Dependency-Based Phrase Alignment

Radu Ion, Alexandru Ceauşu and Dan Tufiş
Romanian Academy Research Institute for AI
13, Calea 13 Septembrie, 050711, Bucharest 5, Romania
radu@racai.ro, alceausu@racai.ro, tufis@racai.ro

Abstract

Phrase alignment is the task that requires the constituent phrases of two halves of a bitext to be aligned. In order to align phrases, one must discover them first and this article presents a method of aligning phrases that are discovered automatically. Here, the notion of a 'phrase' will be understood as being given by a subtree of a dependency-like structure of a sentence called linkage. To discover phrases, we will make use of two distinct, language independent methods: the IBM-1 model (Brown et al., 1993) adapted to detect linkages and Constrained Lexical Attraction Models (Ion & Barbu Mititelu, 2006). The methods will be combined and the resulted model will be used to annotate the bitext. The accuracy of phrase alignment will be evaluated by obtaining word alignments from link alignments and then by checking the F-measure of the latter word aligner.

1. Introduction

When establishing the correspondence between two reciprocal translations, there are at least three levels of alignment to be considered: sentence alignment, phrase alignment and finally word alignment. Sentence alignment of the parallel corpora, represent a resource that is useful in any multilingual NLP setting. Phrase and word alignment are more complex than sentence alignment and also more useful. There are word alignment systems that do phrase alignment at some stage of their processing time (see for instance (Tufiş et al., 2006)) prior to aligning the words belonging to the phrases. This divide et impera strategy is likely to provide better results than purely statistical methods because both phrase identification and aligning words within phrase boundaries may use language specific procedures. In any case, phrase alignment is a useful task especially to the phrase-based machine translation where the translations are generated as phrases rather than words (Koehn et al., 2003).

Traditionally, phrases are taken to be syntactic constituents of a sentence. They are units of a sentence that can be used to generate other sentences of the language and this is the strategy employed in phrase-based translation: instead of generating translation of individual words in the source language, generate translations of the phrases and assemble the final translation by a permutation of these (see (Koehn et al., 2003) or (Yamada & Knight, 2001)).

In a generative framework, a phrase is an ordered list of adjacent words such that they are all leaves of a syntactic tree that is contained by the syntactic analysis of the sentence. Within its dependency framework, Mel'čuk (1988) puts forward another view of the syntactic phrase: a phrase is a syntactic tree rooted at the phrase's head. The tree branches are the surface syntactic dependency relations that are established between the word-forms of the sentence. Evaluating the two views of a syntactic phrase, we find that the dependency formulation has one advantage: the syntactic phrase does not require the adjacency of the word-forms of the sentence, thus also allowing for meaningful discontinuous syntactic units.

Previous work on phrase alignment includes alignment of parse trees (Meyers, 1996; Lavoie, 2001; Gildea, 2003) and synchronous parsing (Alshavi et al., 2000; Lopez et al., 2002). These systems achieve more than we seek to: structural alignment of the parse trees (alignment of their nodes). By doing this, they implicitly align phrases because phrases are given as subtrees of the syntactic analysis. There are also systems that try to align phrases disregarding their structure (Koehn et al., 2003).

In the present article, we will make use of the dependency framework for defining phrases. Since we do not have yet a dependency grammar for Romanian, and neither the resources to build one, it is our hope that the linkages will enable, by exploiting the techniques of annotation transfer in parallel corpora, the induction of a Romanian dependency grammar core. This could be a robust starting point for a wider coverage grammar.

For the current task we will simplify the surface syntactic dependency structure from (Mel'čuk, 1988) by removing the labels on the syntactic relations to obtain a syntactic dependency tree that we call an oriented linkage. Such a linkage will be induced via two distinct, language independent methods:

- IBM-1 model (Brown et al., 1993) geared to detect oriented linkages;
- Constrained Lexical Attraction Models (CLAM) (Ion & Barbu Mititelu, 2006) modified to output oriented linkages.

Our goal is to align phrases made out of word-forms that are not necessary adjacent. This approach builds on easy-to-obtain pseudo syntactic structures, based on which produces the phrase alignment. Oriented linkages (if correct) coupled with good translation lexicons are sufficient for achieving a good phrase alignment.

The rest of the article is structured as follows: in the next section we will briefly describe the IBM-1 and CLAM linkers; the phrase alignment algorithm is presented in a distinct section. Finally, we will assess the accuracy of the phrase alignment on a bitext whose sentences were linked using a combination of the two previously mentioned methods.

2. Linkage Generation

A perfect linkage of a sentence is obtained from the syntactic dependency structure of the respective sentence. It is a connected graph whose vertices are the word-forms of the sentence. This linkage has the following properties:

- it is <u>not oriented</u>: in other words, from a linked pair of word-forms one cannot distinguish the governor from the dependent;

- it has no cycles: there exists only one possible path between any two word-forms of the sentence;
- it is planar: if the vertices of the graph are represented in a linear fashion, then the graph edges are not allowed to cross.

The linkage of a sentence is a simplification of the dependency structure but, on the other hand, it is much easier to obtain it even from raw text (see (Yuret 1998)).

Running the linkers requires each half of the bitext be tokenized, part-of-speech (POS) tagged and lemmatized. We have used in-house tools to do these tasks. The tagger is a PERL implementation of Brants' TnT tagger specifications (Brants, 2000) extended with a few extra heuristics for dealing with unknown words. The lemmatizers are based on lexicon lookup for finding lemmas. When a word-form is not found, we use a set of automatically induced rules to generate candidate lemmas and then Markov modeling to rank the candidates. The one with the highest probability wins.

2.1. The IBM-1 linker

This linker, based on the IBM-1 model (Brown et al., 1993), tries to obtain oriented linkages for phrases of an input sentence[1]. We have chosen the IBM-1 model because it is simple and it fits perfectly within the job description. It offers a training procedure (the EM algorithm) that produces link probabilities and an alignment procedure that can be used to generate the links. In our opinion, the positions of the words to be aligned, encoded by the IBM-2 model, are not useful to this particular task because a phrase can appear at any position in a sentence and what is important to us is the relative order of the words within the phrase and not their absolute position. The IBM 3,4 and 5 models are further refined in order to account (among others) for $m:n$ alignments. Since we are interested only in 1:1 alignments, their power is an unnecessary expense.

For turning IBM-1 into a linkage generator, we have modified the model's training and Viterbi-aligning procedures to run on a 'bitext' that contains pairs of sentences of a single language in which the source one is duplicated as the target. The changes that we performed to the EM algorithm and to the model are:

- removal of the NULL alignments because there always exists a link between any two word-forms of a sentence;
- the estimation step of the EM algorithm for IBM-1 ((Brown et al., 1993), page 272, equation 17) disregards words on the same position (index) because no link can relate a word-form to itself. Thus, we estimate the count $c(f_a|e_b;\mathbf{f},\mathbf{e})$, with $b \neq a$, $1 \leq b \leq l$ and $i > 0$, $i \neq b$.

The modified IBM-1 model has been trained to produce *t*-tables for lemmas and parts of speech. Thus we have obtained linkage probabilities for lemmas and parts of speech, which will serve the Viterbi alignment. IBM-1 Viterbi alignment has been adapted to give more credit to local links. That is, we are trying to discover phrases of whose words usually reside close to each other.

Finding $V(\mathbf{f}|\mathbf{e};1)$ ((Brown et al., 1993), page 276), means that for every $1 \leq j \leq m$, search $1 \leq i \leq l$ so that

$$\frac{t(f_j \mid e_i) + t(e_i \mid f_j)}{(i-j)^2}$$

is maximum. The denominator acts as a locality constraint penalizing high scores that tend to link words at distant positions. Please note that the nature of our monolingual 'bitext' imposes a symmetrical *t*-table and at this stage, we are not concerned with the link orientation. So, naturally, we want that both link probabilities (source to target and target to source) to contribute to the overall link score (hence the sum at the numerator). Also, one should note that the sum in the numerator is computed for both lemmas and parts of speech so the numerator has in fact 4 terms and that the links are generated so that the planarity constraint is preserved[2].

Link orientation is obtained by a set of linguistically universal rules that, given two POSes, specify which is the governor and which is the dependent, For instance, a main verb is always a governor regardless of the combination it enters. Then, a noun is governor if it is linked to any of the following: determiners, adjectives, articles or prepositions[3]. The adjective is a governor when it is modified by an adverb.

2.2. The CLAM linker

The CLAM linker is described at length in (Ion & Barbu Mititelu, 2006). It is an extension of the linker presented by Yuret (1998) and requires the input text be POS-tagged. The CLAM linker uses a set of language specific rules to restrict the set of possible links. It generates complete linkages of sentences when such a linkage can be constructed without violating the linking rules.

Because CLAMs produce complete linkages of sentences, the task of orienting them is not as simple as the similar one of the previous linker. We could use governor-selecting rules but because the number of combination possibilities is much bigger, we cannot say for sure that the rules will impose a tree structure over the linkage.

One solution is to choose an orientation that will impose a similar tree structure for any two word-aligned sentences of a bitext. That is, we have decided to select the first pair of aligned main verbs (that have high chances of being the actual roots as predicates) and to impose them as roots of the oriented linkages of both sentences. Then, the orientation process proceeds recursively: all dependents of the main verb will become in turn, the heads of their connected dependents and so on until a proper tree is obtained.

The combination of the two methods (IBM-1 and CLAM) for linkage generation will constitute the combined linkage generation model. This combination could be filtered in the same way that COWAL word aligner (Tufiş et al. 2006) is filtering the input alignments. But, for our present purposes, we have considered that the simple union of the linkages is sufficient to show that phrase alignment is a worthwhile enterprise

[1] This linker does not compute complete linkages for the sentences. It only determines structures of the phrases such as noun phrase, prepositional phrase, adjectival/adverbial phrase and verbal complexes.

[2] That is, a link is discarded if one of its ends is placed linearly between the indexes of an existing link.

[3] Mel'čuk (1988) chooses the preposition as the head when it enters in a combination with a noun. But, for our purposes it is only necessary to adopt the same conventions for both languages of the bitext.

3. Dependency-based Alignment

One of the features required by a word aligner is the translation equivalence of two tokens. Word translation equivalence by itself does not cover phenomena as collocation, verb phrase composition, etc. To cope with these requirements, the COWAL aligner (Tufiş et al. 2006) defined the collocation feature, but only with respect to adjacent lexical tokens. However, if one considers the lexical tokens, as well as the dependency relations among them as being the basic units subject to the alignment process, they, become altogether the atoms of the parameters estimation procedure. As such, the role of the collocation parameter of COWAL or of the fertility parameter introduced in the IBM models 3-5, is nicely taken over by the dependency relations. The computational complexity of the usual N-M word aligners is considerably reduced because the 1-1 alignment of the dependency links, together with the 1-1 lexical tokens alignment links encode the information necessary to compute the phrase alignments (the N-M alignment in the word alignment jargon).

The estimation of the translation equivalence table is a task that has to find how a target language sentence *t* of *m* words is generated from a source language sentence *s* consisting of *l* words. The IBM model 1 has the assumptions that the order of the sentences words does not matter and that all translations of the source language words can occur, having a uniform probability ε.

$$\Pr(t \mid s) = \frac{\in}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} tr(t_j \mid s_i)$$

Instead of using the words, the dependency-based aligner directly estimates the possible translation of a link between two words. Because the amount of memory needed for parameter estimation in this scenario would be immense, we filtered out linked pairs occurring less than three times in the corpus. All the words for which a dependency relation could not be established are automatically set as dependents of the special token *null-word*.

Because each dependency counts as a single token, the result of the estimation is a translation equivalence table of dependencies.

The same estimation process when applied to the word's part of speech instead of the word-form, results in a affinity table for the part-of-speech dependencies.

The second stage of the aligner uses the estimated translation equivalence tables and several other alignment features. For the alignment of the dependencies, we differentiate between two kinds of features, characterizing a link: *context independent* and *context dependent* features.

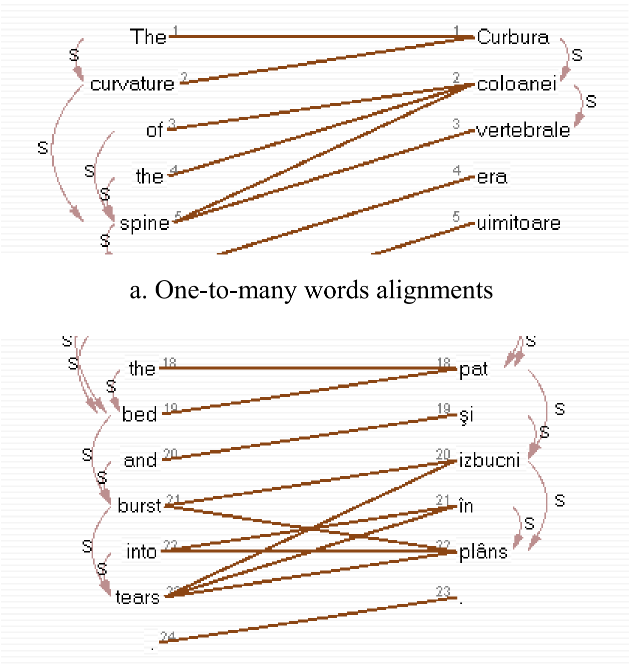
Iteration	Precision	Recall	F-Measure
"Anchor" dependencies	96.45%	43.56%	60.01%
Minimally crossing alignments	93.98%	63.58%	75.85%
Probable dependency links	92.83%	70.67%	80.24%

Table 1: The accuracy of the dependency aligner on different iterations

Context independent features, such as translation equivalence or cognate scores, rely only on the lexical tokens (words, phrases) paired by an alignment link. In our implementation we have 4 context independent features: dependency translation equivalence, category dependency affinity, the obliqueness (the difference on relative position), and cognate score computed on the content words of a dependency link.

Context dependent features for a candidate dependency link contain information from the surrounding links. Among these we have the locality and the links that were crossed by the candidate link.

Using several iterations and manually assigned thresholds and weights for each iteration the aligner aligns each dependency, favouring the ones that have features that show strong alignment information.



a. One-to-many words alignments

b. Many-to-many words alignments

Figure 1: Examples of phrase alignments

As can be seen in figure 1, our system can draw one-to-many and many-to-many word alignments. In the example of figure 1a the word "spine" is correctly linked to "coloanei vertebrale". The preposition "of" and the determiner "the" are correctly aligned with the Romanian gerund "coloanei". In figure 1b we show another case where the verbal phrase "burst into tears" is correctly aligned with the Romanian verbal phrase "izbucni în plâns"

4. Results

The aligner was trained on a 1.6 million words news corpus (TM1). For the evaluation of the system, we used the translation model TM1 and the gold standards provided in the ACL (2003 and 2005) Romanian-English word alignment competitions. Since we didn't have a gold standard of phrases available at the time we performed the experiments, we have estimated the accuracy of the phrase alignment in terms of word alignments derived from the dependencies alignments.

Aligner	Precision	Recall	F-measure
COWAL	86.99%	79.91%	83.30%
Dependency-based aligner	92.83%	70.67%	80.24%

Table 2: The accuracy of the dependency-based aligner

We compared the dependency-based aligner to the COWAL aligner, the best-rated word-aligner in the ACL2005 Romanian-English shared task (Martin et al., 2005). As it can be seen in Table 2, the performance of the dependency-based aligner is lower than the one of the COWAL aligner. The lower F-measure of our dependency aligner may be due to the way the gold standard was made.

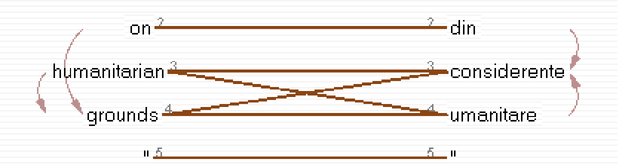


Figure 2: Multiple-word expression missing in the gold standard

In Figure 2 we show an example where a correctly identified alignment of a multi-word expression does not appear in the golden standard. This fact is due to the correct translation of the word “humanitarian” with “umanitare” and “grounds” with “considerente”. But our aligner identifies them as collocations and as such, it links every index of the English one with all the indexes of the Romanian one.

The high precision of word alignment in the case of the dependency-based aligner translates directly in high precision of the phrase alignment because of the way in which phrases can be extracted. Take for instance Figure 2 above: the link “on” – “grounds” is aligned with the link “din” – “considerente” and the link “humanitarian” – “grounds” with “considerente” – “umanitare”. But because “on humanitarian grounds” and “din considerente umanitare” are two sequences of words, these are also two aligned phrases. The low recall of the dependency-based aligner does not necessarily translate into low recall of phrase alignment because even if not all the words between two phrases are aligned, the phrases can still align very well.

5. Conclusions

We have presented a method of phrase alignment based on alignments of word dependencies. To generate dependencies, we have used two language independent methods that are easily implemented using no external resources and inexpensive text annotations like POS-tagging and lemmatization. The methods produce oriented linkages that are good approximations of the actual word dependencies (Yuret, 1998).

The evaluation of the phrase aligner is realized as a comparison between the COWAL word aligner and the word aligner derived by the dependencies alignment. Since we do not have a phrase alignment gold standard, we have compared the results produced by COWAL and the results of the dependency-based word aligner. The dependency-based word aligner has a considerably higher precision that is a suggestive clue of the precision of the phrase alignment. The recall of the dependency-based word aligner need not be the recall of the phrase aligner because even if not all words within a phrase are all aligned, the phrase might still get the proper alignment by following the word alignments and dependencies within that phrase.

6. References

Alshawi, H., Srinivas, B. & Douglas, S. (2000). Learning dependency translation models as collections of finite state head transducers. Computational Linguistics, 26(1), 45—60.

Brown, P.F., Pietra, S.A.D., Pietra, V.J.D. & Mercer, R.L. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics, 19, 263—311.

Gildea, D. (2003). Loosely Tree-Based Alignment for Machine Translation. In Proceedings of the 41st ACL (pp. 80—87), Sapporo, Japan.

Ion, R. & Barbu Mititelu, V. (2006). Constrained Lexical Attraction Models. In Proceedings of the Florida Artificial Intelligence Research Society Conference FLAIRS 2006, Special Track ‘Automatic Annotation by Categories for Text Information Extraction: New Perspectives’, to appear.

Koehn, P., Och, F.J. & Marcu, D. (2003). Statistical Phrase-Based Translation. In Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference NAACL/HLT 2003, Edmonton, Canada.

Lavoie, B., White, M. & Korelsky, T. (2001). Inducing Lexico-Structural Transfer Rules from Parsed Bi-texts. In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics ACL 2001, DDMT Workshop, Toulouse, France.

Joel Martin, Rada Mihalcea, Ted Pedersen. (2005). Word Alignment for Languages with Scarce Resources. In *Proceeding of the ACL2005 Workshop on “Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond”.* June, 2005, *Ann Arbor, Michigan, June,* Association for Computational Linguistics, 65–74

Mel’čuk, I.A. (1988). Dependency Syntax: Theory and Practice. Albany, NY: State University of New York Press.

Meyers, A., Yangarber, R. & Grisham, R. (1996). Alignment of shared forests for bilingual corpora. In Proceedings of 16th International Conference on Computational Linguistics COLING-96 (pp. 460—465), Copenhagen, Denmark.

Tufiş, D., Ion, R., Ceauşu, A. & Ştefănescu, D. (2006). Improved Lexical Alignment by Combining Multiple Reified Alignments. In Proceedings of the European Chapter of the Association for Computational Linguistics EACL 2006, to appear.

Yamada, K., Knight, K. (2001). A syntax-based statistical translation model. In Proceedings of the 39th Meeting of the Association for Computational Linguistics ACL 2001 (pp. 523—530), Toulouse, France.

Yuret, D. (1998). Discovery of linguistic relations using lexical attraction. PhD thesis, Department of Computer Science and Electrical Engineering, MIT.