Probabilistic PP Attachment: Hindle and Rooth's procedure applied to Portuguese

António Horta Branco
Tiago Castro Henriques
Faculty of Sciences, University of Lisbon

1. Ambiguity and PP Attachment

One of the pervasive problems to be dealt with in Natural Language Processing is the resolution of syntactic ambiguities. More often than not, sentences are associated with several different syntactic trees that are all correct from a constituency point of view. This multiplicity of possible syntactic structures typically increases with the number of lexical items contained in the sentence under analysis.

Interestingly, human speakers tend to be unaware of this multiplicity. They are usually able to rapidly select a single preferred syntactic structure from the many possible candidate structures. A common justification for this capacity is the existence of so called "lexical preferences" imposing that a given syntactic analysis is favored with respect to the other possible analyses when certain lexemes are involved. This means that the simple replacement of a lexeme by another one even with the same category and the same syntactic and semantic selectional properties may change the preferred analysis. While these preferences have not been easily handled by deterministic procedures, the combination of this type of solutions with statistical methods has been explored as a powerful tool to capture the subtleties of lexical preferences and enhance real-time automatic disambiguation.

A notorious type of syntactic ambiguity is the so called attachment ambiguity, in which a given sub-tree may be correctly attached to different nodes of the larger syntactic tree to which it belongs. In this respect, Prepositional Phrase attachment is one of the most studied cases. Let us consider a simple but typical example such as "Astronomers saw stars with ears", which can receive the following two structural analyses, underlying two different interpretations, roughly corresponding to the paraphrases "astronomers saw stars that have ears" and "astronomers using their ears saw stars":

(1) a. Astronomers saw stars with ears.
[[Astronomers]$_{NP}$ [saw [stars [with ears]$_{PP}$]$_{NP}$]$_{VP}$]
b. Astronomers saw stars with ears.
[[Astronomers]$_{NP}$ [saw [stars]$_{NP}$ [with ears]$_{PP}$]$_{VP}$]

***Actas do XVII Encontro Nacional da Associação Portuguesa de Linguística*, Lisboa, APL, 2002, pp. 69-79**

This issue of Prepositional Phrase (PP) attachment can be synthesized as follows: when a PP is "near" a head X, is it immediately dominated by the corresponding phrase XP, or is it dominated by higher nodes dominating XP? It is of note that this is an issue of paramount importance since this kind of constructions occurs very frequently both in verbal and in written speech.

In the present article, we will concentrate on a restricted version of this problem: when a PP follows the head of a Direct Object, is it immediately dominated by this NP or by the corresponding VP? This being said, in order to abbreviate, we will proceed with the following simplified formulation for our question when there be no danger of confusion: given a sentence of the form [... V_{Trans} ... $N_{DirectObject}$ ... P ...], does the preposition attach to the noun or to the verb?

In this connection, our specific goal in the study we are reporting here was to use the procedure for PP attachment resolution proposed by Hindle and Rooth in 1993 for English, and check what results can be obtained when using it for Portuguese.

2. Hindle and Rooth's Procedure

The rationale underlying Hindle and Rooth's procedure is the following. If we have an annotated training corpus in which V/N/P attachments have already been correctly resolved, in order to define our probabilistic model, we can resort to simple counts of the result of the attachment for each specific lexical combination of verb, noun and preposition. However, if we do not have such a corpus, we will be forced to resort to some heuristics allowing to determine the probability of a given preposition attaching to a given verb or to a given noun. The basic idea behind the heuristic proposed by Hindle and Rooth consists in identifying those cases where there can be no ambiguity in PP attachment. After these cases are identified, we can then use simple counts to provide a first estimate for the probability of a given preposition attaching to each verb and for the probability of it attaching to each name.

In order to characterize the model proposed by Hindle and Rooth, the space of events under analysis is to be defined. This event space consists of all sentences containing a transitive verb followed by an NP, which in turn is followed by a PP. In order to limit the complexity of the analysis and render the problem more manageable, two simplifications are adopted. Whenever more than one PP follows the Direct Object:

1. The possibility of there being any interaction between different prepositions is ignored.
2. If the same preposition occurs in more than one PP, we only model the behavior of the first occurrence and ignore subsequent PP's containing the same preposition.

Two random variables are defined, corresponding to the following questions:

1. VA_p – Is there a PP headed by p following verb v, which is dominated by that verb?
   Yes: $VA_p=1$
   No: $VA_p=0$
2. NA_p – Is there a PP headed by p following noun n, which is dominated by that noun?
   Yes: $NA_p=1$
   No: $NA_p=0$

The probability of a specific pair of values for these two random variables is then:

$$P(VA_p, NA_p \mid v,n) = P(VA_p \mid v,n)P(NA_p \mid v,n)$$
$$= P(VA_p \mid v)P(NA_p \mid n)$$

In deriving the above expression, the two random variables were assumed to be independent. At first glance, one might consider that they are not independent, and that if e.g. NA_p were true, VA_p would necessarily be false, as a preposition can only attach to one node. That would be the case if there were only one PP following the transitive verb. However, more than one PP may occur after the transitive verb. Let us then consider the case were we have two PP's following the verb. If the first preposition attaches to the noun, the second preposition can attach either to the verb or the noun:

$$P(Attach(p) = n \mid v, n) = P(NA_p = 1 \mid n) \times P(VA_p = 0 \vee VA_p = 1 \mid v)$$
$$= P(NA_p = 1 \mid n)$$

However, if the first preposition attaches to the verb, the other preposition can only attach to the verb, otherwise we would get crossing lines in the phrase structure tree. We then get:

$$P(Attach(p) = v \mid v, n) = P(NA_p = 0, VA_p = 1 \mid v, n)$$
$$= P(NA_p = 0 \mid n)P(VA_p = 1 \mid v)$$

As desired, one is thus able to define the probability of a preposition attaching to the verb or to the noun for a specific combination of verb, noun and preposition in terms of the probability of the preposition attaching to the verb independently of the noun, and the probability of it attaching to the noun independently of the verb.

The Hindle and Rooth's procedure consists then merely in comparing a pair of probabilities and deciding which one is higher than the other, which is equivalent to deciding whether their ratio is greater or smaller than one. Using logarithms, lambda is defined as:

$$\lambda(v, n, p) = \log_2 \frac{P(Attach(p) = v \mid v, n)}{P(Attach(p) = n \mid v, n)}$$
$$= \log_2 \frac{P(NA_p = 0 \mid n) P(VA_p = 1 \mid v)}{P(NA_p = 1 \mid n)}$$

If lambda is higher than zero, the probability of the preposition attaching to the verb is higher than the probability of it attaching to the noun. When analyzing a sentence, we only have thus to calculate lambda and resolve the ambiguity by attaching the PP to the verb if lambda is higher than zero, or attaching it to the noun otherwise.

In order to calculate lambda, one needs to be able to calculate the probabilities conditioned by the verb or by the noun. The escape hatch here in the absence of an unresolved corpus, as mentioned above, consists in identifying the unambiguous cases and counting them (cf. the next section below). Using *C(v,p)* and *C(n,p)* to denote the number of times *p* attaches unambiguously to *v* and to *n*, respectively, we have

$$P(VA_{p_1} = 1 \mid v) = \frac{C(v, p)}{C(v)}$$

$$P(NA_{p_1} = 1 \mid n) = \frac{C(n, p)}{C(n)} = 1 - P(NA_{p_1} = 0 \mid n)$$

$$\lambda(v, n, p) = \log_2 \frac{P(NA_p = 0 \mid n) P(VA_p = 1 \mid v)}{P(NA_p = 1 \mid n)}$$

$$\lambda(v, n, p) = \log_2 \frac{\left(1 - \frac{C(n, p)}{C(n)}\right) \frac{C(v, p)}{C(v)}}{\frac{C(n, p)}{C(n)}}$$

$$\lambda(v, n, p) = \log_2 \frac{C(n) - C(n, p)}{C(v)} \frac{C(v, p)}{C(n, p)}$$

To exemplify how the method is used, we provide an example from Hindle and Rooth, who used the counts from their corpus to resolve the correct attachment in the sentence "Moscow send more than 100,000 soldiers into Afghanistan":

$$P(VA_{p1} = 1 \mid send) = \frac{C(send, into)}{C(sent)} \approx \frac{86}{1742.5} \approx 0.049$$

$$P(NA_{p1} = 1 \mid soldiers) = \frac{C(soldiers, into)}{C(soldiers)} \approx \frac{1}{1478} \approx 0.0007$$

$$P(NA_{p1} = 0 \mid soldiers) \approx 1 - 0.0007 = 0.9993$$

$$\lambda(send, soldiers, into) = \log_2 \frac{0.049 \times 0.9993}{0.0007} \approx 6.13$$

We are then led to predict that the correct attachment of the PP headed by *into* is to the verb *sent*, which is in fact the correct attachment.

3. The Heuristic for Unresolved Corpora

In order to apply Hindle and Rooth's procedure with a corpus with unresolved PP attachments, we need a way to build our model, i.e. a way of calculating *C(v,p)* and *C(n,p)*. Hindle and Rooth proposed the following heuristic for constructing such a model:

- Identify the unambiguous cases by using the following rules

1. Whenever a PP follows an NP occurring in a sentence-initial position, this PP must attach to the NP since there is no preceding verb.

2. Whenever a transitive verb is followed by an accusative pronoun (e.g. *him*), which in turn is followed by a PP, this PP attaches to the verb, since it can hardly attach to the pronoun.

- Using this initial model, calculate lambda for all ambiguous cases. If the absolute value of lambda exceeds a threshold, resolve the attachment according to lambda and increase the appropriate count (*C(v,p)* or *C(n,p)*).

- Divide the remaining ambiguous cases evenly between both counts, i.e. increase both *C(v,p)* and *C(n,p)* by 0.5.

The rules we adopted for Portuguese are similar to these proposed for English. In Portuguese, the accusative clitic pronoun, if in proclisis, occurs immediately after the verb, preceded by a hyphen. The counts in steps 1 and 2 of the heuristic above are then the following:

- $C(n,p)$ – count occurrences of pattern "[... N P ...]$_{SN}$ ..."
- $C(v,p)$ – count occurrences of pattern "... v_{trans} – clitic P ..."

4. Choosing a Corpus

Given there are no annotated corpora available for the Portuguese language, we had to turn to raw text corpora. Of those freely available, the CetemPúblico[1] corpus is at present the largest one. This corpus consists of the two first paragraphs of many articles published between 1991 and 1998 in the daily newspaper "Público". This corpus is available on CD-ROM and contains 478 Mbytes of compressed text, with a compression ratio of 1 to 3. The uncompressed extracts contain around 30 million sentences corresponding to 180 million words. The text uses a simplified markup language derived from SGML to convey information about the extracts (e.g. which section they were taken from, which year, etc.). Since we were not interested in this information, we used the UNIX tool sgrep to extract the plain text from the corpus and compile files containing one sentence per line.

The CetemPúblico corpus includes many lines that contain spurious characters that serve no visible purpose. These characters include escape characters and non-printable characters, as well as many non-alphabetic characters (excluding punctuation). We excluded all such lines from our analysis. To render our task more simple, we also left out all lines containing digits and any of the punctuation characters ", ', (), --, :, and ;.

Additionally, we excluded lines that did not terminate in a period, an exclamation mark or an interrogation mark. We were left with lines that only contained alphabetic characters and accented characters. The only punctuation signs allowed were the full stop or period, the comma and the exclamation and interrogation marks. Finally, we surrounded every punctuation sign with spaces, to simplify parsing.

In order to filter out unwanted lines and perform the necessary transformations on the text, we resorted to the common arsenal of UNIX text-processing tools, which includes programs such as grep, sed, awk, tr, paste, uniq, sort and cut.

The filtered corpus that we finally obtained contained nearly 3 million sentences and 72 million words.

5. Tagging the Corpus

In order to apply Hindle and Rooth's procedure to the CetemPúblico corpus, we first had to devise a way of tagging this corpus. Tagging a corpus inevitably requires that one possesses some kind of dictionary that associates each word with its morpho-syntactic category or categories. To avoid re-inventing the wheel, we

[1] Availability details can be found in: http://cgi.portugues.mct.pt/cetempublico/whatisCETEMP.html

looked for existing freely available dictionaries. Fortunately, the Natura Project[2] has made one such dictionary available, as part of their generic morphological analyzer Jspell. The Jspell dictionary contains a base dictionary (spread out among several *.dic files), which contains all closed classes, verb infinitival forms, irregular verbal forms and base words, together with their category and an optional set of morphological flags. These morphological flags point towrad productive rules, which are used to generate additional words. The productive rules are specified in a file named port.aff, which uses regular expressions to specify how words are changed in order to accommodate the various prefixes and suffixes. These rules include all suffixes that produce the various forms of regular verbs.

We used the morphological tags corresponding to verb inflexion and their associated rules contained in the file port.aff to construct a generator of verbal forms. This generator, written in the Python programming language[3], was applied to all verbs in the dictionary except intransitive ones to produce all verb forms (gerunds, past participles and the 2nd person plural were left out). We produced a file containing a total of 234,800 verb tokens, corresponding to 3,692 verb types. Each line of the file contained the inflected verb form followed by its base form, in order to allow us to perform counts based on the base form.

The next step consisted in constructing a program capable of deriving every possible noun form from the base nouns contained in the dictionary. Since the set of morphological flags and corresponding rules used by the dictionary is very large and complex, we limited ourselves to the morphological flags used to produce plural and feminine forms. We obtained a total of 24,104 noun forms, corresponding to 12,438 base nouns, which were also kept in a file containing one noun form plus its base form per line. We used an additional procedure to help us identify nouns: since the corpus contains written text, it contains many words with an initial capital letter (we ignored words at the beginning of sentences). This distinction, which is not present in spoken language, conveys useful information about the syntactic category of some words: It is a labeling of proper nouns performed automatically by human writers. If we look at words occurring in written text that start with a capital letter, almost invariably they correspond to proper nouns. We added every word in the corpus starting with a capital letter to our set of nouns, after converting every initial word of every sentence to lowercase, to avoid adding false proper nouns.

We further limited our analysis by looking only at the 5 most frequent prepositions: *de*, *em*, *para*, *com* e *por*. The next most frequent preposition, *sobre*, occurs only once for every thirty occurrences of the most frequent preposition and once for every 3.5 occurrences of the fifth most frequent preposition. However, an additional operation had to be performed. Prepositions in Portuguese often occur in

[2] http://shiva.di.uminho.pt/~jj/pln/pln.html

[3] http://www.python.org

contracted form. The contracted form joins the preposition and the following word (usually an article or a demonstrative) into one single word. We had to pre-process the entire corpus and replace the contracted forms with the corresponding non--contracted words.

We then tagged the corpus, using the set of verb forms and the set of noun forms that we created, together with the set of unambiguous words belonging to the closed classes. Tagging the corpus consisted in replacing each verb form with its infinitive and every noun form with its base noun, and then appending a colon to the word, followed by the corresponding category. For instance, the sentence:

> George Bush dá palmada nas costas ao seu enviado, Jim Woolsey.

became, after the initial transformations and tagging,

> George Bush:pn dar:v palmada:n em:prep as costa:n a o seu enviado, Jim:pn Woolsey:pn .

Once we obtained this annotated corpus, we were able to apply the Hindle and Rooth's procedure.

6. Applying Hindle and Rooth's Heuristic

Since the annotated corpus only contains base forms, we were able to perform the $C(v)$ and $C(n)$ counts directly, using the text2wfreq tool from the CMU toolkit.[4] To perform the $C(n,p)$ counts, we first searched the corpus for sentences beginning with a noun (optionally preceded by an article, a determiner or a possessive) and followed by a preposition (thus ignoring NP's where some material intervenes between its head noun and the PP). We found 335,300 sentences exhibiting this pattern. We reduced every such sentence to a line of the form: basenoun: n_preposition:p (for instance, conversa:n_com:p). We then used text2wfreq to obtain the number of occurrences of each (n,p) pair, and obtained a total of 29,000 pairs. To perform the $C(v,p)$ counts, we searched the corpus for sentences containing a verb which was followed by a hyphen and a clitic pronoun, which in turn was followed by a preposition. We found 4,029 instances of this pattern. All sentences containing the pattern were reduced to the form verb:v_preposition:p, and text2wfreq was used to obtain the number of occurrences of each (v,p) pair, giving us a total of 1,291 pairs.

[4] Cambridge Statistical Language Modeling Toolkit, a set of command-line UNIX tools written in C: http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html.

7. Building an Attachment Resolver

The next step consisted in constructing an Attachement Resolver, a program able to detect sentences with PP attachment ambiguities and use the probabilistic model constructed from the corpus to resolve these ambiguities. Our Resolver is very simple: it examines every sentence, looking for the pattern vnp — more exactly the pattern word* word:v word* word:n word* word:p. It looks only at the categorial tags that follow the colon, and searches for the sequence of tags vnp, ignoring every word in between these tags. The program performs a "greedy" search, where the first sub-sequence of tags matching the pattern is used. Of course, this means that many sentences will be false positives, i.e. false cases of PP attachment ambiguity since we are deliberately ignoring everything that occurs between these three tags we are searching for. In the present case this does not really poses a problem since we will later eliminate all false positives manually, when evaluating the resolved text produced by the Resolver. The important thing is that we let no true positive escape our analysis. Once a suitable sequence is found, the words that precede the tags are identified, and lambda is calculated using the expression:

$$\lambda(v, n, p) = \log_2 \frac{C(n) - C(n, p)}{C(v)} \frac{C(v, p)}{C(n, p)}$$

If the absolute value of lambda is greater than the threshold value, we use it to decide where the PP attaches: to the verb if lambda is positive, to the noun if negative. The Resolver then outputs a line with its verdict followed by the value of lambda, followed by another line containing the entire matching sentence. The words corresponding to the verb, noun and preposition responsible for the match are capitalized to facilitate the manual verification of the results.

If our Resolver is not able to calculate lambda or if lambda is below the threshold, it outputs a line saying "# sentenced not processed" followed by the entire sentence in another line. The following lines are examples of the output produced by the Resolver:

> # p attaches to noun – lambda ~= -10
> George Bush:pn **DAR:V PALMADA:N EM:PREP** as costa:n a o seu enviado, Jim:pn Woolsey:pn .
>
> # p attaches to noun – lambda ~= -8
> no que a o Zimbabwe:pn, este estudioso de:prep as coisa:n de:prep a África:pn Austral:pn declarar:v a o PÚBLICO:pn que as recentes eleição:n **DEMONSTRAR:V FALTA:N DE:PREP** cultura:n democrática e de:prep uma oposição:n forte:n .

sentence not processed
as jovem:an, obrigadas a prostituir:v – se, **FACTURAR:V CENTENA:N DE:PREP** conto:n mensais, mas quase nada receber:v.

p attaches to verb – lambda ~= 2
ao procurar:v deter De:pn Klerk:pn, os partidário:an de:prep Treurnicht:pn tentar:v afastar os negro:an de:prep algumas com:prep recurso:n a, **COLOCAR:V BOMBA:N EM:PREP** uma escola:n que iria servir para:prep alojar órfão:an negro:an e poder:v ter contribuído, sob:prep o de:prep as Forças:pn de:prep Segurança:pn, para:prep a os diferentes grupo:n de:prep negro:an .

8. Results

We processed the entire corpus with the Resolver described above, using a value of 2 as threshold. We followed a simplified version of Hindle and Rooth's heuristic, using the initial model to resolve ambiguities immediately. The cases where lambda is below threshold were not split between counts as proposed by the original Hindle and Rooth's heuristic for unresolved corpora.

As we discussed previously, our "vnp" sieve uses a simple approach that makes false matches inevitable. It is therefore necessary to manually examine the result of the procedure to eliminate all false positives. After the false positives were removed, we examined each sentence and its proposed resolution, comparing it with a human verdict.

We compiled three counts: I) the number *A* of valid sentences; II) the number *B* of sentences where the procedure chose an attachment; III) the number *C* of sentences where the procedure chose the correct attachment. Using these three measures, we were able to calculate two quantitative measures for our resolver: precision *Pr* and recall *Re*. Precision is the ratio between the number of correct guesses and the total number of guesses. Recall, a related quantity, is defined as the ratio between the number of correct guesses and the number of sentences that we would like to guess. Using the counts we defined above, we then have:

$Pr=C/B$
$Re=C/A$
Since $A>B>C$, it follows that $Pr>Re$

From the processed corpus, 225 sentences taken randomly were manually verified. The counts thus obtained indicate a Precision of 90% and a Recall of 55% for the threshold of 2. These results are strikingly similar to those obtained by Hindle and Rooth, who obtained a precision of 92% and a recall of 55% for a threshold of 3.

9. Future Developments

The rationale behind this study was to experimentally explore the application of the Hindle and Rooth's procedure to Portuguese. The main objective was not to produce a detailed analysis of this procedure when applied to Portuguese, but rather to find out whether a coarse grained approach could produce useful results. The preliminary measures obtained show that this was indeed the case. This now stimulates us to further improve the sophistication of our study.

There are several avenues that can be pursued to improve our study:

1. **Better Tagging:** Most likely, the most important improvement consists in either obtaining a fully and correctly tagged corpus or creating a more sophisticated automatic tagger. Our tagging device used a limited dictionary (the Jspell dictionary) that was not even fully explored (e.g. not all productive morphological rules were used), meaning that many words in our corpus were left untagged. The use of a more complete dictionary may also improve the quality of our results. For the sake of simplicity, we did not label all categories, opting for labeling only most of the closed classes and all verbs and nouns. Adjectives and adverbs were left out. Also, we did not provide a mechanism for resolving lexical ambiguity and we were therefore forced to ignore all "vnp" matches where the verb or the noun were categorially ambiguous.

2. **Tuning the Threshold :** It would certainly be important to know how the variation of the threshold might influence the results. Often, no guess is delivered for a given sentence simply because the value obtained for lambda is below the threshold. This has a direct impact on the value of Recall.

3. **Data Sparseness:** Another factor that may have a serious impact on Recall is the problem of data sparseness. For many sentences, no guess is produced simply because no counting is available in the corpus for the specific (v,p) or (n,p) pair at stake. One way of handling this problem is to introduce statistical estimation techniques. An important step in that direction would be also to use the full Hindle and Rooth's heuristic for non annotated corpora as its application might help to increase counts for rare combinations.

References

Hindle, Donald and Mats Rooth. 1993. Structural Ambiguity and Lexical Relations. *Computational Linguistics* **19**:103-120.