Constrained Lexical Attraction Models

Radu Ion and Verginica Barbu Mititelu

Romanian Academy Research Institute for Artificial Intelligence
13 Septembrie, 13, Bucharest 5, 050711, Romania
radu@racai.ro, vergi@racai.ro

Abstract

Lexical Attraction Models (LAMs) were first introduced by Deniz Yuret in (Yuret 1998) to exemplify how an algorithm can learn word dependencies from raw text. His general thesis is that lexical attraction is the likelihood of a syntactic relation. However, the lexical attraction acquisition algorithm from (Yuret 1998) does not take into account the morpho-syntactical information provided by a part-of-speech (POS) tagger and, thus, is unable to impose certain linguistically motivated restrictions on the creation of the links. Furthermore, it does not behave well when encountering unknown words. The present article presents a new link discovery algorithm using the annotation provided by a POS-tagger. The results show an F-measure of approximately 70% when comparing the links produced by this algorithm with those produced by a fully-fledged parser.

Introduction

Lexical Attraction Models (LAMs) were first introduced by Deniz Yuret in (Yuret 1998) to exemplify how an algorithm can learn word dependencies from raw text. The 'syntactic structure' of a sentence is given by a dependency structure which is an undirected, connected and planar graph with no cycles. In what follows, we will refer to this structure using the term 'linkage' (not a proper syntactic structure as defined by (Mel'čuk 1988) because the links are not oriented and they have no names; what has been retained from Mel'čuk's definition is the planarity condition which states that two links are not allowed to intersect except at a word position in a sentence).

Yuret's general thesis is that lexical attraction is the likelihood of a syntactic relation. Therefore, the highest probability assigned by the model to a sentence is to be obtained by the syntactically correct linkage. He proves the result by formalizing the dependency structure of a sentence as a Markov network (or Markov random field; for a description see (Kindermann and Snell 1950)). A node of the network is a word of the sentence and the potential function is the pointwise mutual information (MI) of a link.

There are other papers that describe linkages. The first to be mentioned is the link grammar (LG) of (Sleator and Temperley 1991) which produces a named linkage of a sentence. LG makes use of POS information and of linkage restrictions to ensure correct linking. Every word in the parser's dictionary has a set of restrictions telling the parser in which kind of links the word may participate and where to look for the pair: in the left-hand side context of the word or in its right-hand side context. For instance the transitive verb '*chased*' requires a subject that is usually located in front of it (restriction `S-`) and a direct object that is placed after it (restriction `O+`). Thus, the dictionary-based restrictions of '*chased*' are as follows: `S-&O+`. The parser generates the linkage of the sentence by resolving the restrictions among the words of the sentence.

Another example of sentence linkage is provided by the grammatical bigrams of (Paskin 2001). His model introduces a linkage similar to that of Yuret but with a significant difference: the links are oriented. The probabilistic model of this oriented linkage assumes that the dependent along with its position (left/right) relative to its direct governor is probabilistically dependent only on its direct governor. Consequently, the posterior probability of a sentence given a parse for it can be decomposed in the product of the link probabilities. (Paskin 2001) also gives an estimation of the model's ability to induce oriented linkages. Model's link precision is 39.7% which is below the 61% of Yuret's experiments. Yet, this model also takes into account the link orientation, which may be an important differentiating factor.

This article presents an algorithm that produces a linkage with the same properties as that of (Yuret 1998). It will be generated using the same basic idea: intermixing learning with parsing. The novelty of our approach resides in the use of with two additional devices:

1. linking rules that allow only for certain links, while rejecting others;
2. use of POS information to increase the score of a weak link.

Two experiments will be considered: the English setting in which we will compare the output of our linker with the output of a recognized dependency parser for English (Tapanainen and Järvinen 1997), and the Romanian setting for which the gold standard annotation was performed by the authors, but on a much smaller test file.

Text Preprocessing

The link discovery algorithm proposed by (Yuret 1998) does not require any preprocessing of the corpus. Yuret used very large corpora (100 million words) in order to train his linker and to be sure that pairs gain sufficient statistical information. The algorithm proposed here does not beneficiate of very large corpora to train on and because of that, it requires the text to be lemmatized. In order to obtain the lemmatization, one needs to also POS-tag the input text.

We used the TnT POS-tagger developed by (Brants 2000). The Romanian tagger was trained on a 1 million word corpus comprising the *1984* novel of George Orwell and a collection of newspaper materials (we will refer to this corpus by the name ROC). The POS annotation of this corpus was manually checked to ensure correct trigram sequences. The tagger's dictionary was augmented with a 500 thousand Romanian word form lexicon that contains for every word form its correct POS-tag. The English tagger was trained on the English translation of ROC (we call it ENC).

The lemmatization was performed with in-house developed Romanian and English lemmatizers. They rely on dictionary lookup for finding lemmas and if the word form isn't found there, they use automatically acquired lemma finding rules to generate a list of candidate lemmas and then fourgram Markov models (trained on lemmas from dictionary) to rank the candidates. The best-ranked candidate wins and becomes the sought lemma for the word form.

The Link Discovery Algorithm

The algorithm's general architecture is depicted in Figure 1 (taken from (Yuret 1998)):

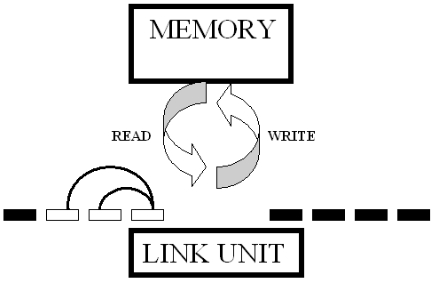

Figure 1. The general architecture of the linker

The link unit (or processor in Yuret's terminology) is responsible for reading the tokens of the input sentence and for assigning links to pairs of tokens. The memory stores frequency information for words and pairs of words.

Yuret's linker begins reading the tokens of the sentence from left to right, updates memory with the frequency information of the pairs that it sees (using different update procedures) and on the basis of the frequency information provided by the memory, it constructs links ensuring that the planarity of the resulted graph is preserved. That is, if two links cross, the weakest one (as given by the pointwise mutual information of the pair) is removed.

The two controlling factors of the accuracy of the linker are the order in which tokens are read and the memory update procedure (which pairs are to be recorded in memory). It was pointed out by (Yuret 1998) that a high percentage of the syntactic relations are between adjacent words. This statement leads naturally to the idea that the parse tree should be constructed bottom-up: first find the most probable links that are between adjacent words then find the most probable links that are between adjacent groups of linked words and so on until a full parse tree is obtained. In what follows we will give a more precise description of this algorithm along with the memory update procedure.

Given a sentence *S* with *n* words (*S* is an array of *n* word forms), we will understand by *s*(*i*), *p*(i) and *l*(i) the word form at position *i* in the sentence, its POS-tag and its lemma, respectively. At any given moment, the array *G* holds the groups of adjacent words of *S* that are bound by complete linkages (*g*(*i*) gives the group at index *i* in *G*). The algorithm stops when *G* has the size equal to 1 (has the complete linkage of S) or when its size does not decrease anymore (because all new links are forbidden by the linking rules; in this case, an incomplete parse tree is obtained). The algorithm is:

```
G = S;
while ( sizeof(G) > 1 ) {
   Gnew = the empty array;
   foreach consecutive a,b,c < sizeof(G) {
      lnkab = best link between g(a),g(b);
      lnkbc = best link between g(b),g(c);
      if ( lnkab > lnkbc ) {
         add( Gnew, makegroup( g(a), g(b) ) );
      }
      else {
         add( Gnew, g(a) );
         add( Gnew, makegroup( g(b), g(c) ) );
      }
   }
   exit loop if ( sizeof(G) == sizeof(Gnew) );
   G = Gnew;
}
```

The function `sizeof()` gives the length of an array. At line 5, `lnkab` is constructed between `g(a)` and `g(b)` with the constraints that it does not cross any of the links in `g(a)` and `g(b)` and that it is allowed by the rule filter. If `g(a)` and `g(b)` can be linked by more than one link, the best one (highest MI score) is chosen. The function `makegroup()` constructs a new group from its arguments given the winning link between them. At line 8 above, a new group is constructed from `g(a)`, `g(b)` and `lnkab` and this group is guaranteed to have the properties of a linkage. This group consists of a sequence of adjacent words from S that are completely linked. The comparison at line 7 is based on the MI score of the links. At line 15 if the newly constructed array of groups `Gnew` is the same as the old one `G`, the algorithm stops. Finally, at line 8 it should be pointed out that the next three consecutive indices read begin with the old `c`.

The memory updating happens at lines 5 and 6. The memory is updated with every possible link $\{s(i), s(j)\}$ that can connect two groups (the link does not violate the planarity restriction and it is not rejected by the rule filter). What is actually recorded is frequency information for $\{l(i), l(j)\}$, $\{p(i), p(j)\}$, $l(i)$, $l(j)$, $p(i)$, $p(j)$ and the score of the link is computed as the sum

$$\mathrm{MI}(\{l(i), l(j)\}) + \mathrm{MI}(\{p(i), p(j)\})$$

This way, if $\{l(i), l(j)\}$ has not been seen, $\{p(i), p(j)\}$ has certainly been seen and the link will not have a 0 score. This is the way our algorithm generalizes and copes with unknown words.

We called this class of LAMs *constrained LAMs* because of the rule filter module that reduces significantly the number of pairs recorded by the memory and increases the likeliness that they can form syntactically correct links. The rule filter is a module that reads a rule file and accepts links if they pass all of rejection clauses, if an `allow` clause is encountered or if that link is stipulated. A rule may be of the following types:

```
deny from:POS to:POS [if condition]
enforce from:POS to:POS [if condition]
allow from:POS to:POS [if condition]
link from:POS to:POS [if condition]
```

The condition is optional and if it holds true the rule is applied. If not, no decision is made except for `enforce` which denies a link if its left side matches but its right side does not. The default policy is to accept the link (that is if no rule applied). For instance we may have the following rule that enforces agreement between a determiner and its noun:

```
deny from:det.sg to:noun.pl
```

which says that a link from a singular determiner cannot end in a plural noun. That is, "this cats" is not grammatically correct.

Let us exemplify the algorithm by running it step by step on the following noun phrase: *the/the/det big/big/adje brown/brown/adje cat/cat/noun* with the format wordform/lemma/POS-tag. Also, let's assume that the correct links have the best MI score when read from memory and that we have the following rules:

```
deny from:adje to:adje
deny from:det  to:adje
```

At the beginning there is no link and the G array contains four groups:

[the/det] [big/adje] [brown/adje] [cat/noun]

The algorithm tries to link the groups *the/det*, *big/adje* and *big/adje*, *brown/adje*. Since the links *det-adje* and *adje-adje* are forbidden by deny clauses, the algorithm shifts and tries to link *big/adje*, *brown/adje* and *brown/adje*, *cat/noun*. It will link the latter group since the former is forbidden. It also updates the memory with the pair *brown/adje*, *cat/noun*. We now have the following G array consisting of three groups:

[the/det] [big/adje] [brown/adje cat/noun]

The only link that is allowed here is *big/adje*, *cat/noun* and the memory is again updated. The G array becomes:

[the/det] [big/adje brown/adje cat/noun]

Finally, the link *the/det*, *cat/noun* is added as the only one possible, the memory is updated with the new pair and the algorithm finishes with the final parse for the input phrase:

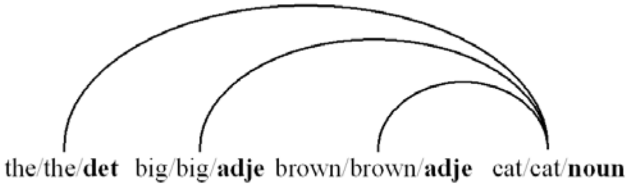
the/the/**det** big/big/**adje** brown/brown/**adje** cat/cat/**noun**

Figure 2. The final parse tree for the input phrase

It is obvious that with a comprehensive collection of rules, the linker has greater chances to detect the correct links (only). In the following section we will give an analysis of the linker performance for Romanian with suggestions aimed at rule improving and addition.

Output Analysis for Rule Improving

Analyzing the results of the linker we found that mistakes do occur, so there is a need for rule inspection and evaluation. We considered that it would be useful to identify the mistakes and create further rules whose restrictions will improve the results of a parser in a further version.

In this section we exemplify some of the errors identified, when possible we give an explanation for their occurrence, and we also suggest some rules aiming at making the linker perform better. Such rules are created through linguistic introspection of native speakers with appropriate training (linguists).

Consider the following example illustrated in Figure 3:

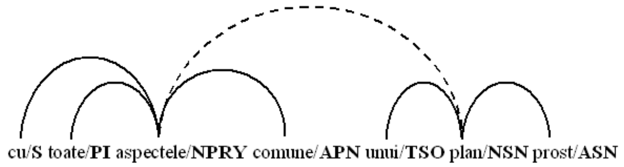
cu/S toate/PI aspectele/NPRY comune/APN unui/TSO plan/NSN prost/ASN

Figure 3. Oblique nouns with adjective as heads

This entire string is separated by commas from the rest of the sentence (*E mură-n gură, cu toate aspectele comune unui plan prost, dar care ia în calcul practici familiare doar în poveştile cu spioni ale românilor*.). This remark is worth being done, as it explains why dependency relations are to be searched for only within the limits of this string. The noun phrase *unui plan prost* has an oblique case (either genitive or dative), visible only in the form of the indefinite article (TSO). The linker analyzes this (see the dotted line) as being dependent on the definite N *aspectele* (thus the NP being in the genitive case). However, such a relation is not possible: Romanian genitive is licensed by a definite noun only when it immediately follows it (*pălăria fetei* "the girl's hat"). If there is any intervening word between the definite noun and the N in genitive, then a possessive article (TP) is necessary to license the genitive (*pălăria neagră a fetei* "the girl's black hat", where *a* is a possessive article). As such an article is not present in the structure, we can assume that the only possible governor of the noun phrase *unui plan prost* is the adjective *comune* (it is common knowledge that adjectives may have predicates, so they may take arguments). So, the implementation of a rule postulating that in case a structure as the following

N1+ A1+ TS+N2 (+A2)

is encountered (where parentheses are indicators of the optionality of the element they contain), then link the latter noun phrase (i.e. T+N2 (+A2)) to A1, instead of to N1. In case the structure is

N1+ A1+ TP+N2 (+A2)

then link the latter noun phrase (i.e. T+N2 (+A2)) to N1.

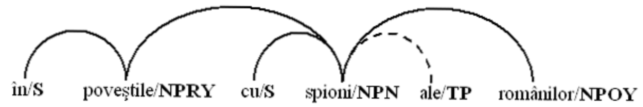

Figure 4. Oblique nouns with nouns as heads

Another example that involves attachment of oblique noun phrases is presented in Figure 4. A TP cannot be attached to a previous noun in the sentence. TPs are licensers of oblique nouns (more precisely of genitives), so they are heads of the nouns immediately following them, although the occurrence of some modifying adjectives between the TP and the noun is also possible, although more rarely and characteristic of stylistically marked discourse[1]. However, this example is also suggestive of the fact that attachment of TP phrases (as one can call them) has to consider some morphological aspects as well: the TP agrees in gender and number with the noun it refers to. *Ale* is a feminine plural form, *spioni* is a masculine plural form, and *poveştile* is a feminine plural form. So, the phrase *ale românilor* should be attached to *poveştile*, not to *spioni*. Gender information is, unfortunately, not present in the tagset we are working with. So, a correct attachment of such genitive phrases headed by a TP needs a tagset in which such grammatical information (i.e. gender) is present.

Regarding the attachment of adjectives let us consider the example below (taken from the larger context *Aruncând inepția spre Cotroceni, dar cu zgomot, prin piața publică, Ion_Iliescu riscă să obțină exact contrariul*):

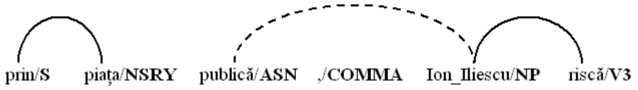
prin/S piața/NSRY publică/ASN ,/COMMA Ion_Iliescu/NP riscă/V3

Figure 5. Linking adjectives

In such cases, the adjective depends on the N it follows, not on the N coming after it, especially that there is a comma between the A and the N following it. This problem is linked with the tagset problem. If the linker could know the nouns' gender, then agreement (gender, number, case) may be helpful in establishing the right head of adjectives. However, our claim is that considering punctuation is also important. The commentary of example in Figure 3 is also relevant in this respect. So, a further study on the way punctuation is helpful for improving the linker's results is, in our opinion, worth being made.

Adjectives may also raise the following situation (the larger context is the following: *Vadim a afirmat vineri că România ar fi instruit nişte terorişti palestinieni şi l-a sfătuit pe preşedintele Ion_Iliescu să facă mai puțină agitație cu presupusa luptă împotriva terorismului.*):

[1] The unmarked word order in Romanian is noun followed by adjective. Only in some exceptional cases the unmarked order prefers the adjective before the noun. These cases can be dealt with via a lexical rule.

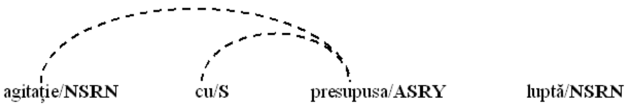

Figure 6. Definite adjectives

A definite adjective cannot attach to a noun preceding it. When the noun and the adjective appear in unmarked word order (see footnote 1), then the noun is the carrier of the definite article (in case it exists). However, in marked word order (i.e. the adjective precedes the noun it modifies) the adjective bears the definiteness marker (in case it exists). With that in mind, a correct linkage would be from the adjective *presupusa* (with *-a* the definiteness marker) to the noun *luptă*. In this case, we can also observe that after making the wrong link between *agitație* and *presupusa*, the linker is then forced to link (again, wrongly) the preposition *cu* with the adjective in order to satisfy the planarity constraint.

Another problem concerns the Romanian demonstratives. Consider the example in Figure 7 (the larger context is: *Nu e şi acesta un fel de terorism la adresa structurilor statului de drept…*):

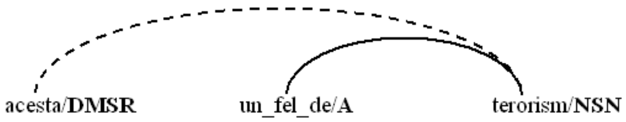

Figure 7. Demonstratives' attachment

Romanian has two sets of demonstratives (i.e. they have different forms): one set appears before nouns and the other after the noun. The former set of forms function only as (prenominal) determiners, while the latter may function as either (postnominal) determiners or as pronouns. Once this information available in the resources of a linker, the form *acesta* from Figure 7 would not be linked to the following noun, *terorism*, as it belongs to the second set of Romanian demonstratives.

Attachment of the negative marker *nu* also seems to raise problems. The literature devoted to the study of negation distinguishes between sentential negation and constituent negation[2]. Studies on sentential negation in Romanian have shown that the negative marker has a fixed position within the verbal complex (see Barbu 2004). According to these studies, the negative marker never follows the verb it negates. So, if a string as the following:

să/QS se/PXA poarte/V3 nu/QZ neapărat/R în/S buna/ASRY tradiție/NSRN a/TS mândriei/NSOY

the negative marker (having the tag QZ here) should not be linked to the verb *poarte*, but should be interpreted as expressing constituent negation and, thus, be linked with the phrase immediately following it.

A problem for all parsers, and for our linker as well, is the attachment of prepositional phrases (PP attachment). We consider that, besides some structures that can be dealt with at a structural level, most of the times the PP attachment has to consider semantic aspects of the sentence to be processed. That is why, the size of the training corpus should be very large to ensure that verbs or nouns can be associated with the relevant dependents in a statistically relevant way. Also, we may aim at solving those situations in which PP attachment follows straightforwardly from structural bases using specially handcrafted rules.

Experimental Results and Evaluation

We have trained the linker on the previously mentioned Romanian and English corpora: ROC and ENC. The sizes of the training corpora are very small compared with the experiments presented in (Yuret 1998) and (Paskin 2001). The set of rules was not improved by the observations in the previous section. Comparison of the English output was made against the *1984* novel of George Orwell parsed with the non-projective FDG parser of (Tapanainen and Järvinen 1997) where the link names and their orientation were stripped off. On the Romanian side, the authors manually annotated a set of 20 sentences with linkages and the comparison was made against this gold standard annotation. Of course, further evaluation on a larger corpus is necessary both for English and Romanian with a special emphasis on Romanian.

The next table gives the results of the evaluation using the F-measure score. Every correct link found by the linker adds to its precision and recall.

	Romanian		
	P	**R**	**F**-m
MI	76.94%	72.26%	74.53%
DICE	76.94%	74.70%	75.80%

Table 1. Romanian results against 20 annotated sentences

	English		
	P	**R**	**F**-m
MI	68.61%	71.23%	69.89%
DICE	68.25%	72.00%	70.07%

Table 2. English results against automatically parsed George Orwell's *1984*

We have used both the pointwise mutual information (MI) association score and the DICE coefficient. The coverage is better in the case of the DICE coefficient perhaps

[2] Classification of negation can be done from many perspectives, thus distinguishing between different types of negation. For the purpose of our study we are only interested in the distinction between constituent versus sentential negation (in respect to Romanian, see (Barbu Mititelu and Maftei Ciolăneanu 2004)).

because the objection on pointwise MI given in (Manning and Schütze 1999). We think that the coverage for English is encouraging given the size of the training corpus. We look forward to training the linker on corpora of at least 20 million words and check the coverage on the parsed text.

Conclusions and Further Work

We have presented in this paper an algorithm that aims at discovering syntactic links between words in two pre-processed (i.e. lemmatized and POS-tagged) corpora. In our experiment we made use of the Lexical Attraction Models introduced by (Yuret 1998), but we developed morpho-syntactic constrains for obtaining better results. The assumption behind the use of such constrains is that words have preferences regarding the combinations they enter. For instance, prepositions do not combine with adjectives, but with nouns. So, if a string containing a preposition followed by an adjective and then by a noun is encountered, then the first established link will be between the adjective and the noun and only afterwards the preposition can be linked with the noun phrase created in the previous step.

The partial results presented here are obtained by working with small corpora. We are aware that testing our algorithm on larger corpora is essential and that is the next stage in our experiment. Using larger texts can be helpful for solving problems commonly encountered by parsers, such as PP attachment.

Among other uses of it[3], the linker developed here will help us in an experiment (see Barbu Mititelu and Ion 2005) of automatically transferring syntactic relations from English into Romanian using as input data a parallel corpus aligned at the word level[4] and whose English part is parsed using the FDG parser presented in (Tapanainen and Järvinen 1997). The aim is the bootstrapping of a dependency grammar for a Romanian parser. Sometimes there is a 1:2 correspondence between words in the two languages. If one is able to identify the way words link with each other in the two languages, then it makes the transfer procedure easier provided that the aligned Romanian words are already linked. The linkage acts as a support tree which will be oriented and named properly by the English parse tree.

Acknowledgements

The authors would like to thank the anonymous reviewers of this paper for their useful comments that were taken into consideration for the final form of the paper.

References

Barbu, A.-M. 2004. The negation NU: Lexical or affixal item? In E. Ionescu ed. *Understanding Romanian negation: syntactic and semantic approaches in a declarative perspective*, Bucureşti, Editura Universităţii din Bucureşti.

Barbu Mititelu, V. and Maftei Ciolăneanu, R. 2004. The Main Aspects of the Grammar of Negation in Romanian. In E. Ionescu ed. *Understanding Romanian negation: syntactic and semantic approaches in a declarative perspective*, Bucureşti, Editura Universităţii din Bucureşti.

Barbu Mititelu, V. and Ion, R. 2005. Automatic Import of Verbal Syntactic Relations Using Parallel Corpora. In *Proceedings of Recent Advances in Natural Language Processing*, Borovets, Bulgaria.

Brants, T. 2000. TnT a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference*, Seattle, WA.

Kindermann, R. and Snell, J. L. 1950. Markov Random Fields and their Applications. American Mathematical Society, Providence Rhode Island.

Manning, C. D. and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA.

Mel'čuk, I. 1988. *Dependency Syntax: Theory and Practice*, New York , SUNY Press.

Paskin, M. A. 2001. Grammatical Bigrams. In T. Dietterich, S. Becker, and Z. Gharahmani eds. *Advances in Neural Information Processing Systems 14 (NIPS-01)*. Cambridge, MA: MIT Press.

Sleator, D. and Temperley, D. 1991. Parsing English with a Link Grammar, Technical report CMUCS-91-196, Department of Computer Science, Carnegie Mellon University.

Tapanainen, P. and Järvinen, T. 1997. A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP'97)*, ACL, Washington, D.C.

Yuret, D. 1998. Discovery of linguistic relations using lexical attraction. Ph.D. diss., Department of Computer Science and Electrical Engineering, MIT.

[3] Such as word alignment if appropriate rule files are written for other languages.

[4] Development of a parser grammar is a time- and money-consuming task. Nowadays, when there are languages (such as English) for which there have been created a lot of good resources and when parallel corpora also exist, it may seem a good alternative to transfer knowledge from the resource-rich language into the resource-poor one via a parallel corpus aligned at word level.