Distributional Thesaurus vs. WordNet: A Comparison of Backoff Techniques for Unsupervised PP Attachment*

Hiram Calvo,[1] Alexander Gelbukh,[1] and Adam Kilgarriff [2]

[1] Center for Computing Research, National Polytechnic Institute, Mexico
hcalvo@sagitario.cic.ipn.mx, www.Gelbukh.com

[2] Lexical Computing Ltd., United Kingdom
adam@lexmasterclass.com

Abstract. Prepositional Phrase (PP) attachment can be addressed by considering frequency counts of dependency triples seen in a non-annotated corpus. However, not all triples appear even in very big corpora. To solve this problem, several techniques have been used. We evaluate two different backoff methods, one based on WordNet and the other on a distributional (automatically created) thesaurus. We work on Spanish. The thesaurus is created using the dependency triples found in the same corpus used for counting the frequency of unambiguous triples. The training corpus used for both methods is an encyclopaedia. The method based on a distributional thesaurus has higher coverage but lower precision than the WordNet method.

1 Introduction

The Prepositional Phrase (PP) attachment task can be illustrated by considering the canonical example *I see a cat with a telescope*. In this sentence, the PP *with a telescope* can be attached to *see* or *cat*. Simple methods based on corpora address the problem by looking at frequency counts of word-triples or dependency triples: *see with telescope* vs. *cat with telescope*. In order to find enough occurrences of such triples, a very large corpus is needed. Such corpora are now available, and the Web can also be used [4, 27]. However, even then some combinations of words do not occur. This is a familiar effect of Zipf's law: few words are very common and there are many words that occur with a low frequency [14], and the same applies to word combinations.

To address the problem, several backoff techniques have been explored. In general, 'backing off' consists of looking at statistics for a set of words, when there is insufficient data for the particular word. Thus *cat with telescope* turns into ANIMAL *with* INSTRUMENT and *see with telescope* turns into *see with* INSTRUMENT (capitals denote sets of instrument-words, animal-words, etc.) One way to identify

* Work done under partial support of Mexican Government (CONACyT, SNI, PIFI-IPN, CGEPI-IPN) and RITOS-2.

H. Calvo, A. Gelbukh, A. Kilgarriff. Distributional Thesaurus vs. WordNet: A Comparison of Backoff Techniques for Unsupervised PP Attachment. Computational Linguistics and Intelligent Text Processing (CICLing-2005), Lecture Notes in Computer Science, N 3406, Springer-Verlag, 2005, pp. 172–183.

Table 1. State of the art for PP attachment disambiguation

Human (without context)			Use WordNet backoff			Use thesaurus backoff		
Ratnaparkhi	[20]	88.2	Stetina and Nagao	[24]	88.1	Pantel and Lin	[19]	84.3
Mitchell	[16]	78.3	Li and Abe 1998	[12]	85.2	McLauchlan	[15]	85.0

the set of words associated with a given word is to use WordNet, and another is to use a distributional thesaurus. A distributional thesaurus is a thesaurus generated automatically from a corpus by finding words which occur in similar contexts to each other [8, 25, 26]. Both approaches have already been explored (for English) and have been shown to yield results close to human disambiguation, see Table 1.

Experiments using different techniques have been carried out independently, and to date there are no evaluations which compare WordNet with distributional thesauruses. In this paper we compare those two approaches, as proposed in [10]. We use a single corpus in both cases to enable us to compare results. The same corpus is used for generating the thesaurus and the WordNet generalizations. The corpus is also used for counting the dependency triples.

Our work is on Spanish. This is, to the best of our knowledge, the first work exploring backoff methods for PP attachment for a language other than English.

2 PP Attachment with no Backoff

2.1 Building the Resources

The main resource is the count of dependency triples (DTC). In order to increase coverage, instead of considering strictly adjacent words, we consider dependency relations between word types (lemmas). Only unambiguous dependency relations are considered. For example the following two sentences: *I see with a telescope. A cat with three legs is walking*, will provide the dependency triples *see, with, telescope* and *cat, with, legs*, respectively. However, the sentence *I see a cat with a telescope* will not provide any dependency triple, as it is an ambiguous case.

We extract all dependency triples from our corpus in a batch process. We first tag the text morphologically and then group adjectives with nouns, and adverbs with verbs. Then, we search for the patterns *verb preposition noun*, *noun preposition noun*, *noun verb*, and *verb noun*. Determiners, pronouns and other words are ignored.

Following Lin [13], dependency triples consist of two words and the grammatical relationship, including prepositions, between two words in the input sentence. To illustrate the kind of dependency triples extracted, consider a micro-corpus (μC) consisting of two sentences: *A lady sees with a telescope*; and *The lady with a hat sees a cat*. The triples corresponding to this μC are shown in Figure 1. We then denote the number of occurrences of a triple <*w,r,w'*> as |*w,r,w'*|. From μC, |*lady,SUBJ,see*|=2 and |*lady*,*with*,*hat* |=1. |*,*,*| denotes the total number of triples (10 in μC), an asterisk * represents any word or relation. In μC, |*see,*,**| = 4, |*,*with,**| = 2, |*,*,lady*| = 2.

see, SUBJ, lady	see, SUBJ, lady	see, OBJ, cat
lady, SUBJ-OF, see	lady, SUBJ-OF, see	cat, OBJ-OF, see
see, with, telescope	lady, with, hat	
telescope, with_r, see	hat, with_r, lady	

Figure 1. Dependency triples extracted from μC

$$x=|x,*,*| \quad p=|*,p,*| \quad n=|*,*,n_2| \quad t=|*,*,*| \quad \bar{x}=t-x,\quad \bar{p}=t-p,\quad \bar{n}=t-n$$
$$xpn=|x,p,n_2|,\quad \bar{x}pn=|*,p,n_2|-xpn,\quad x\bar{p}n=|x,*,n_2|-xpn,\quad xp\bar{n}=|x,p,*|-xpn$$
$$\bar{x}\bar{p}n=n-xpn-\bar{x}pn-x\bar{p}n,\quad \bar{x}p\bar{n}=p-xpn-\bar{x}pn-xp\bar{n}$$
$$x\bar{p}\bar{n}=x-xpn-x\bar{p}n-xp\bar{n},\quad \bar{x}\bar{p}\bar{n}=t-\left(xpn+\bar{x}pn+x\bar{p}n+xp\bar{n}+\bar{x}\bar{p}n+\bar{x}p\bar{n}+x\bar{p}\bar{n}\right)$$
$$\begin{aligned} score = {} & xpn\cdot\log[xpn/(x\cdot p\cdot n/t^2)]+\bar{x}pn\cdot\log[\bar{x}pn/(\bar{x}\cdot p\cdot n/t^2)]+\\ & x\bar{p}n\cdot\log[x\bar{p}n/(x\cdot\bar{p}\cdot n/t^2)]+xp\bar{n}\cdot\log[xp\bar{n}/(x\cdot p\cdot\bar{n}/t^2)]+\\ & \bar{x}p\bar{n}\cdot\log[\bar{x}p\bar{n}/(\bar{x}\cdot p\cdot\bar{n}/t^2)]+\bar{x}\bar{p}n\cdot\log[\bar{x}\bar{p}n/(\bar{x}\cdot\bar{p}\cdot n/t^2)]+\\ & x\bar{p}\bar{n}\cdot\log[x\bar{p}\bar{n}/(x\cdot\bar{p}\cdot\bar{n}/t^2)]+\bar{x}\bar{p}\bar{n}\cdot\log[\bar{x}\bar{p}\bar{n}/(\bar{x}\cdot\bar{p}\cdot\bar{n}/t^2)]\end{aligned}$$

for VScore, x is v, for NScore, x is n_1

Figure 2. Formulae for calculating three-point log-likelihood

The grammatical relationships without prepositions will be useful later for thesaurus-building, where word similarity will be calculated based on contexts shared between two words. By now, we will use this resource (DTC) only to count triples of (*verb*, *preposition*, *noun*$_2$) and (*noun*$_1$, *preposition*, *noun*$_2$) to decide a PP attachment. This is explained in the following section.

2.2 Applying the Resources

The task is to decide the correct attachment of p,n_2 given a 4-tuple of verb, noun$_1$, preposition, noun$_2$: (v,n_1,p,n_2). The attachment of p,n_2 can be either to the verb v or the noun n_1. The simplest unsupervised algorithm attaches according to which is the highest of VScore = $|v,p,n_2|$ and NScore = $|n_1,p,n_2|$. When both values are equal we say that this attachment is not decidable by this method.

The corpus used for counting dependency triples (DTC) in this experiment was the whole Encarta encyclopaedia 2004 in Spanish [1]. It has 18.59 M tokens, 117,928 types in 73MB of text, 747,239 sentences, and 39,685 definitions. The corpus was tagged using the TnT Tagger trained with the manually tagged (morphologically) corpus CLiC-TALP[1] and lemmatized using the Spanish Anaya dictionary [11].

Once the corpus is morphologically tagged and lemmatized, the dependency triples are extracted. Encarta produced 7M dependency triple tokens, amongst which there were 3M different triples, i.e. 3M dependency-triple types. 0.7M tokens (0.43M types) involved prepositions.

[1] http://clic.fil.ub.es. The TnT tagger trained with the CLiC-TALP corpus has a performance of over 92% 17.

Table 2. Different formulae for calculating VScore and NScore

	description	VScore	NScore
S	the simplest one	$\|v,p,n_2\|$	$\|n_1,p,n_2\|$
S2	considering doubles too	$\|v,p,n_2\| \times \|v,p,*\|$	$\|n_1,p,n_2\| \times \|n_1,p,*\|$
LL3	Log likelihood ratio	See Figure 2	
Feat	Simplified Roth features 19 and 23	$\log(\|*,p,*\|/\|*,*,*\|) +$ $\log(\|v,p,n_2\|/\|*,*,*\|) +$ $\log(\|v,p,*\|/\|v,*,*\|) +$ $\log(\|*,p,n_2\|/\|*,*,n_2\|)$	$\log(\|*,p,*\|/\|*,*,*\|) +$ $\log(\|n_1,p,n_2\|/\|*,*,*\|) +$ $\log(\|n_1,p,*\|/\|v,*,*\|) +$ $\log(\|*,p,n_2\|/\|*,*,n_2\|)$

We used four different formulae for calculating VScore and NScore, listed in Table 2. The first two formulae can be seen as the calculus of the probability of each triplet, e.g. $p(v,p,n_2)=|v,p,n_2|/|*,*,*|$. Since both VScore and NScore are divided by the same number |*,*,*|, it can be omitted without any difference. For log-likelihood[2] formulae, see Figure 2.

Following the PP attachment evaluation method by Ratnaparkhi *et al.* [20], the task is to determine the correct attachment given a 4-tuple (v,n_1,p,n_2). We extracted 1,137 4-tuples, along with their correct attachment (N or V), from the manually tagged corpus Cast-3LB[3] [18]. Sample 4-tuples are shown in Table 3.

Table 3. Example of 4-tuples (v,n_1,p,n_2) used for evaluation

4-tuples	English gloss
informar comunicado del Banco_Central N	inform communication of Central_Bank N
producir beneficio durante periodo V	produce benefit during period V
defender resultado de elección N	defend results of election N
recibir contenido por Internet V	receive contents by Internet V
planchar camisa de puño N	iron shirt of cuff N

The baseline can be defined in two ways. The first is to assign all attachments to $noun_1$. This gives precision of 0.736. The second is based on the fact that the preposition *de* 'of' attaches to a noun in 96.9% of the 1,137 4-tuples.[4] This gives a precision of 0.855, a high value for a baseline, considering that the human agreement level is 0.883. To avoid this highly biased baseline, we opted for excluding all 4-tuples with preposition *de*—no other preposition presents such a high bias. Then all

Table 4. Comparison of formulae for calculating VScore and NScore

Method	Coverage	Precision
Baseline	1.000	0.661
S	0.127	0.750
S2	0.127	**0.773**
LL3	0.127	0.736
Feat	0.127	0.717

2 Log-likelihood was calculated using the Ngram statistics package, see [2].

3 Cast-3LB is part of the 3LB project, financed by the Science and Technology Ministry of Spain. 3LB, (FIT-150500-2002-244 and FIT 150500-2003-411)

4 This is valid also for English. For the training set provided by Ratnaparkhi, the preposition *of* attaches to a noun in 99.5% of the 20,801 4-tuples.

our evaluations are done using only 419 of the 1,137 4-tuples extracted. The baseline in this case consists of assigning all attachments to the verb, which gives 66.1% precision. The human inter-tagger agreement for 4-tuples excluding preposition *de* is 78.7%, substantially lower than human agreement for all 4-tuples. Results are shown in Table 4.

The highest precision is provided by formula S2, so from now on we will use this formula to compare results with backoff methods.

3 WordNet Backoff

3.1 Building the Dictionary

We are looking for a wider coverage of dependency relations in order to decide a correct PP attachment. To achieve this, we construct a dictionary which uses WordNet to find a generalization of dependency relations. For example, we seek the generalization of *eat with fork*, *eat with spoon* and *eat with knife* into *eat with* {*tableware*}. Note that {*tableware*} is not a word, but a concept in WordNet. WordNet provides the knowledge that *fork*, *spoon* and *knife* are {*tableware*}. This way, if an unseen triple is found, such as *eat with chopsticks,* WordNet can help by saying that *chopsticks* are a {*tableware*} too, so that we can apply our knowledge about *eat with* {*tableware*}.

Before we describe our method, let us introduce some notation. Every word *w* is linked to one or more synsets in WordNet corresponding to its different senses. W_n denotes the synset corresponding to the *n*-th sense of *w,* and *N* the total number of senses. Each one of these synsets has several paths to the root by following their hypernyms. W_n^m denotes the *m-th* hypernym of the *n-th* sense of *w',* and M_n the depth, i.e. the number of hypernyms to the root for sense number *n*.

For example, *glass* in WordNet has 7 senses. The third hypernym of the fourth sense of *glass* is denoted by W_4^3 = *astronomical_telescope*. See below an extract for *glass* from WordNet to illustrate this.

sense 2: *glass* (drinking glass) → container → instrumentality → artifact → object → whole → object → entity

sense 4: *glass* (spyglass) → refracting_telescope → optical_telescope → **astronomical_telescope** → telescope → magnifier → scientific_instrument → instrument → device → instrumentality → artifact → object → entity

Our WordNet backoff method is based on [5] and [6]. To extend a score (NScore or VScore) through WordNet, we must consider all triples involving the same *w* and *r*, varying *w'* (as in the case of learning *eat with* {*tableware*} from several examples of *eat with* *). This set of triples is denoted by <*w,r,**>. For each involved w', we distribute evenly[5] each score s(*w,r,w'*) among each one of its senses of *w'* (as in [22]).

[5] We assume an equiprobable distribution, which is problematic. However, there are currently no comprehensive sense tagged texts for Spanish from which we could extract sense distributions.

Then this result is propagated to all hypernyms W_n^m. This value is accumulative: higher nodes in WordNet collect information from all their daughters. This way, more general concepts summarize the usage (frequency of triples) of their specific concepts (hyponyms).

To avoid over-generalization (that is, the excessive accumulation at top levels,) depth must be considered. Sometimes the depth of hypernyms' chain is very large (as in *glass*' sense 4) and sometimes small (sense 2 of *glass*). A useful propagation formula that allows generalization and considers depth of chains of hypernyms is:

$$s(w,r,W_n^m) = [s(w,r,w')/\mathrm{N}] \times [1-(m-1/M_n)] \quad (1)$$

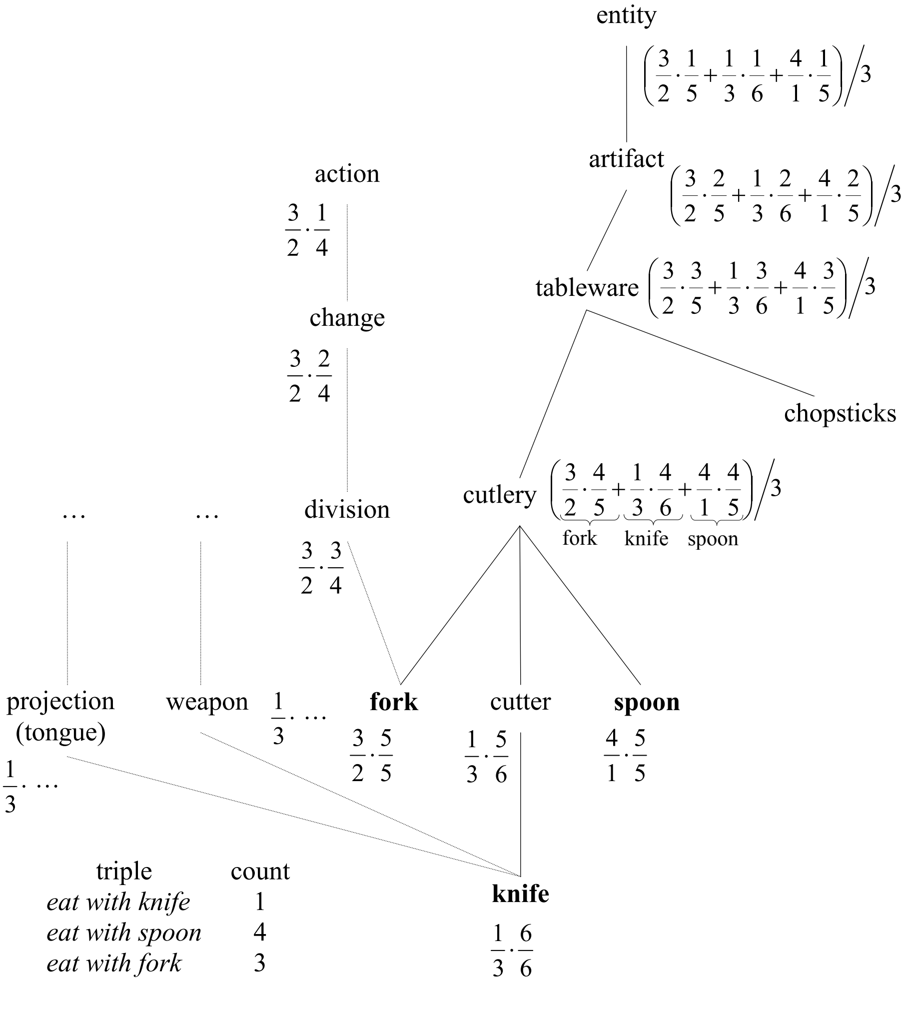


Figure 3. Example of propagation of triple's counts in WordNet

Table 5. Examples of relation triples (*w,r,w'*) with WordNet backoff

w	*r*	*w'*	English	score
comer	*con*	*mano*	hand	3.49
'eat'	'with'	*cubiertos*	cutlery	1.31
		tenedor	fork	1.19
matar	*con*	*arma*	weapon	0.27
'kill'	'with'	*armamento*	armaments	0.23
		utillaje	utensil	0.18

In addition, the number of triples contributing to a certain WordNet node is counted for averaging at upper levels. That is, after considering the *k* triples <*w,r,**>, we count the number of triple types contributing to each node. Then, the value of each node is divided by such number.

To illustrate our algorithm, see Figure 3. For this example suppose we only have three triples—each one is listed along with its count in Figure 3. The frequency count for each triple is added to the corresponding word in WordNet. For *eat with fork*, the node for the word *fork* is labeled with 3 counts for *eat with*. *fork* may be used with other combinations of words, but we show here only values for *eat with,* i.e., <*w,r,**>. Accordingly to Formula (1), this value is divided by the number of senses of *fork*. In this example we assume two different senses of *fork*, with different hypernyms each: {*division*} and {*cutlery*}. Focusing on the {*cutlery*} branch, we can see how this value is propagated towards to {*entity*}. For this branch there are 5 levels of depth from {*entity*} to *fork* (M_2=5)—the other branch has 4 levels (M_1=4). Following the propagation of *fork* up in the tree, it can be seen how each level has a lower weight factor—for {*tableware*} is 3/5 and for {*entity*} only 1/5. Each node is accumulative; because of this, {*cutlery*} accumulates the values for *fork*, *knife* and *spoon.* The value for {*cutlery*} is divided by 3 because the number of types of contributing triples to this node is 3. If we had another triple *eat with chopsticks* then {*cutlery*} would remain untouched, but {*tableware*} would be divided by 4.

For this experiment we used Spanish EuroWordNet[6] 1.0.7 (S-EWN) [7]. It has 93,627 synsets (62,545 nouns, 18,517 adjectives, 12,565 verbs), 51,593 hyponym/ hypernym relations, 10,692 meronym relations and 952 role information entries (noun agent, instrument, location or patient). We propagated all dependency triples in DTC using Formula (1) (creation of DTC was explained in Section 2.1.)

The WordNet backoff algorithm presented in this section produces subjectively good results. In Table 5 the first three top qualifying triples with *con* as relation for two common Spanish verbs are listed.

3.2 Using the Dictionary

To decide a PP attachment in a 4-tuple (v,n_1,p,n_2), we calculate NScore for (n_1,p,n_2), and VScore for (v,p,n_2) as in Section 2.2. The highest score determines the

[6] S-EWN was Developed jointly by the University of Barcelona (UB), the Nat University of Open Education (UNED), and the Polytechnic University of Cataluña (UPC), Spain.

attachment. WordNet backoff is applied when a triple is not found. In this case, n_2 is substituted by its hypernyms until the score from the new triple (x,p, W_n^m) is found in the previously calculated WordNet-extended-scores. When calculating NScore, x is n_1, and when calculating VScore, x is v. The highest score determines the attachment. Note that we are backing off only n_2. We decided not to back off v because the verb structure in S-EWN has very few hypernym relations for verbs (7,172) and the definition of a hypernym for a verb is not clear in many cases. Since we do not back off v, we cannot back off n_1 as this would introduce a bias of NScores against VScores. Also note that W_n^m is a specific synset in the WordNet hierarchy, and hence it has a specific sense. The problem of disambiguating the sense of n_2 is solved by choosing the highest value from each set of senses in each hypernym layer; see [5] and [24] for WSD using PP attachment information. Results for this method will be presented in Section 5.

Following the example from Figure 3, suppose we want to calculate the VScore for *eat with chopsticks*. Since this triple is not found in our corpus of frequency counts, we search for the hypernyms of *chopsticks*, in this case, {*tableware*}. Then, the value of this node is used to calculate VScore.

4 Thesaurus Backoff

4.1 Building the Dictionary

Here we describe the automatic building of a thesaurus so that words not found in the dependency triples can be substituted by similar words. This similarity measure is based on Lin's work [19]. This thesaurus is based on the similarity measure described in [13]. The similarity between two words w_1 and w_2 as defined by Lin is:

$$sim_{lin}(w_1, w_2) = \frac{\sum_{(r,w)\in T(w_1)\cap T(w_2)} (I(w_1,r,w) + I(w_2,r,w))}{\sum_{(r,w)\in T(w_1)} I(w_1,r,w) + \sum_{(r,w)\in T(w_2)} I(w_2,r,w)}$$

$$I(w,r,w') = \log \frac{|w,r,w| \times |*,r,*|}{|w,r,*| \times |*,r,w'|}$$

$T(w)$ is the set of pairs (r,w') such that $I(w,r,w')$ is positive. The algorithm for building the thesaurus is the following:

```
for each word type w1 in the corpus
    for each word type w2 in the corpus
          sims(w1) ← {simlin(w1,w2), w2}
    sort sims(w1) by similarity in descending order
```

Table 6. Example of similar words using Lin similarity method

word *w*	similar word *w'*	English	$sim_{lin}(w,w')$
guitarrista	*pianista*	pianist	0.141
'guitarist'	*fisiólogo*	physiologist	0.139
	educador	teacher	0.129
devoción	*afecto*	affection	0.095
'devotion'	*respeto*	respect	0.091
	admiración	admiration	0.078
leer	*editar*	to edit	0.078
'to read'	*traducir*	to translate	0.076
	publicar	to publish	0.072

Like the WordNet method, this gives subjectively satisfactory results: Table 6 lists the 3 most similar words to *guitarrista* 'guitarrist', *devoción* 'devotion', and *leer* 'to read'.

4.2 Using the Dictionary

To decide a PP attachment in a 4-tuple (v,n_1,p,n_2), our algorithm calculates NScore for (n_1,p,n_2), and VScore for (v,p,n_2) as in Section 2.2. The highest score determines the attachment. When a triple is not found, the backoff algorithm is applied. In this case, n_2 is substituted by its most similar word n'_2 calculated using $sim_{lin}(n_2, n'_2)$. If the new triple (x,p,n'_2) is found in the count of dependency triples (DTC), then it is used for calculating the score. If it is not found, then the next most similar word is tried for a substitution, until the new triple (x,p,n'_2) is found. When calculating NScore, x is n_1; when calculating VScore, x is v. The highest score determines the attachment. The algorithm is shown below. When n=1, the n-th most similar word corresponds to the first most similar word—for example *pianist* for *guitarist*. For n=2 it would be *physiologist*, and so on.

```
To decide the attachment in (v,n1,p,n2):

      VSCore = count(v,p,n2)
      NScore = count(n1,p,n2)
      n, m ← 1
      if NScore = 0
      |  while NScore = 0 & exists n-th word most similar to n2
      |  |      simn2  ← n-th word most similar to n2
      |  |      factor ← sim(n2,simn2)
      |  |      NScore ← count(n1,p,simn2) × factor
      └  └      n      ← n + 1
      if VScore = 0
      |  while VScore = 0 & exists n-th word most similar to n2
      |  |      simn2  ← m-th word most similar to n2
      |  |      factor ← sim(n2,simn2)
      |  |      VScore ← count(n1,p,simn2) × factor
      └  └      m      ← m + 1

      if NScore = VScore then cannot decide
      if NScore > Vscore then attachment is to n1
      if NScore < Vscore then attachment is to v
```

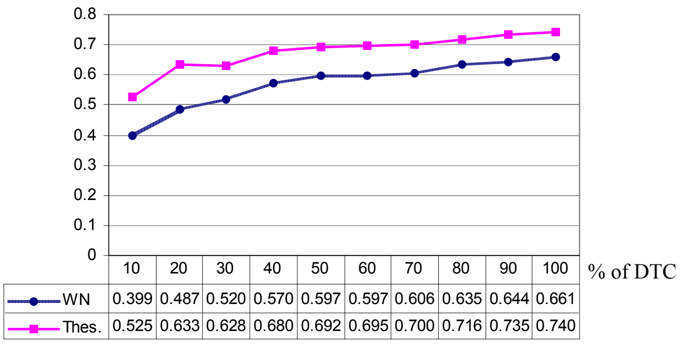
Table 7. Results of our experiments for PP attachment disambiguation

Method	Coverage	Precision	Average
Manual agreement (human)	1.000	0.787	0.894
Default to verb (baseline)	1.000	0.661	0.831
No backoff	0.127	**0.773**	0.450
WordNet backoff	0.661	0.693	0.677
Distributional thesaurus backoff	**0.740**	0.677	**0.707**

5 Comparison of Methods

In this section we compare results of the three methods: no backoff, WordNet backoff and thesaurus backoff. The results are listed in Table 7, along with the baseline and manual agreement results. The third column shows the average between coverage and precision. Note that the baseline shown in Table 7 involves some supervised knowledge: most of attachments, after excluding *de* cases, are to noun. The highest precision, coverage and average values are in boldface. After excluding *de* cases, we have 419 instances. For 12.7% of them all three algorithms do the same thing, so the

Coverage



Precision

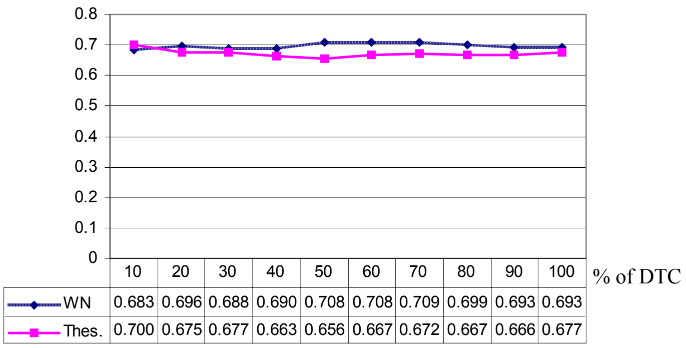


Figure 4. Precision and coverage using different percentages of triple counts (0–100%)

differences between WordNet backoff and distributional thesaurus backoff are based on the remaining 366 cases.

Not all cases are covered by these backoff methods either because no substitution can be found for a certain word (such as several acronyms or proper names), or because even after trying all possible substitutions the triple was not found in DTC. In general, this coverage is low because of the size of the corpus for counting attachment frequencies. Although an encyclopaedia provides a text with many different words, the number of prepositional attachments extracted is rather low. We believe that using a bigger corpus will yield higher coverage measures but will keep the same relationship between the backoff methods studied, as suggested by our experiments which use only randomly chosen partial percentages of the DTC corpus. This is shown in Figure 4. Note that we are using a totally unsupervised model. That is, in both algorithms we do not use any other backoff technique for not covered cases.

6 Conclusions

Amongst the three methods evaluated for PP attachment, the best average measure was 0.707 using thesaurus backoff, due to its greater coverage compared with other methods. However, it has lower precision than WordNet backoff. The method with no backoff had a very low coverage (0.127) but for the attachments covered the results were the best, at 0.773 close to manual agreement. (Remember that this agreement is calculated excluding a highly biased preposition: *de* 'of', which practically is always attached to nouns.) Performance of WordNet backoff could be increased by adding information of the sense distribution for each word, instead of assuming an equiprobable distribution, although this would render this method closer to a supervised approach, and moreover no resource providing sense distributions for Spanish is available.

Our results indicate that an automatically built resource (in this case, a thesaurus) can be used instead of a manually built one and still obtain similar results.

In our future work we shall explore using much larger corpora for gathering counts of triples, and we shall experiment with more sophisticated algorithms for using the thesaurus to determine attachments.

References

1. Biblioteca de Consulta Microsoft Encarta 2004, Microsoft Corporation, 1994–2004
2. Banerjee, S. and Ted Pedersen, T., The Design, Implementation, and Use of the Ngram Statistic Package, in *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, 2003, 370–381
3. Brants, Thorsten. TnT: A Statistical Part-of-Speech Tagger. *Proceedings of the 6th Applied Natural Language Processing Conference*, Seattle, WA, USA, 2000
4. Calvo, Hiram and Alexander Gelbukh. Improving Disambiguation of Prepositional Phrase Attachments Using the Web as Corpus, Procs. of CIARP'2003, Cuba, 592–598, 2003
5. Calvo, Hiram and Alexander Gelbukh. Unsupervised Learning of Ontology-Linked Selectional Preferences, Procs. *of* CIARP'2004, Puebla, Mexico, 2004

6. Clark, Stephen and David Weir. Class-based Probability Estimation Using a Semantic Hierarchy, *Computational Linguistics* 28(2), 2002
7. Farreres, X., G. Rigau and H. Rodríguez. Using WordNet for Building WordNets. In *Proceedings of COLING-ACL Workshop "Usage of WordNet in Natural Language Processing Systems*", Montreal, Canada. 1998.
8. Grefenstette, G. *Explorations in Automatic Thesaurus Discovery*. Kluwer, 1994
9. Hindle, Don and Mats Rooth. Structural ambiguity and lexical relations. *Computational Linguistics* 19:103–120, 1993
10. Kilgarriff, Adam. Thesauruses for Natural Language Processing. *Proceedings of NLP-KE 03*, Beijing, China, 5–13, 2003
11. Lázaro Carreter, F. (Ed.) Diccionario Anaya de la Lengua, Vox, 1991
12. Li, Hang and Naoki Abe. Word clustering and disambiguation based on co-ocurrence data. *Proceedings of COLING '98*, 749–755, 1998
13. Lin, Dekang. An information-theoretic measure of similarity. *Proceedings of ICML'98*, 296–304, 1998
14. Manning, Chris and Hinrich Schutze. Chapter 1 of *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA, 1999
15. McLauchlan, Mark. Thesauruses for Prepositional Phrase Attachment. *Proceedings of CoNLL-2004*, Boston, MA, USA, 73–80, 2004
16. Mitchell, Brian. *Prepositional phrase attachment using machine learning algorithms*. Ph.D. thesis, University of Sheffield, 2003
17. Morales-Carrasco, Raúl and Alexander Gelbukh. Evaluation of TnT Tagger for Spanish. *Proc. Fourth Mexican International Conference on Computer Science*, Mexico, 2003
18. Navarro, Borja, Montserrat Civit, M. Antonia Martí, R. Marcos, B. Fernández. Syntactic, semantic and pragmatic annotation in Cast3LB. *Shallow Processing of Large Corpora (SProLaC)*, a Workshop of Corpus Linguistics, Lancaster, UK, 2003
19. Pantel, Patrick and Dekang Lin, An Unsupervised Approach to Prepositional Phrase Attachment using Contextually Similar Words. *Proceedings of Association for Computational Linguistics (ACL-00)*, Hong Kong, 101–108, 2000
20. Ratnaparkhi, Adwait, Jeff Reynar and Salim Roukos. A maximum entropy model for prepositional phrase attachment. *Proceedings of the ARPA Workshop on Human Language Technology*, 250–255, 1994
21. Ratnaparkhi, Adwait. Unsupervised Statistical Models for Prepositional Phrase Attachment. *Proceedings of COLINGACL98*, Montreal, Canada, 1998
22. Resnik, Philip. Selectional preference and sense disambiguation, *ACL SIGLEX Workshop on Tagging Text with Lexical Semantics*, Washington, D. C., USA, 1997
23. Roth, D. Learning to Resolve Natural Language Ambiguities: A Unified Approach. Ini *Proceedings of AAAI-98*, Madison, Wisconsin, 806–813, 1998
24. Stetina, Jiri and Makoto Nagao. Corpus based PP attachment ambiguity resolution with a semantic dictionary. *Proceedings of WVLC '97*, 66–80, 1997
25. Sparck Jones, Karen. *Synonymy and Semantic Classification*. Edinburgh University. Press, 1986
26. Weeds, Julie, *Measures and Applications of Lexical Distributional Similarity*. Julie Weeds, Ph.D. thesis. University of Sussex. 2003
27. Volk, Martin. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. Proceeding of *Corpus Linguistics 2001*, Lancaster, 2001