# Evaluating Two Different Methods for the Task of Extracting Bilingual Lexicons from Comparable Corpora

**Pablo Gamallo Otero**

Departamento de Língua Espanhola
Universidade de Santiago de Compostela
Galiza, Spain
pablogam@usc.es

**Abstract**

In this paper, two different approaches to extract bilingual lexicons from comparable corpora are evaluated and compared. One uses syntactic contexts, and the other windows of tagged words. On a Spanish-Galician comparable corpus of $2 \times 10$ million words, syntactic contexts produce significantly better results for both frequent and less frequent words.

## 1. Introduction

In the last ten years, some methods have been proposed to acquire bilingual lexicons from non-parallel and comparable corpora. A non-parallel, comparable corpus (hereafter ”comparable corpus”) consists of sets of documents in several languages dealing with a given topic or domain, but in which the documents have been composed independently of each other in the different languages. As comparable texts are much easier to collect than parallel corpora, especially for minority languages and for a given domain, there is a growing interest in acquiring bilingual lexicons from comparable corpora. Indeed, they are more abundant, less expensive, and easily available via web than parallel texts. The main assumption underlying the approaches using comparable corpora is that a word in the target language is a candidate translation of a word in the source language, if the former tends to co-occur with expressions that are also translations of expressions co-occurring with that word in the source language. That is, the associations between a word and its context seed words are preserved in comparable texts of different languages.

The main contribution of this paper is to describe and compare two different approaches for extracting bilingual lexicons from comparable corpora. One of the tested approaches uses as contexts syntactic dependencies that can be extracted for each word in a corpus by robust parsers. The other approach uses the classic windowing technique around each word. Both techniques are applied to the same non-parallel, comparable corpus. A somehow related evaluation was performed by (Grefenstette, 1993), but on a monolingual corpus. According to the experiments we will describe later, the dependency-based method provides much better results than the windowing approach, very especially if only the top translation candidate is considered. In addition, further experiments will be performed to compare the efficiency of different similarity measures.

The paper is organized as follows: Section 2. introduces some comparable-based strategies to learn translation equivalents. Then, sections 3. and 4. describe a window and a syntax based method, respectively. The former is inspired by the Rapp approach (Rapp, 1999), and the later relies on a very simple dependency parser. Finally, in Section 5., some experiments will be performed against the same comparable corpus in order to evaluate several features of the 2 methods described in the previous sections.

## 2. Some Related Work

There is a growing interest in approaches focused on extracting word translations from comparable corpora (Fung and McKeown, 1997; Fung and Yee, 1998; Rapp, 1999; Chiao and Zweigenbaum, 2002; Dejean et al., 2002; Kaji, 2005; Gamallo, 2007; Saralegui et al., 2008). Most of them share a standard strategy based on context similarity. This strategy can be described as follows: a word $w_2$ in the target language is a candidate translation of $w_1$ in the source language if the context expressions with which $w_2$ co-occurs tend to be translations of the context expressions with which $w_1$ co-occurs. The basis of the method is to find the target words that have the most similar distributions with a given source word. The starting point of this strategy is a list of bilingual expressions that are used to build the context vectors of all words in both languages. This list is usually provided by an external bilingual dictionary. In Gamallo (2007), however, the starting list is provided by bilingual correlations previously extracted from a parallel corpus. In Dejean (2002), the method relies on a multilingual thesaurus instead of an external bilingual dictionary. In all cases, the starting list contains the “seed expressions” required to build context vectors.

There exist other approaches to bilingual lexicon extraction which do not use a starting list of seed expressions (Fung, 1995; Rapp, 1995; Diab and Finch, 2001). Yet, Fung (1995) failed to reach an acceptable accuracy rate for actual use, Rapp (1995) had strong computational limitations, and Diab et al. (2001) was applied only to non-parallel texts in the same language.

As far as the standard approach is concerned, works mainly differ in the coefficients (Dice, Jaccard, Cosine, City-Block, Lin ...) used to measure the similarity between context vectors. One of the contributions of this paper is to evaluate the efficiency of these coefficients to extract

This work has been supported by the Galician Government, within the project EXTRALEX, ref: PGIDIT07PXIB204015PR.

translation equivalents from comparable corpora. Moreover, works based on the standard approach also differ in the way they define word contexts. Most of them model contexts as a window of words of size $N$ (*window-based paradigm*). Another technique (*syntax-based paradigm*) defines contexts by means of dependency relationships (Gamallo, 2007). The two techniques are very similar except that in one case a partial syntactic analysis is performed. As have been said, the main contribution of this paper is to evaluate and compare the results of each technique against the same comparable corpus.

## 3. Window-Based Method

The first technique for extracting bilingual lexicons does not perform any kind of syntactic analysis, but simply consider some window of words as forming the context of the compared words. We follow the method described in Rapp (1999), which is one of the most cited works on this topic.

### 3.1. Building Context Vectors

It is assumed that there is a small bilingual dictionary available at the beginning. The entries of the dictionary are considered as the starting list of seed words. Texts in both languages are lemmatized and POS tagged, and function words are removed. Then, for each lemma we build a context vector whose dimensions are seed words in different window positions with regard to the lemma. For instance, if we have chosen the window size 2, we compute a first context vector of lemma A whose dimensions are the seed words co-occurring 2 positions to the left of A. We also compute a second vector counting co-occurrences between A and the seed words appearing 1 position to the left of A. The same for the 2 positions following lemma A. Finally, we combine the 4 vectors of length *n* (where *n* is the size of the seed lexicon) into a single vector of length *4n*. This method takes into consideration word order to define contexts.

Each vector dimension of a lemma takes as value the number of co-occurrences between the lemma and a seed word in a given window position. Besides simple context frequency, additional weights can be considered, namely, a statistical degree of association between the lemma and each seed word. In the experiments described later, we will make use of log-likelihood ratio. This procedure is performed on the two monolingual texts.

### 3.2. Vector Similarity

Given a context vector defining a lemma of the source language, we compute a similarity score for each target vector. Then, a ranking list is built according to this score. The lemmas represented by the best-ranked target vectors are considered candidate translations of the given source lemma. We used several similarity coefficients for comparing pairs of vectors: *city-block* (Rapp, 1999), *cosine* (Fung and McKeown, 1997; Fung and Yee, 1998; Chiao and Zweigenbaum, 2002; Saralegui et al., 2008), *lin* (Lin, 1998a), and two different versions of both *jaccard* and *dice*. This way, the similarity of two lemmas, $w_1$ and $w_2$, is computed as follows:

$$\mathsf{city\text{-}block}(w_1, w_2) = \sum_j |A(w_1, c_j) - A(w_2, c_j)|$$

$$\mathsf{cosine}(w_1, w_2) = \frac{\sum_j A(w_1, c_j)A(w_2, c_j)}{\sqrt{\sum_j (A(w_1, c_j))^2}\sqrt{\sum_k (A(w_2, c_k))^2}}$$

$$\mathsf{diceMin}(w_1, w_2) = \frac{2\sum_j \min(A(w_1, c_j), A(w_2, c_j))}{\sum_j A(w_1, c_j) + \sum_k A(w_2, c_k)}$$

$$\mathsf{diceProd}(w_1, w_2) = \frac{2\sum_j A(w_1, c_j)A(w_2, c_j)}{\sum_j (A(w_1, c_j))^2 + \sum_k (A(w_2, c_k))^2}$$

$$\mathsf{jaccardMin}(w_1, w_2) = \frac{\sum_j \min(A(w_1, c_j), A(w_2, c_j))}{\sum_j \max(A(w_1, c_j), A(w_2, c_j))}$$

$$\mathsf{jaccardProd}(w_1, w_2) =$$

$$\frac{\sum_j A(w_1, c_j)A(w_2, c_j)}{\sum_j (A(w_1, c_j))^2 + \sum_k (A(w_2, c_k))^2 - \sum_i A(w_1, c_i)A(w_2, c_i)}$$

$$\mathsf{lin}(w_1, w_2) = \frac{\sum_{c_i \in C_{1,2}} (A(w_1, c_j) + A(w_2, c_j))}{\sum_j A(w_1, c_j) + \sum_k A(w_2, c_k)}$$

Where $A(w_1, c_j)$ is an association value of a vector of length $n$, with $j$, $i$, and $k$ ranging from 1 to $n$. In our experiments, the association value stands for either the simple co-occurrences of lemma $w_1$ with a contextual seed word $c_j$, or the log-likelihood ratio between the lemma and its context. For both $jaccardProd$ and $diceProd$ metrics, the association values of two lemmas with the same context are joined using their product (Chiao and Zweigenbaum, 2002; Saralegui et al., 2008), while for $jaccardMin$ (Grefenstette, 1994; Kaji and Aizono, 1996) and $diceMin$ (Curran and Moens, 2002; van der Plas and Bouma, 2004; Gamallo, 2007) only the smallest association weight is considered. As regards $lin$ coefficient, the association values of common contexts are summed (Lin, 1998a), where $c_j \in C_{1,2}$ if only if $A(w_1, c_j) > 0$ and $A(w_2, c_j) > 0$.

## 4. Syntax-Based Method

The second technique to extract translation equivalents relies on the identification of syntactic dependencies. So, context vectors will be provided with syntactic information.

### 4.1. Partial Parsing with Regular Expressions

As in the previous method, monolingual texts are lemmatized and POS tagged. Then, instead of searching for windows positions around lemmas, we make use of regular expressions to identify syntactic dependencies. Regular expressions represent basic patterns of POS tags which are supposed to stand for binary dependencies between two

| Dependencies | Patterns of POS tags |
|---|---|
| (green$_5$, $mod_<$, jacket$_6$)<br>(big$_{10}$, $mod_<$, $d$dog$_{11}$) | *R$_1$: $s/(\mathbf{A_i})(\mathbf{N_j})/\mathbf{N_j}/$ |
| () | *R$_2$: $s/(\mathbf{N_i})(\mathbf{N})_\mathbf{j}/\mathbf{N_i}/$ |
| (man$_2$, $with_3$, jacket$_5$) | *R$_3$: $s/(\mathbf{N_i})(\mathbf{P_k})(\mathbf{N})_\mathbf{j}/\mathbf{N_i}/$ |
| (see$_6$, $obj_>$, dog$_{11}$) | R$_4$: $s/(\mathbf{V_i})(?:D_k\vert R_n)*(\mathbf{N})_\mathbf{j}/\mathbf{V_i}/$ |
| (see$_6$, $obj_<$, man$_2$) | R$_5$: $s/(?:D_k)*(\mathbf{N_i})(?:R_n)*(\mathbf{V})_\mathbf{j}/\mathbf{V_j}/$ |
| () | R$_6$: $s/(\mathbf{V_i})(?:R_n)*(\mathbf{P_k})(?:\vert D_m\vert R_r)*(\mathbf{N})_\mathbf{j}/\mathbf{V_i}/$ |

Table 1: Dependency triplets and patterns of POS tags

lemmas. Our experiments are focused on dependencies with verbs, nouns, and adjectives. Our parsing strategy consists of a sequence of syntactic rules, each rule being defined by a specific pattern of tags that stands for a binary dependency. This strategy is implemented as a finite-state cascade (Abney, 1996). Let's take an example. Suppose our corpus contains the following tagged sentence:

> a_D$_1$ man_N$_2$ with_P$_3$ a_D$_4$ green_A$_5$ jacket_N$_6$ see_V$_7$ yesterday_R$_8$ a_D$_9$ big_A$_{10}$ dog_N$_{11}$

The aim is to identify dependencies between lemmas using basic patterns of POS tags. Dependencies are noted as triplets: $(head, rel, dependent)$. The first column of Table 1 shows the 5 triplets generated from the sentence above using the patterns appearing in the second column. Patterns are organized in a sequence of substitution rules in such a way that the input of a rule $R_n$ is the output of a rule $R_m$, where $m \leq n$. A rule substitutes the POS tag of the head word (right side) for the whole pattern of tags representing the head-dependent relation (left side). The first rule, $R_1$, takes as input a string containing the ordered list of all tags in the sentence:

> D$_1$N$_2$P$_3$D$_4$A$_5$N$_6$V$_7$R$_8$D$_9$A$_{10}$N$_{11}$

The left pattern in this rule identifies two specific adjective-noun dependencies, namely "A$_5$N$_6$" and "A$_{10}$N$_{11}$". As a result, it removes the two adjective tags from the input list. Then, rule $R_3$ is applied to the output of $R_1$. The left pattern of this rule matches "N$_2$P$_3$D$_4$A$_5$" and rewrites the following ordered list of tags:

> D$_1$N$_2$V$_7$R$_8$D$_9$N$_{11}$

This list is the output of the following applicable rule, $R_4$, which produces "D$_1$N$_2$V$_7$". Finally, rule $R_5$ is applied and gives as result only one tag, V$_7$, which is associated to the root head of the sentence: the verb "see". As this verb does not modify any word, no rule can be applied and the process stops. This is in accordance with the main assumption of dependency-based analysis, namely, a word in the sentence may have several modifiers, but each word may modify at most one word (Lin, 1998b). In sum, each application of a rule, not only rewrites a new version of the list of tags, but also generates the corresponding dependency triplet. So, even if we do not get the correct root head at the end of the analysis, the parser generates as many triplets as possible. This strategy can be seen as partial and robust parsing, as faster as identifying contextual words with a window-based technique.

The 5 triplets in Table 1 where generated from 4 substitution rules, each matching a type of dependency: adjective-noun, noun-prep-noun, verb-noun, and noun-verb. The sentence analysed above does not contain triplets instantiating noun-noun and verb-prep-noun dependencies. Wildcards $(?:\ D|R)*$ stand for optional determiners and adverbs, that is, they represent optional sequences of determiners or/and adverbs that are not considered for triplets. Rules with an asterisk can be applied several times before applying the next rule (e.g., when a noun is modified by several adjectives). Subscript numbers allow us to link tags in the patterns with their corresponding lemmas in the sentence.

To represent triplets, we use 4 types of binary relations: prepositions, left modifiers (noted as $mod_<$), right objects ($obj_>$), and left objects ($obj_<$). The latter two are generic dependencies between verb and nouns. They are likely to be specified with further linguistic information. For instance, a left object can be seen as a *direct object* if there is a passive form of a transitive verb; otherwise the left object is a *subject*. As we are not provided with information on transitivity, our list of dependencies does not contain subjects nor direct objects. Furthermore, long-distance dependencies are not taken into account. This is because rules are organised in such a way that they resolve attachment ambiguities by "Minimal Attachment" and "Right Association". Finally, relative clauses are also considered. However, for the sake of simplicity, Table 1 does not show the rules dealing with this phenomenon.

Note that the patterns of tags in Table 1 work well with English texts, but they are so generic that they can be used for many languages. To extract triplets from texts in Romance languages such as Spanish, French, Portuguese, or Galician, at least, 2 tiny changes are required: to provide a new pattern with dependent adjectives at the right position of nouns ($mod_>$), and to take as the head of a noun-noun dependency the noun appearing at the left position. Our main grammar only contains 10 generic rules suitable for Romance languages while the English grammar was provided with 9 rules. The linguistic knowledge required is then very low. The experiments that will be described later were performed over Spanish and Galician text corpora.

### 4.2. Lexico-Syntactic Contexts

The second step of our syntax-based method consists in extracting lexico-syntactic contexts from the dependencies and counting the occurrences of lemmas in those contexts. This information is stored in a collocation database. The extracted triplets of our example allow us to easily build the collocation database depicted in Table 2. The first line of

| Lemmas | Lexico-Syntactic Patterns and freqs. |
|---|---|
| man | $<$ (see, $obj_<, N), 1 >$<br>$< (N, with,$ jacket$), 1 >$ |
| see | $< (V, obj_<,$ man$), 1 >$<br>$< (V, obj_>,$ dog$), 1 >$ |
| big | $<$ (dog, $mod_<, A), 1 >$ |
| dog | $< (N, mod_<,$ big$), 1 >$<br>$<$ (see, $obj_>, N), 1 >$ |
| green | $<$ (jacket, $mod_<, A), 1 >$ |
| jacket | $< (N, mod_<,$ green$), 1 >$<br>$<$ (man, $with, N), 1 >$ |

Table 2: Collocation database of lemmas and lexico-syntactic contexts

the table describes the entry "man". This noun occurs once in two lexico-syntactic contexts, namely that representing the left position ($obj_<$) of the verb "see", (see, $obj_<, N$), and that denoting the noun position being modified by the prepositional complement "with a jacket". The second line describes the entry "see", which also occurs once in two different lexico-syntactic contexts: $(V, obj_<, man)$ and $(V, obj_>, dog)$, i.e., it co-occurs with both a left object, "man", and a right object: "dog". The remaining lines describe the collocation information of the remaining nouns and adjectives appearing in the sentence above.

Notice we always extract 2 complementary lexico-syntactic contexts from a triplet. For instance, from (man, $with$, jacket), we extract:
$(N, with,$ jacket$)$ (man, $with, N)$
This is in accordance with the notion of co-requirement defined in (Gamallo et al., 2005). In this work, two syntactically dependent words are no longer interpreted as a standard "predicate-argument" structure, where the predicate is the active function imposing syntactic and semantic conditions on a passive argument, which matches such conditions. On the contrary, each word in a binary dependency is perceived simultaneously as a predicate and an argument. In the example above, (man, $with, N$) is seen as an unary predicate that requires nouns denoting parts of men (e.g. jackets), and simultaneously, $(N, with,$ jacket$)$ is another unary predicate requiring entities having jackets (e.g. men).

### 4.3. Building Syntax-Based Context Vectors

In this approach, the seed expressions used as cross-language contexts are not bilingual pairs of words as in the window-based approach, but bilingual pairs of lexico-syntactic contexts. The process of building a list of seed syntactic contexts consists of two steps: first, we generate a large list from an external bilingual dictionary Second, this starting list is used to build the context vectors of the lemmas appearing in the comparable corpus.

To show how we generate bilingual correlations between lexico-syntactic contexts using bilingual dictionaries, let's take an example. Suppose that an English-Spanish dictionary translates the noun "import" into the Spanish counterpart "importación". To generate bilingual pairs of lexico-syntactic contexts from these two nouns, we follow basic linking rules such as: (1) if "import" is the left object of a verb (i.e, if it is the subject of the verb), then its Spanish equivalent, "importación", is also the left object; (2) if "import" is modified by an adjective at the left position, then its Spanish equivalent is modified by an adjective at the right position; (3) if "import" is restricted by a prepositional complement headed by the preposition *in*, then its Spanish counterpart is restricted by a prepositional complement headed by the preposition *en*. The third rule needs a closed list of English prepositions and their more usual Spanish translations. For each entry (noun, verb, or adjective), we only generate a subset of all possible lexico-syntactic contexts. Table 3 depicts the contexts generated from the bilingual pair "import-importación" by making use of 6 basic linking rules for English-Spanish. As regards the other language pairs, we use a very similar set of rules. The human effort required to develop such rules is very low.

The second step consists in building a context vector for each lemma appearing in the comparable corpus. Vector dimensions are constituted by those contexts of the collocation database created above that also appear in the list of bilingual contexts generated from the external dictionary. For instance, if (import, $of, N$) both occurs in the corpus (i.e, it is in the collocation database), and belongs to the list of bilingual pairs, then it must be taken as a dimension in a context vector.

Finally, vector similarity between lemmas is computed as in the window-based approach.

## 5. Experiments and Evaluation

Three experiments were performed in order to evaluate three different parameters of the extraction techniques described in this paper: First, the quality of dependency relationships was compared to the linguistic relevance of relations between words co-occurring the same window. Second, we compared the efficiency of different similarity coefficients. And third, we evaluated the accuracy of both the syntax and the window based approaches described above.

### 5.1. Experiment 1

We first evaluated the triplets generated by our dependency-based parser. For this purpose, we manually analysed a Spanish text containing 200 dependency triplets. We considered only those types of dependencies likely to be identified by our parser, namely, prepositional complements, left and right verbal objects, and nominal modifiers. Among the verbal complements and objects, we also include the relationships between a noun and the main verb in a relative clause modifying the noun. As in Lin (1998b), the gold standard dependencies are called *key*. On the other hand, the triplets generated by our parser from the same text are called *answer*. Once the key and the answer are both represented as dependency triplets, we can compare and calculate *precision* and *recall*. Precision is the percentage of dependency relationships in the answer that are also found in the key. Recall is the percentage of dependency relationships in the key that are also found in the answer.

Table 4 summarizes the evaluation results considering the different types of dependency relationships. The total precision is $74\%$ while recall reaches $64\%$. These results are not far from baseline dependency parsers for English. For instance, in Lin (1998b), if we only consider the precision

| English | Spanish |
|---|---|
| (import, $of \mid to \mid in \mid for \mid by \mid with, N$) | (importación, $de \mid a \mid en \mid para \mid por \mid con, N$) |
| ($N, of \mid to \mid in \mid for \mid by \mid with$, import) | ($N, de \mid a \mid en \mid para \mid por \mid con$, importación) |
| ($V, obj_>$, import) | ($V, obj_>$, importación) |
| ($V, obj_<$, import) | ($V, obj_<$, importación) |
| ($V, of \mid to \mid in \mid for \mid by \mid with$, import) | ($V, de \mid a \mid en \mid para \mid por \mid con$, importación) |
| (import, $mod_<, A$) | (importación, $mod_>, A$) |

Table 3: Bilingual correlations between contexts generated from the translation pair: import-importación.

| Dependency type | Precision | Recall |
|---|---|---|
| modification | 78% | 94.5% |
| left object | 67% | 45% |
| right object | 90% | 79% |
| pp attachment | 68% | 55% |
| **Total** | **74%** | **64%** |

Table 4: Evaluation of different types of dependency relations.

| Type of strategy | Precision | Recall | F-Meas. |
|---|---|---|---|
| Dependency-Based | 74% | 64% | 69% |
| Window-Based | 32% | 91% | 47% |

Table 5: Evaluation of dependency and window based relationships.

of dependencies such as subject, complement, pp attachment, and relative clause, the average score is 76%, with 70% of recall.

The linguistic relevance of dependency triplets was compared to that of window-based contexts. For this purpose, we computed precision and recall of the relationship between window-based contexts and their co-occurrence lemmas. More precisely, we used the same Spanish text to generate an answer consisting of binary relations between lemmas and their context lemmas within a window of size $N$ (where $N = 2$, see Section 3.). Here, types of dependencies cannot be taken into account. So, if a relationship between a lemma and a context lemma is instantiated by one of the specific dependencies in the key, then such a relation is considered to be correct. Results are depicted in Table 5 . We used the same key as in the previous evaluation.

These results show that a rudimentary dependency parser allows us to extract much more precise contexts than a window-based strategy. However, the latter reaches a greater recall. Regarding computational efficiency, the two strategies turned out to be similar. Identifying dependency triplets takes the same time as extracting window-based contexts: about $9,000$ words per second, using a 2.33GHz CPU. We will see in the third experiment which contexts are more significant for translation equivalents extraction.

## 5.2. Experiment 2

The aim of the second experiment was to compare the efficiency of several similarity metrics in the task of bilingual lexicon extraction. Each metric was combined with two weighting schemes: simple occurrences and log likelihood. The strategy used here was the window-based method described in Section 3.. For each source lemma, we obtain a ranked list of 10 target lemmas considered as their translation equivalents.

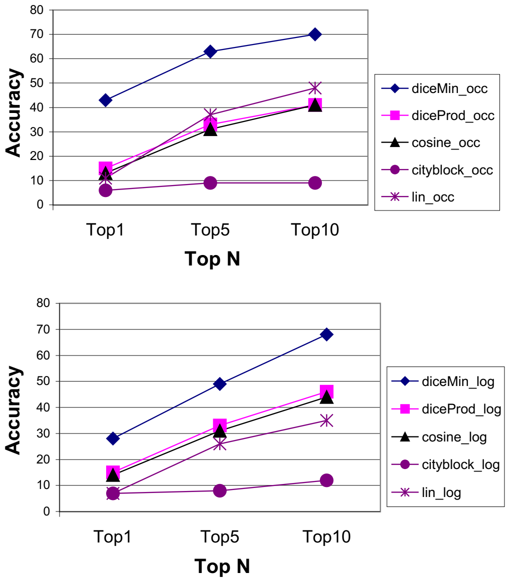

Figure 1: Percentile rank of the measures weighted with occurrences and log-like

### 5.2.1. Training Corpus and Bilingual Dictionary

The experiment was performed on a Spanish and Galician comparable corpus being constituted by news from on-line journals published between 2005 and 2006. As the Spanish corpus, we used $10,5$ million words of two newspapers: *La Voz de Galicia* and *El Correo Gallego*, and as Galician corpus 10 million words from *Galicia-Hoxe*, *Vieiros* and *A Nosa Terra*. The Spanish and Galician texts were lemmatized and POS tagged using a multilingual free software: Freeling (Carreras et al., 2004). Since the orientation of the newspapers is quite similar, the two monolingual texts can be considered as more or less comparable. The bilingual dictionary used to select seed words is the lexical resource integrated in OpenTrad, an open source machine translation system for Spanish-Galician (Armentano-Oller et al., 2006). The dictionary contains about $25,000$ entries.

Table 6: Syntax-Based Approach

| **Cov(%)** | **Nouns** $(74,205$ cntxs) | | | **Adjs** $(13,047$ cntxs) | | | **Verbs** $(39,985$ cntxs) | | |
|---|---|---|---|---|---|---|---|---|---|
| | **acc-1** | **acc-10** | **freq** | **acc-1** | **acc-10** | **freq** | **acc-1** | **acc-10** | **freq** |
| 50 | .87 | .89 | $> 1,221$ | .95 | .97 | $> 1,239$ | .99 | .99 | $> 3,290$ |
| 80 | .60 | .72 | $> 123$ | .71 | .76 | $> 187$ | .89 | .94 | $> 770$ |
| 90 | .38 | .45 | $> 28$ | .58 | .63 | $> 49$ | .84 | .94 | $> 266$ |

Table 7: Window-Based Approach

| **Cov(%)** | **Nouns** $(128,504$ cntxs) | | | **Adjs** $(94,669$ cntxs) | | | **Verbs** $(111,007$ cntxs) | | |
|---|---|---|---|---|---|---|---|---|---|
| | **acc-1** | **acc-10** | **freq** | **acc-1** | **acc-10** | **freq** | **acc-1** | **acc-10** | **freq** |
| 50 | .49 | .80 | $> 1,221$ | .72 | .86 | $> 1,239$ | .62 | .84 | $> 3,290$ |
| 80 | .26 | .51 | $> 123$ | .43 | .70 | $> 187$ | .56 | .78 | $> 770$ |
| 90 | .14 | .36 | $> 28$ | .27 | .51 | $> 49$ | .47 | .65 | $> 266$ |

#### 5.2.2. Evaluation

To evaluate the efficiency of the different coefficients in the process of extracting bilingual lexicons, we elaborated an evaluation protocol with the following characteristics. A random sample of 200 test adjectives was selected from a list of adjectives occurring in the Spanish corpus. This list consists of those adjectives whose frequency achieves 80% of the total occurrences of adjectives in the corpus (80% of coverage). At this level of coverage, we computed 3 types of accuracy: *accuracy-1* is the number of correct translation candidates ranked first divided by the number of test lemmas. Then, *accuracy-5* and *accuracy-10* represent the number of correct candidates appearing in the top 5 and top 10, respectively, divided by the number of test lemmas. Indirect associations are judged to be incorrect.

#### 5.2.3. Results

Figure 1 shows results using 7 different metrics combined with two types of weighted context vectors: simple occurrences and log-likelihood. In sum, we performed 14 different experiments. As the scores obtained using jaccard and dice coefficients were very similar, for the sake of simplicity, only dice scores ($diceMin$ and $diceProd$) are depicted in the figure.

These results show that the use of log-likelihood improves slightly *cityblock*, *cosine*, and *diceProd*, compared to the use of simple occurrences. However, *diceMin* (and so *jaccardMin*) as well as *lin* get better scores when simple occurrences are considered. On the other hand, there is a significant difference between *diceMin* compared to the other coefficients, regardless of the weight employed. With *diceMin*, 70% of the adjectives find their correct translation within the top 10 words, which is much better than the score achieved by $lin_{occ}$ (49%), the second better coefficient. The reason of such a difference is that the product (or the sum as in *lin*) of association values maximizes odd similarities whereas the choice of the smallest value minimizes them. This is in accordance with the results obtained by (Curran and Moens, 2002) and (van der Plas and Bouma, 2004) Finally, the distance coefficient *city-block* seems to be unsuitable for this type of data.

### 5.3. Experiment 3

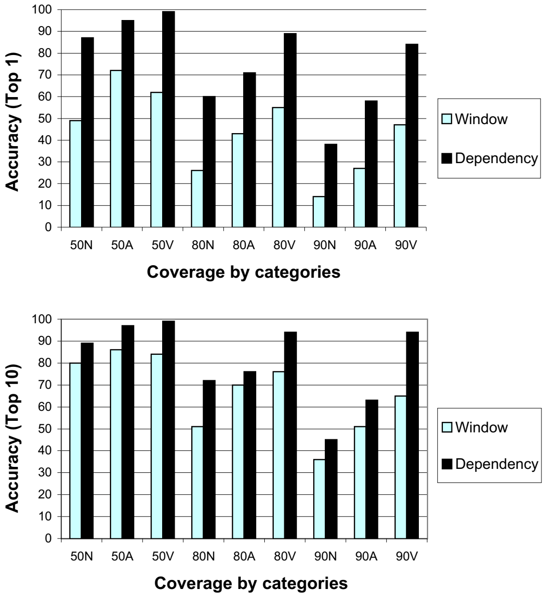

Figure 2: Comparison of accuracy between the two approaches considering both top 1 (above) and top 10 (below) translation equivalents

The aim of the third experiment was to compare the accuracy of both window and syntax based methods to extract bilingual lexicons. For this purpose, we used the same comparable corpus and bilingual dictionary as in the previous experiment. Similarity measure was computed with the most effective metric/weight combination: $diceMin_{occ}$. The evaluation protocol was more elaborated. We evaluated both *accuracy-1* and *accuracy-10* at three levels of coverage: 50%, 80%, and 90%, taking into account three POS categories: nouns, adjectives, and verbs. As nouns, we included proper nouns constituted by both mono and multi-word lemmas. Results are depicted in two tables: 6 and 7. They convey information on accuracy of three POS categories at different levels of coverage. They also show the number of contexts (i.e., vector size) used to define the lemmas of each category. Notice the number of syntactic contexts is much smaller than the number of contexts based on windows. As the size of context vectors in the syntactic approach is not very large, the process of computing similarities turns out to be more efficient. In addition, in order to analyze the impact the frequency has on the results, we include lemma frequencies of each category at each level of coverage. For instance, the nouns evaluated at 80% of coverage have more than 123 occurrences in the source corpus. This is not far from the usual threshold used in related work, where only words with frequency $> 100$ are evaluated.

It can be seen in tables 6 and 7 that the approach based on syntactic contexts (i.e., dependencies) works much better than that based on the windowing technique, at whatever level of coverage and for the three POS categories. The reason is that syntactic dependencies allow us to define finer-grained contexts which are semantically motivated. It can also be seen that the differences between both approaches are more significant when we only consider *accuracy-1* (see Figure 2): for instance, .87 against .49 percent considering nouns at 50% of coverage. If we look among the top 10 ranked lemmas (*accuracy-10*), differences are not so important: .89 against .80.

## 6. Conclusion

In this paper, we described and compared two techniques focused on bilingual extraction from comparable corpora. The syntax-based method produced better results than the window-based technique for very frequent ($> 1,221$), less frequent ($> 123$), and low frequent ($> 28$) nouns, adjectives, and verbs. In addition, the former method is more computationally efficient since it defines and uses smaller context vectors. On the other hand, the syntactic method can be seen as a knowledge-poor strategy (as the window-based approach), because our partial parsing relies on few generic regular expressions. Moreover, as the generic knowledge underlying the parsing technique is used to identify basic dependencies for the same family of natural languages, our syntax-based strategy turns out to be almost as language-independent as any windowing technique. Finally, we compared many similarity coefficients and discovered that two specific versions of Dice and Jaccard, *diceMin* and *jaccardMin*, are the best suited metrics for this specific task.

## 7. References

Steven Abney. 1996. Part-of-speech tagging and partial parsing. In Ken Church, Steve Young, and Gerrit Bloothooft, editors, *Corpus-Based Methods in Language and Speech.* Kluwer Academic Publishers, Dordrecht.

Carme Armentano-Oller, Rafael C. Carrasco, Antonio M. Corb-Bellot, Mikel L. Forcada, Mireia Ginest-Rosell, Sergio Ortiz-Rojas, Juan Antonio Prez-Ortiz, Gema Ramrez-Snchez, Felipe Snchez-Martnez, and Miriam A. Scalco. 2006. Open-source portuguese-spanish machine translation. In *Lecture Notes in Computer Science, 3960*, pages 50–59.

X. Carreras, I. Chao, L. Padró, and M. Padró. 2004. An open-source suite of language analyzers. In *4th International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal.

Y-C. Chiao and P. Zweigenbaum. 2002. Looking for candidate translational equivalents in specialized, comparable corpora. In *19th COLING'02*.

James R. Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In *ACL Workshop on Unsupervised Lexical Acquisition*, pages 59–66, Philadelphia.

H. Dejean, E. Gaussier, and F. Sadat. 2002. Bilingual terminology extraction: an approach based on a multilingual thesaurus applicable to comparable corpora. In *COLING 2002*, Tapei, Taiwan.

Mona Diab and Steve Finch. 2001. A statistical word-level translation model for comparable corpora. In *Proceedings of the Conference on Content-Based Multimedia Information Access (RIAO).*

Pascale Fung and Kathleen McKeown. 1997. Finding terminology translation from non-parallel corpora. In *5th Annual Workshop on Very Large Corpora*, pages 192–202, Hong Kong.

Pascale Fung and Lo Yuen Yee. 1998. An ir approach for translating new words from nonparallel, comparable texts. In *Coling'98*, pages 414–420, Montreal, Canada.

Pascale Fung. 1995. Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. In *14th Annual Meeting of Very Large Corpora*, pages 173–183, Boston, Massachusettes.

Pablo Gamallo, Alexandre Agustini, and Gabriel Lopes. 2005. Clustering syntactic positions with similar semantic requirements. *Computational Linguistics*, 31(1):107–146.

Pablo Gamallo. 2007. Learning bilingual lexicons from comparable english and spanish corpora. In *Machine Translation SUMMIT XI*, Copenhagen, Denmark.

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

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

Hiroyuki Kaji and Toshiko Aizono. 1996. Extracting word correspondences from bilingual corpora based on word co-occurrence information. In *16th Conference on Computational Linguistics (Coling'96)*, pages 23–28, Copenhagen, Danmark.

Hiroyuki Kaji. 2005. Extracting translation equivalents from bilingual comparable corpora. In *IEICE Transactions 88-D(2)*, pages 313–323.

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

Dekang Lin. 1998b. Dependency-based evaluation of minipar. In *Workshop on Evaluation of Parsing Systems*, Granada, Spain.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *33rd Conference of the ACL'95*, pages 320–322.

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated english and german corpora. In *ACL'99*, pages 519–526.

X. Saralegui, I. San Vicente, and A. Gurrutxaga. 2008. Automatic generation of bilingual lexicons from comparable corpora in a popular science domain. In *LREC 2008 Workshop on Building and Using Comparable Corpora*.

Lonneke van der Plas and Gosse Bouma. 2004. Syntactic contexts for finding semantically related words. In *Meeting of Computational Linguistics in the Netherlands (CLIN2004)*.