between `[`$_{vp}$ `assistir]` and `[`$_{pp}$ `por o representante]`. Let us analyse the behavior of the three methods. LA_{sim} incorrectly suggests that there is no attachment. The score of two internal requirements is 0, so the final decision is a false negative: *fn*. The system has not information on requirements because, on the one hand, the two phrases at stake do not co-occur in the training corpus, and, on the other, co-occurrences of phrases with similar words were not attested (and then no generalisation was allowed). CR, by contrast, is endowed with the appropriate requirements to correctly suggest an attachment (*tp*) between the two phrases, even in they are not attested in the training corpus. The clustering strategy allowed to learn that both $< iobj\_por\_D, assistir >$ requires `representante` and $< iobj\_por\_H, representante >$ requires `assistir`. Note that in order to suggest the attachment, it is not necessary to learn the two complementary requirements. As has been said in subsection 8.2, only one of them is enough to make the suggestion. Finally, RA also suggests the correct attachment. Indeed, the two phrases in 33 are related by right association.

As regards Decision B is concerned, the three methods correctly suggest that there is an attachment (*tp*) between `[`$_{np}$ `o representante]` and `[`$_{pp}$ `de o Estado-Membro]`.

BASELINE (RA)						
sequences	$\mathbf{Pr}_{EC}$	$\mathbf{Pr}_{PGR}$	$\mathbf{Rec}_{EC}$	$\mathbf{Rec}_{PGR}$	$\mathbf{F\text{-}S}_{RV}$	$\mathbf{F\text{-}S}_{PGR}$
$np-pp-pp$	.71	.72	.71	.72	.71	.72
$vp-np-pp$	.83	.80	.83	.80	.83	.80
$vp-pp-pp$	.75	.74	.75	.74	.75	.74
LEXICAL ASSOCIATION (LA$_{sim}$)						
sequences	$\mathbf{Pr}_{EC}$	$\mathbf{Pr}_{PGR}$	$\mathbf{Rec}_{EC}$	$\mathbf{Rec}_{PGR}$	$\mathbf{F\text{-}S}_{RV}$	$\mathbf{F\text{-}S}_{PGR}$
$np-pp-pp$	.77	.82	.66	.72	.71	.77
$vp-np-pp$	.90	.86	.75	.74	.82	.79
$vp-pp-pp$	.85	.89	.65	.70	.74	.78
BOOLEAN REQUIREMENTS (CR)						
sequences	$\mathbf{Pr}_{EC}$	$\mathbf{Pr}_{PGR}$	$\mathbf{Rec}_{EC}$	$\mathbf{Rec}_{PGR}$	$\mathbf{F\text{-}S}_{RV}$	$\mathbf{F\text{-}S}_{PGR}$
$np-pp-pp$	.85	.86	.73	.76	.78	.81
$vp-np-pp$	.92	.93	.75	.78	.83	.85
$vp-pp-pp$	.86	.91	.69	.75	.77	.82

Table 10
Evaluation taking into account three types of sequences and two corpora: *EC* and *PGR*.

8.3.6 Results Table 10 reports the test scores concerning the precision and recall of the experiments performed. These scores concern three methods, namely RA, LA$_{sim}$ and CR, two text corpora (*EC* and *PGR*), and three types of phrase sequences. There are no significant differences between the scores obtained from corpus *EC* and those from *PGR*. CR, for instance, obtains very similar F-Scores over the two corpora. However, there are important differences among the precision values associated to the three phrase sequences. In particular, the scores of sequence $vp - np - pp$ are significatively higher than those of the other sequences, regardless of the method employed. This is motivated by the fact that, in most $vp - np - pp$ sequences ($\approx 95\%$), there is a true attachment between *np* and *vp*. So, the precision score reached by the three methods with regard to this particular attachment decision is very high. Prepositional phrase attachments, by contrast, are more ambiguous. This situation leads sequences $vp-pp-pp$ and $np-pp-pp$ to be less predictable. Indeed, such sequences have two prepositional phrases involved in the attachment decisions.

Concerning the differences among the three methods, see Table 11. It averages the results of the three methods over the two corpora and the three phrase sequences. The total precision of our method (CR) reaches 0.89, i.e. 4 points more than LA_{sim}. Note that the precision value of LA_{sim} is not far from the values reached by other approaches to attachment resolution, based on the similarity-based lexical association strategy. For instance, the method described in (Grishman and Sterling, 1994) scores a precision of ≈ 0.84. Concerning the recall, CR also reaches 4 points more than LA_{sim}. It entails that, on the one hand, the ability of CR to learn accurate subcategorisation information is higher than that of LA_{sim}, and on the other hand, the ability of CR to learn from sparse data and to generalise is, at least, no lower than that of LA_{sim}.

	Prec.	**Recall**	**F-Score**
Baseline	.76	.76	.76
LA_{sim}	.85	.71	.77
CR	.89	.75	.81

Table 11
Total scores of the three methods. For each method, we compute the average of the three sequences and the two corpora.

The baseline score informs us that about 76% attachments are links by proximity. The remainder (24%) are either long distance attachments between *phrase3* and *phrase1*,

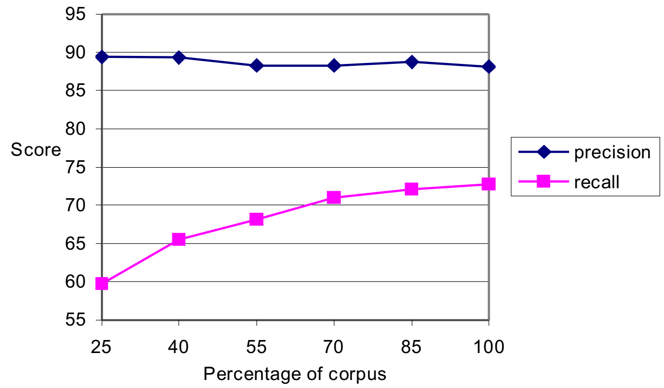

Figure 7
Variation of recall and precision as a function of corpus size

other attachments such as adjective complements, sentence modifiers, etc., or finally tagger errors. Note that there is no difference between precision and recall since RA always takes a (true or false) positive decision. So, there cannot be (true nor false) negative decisions.

Some tagger errors, especially those that appear systematically and regularly in the training corpus, have a negative influence on the precision of both LA_{sim} and CR. These methods are sensitive to noisy data.

In order to measure recall and precision stability, we ran the clustering process over 6 partitions (25%, 40%, 55%, 70%, 85% and 100%) of the of *E.C.* corpus. Figure 7 shows how recall improves with corpus size. However, the recall growth is more significant in the smaller partitions. In this particular corpus, recall stability seems to be achieved when the corpus contains 3 millions words. It follows that, in order to improve recall, we must use, not only a bigger training corpus, but also a more efficient clustering strategy, that is, a strategy that would be able to make more correct generalisations. Finally, note that precision neither increases nor decreases with the corpus size.

9 Conclusion and Future Work

This paper has presented a particular unsupervised strategy to automatically acquire syntactic and semantic requirements. Given a word, our aim was to learn two types of information: the syntactic positions in which the word appears and the semantic requirements associated with each syntactic position. Besides that, this strategy also allowed us to discriminate word senses. The strategy is mainly based on some linguistic assumptions. First, it was assumed that not only the syntactic Head imposes restrictions on its Dependent word, but also the latter may select for a specific type of Head. This phenomenon was called "co-requirement". Second, we claimed that similar syntactic positions share the same semantic requirements. So, we measured not similarity between words on the basis of their syntactic distribution, but similarity between syntactic positions on the basis of their word distribution. It was assumed that the latter kind of similarity conveys more pertinent information on linguistic requirements than the former one. The learning process allowed us to provide a lexicon with, among other information, both syntactic subcategorisation and selection restrictions. This information was used to constrain attachment heuristics.

In current work, we are using the learnt clusters in other NLP applications than attachment resolution. They are being used to automatically select word senses in context (Word Sense Disambiguation task). For this purpose, we are performing new experiences on less domain-specific text corpora, since they increase the number of senses per word. On the other hand, these clusters turn out to be very useful to check whether two or more different morphological forms of a word are semantically related or not. For instance, if "ratification of [_]" is similar to "ratify [_]", we may infer that the verb and the noun are semantically related.

In future work, we aim at extending the lexicon in order to increase the coverage of the parser. To do it, parsing and learning can be involved in a bootstrapping process. The dependencies proposed by heuristic CR will be used as input to discover new linguistic requirements. This new information will enable to update the lexicon, and then to propose new dependencies. At each cycle, the lexicon will be provided with new requirements and thereby the parser coverage will be higher. The successive "learning + parsing" cycles will stop as no more new information is acquired and no more new dependencies are proposed.

References

Allegrini, P., S. Montemagni, and V. Pirrelli. 2003. Example-based automatic induction of semantic classes through entropic scores. *Linguistica Computazionale*, pages 1–45.

Allegrini, Paolo, Simonetta Montemagni, and Vito Pirrelli. 2000. Learning word clusters from data types. In *Coling-2000*, pages 8–14.

Androutsopoulos, Ion and Robert Dale. 2000. Selectional restrictions in hpsg. In *18th Conference on Computational Linguistics (COLING)*, pages 15–20, Saarbrucken, Germany.

Basili, Roberto, Maria Pazienza, and Paola Velardi. 1992. Comptuational lexicons: the neat examples and the odd exemplars. In *Proc. of the 3rd ANLP*.

Beale, S., S. Niremburg, and E. Viegas. 1998. Constraints in computational semantics. In *COLING*.

Brent, Michel R. 1991. Automatic acquisition of subcategorization frames from untagged text. In *29th Annual Meeting of ACL*, pages 209–214.

Ciaramita, Massimiliano and Mark Johnson. 2000. Explaining away ambiguity: Learning verb selectional preference with bayesian networks. In *COLING-00*.

Collins, Michael and James Brooks. 1995. Prepositional phrase attachment through a backed-off model. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 27–38, Cambridge.

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1998. Similarity-based methods of word coocurrence probabilities. *Machine Learning*, 43.

Dagan, Ido, Shaul Marcus, and Shaul Markovitch. 1995. Contextual word similarity and estimation from sparse data. *Computer Speech and Language*, 9(2):123–152.

de la Clergerie, Eric. 2002. Construire des analyseurs avec dyalog. In *Proceedings of TALN'02*.

Faure, David and Claire Nédellec. 1998. Asium: Learning subcategorization frames and restrictions of selection. In *ECML98, Workshop on Text Mining*.

Framis, Francesc Ribas. 1995. On learning more appropriate selectional restrictions. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics*, Dublin.

Gamallo, Pablo. 2003. Cognitive characterisation of basic grammatical structures. *Pragmatics and Cognition*, 11(2):209–240.

Gamallo, Pablo, A. Agustini, and Gabriel P. Lopes. 2003. Learning subcategorisation information to model a grammar with co-restrictions. *Traitement Automatic de la Langue*, 44(1):93–117.

Gamallo, Pablo, Caroline Gasperin, Alexandre Agustini, and Gabriel P. Lopes. 2001. Syntactic-based methods for measuring word similarity. In V. Mautner, R. Moucek, and K. Moucek, editors, *Text, Speech, and Discourse (TSD-2001)*. Berlin:Springer Verlag, pages 116–125.

Gildea, Daniel and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Grefenstette, Gregory. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, USA.

Grishman, Ralph and John Sterling. 1994. Generalizing automatically generated selectional patterns. In *Proceedings of the 15th International on Computational Linguistics (COLING-94)*.

Harris, Z. 1985. Distributional structure. In J.J. Katz, editor, *The Philosophy of Linguistics*. New York: Oxford University Press, pages 26–47.

Hindle, Donald and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

Hudson, Richard. 2003. The psychological reality of syntactic dependency relations. In *MTT2003*, Paris.

Langacker, Ronald W. 1991. *Foundations of Cognitive Grammar: Descriptive Applications*, volume 2. Stanford University Press, Stanford.

Li, Hang and Naoki Abe. 1998. Word clustering and disambiguation based on co-occurrence data. In *Coling-ACL'98*, pages 749–755.

Lin, Dekang. 1998. Automatic retrieval and clustering of similar words. In *COLING-ACL'98*, Montreal.

Lin, Dekang and Patrick Pantel. 2001. Induction of semantic classes from natural language text. In *SIGKDD-01*, San Francisco.

L.Kovacs and P. Baranyi. 2002. Document clustering based on concept lattice. In *IEEE Int. Conf. System Man and Cybernetics (SMC'02)*, Hammamet, Tunisia.

Manning, Chris and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge.

Manning, Cristopher. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *31th Annual Meeting of ACL*, pages 235–242.

Marques, N. and G.P. Lopes. 2001. Tagging with small training corpora. In F. Hoffmann, D. Hand, N. Adams, D. Fisher, and G. Guimaraes, editors, *Advances in Intelligent Data Analysis*. LNCS, Springer Verlag, pages 62–72.

Marques, Nuno, Gabriel P. Lopes, and Carlos Coelho. 2000. Mining subcategorization information by using multiple feature loglinear models. In *10th CLIN*, pages 117–126, UILOTS Utrecht.

Niemann, Michael. 1998. Determining pp attachment through semantic associations and preferences. In *ANLP Post Graduate Workshop*, pages 25–32.

Pantel, Patrick and Dekan Lin. 2000. An unsupervised approach to prepositional phrase attachment using contextually similar words. In *ACL'00*, pages 101–108, Hong Kong.

Pantel, Patrick and Dekan Lin. 2002. Discovering word senses from text. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 613–619, Edmonton, Canada.

Pustejovsky, James. 1995. *The Generative Lexicon*. MIT Press, Cambridge.

Ratnaparkhi, Adwait, Jeff Reymar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the ARPA Human Language Technology Workshop)*, pages 250–255.

Reinberger, M-L. and W. Daelemans. 2003. Is shallow parsing useful for unsupervised learning of semantic clusters? In *4th Conference on Intelligent Text Processing and Computational Linguistics (CICLing-03)*, pages 304–312, Mexico City.

Resnik, Philip. 1997. Selectional preference and sense disambiguation. In *ACL-SIGLEX Workshop on Tagging with Lexical Semantics*, Washinton DC.

Rocio, V., E. de la Clergerie, and J.G.P. Lopes. 2001. Tabulation for multi-purpose partial parsing. *Journal of Grammars*, 4(1).

Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–124.

Sekine, Satoshi, Jeremy Carrol, Sofia Ananiadou, and Jun'ichi Tsujii. 1992. Automatic learning for semantic collocation. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, pages 104–110.