Dependency Triples vs. Trigrams for PP Disambiguation

Hiram Calvo and Alexander Gelbukh
Center for Computing Research, CIC
National Polytechnic Institute, IPN
Mexico City, Mexico
likufanele@likufanele.com; www.Gelbukh.com

Abstract

The standard problem of PP attachment disambiguation by Ratnaparkhi consists on deciding for a given quadruple (v,n_1,p,n_2) the attachment of p,n_2 to v or to n_2. One of the most successful techniques for PP attachment disambiguation is to use large databases of corpora on which statistical methods are applied to find the most probable attachment, by querying the number of occurrences of (v,p,n_2) and (n_1,p,n_2) and then deciding by the highest normalized number of counts. Usually this is done by counting adjunct trigrams in such corpora. We believe that using dependency triples instead of adjunct trigrams can help to improve the performance of current PP disambiguation methods. In this paper we experiment with dependency triples for PP attachment disambiguation, and then we compare with the standard trigram count.

1. Introduction

The Prepositional Phrase (PP) attachment task is an important part of syntactic analysis particularly when recognizing entities in a text is needed. We illustrate this problem with 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*, so that the entity is only *a cat*, or *a cat with a telescope* (which is rather unusual). Methods based on corpora statistics address the problem by querying the frequency counts of word-triples, such as *see with telescope* and *cat with telescope*. If we try this search in google, we find 24 occurrences for *"see with telescope"*, and 10 occurrences for *"cat with telescope"*, which leads to the correct answer that is to attach *with telescope* to *see*, and not to *cat*. However, looking at the low number of occurrences returned by Google for the previous example, we have the hunch that one of the main drawbacks of this basic form of PP disambiguation approach is coverage. This particular problem has been solved in many different ways. Zhao and Lin [11] propose a nearest-neighbor algorithm for PP disambiguation. Calvo *et al.* [2] compare two different smoothing methods, one of them consisting on using a distributional thesaurus based on Lin [6] and other using WordNet for substituting words that cannot be found directly within the corpus. Volk [10] combines supervised methods with unsupervised methods for PP attachment disambiguation using decision levels and attaining a coverage of 100%.

In this work we will concentrate on unsupervised methods and we will not focus on smoothing techniques (see [2] for a comparison of different smoothing techniques).

When counting adjacent trigrams, there are many combinations that are not taken into account. Consider for example a corpus containing the phrase *I see a cat with a large telescope*. The trigrams extracted for this phrase will be *I see a*, *see a cat*, *cat with a*, *with a large*, *a large telescope*, and we will never find *see with telescope* nor *cat with telescope* from this sentence. One solution for this is to broaden the window for obtaining the trigrams to, say, 4 or 5 words. Balázs *et al.* point out [5] that a narrow window produces little variations, while larger windows introduce a significant amount of noise. Another solution is to use syntactic information. Particularly, using information about dependency syntactic relationships helps to overcome this problem. In this work we compare both solutions.

In dependency approach, words are considered "dependent" from, or modifying, other words [4]. A word modifies another word (governor) in the sentence if it adds details to the latter, while the whole combination inherits the syntactic (and semantic) properties of the governor: *large telescope* is a kind of *telescope* (and not a kind of large); *I see cat* is a kind of (situation of)

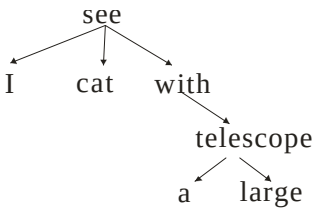
Table 1. Example of Dependency Triples Extracted.

w1	w2	w3	a	b	c	d	e	f	g
ábaco	*por*	*capitel*	1	1	31906	6	1	1	1
ábrete	*de*	*oreja*	1	1	406700	37	1	1	18
ábside	*con*	*arquería*	1	7	32034	15	1	1	5
ábside	*de*	*basílica*	2	7	406700	39	5	2	23
ábside	*de*	*capilla*	1	7	406700	92	5	1	56
ábside	*de*	*iglesia*	1	7	406700	1157	5	1	705
ábside	*de*	*maria*	1	7	406700	38	5	1	27
ábside	*en*	*cabecera*	1	7	136503	53	1	1	16
ácaro	*de*	*agua*	1	8	406700	2445	8	1	1597

Table 2. Example of Trigrams Extracted

w1	w2	w3	a	b	c	d	e	f	g
abad	*y*	*prior*	1	1	7832	2	1	1	1
abad	*y*	*santo*	1	13	7899	323	1	1	16
abaddón	*y*	*exterminador*	1	1	7989	1	1	1	1
abadejo	*alimentar*	*de*	1	4	30	26089	1	1	17
abadejo	*de*	*alaska*	1	1	30747	4	1	1	1
abadejo	*de*	*pacífico*	1	4	30188	24	1	1	14
abadejo	*pertenecer*	*a*	1	4	131	5734	1	1	112
abadejo	*ser*	*pollachius*	1	4	3586	2	1	1	2
abadejo	*y*	*lenteja*	1	1	7992	1	1	1	1
abadejo	*y*	*platija*	1	1	8087	1	1	1	1
abadengo	*destacar*	*localidad*	1	1	193	39	1	1	1

see (and not, say, a kind of *cat*). Such dependency is represented by an arrow from the governor to the governed word:



where the arcs represent the dependency relation between individual words, the words of the lower levels contributing details to those of the upper levels while preserving the syntactic properties of the latter.

A dependency tree like the previous one, yields the following dependency triples: (*I,<,see*), (*see,>cat*), (*see, with, telescope*), (*a, DET, telescope*) and (*large, ADJ, telescope*). Here <, >, *DET* and *ADJ* denote relationships of subject, object, determiner and adjective, respectively.

In this paper we explore the degree of impact of using dependency triples like the ones shown previously, instead of using trigrams. In the following sections we will describe in detail our experiment.

2. Experimental setup

2.1. The dependency parser

To obtain the dependency triples we use the dependency parser DILUCT [3]. This parser uses an ordered set of simple heuristic rules to iteratively determine the dependency relationships between words not yet assigned to a governor. In case of ambiguities of certain

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

Table 4. Results of PP attachment disambiguation

	All prepositions			**Without preposition *de* 'of'**								
Smoothing Method	no smoothing			no smoothing			Method A			Method B		
	P	**R**	**Cov.**	**P**	**R**	**Cov.**	**P**	**R**	**Cov.**	**P**	**R**	**Cov.**
Baseline	73.6	73.6	100	66.1	66.1	100						
Dependency Triples	73.8	16.3	22.1	**77.4**	9.8	12.6	**67.7**	**50.1**	74.0	**67.1**	**52.0**	77.6
Trigrams	75.9	19.3	25.5	68.0	4.1	6.0	56.6	31.7	56.1	50.0	32.9	65.9
Lemmatized Trigrams	**88.7**	11.0	12.4	76.3	10.7	14.1	61.4	47.0	76.6	64.6	51.3	79.5
Lem. Trigrams wnd 4	57.6	20.3	35.2	64.6	15.3	23.6	60.8	48.4	79.7	61.7	50.8	82.3
Lem. Trigrams wnd 5	57.5	**23.5**	**40.7**	63.4	**18.6**	**29.4**	59.0	48.7	**82.6**	60.6	51.1	**84.2**

types, word co-occurrences statistics gathered in an unsupervised manner from a large corpus or from the Web (through querying a search engine) are used to select the most probable variant. No manually prepared tree-bank is used for training. If a complete parse cannot be produced, a partial structure is built with some (if not all) dependency relations identified. Evaluations of this parser show that in spite of its simplicity, the parser's accuracy is superior to the available existing parsers for Spanish.

Through the use of heuristics, as placing an adjective as modifier of a noun, this parser produces dependency triples suitable for PP attachment disambiguation. The result of this work can be used to improve the parser itself, by refining iteratively its module for syntactic disambiguation.

2.2. The corpus

The corpus used for obtaining dependency triples 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. 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.

2.3. The gold standard

Following the PP attachment evaluation method by Ratnaparkhi *et al.* [8], 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[1] [7]. Sample 4-tuples are shown in Table 3.

2.4. The baseline

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. 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

[1] 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)

$$\frac{1}{2500}\sum_{i=1}^{50}\sum_{j=1}^{50}\left|w1_{sim_i}, w2, w3_{sim_j}\right| \cdot sim\left(w1, w1_{sim_i}\right) \cdot sim\left(w3, w3_{sim_j}\right)$$

Figure 1. Formula for smoothing frequency count (Smoothing Method B)

for excluding all 4-tuples with preposition de—no other preposition presents such a high bias. Then 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; see Table 4.

2.5. Dependency triples vs. trigrams

Examples of dependency triples and trigrams are shown in Tables 1 and 2. Note that trigrams include all kinds of words as the second word, while dependency relationships include only prepositions. (a) is the number of occurrences of the triple, |w1,w2,w3|, (b) is the number of occurrences of word 1 with any word 2 and any word 3 |w1,*,*|, (c) is |*,w2,*|, (d) |*,*,w3|, (e) |w1,w2,*|, (f) |w1,*,w3|, (g) |*,w2,w3|.

2.6. Smoothing

Since not all PP attachment cases are found in the triples or trigrams database, we used two smoothing methods based on Lin's similarity. In **Method A**, we substitute w3 beginning with its most similar word, until the count for the triple |w1,w2,$w3_{sim}$| is found. This method was previously described in [2].

Method B is a modification of this latter. Method B consists on finding the 50 x 50 (2500) different combinations for the topmost 50 similar words[2] of w1 and w2 and then we sum its frequency weighted by the similarity of w1 and w2. The formula for Method B is shown in Figure 1.

3. Results

Table 4 shows the comparison of the results obtained for PP attachment disambiguation with the considered methods. It includes the measures of precision P, recall R and coverage Cov.

In Spanish, the preposition *de* 'of' is very often and is simple to disambiguate: it almost always is attached to a noun. So the performance figures of the methods which deal well with this particular preposition (like the baseline heuristic just mentioned: always noun attachment) are unjustly too high. To compensate for this effect, we report separately the results for the prepositions other than *de*.

3. Conclusions

As the experimental results show, coverage almost always is best with lemmatized trigrams. Contrary to expectation, lemmatized trigrams also show the best performance on all prepositions—which in fact means they work very well on the preposition *de* 'of'. However, dependencies almost always show the best precision and recall for the prepositions other than *de*. The highest performance is achieved by Dependency Triples with Method B smoothing (52% recall), but it is closely followed by lemmatized trigrams obtained within a window of 5 words: 51.1% recall, with a lower precision, but helped by a greater coverage (84.2%), which is in fact, the best coverage for all compared methods.

In our future work we will investigate the implications of these findings on the methods of syntactic disambiguation. In particular, we will consider a combined disambiguation technique that uses different approaches for the preposition *de* and other prepositions. Generalizing this idea, we can suggest building a specific disambiguation method for each one of the most used prepositions. In addition, we plan on experimenting with greater windows above 5 to probe if the trend is to produce better results than dependency triples, as well as with taking into account different statistical measures of importance of individual words [9].

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

[2] We have chosen 50 empirically, because lowering values provides lower performance, while higher values do not improve performance significatively but they increase more noticeably processing time.

References

[1] Biblioteca de Consulta Microsoft Encarta 2004, Microsoft Corporation, 1994–2004

[2] Calvo, Hiram, Alexander Gelbukh, A. Kilgarriff, Distributional Thesaurus Versus WordNet: A Comparison of Backoff Techniques for Unsupervised PP Attachment. In *Computational Linguistics and Intelligent Text Processing* (CICLing 2005), Lecture Notes in Computer Science 3406, Springer 2005, pp. 177–188.

[3] Calvo, Hiram, Alexander Gelbukh, DILUCT: An Open-Source Spanish Dependency Parser Based on Rules, Heuristics, and Selectional Preferences. En Natural Language Processing and Information Systems, Lecture Notes in Computer Science 3999, Springer 2006, pp. 164–175. ISSN: 0302-9743.

[4] Chomsky, Noam. 1957. *Syntactic Structures*. The Hague: Mouton & Co.

[5] Kis, Balázs, Begoña Villada, Gosse Bouma, Gábor Ugray, Tamás Bíró, Gábor and Pohl and John Nerbonne. A New Approach to the Corpus-based Statistical Investigation of Hungarian Multi-Word Lexemes. In: *Proceedings of the 4th International Conference on Language Resources and Evaluation* pp. 1677-1681, Lisbon, Portugal, May 26-28, 2004

[6] Lin, Dekang. An information-theoretic measure of similarity. *Proceedings of ICML'98*, 296–304, 1998.

[7] 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

[8] 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

[9] Pérez, D., J. Tecuapacho, H. Jiménez-Salazár, G. Sidorov. A term frequency range for text representation. In: Advances in artifical intelligence and computer science, J. *Research on computing science*, vol. 20, 2006, pp. 113–118.

[10] Volk, Martin. *Combining Unsupervised and Supervised Methods for PP Attachment Disambiguation.* In: *Proc. of COLING-2002.* Taipeh. 2002.

[11] Zhao, Shaojun, Dekang Lin. A Nearest-Neighbor Method for Resolving PP-Attachment Ambiguity. In *Proceedings of the First International Joint Conference on Natural Language Processing* (IJCNLP-04). Lecture Notes in Artificial Intelligence (LNAI), Springer-Verlag, 2004.