

Improved Lexical Alignment by Combining Multiple Reified Alignments

| | | | |
|-------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Dan Tufiş Institute for Artificial Intelligence 13, “13 Septembrie”, 050711, Bucharest 5, Romania tufis@racai.ro | Radu Ion Institute for Artificial Intelligence 13, “13 Septembrie”, 050711, Bucharest 5, Romania radu@racai.ro | Alexandru Ceauşu Institute for Artificial Intelligence 13, “13 Septembrie”, 050711, Bucharest 5, Romania alceausu@racai.ro | Dan Ştefănescu Institute for Artificial Intelligence 13, “13 Septembrie”, 050711, Bucharest 5, Romania danstef@racai.ro |
|-------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|

Abstract

We describe a word alignment platform which ensures text pre-processing (tokenization, POS-tagging, lemmatization, chunking, sentence alignment) as required by an accurate word alignment. The platform combines two different methods, producing distinct alignments. The basic word aligners are described in some details and are individually evaluated. The union of the individual alignments is subject to a filtering post-processing phase. Two different filtering methods are also presented. The evaluation shows that the combined word alignment contains 10.75% less errors than the best individual aligner.

1 Introduction

It is almost a truism that more decision makers, working together, are likely to find a better solution than when working alone. Dieterich (1998) discusses conditions under which different decisions (in his case classifications) may be combined for obtaining a better result. Essentially, a successful automatic combination method would require comparable performance for the decision makers and, additionally, that they should not make similar errors. This idea has been exploited by various NLP researchers in language modeling, statistical POS tagging, parsing, etc.

We developed two quite different word aligners, driven by two distinct objectives: the first one was motivated by a project aiming at the development of an interlingually aligned set of wordnets while the other one was developed within an SMT ongoing project. The first one

was used for validating, against a multilingual corpus, the interlingual synset equivalences and also for WSD experiments. Although, initially, it was concerned only with open class words recorded in a wordnet, turning it into an “all words” aligner was not a difficult task. This word aligner, called **YAWA** is described in section 3.

A quite different approach from the one used by **YAWA**, is implemented in our second word aligner, called **MEBA**, described in section 4. It is a multiple parameter and multiple step algorithm using relevance thresholds specific to each parameter, but different from each step to the other. The implementation of **MEBA** was strongly influenced by the notorious five IBM models described in (Brown et al. 1993). We used **GIZA++** (Och and Ney 2000; Och and Ney, 2003) to estimate different parameters of the **MEBA** aligner.

The alignments produced by **MEBA** were compared to the ones produced by **YAWA** and evaluated against the Gold Standard (GS)¹ annotations used in the Word Alignment Shared Tasks (Romanian-English track) organized at HLT-NAACL2003 (Mihalcea and Pedersen 2003).

Given that the two aligners are based on quite different models and that their F-measures are comparable, it was quite a natural idea to combine their results and hope for an improved alignment. Moreover, by analyzing the alignment errors done by each word aligner, we found that the number of common mistakes was small, so

¹ We noticed in the GS Alignment various errors (both sentence and word alignment errors) that were corrected. The tokenization of the bitexts used in the GS Alignment was also modified, with the appropriate modification of the reference alignment. These reference data are available at <http://www.racai.ro/res/WA-GS>

the premises for a successful combination were very good (Dieterich, 1998). The Combined Word Aligner, **COWAL**-described in section 5, is a wrapper of the two aligners (YAWA and MEBA) merging the individual alignments and filtering the result. At the Shared Task on Word Alignment organized by the ACL2005 Workshop on “Building and Using Parallel Corpora: Data-driven Machine Translation and Beyond” (Martin, et al. 2005), we participated (on the Romanian-English track) with the two aligners and the combined one (COWAL). Out of 37 competing systems, COWAL was rated the first, MEBA the 20th and TREQ-AL (Tufiş et al. 2003), the former version of YAWA, was rated the 21st. The usefulness of the aligner combination was convincingly demonstrated.

Meanwhile, both the individual aligners and their combination were significantly improved. COWAL is now embedded into a larger platform that incorporates several tools for bitexts pre-processing (briefly reviewed in section 2), a graphical interface that allows for comparing and editing different alignments, as well as a word sense disambiguation module.

2 The bitext processing

The two base aligners and their combination use the same format for the input data and provide the alignments in the same format. The input format is obtained from two raw texts that represent reciprocal translations. If not already sentence aligned, the two texts are aligned by our sentence aligner that builds on Moore’s aligner (Moore, 2002) but which unlike it, is able to recover the non-one-to-one sentence alignments. The texts in each language are then tokenized, tagged and lemmatized by the TTL module (Ion, 2006). More often than not, the translation equivalents have the same part-of speech, but relying on such a restriction would seriously affect the alignment recall. However, when the translation equivalents have different parts of speech, this difference is not arbitrary. During the training phase, we estimated *POS affinities*: $\{p(\text{POS}_m^{\text{RO}}|\text{POS}_n^{\text{EN}})\}$ and $\{p(\text{POS}_n^{\text{EN}}|\text{POS}_m^{\text{RO}})\}$ and used them to filter out improbable translation equivalents candidates.

The next pre-processing step is represented by sentence chunking in both languages. The chunks are recognized by a set of regular expressions defined over the tagsets and they correspond to (non-recursive) noun phrases, adjectival phrases, prepositional phrases and verb com-

plexes (analytical realization of tense, aspect mood and diathesis and phrasal verbs). Finally, the bitext is assembled as an XML document (Tufiş and Ion, 2005), which is the standard input for most of our tools, including COWAL alignment platform.

3 YAWA

YAWA is a three stage lexical aligner that uses bilingual translation lexicons and phrase boundaries detection to align words of a given bitext. The translation lexicons are generated by a different module, TREQ (Tufiş, 2002), which generates translation equivalence hypotheses for the pairs of words (one for each language in the parallel corpus) which have been observed occurring in aligned sentences more than expected by chance. The hypotheses are filtered by a log-likelihood score threshold. Several heuristics (string similarity-cognates, POS affinities and alignments locality²) are used in a competitive linking manner (Melamed, 2001) to extract the most likely translation equivalents.

YAWA generates a bitext alignment by incrementally adding new links to those created at the end of the previous stage. The existing links act as contextual restrictors for the new added links. From one phase to the other new links are added without deleting anything. This monotonic process requires a very high precision (at the price of a modest recall) for the first step. The next two steps are responsible for significantly improving the recall and ensuring an increased F-measure.

In the rest of this section we present the three stages of YAWA and evaluate the contribution of each of them to the final result.

3.1 Phase 1: Content Words Alignment

YAWA begins by taking into account only very probable links that represent the skeleton alignment used by the second phase. This alignment is done using outside resources such as translation lexicons and involves only the alignment of content words (nouns, verbs, adjective and adverbs).

The translation equivalence pairs are ranked according to an association score (i.e. log-likelihood, DICE, point-wise mutual informa-

² The *alignments locality* heuristics exploits the observation made by several researchers that adjacent words of a text in the source language tend to align to adjacent words in the target language. A more strict alignment locality constraint requires that all alignment links starting from a chunk in the one language end in a chunk in the other language.

tion, etc.). We found that the best filtering of the translation equivalents was the one based on the log-likelihood (LL) score with a threshold of 9.

Each translation unit (pair of aligned sentences) of the target bitext is scanned for establishing the most likely links based on a competitive linking strategy that takes into account the LL association scores given by the TREQ translation lexicon. If a candidate pair of words is not found in the translation lexicon, we compute their orthographic similarity (cognate score (Tufiş, 2002)). If this score is above a predetermined threshold (for Romanian-English task we used the empirically found value of 0.43), the two words are treated as if they existed in the translation lexicon with a high association score (in practice we have multiplied the cognate score by 100 to yield association scores in the range 0 .. 100). The Figure 1 exemplifies the links created between two tokens of a parallel sentence by the end of the first phase.

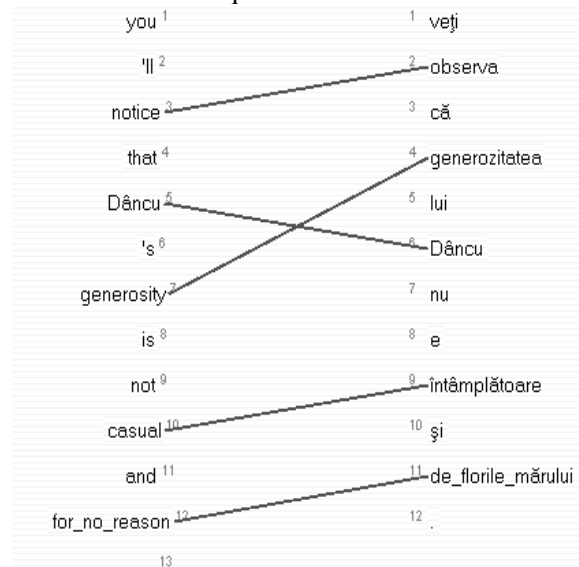


Figure 1: Alignment after the first step

3.2 Phase 2: Chunks Alignment

The second phase requires that each part of the bitext is chunked. In our Romanian-English experiments, this requirement was fulfilled by using a set of regular expressions defined over the tagsets used in the target bitext. These simple chunkers recognize noun phrases, prepositional phrases, verbal and adjectival or adverbial groupings of both languages.

In this second phase YAWA produces first chunk-to-chunk matching and then aligns the words within the aligned chunks. Chunk alignment is done on the basis of the skeleton alignment produced in the first phase. The algorithm is simple: align two chunks $c(i)$ in source lan-

guage and $c(j)$ in the target language if $c(i)$ and $c(j)$ have the same type (noun phrase, prepositional phrase, verb phrase, adjectival/adverbial phrase) and if there exist a link $\langle w(s), w(t) \rangle$ so that $w(s) \in c(i)$ then $w(t) \in c(j)$.

After alignment of the chunks, a language pair dependent module takes over to align the unaligned words belonging to the chunks. Our module for the Romanian-English pair of languages contains some very simple empirical rules such as: if b is aligned to c and b is preceded by a , link a to c , unless there exist d in the same chunk with c and the POS category of d has a significant affinity with the category of a . The simplicity of these rules derives from the shallow structures of the chunks. In the above example b and c are content words while a is very likely a determiner or a modifier for b . The result of the second alignment phase, considering the same sentence in Figure 1, is shown in Figure 2. The new links are represented by the double lines.

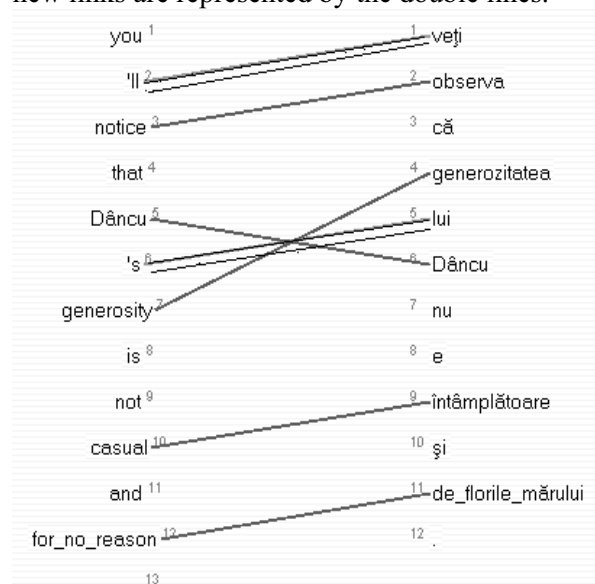


Figure 2: Alignment after the second step

3.3 Phase 3: Dealing with sequences of unaligned words

This phase identifies contiguous sequences of words (blocks) in each part of the bitext which remain unaligned and attempts to heuristically match them. The main criteria used to this end are the POS-affinities of the remaining unaligned words and their relative positions. Let us illustrate, using the same example and the result shown in Figure 2, how new links are added in this last phase of the alignment. At the end of phase 2 the blocks of consecutive words that remain to be aligned are: English $\{en_0 = (\text{you}), en_1 = (\text{that}), en_2 = (\text{is, not}), en_3 = (\text{and}), en_4 = (\text{.})\}$ and

Romanian $\{ro_0 = () , ro_1 = (c\ddot{a}), ro_2 = (nu, e), ro_3 = (\text{\textasciitilde{s}}i), ro_4 = (.)\}$. The mapping of source and target unaligned blocks depends on two conditions: that surrounding chunks are already aligned and that pairs in candidate unaligned blocks have significant POS-affinity. For instance in the figure above, blocks $en_1 = (\text{that})$ and $ro_1 = (c\ddot{a})$ satisfy the above conditions because they appear among already aligned chunks ($\langle \text{‘Il notice} \rangle \Leftrightarrow \langle \text{ve\textasciitilde{t}i observa} \rangle$ and $\langle \text{D\ddot{a}ncu ‘s generosity} \rangle \Leftrightarrow \langle \text{generozitatea lui D\ddot{a}ncu} \rangle$) and they contain words with the same POS.

After block alignment³, given a pair of aligned blocks, the algorithm links words with the same POS and then the phase 2 is called again with these new links as the skeleton alignment. In Figure 3 is shown the result of phase 3 alignment of the sentence we used as an example throughout this section. The new links are shown (as before) by double lines.

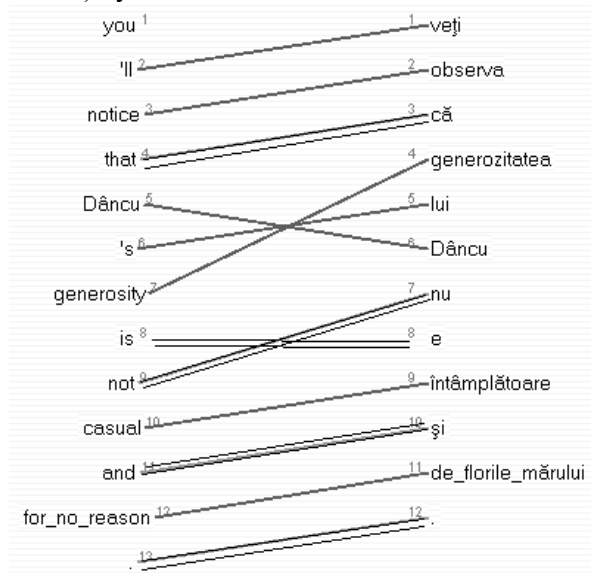


Figure 3: Alignment after the third step

The third phase is responsible for significant improvement of the alignment recall, but it also generates several wrong links. The detection of some of them is quite straightforward, and we added an additional correction phase 3.f. By analysing the bilingual training data we noticed the translators’ tendency to preserve the order of the phrasal groups. We used this finding (which might not be valid for any language pair) as a removal heuristics for the links that cross two or more aligned phrase groups. One should notice that the first word in the English side of the example in Figure 3 (“you”) remained unaligned (interpreted as not translated in the Romanian side). According to the Gold Standard used for

³ Only 1:1 links are generated between blocks.

evaluation in the ACL2005 shared task, this interpretation was correct, and therefore, for the example in Figure 3, the F-measure for the YAWA alignment was 100%.

However, Romanian is a pro-drop language and although the translation of the English pronoun is not lexicalized in Romanian, one could argue that the auxiliary “ve\textasciitilde{t}i” should be aligned also to the pronoun “you” as it incorporates the grammatical information carried by the pronoun. Actually, MEBA (as exemplified in Figure 4) produced this multiple token alignment (and was penalized for it!).

3.4 Performance analysis

The table that follows presents the results of the YAWA aligner at the end of each alignment phase. Although the Precision decreases from one phase to the next one, the Recall gains are significantly higher, so the F-measure is monotonically increasing.

| | Precision | Recall | F-Measure |
|-----------------|---------------|---------------|---------------|
| Phase 1 | 94.08% | 34.99% | 51.00% |
| Phase 1+2 | 89.90% | 53.90% | 67.40% |
| Phase 1+2+3 | 88.82% | 73.44% | 80.40% |
| Phase 1+2+3+3.f | 88.80% | 74.83% | 81.22% |

Table 1: YAWA evaluation

4 MEBA

MEBA uses an iterative algorithm that takes advantage of all pre-processing phases mentioned in section 2. Similar to YAWA aligner, MEBA generates the links step by step, beginning with the most probable (*anchor links*). The links to be added at any later step are supported or restricted by the links created in the previous iterations. The aligner has different weights and different significance thresholds on each feature and iteration. Each of the iterations can be configured to align different categories of tokens (named entities, dates and numbers, content words, functional words, punctuation) in decreasing order of statistical evidence.

The first iteration builds *anchor links* with a high level of certainty (that is cognates, numbers, dates, pairs with high translation probability). The next iteration tries to align content words (open class categories) in the immediate vicinity of the anchor links. In all steps, the candidates are considered if and only if they meet the minimal threshold restrictions.

A link between two tokens is characterized by a set of features (with values in the [0,1] interval). We differentiate between *context independ-*

ent features that refer only to the tokens of the current link (translation equivalency, part-of-speech affinity, cognates, etc.) and *context dependent features* that refer to the properties of the current link with respect to the rest of links in a bi-text (locality, number of traversed links, tokens indexes displacement, collocation). Also, we distinguish between bi-directional features (translation equivalence, part-of-speech affinity) and non-directional features (cognates, locality, number of traversed links, collocation, indexes displacement).

| | Precision | Recall | F-measure |
|------------------------------|---------------|---------------|---------------|
| “Anchor” links | 98.50% | 26.82% | 42.16% |
| Words around “anchors” | 96.78% | 42.41% | 58.97% |
| Funct. words and punctuation | 94.74% | 59.48% | 73.08% |
| Probable links | 92.05% | 71.00% | 80.17% |

Table 2: MEBA evaluation

The score of a candidate link (LS) between a source token i and a target token j is computed by a linear function of several features scores (Tiedemann, 2003).

$$LS(i, j) = \sum_{i=1}^n \lambda_i * ScoreFeat_i ; \sum_{i=1}^n \lambda_i = 1$$

Each feature has defined a specific significance threshold, and if the feature’s value is below this threshold, the contribution to the LS of the current link of the feature in case is nil.

The thresholds of the features and lambdas are different from one iteration to the others and they are set by the user during the training and system fine-tuning phases. There is also a general threshold for the link scores and only the links that have the LS above this threshold are retained in the bitext alignment. Given that this condition is not imposing unique source or target indexes, the resulting alignment is inherently many-to-many.

In the following subsections we briefly discuss the main features we use in characterising a link.

4.1 Translation equivalence

This feature may be used for two types of pre-processed data: lemmatized or non-lemmatized input. Depending on the input format, MEBA invokes GIZA++ to build translation probability lists for either lemmas or the occurrence forms of

the bitext⁴. Irrespective of the lemmatisation option, the considered token for the translation model build by GIZA++ is the respective lexical item (lemma or wordform) trailed by its POS tag (eg. plane_N, plane_V, plane_A). In this way we avoid data sparseness and filter noisy data. For instance, in case of highly inflectional languages (as Romanian is) the use of lemmas significantly reduces the data sparseness. For languages with weak inflectional character (as English is) the POS trailing contributes especially to the filtering the search space. A further way of removing the noise created by GIZA++ is to filter out all the translation pairs below a LL-threshold. We made various experiments and, based on the estimated ratio between the number of false negatives and false positive, empirically set the value of this threshold to 6. All the probability losses by this filtering were redistributed proportionally to their initial probabilities to the surviving translation equivalence candidates.

4.2 Translation equivalence entropy score

The translation equivalence relation is a semantic one and it directly addresses the notion of word sense. One of the Zipffian laws prescribes a skewed distribution of the senses of a word occurring several times in a coherent text. We used this conjecture as a highly informative information source for the validity of a candidate link. The translation equivalence entropy score is a favouring parameter for the words that have few high probability translations. Since this feature is definitely sensitive to the order of the lexical items, we compute an average value for the link: $\alpha ES(A) + \beta ES(B)$. Currently we use $\alpha = \beta = 0.5$, but it might be interesting to see, depending on different language pairs, how the performance of the aligner would be affected by a different settings of these parameters.

$$ES(W) = 1 - \frac{-\sum_{i=1}^N p(W, TR_i) * \log p(W, TR_i)}{\log N}$$

4.3 Part-of-speech affinity

In faithful translations the translated words tend to be translated by words of the same part-of-speech. When this is not the case, the different POSes, are not arbitrary. The part of speech affinity, $P(cat(A)|cat(B))$, can be easily computed from a gold standard alignment. Obviously, this

⁴ Actually, this is a user-set parameter of the MEBA aligner; if the input bitext contain lemmatization information, both translation probability tables may be requested.

is a directional feature, so an averaging operation is necessary in order to ascribe this feature to a link: $PA = \alpha P(\text{cat}(A)|\text{cat}(B)) + \beta P(\text{cat}(B)|\text{cat}(A))$. Again, we used $\alpha = \beta = 0.5$ but different values of these weights might be worthwhile investigating.

4.4 Cognates

The similarity measure, $\text{COGN}(T_S, T_T)$, is implemented as a Levenstein metric. Using the COGN test as a filtering device is a heuristic based on the *cognate conjecture*, which says that when the two tokens of a translation pair are orthographically similar, they are very likely to have similar meanings (i.e. they are cognates). The threshold for the $\text{COGN}(T_S, T_T)$ test was empirically set to 0.42. This value depends on the pair of languages in the bitext. The actual implementation of the COGN test includes a language-dependent normalisation step, which strips some suffixes, discards the diacritics, reduces some consonant doubling, etc. This normalisation step was hand written, but, based on available lists of cognates, it could be automatically induced.

4.5 Obliqueness

Each token in both sides of a bi-text is characterized by a position index, computed as the ratio between the relative position in the sentence and the length of the sentence. The absolute value of the difference between tokens' position indexes, subtracted from 1⁵, gives the link's "obliqueness".

$$OBL(SW_i, TW_j) = 1 - \left| \frac{i}{\text{length}(\text{Sent}_S)} - \frac{j}{\text{length}(\text{Sent}_T)} \right|$$

This feature is "context free" as opposed to the locality feature described below.

4.6 Locality

Locality is a feature that estimates the degree to which the links are sticking together.

MEBA has three features to account for locality: (i) *weak locality*, (ii) *chunk-based locality* and (iii) *dependency-based locality*.

The value of the *weak locality* feature is derived from the already existing alignments in a window of N tokens centred on the focused token. The window size is variable, proportional to the sentence length. If in the window there exist k linked tokens and the relative positions of the

tokens in these links are $\langle i_1 j_1 \rangle, \dots, \langle i_k j_k \rangle$ then the locality feature of the new link $\langle i_{k+1}, j_{k+1} \rangle$ is defined by the equation below:

$$LOC = \min\left(1, \frac{1}{k} \sum_{m=1}^k \frac{|i_{k+1} - i_m|}{|j_{k+1} - j_m|}\right)$$

If the new link starts from or ends in a token already linked, the index difference that would be null in the formula above is set to 1. This way, such candidate links would be given support by the LOC feature (and avoid overflow error). In the case of *chunk-based locality* the window span is given by the indexes of the first and last tokens of the chunk.

Dependency-based locality uses the set of the dependency links of the tokens in a candidate link for the computation of the feature value. In this case, the LOC feature of a candidate link $\langle i_{k+1}, j_{k+1} \rangle$ is set to 1 or 0 according to the following rule:

if between i_{k+1} and i_α there is a (source language) dependency and if between j_{k+1} and j_β there is also a (target language) dependency then LOC is 1 if i_α and j_β are aligned, and 0 otherwise. Please note that in case $j_{k+1} \equiv j_\beta$ a trivial dependency (identity) is considered and the LOC attribute of the link $\langle i_{k+1}, j_{k+1} \rangle$ is set to always to 1.

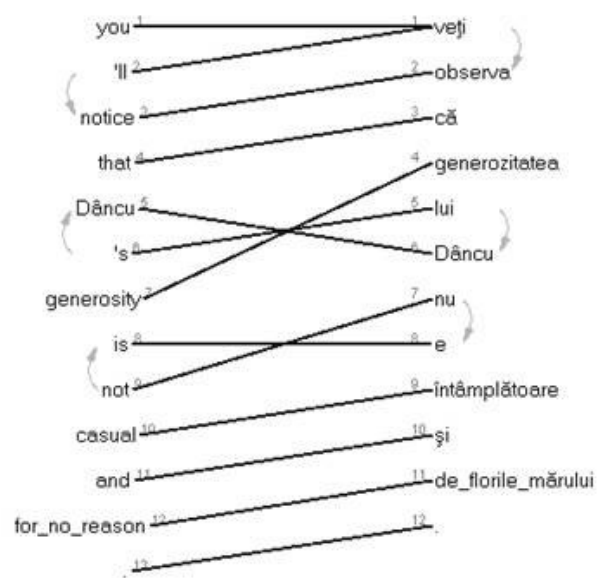


Figure 4: Chunk and dependency-based locality

4.7 Collocation

Monolingual collocation is an important clue for word alignment. If a source collocation is translated by a multiword sequence, very often the lexical cohesion of source words can also be found in the corresponding translated words. In this case the aligner has strong evidence for

⁵ This is to ensure that values close to 1 are "good" ones and those near 0 are "bad". This definition takes into account the relatively similar word order in English and Romanian.

many to many linking. When a source collocation is translated as a single word, this feature is a strong indication for a many to 1 linking.

Bi-gram lists (only content words) were built from each monolingual part of the training corpus, using the log-likelihood score (threshold of 10) and minimal occurrence frequency (3) for candidates filtering.

We used the bi-grams list to annotate the chains of lexical dependencies among the contents words. Then, the value of the collocation feature is computed similar to the dependency-based locality feature. The algorithm searches for the links of the lexical dependencies around the candidate link.

5 Combining the reified alignments

From a given alignment one can compute a series of properties for each of its links (such as the parameters used by the MEBA aligner). A link becomes this way a structured object that can be manipulated in various ways, independent of the bitext (or even of the lexical tokens of the link) from which it was extracted. We call this procedure *alignment reification*. The properties of the links of two or more alignments are used for our methods of combining the alignments.

One simple, but very effective method of alignment combination is a heuristic procedure, which merges the alignments produced by two or more word aligners and filters out the links that are likely to be wrong. For the purpose of filtering, a link is characterized by its type defined by the pair of indexes (i, j) and the POS of the tokens of the respective link. The likelihood of a link is proportional to the POS affinities of the tokens of the link and inverse proportional to the *bounded relative positions* (BRP) of the respective tokens: $BRP = 1 + ||i - j| - avg|$ where avg is the average displacement in a Gold Standard of the aligned tokens with the same POSes as the tokens of the current link. From the same gold standard we estimated a threshold below which a link is removed from the final alignment.

A more elaborated alignment combination (with better results than the previous one) is modelled as a binary statistical classification problem (good / bad) and, as in the case of the previous method, the net result is the removal of the links which are likely to be wrong. We used an “off-the-shelf” solution for SVM training and classification - LIBSVM⁶ (Fan et al., 2005) with

the default parameters (C-SVC classification and radial basis kernel function). Both context independent and context dependent features characterizing the links were used for training. The classifier was trained with both positive and negative examples of links. A set of links extracted from the Gold Standard alignment was used as positive examples set. The same number of negative examples was extracted from the alignments produced by COWAL and MEBA where they differ from the Gold Standard.

It is interesting to notice that for the example discussed in Figures 1-4, the first combiner didn’t eliminate the link <you veți> producing the result shown in Figure 4. This is because the relative positions of the two words are the same and the POS-affinity of the English personal pronouns and the Romanian auxiliaries is significant. On the other hand, the SVM-based combiner deleted this link, producing the result shown in Figure 3. The explanation is that, according to the Gold Standard we used, the links between English pronouns and Romanian auxiliaries or main verbs in pro-drop constructions were systematically dismissed (although we claim that they shouldn’t and that the alignment in Figure 4 is better than the one in Figure 3). The evaluation (according to the Gold Standard) of the SVM-based combination (COWAL), compared with the individual aligners, is shown in Table 3.

| Aligner | Precision | Recall | F-measure |
|---------|---------------|---------------|---------------|
| YAWA | 88.80% | 74.83% | 81.22% |
| MEBA | 92.05% | 71.00% | 80.17% |
| COWAL | 86.99% | 79.91% | 83.30% |

Table 3: Combined alignment

6 Conclusions and further work

Neither YAWA nor MEBA needs an a priori bilingual dictionary, as this will be automatically extracted by TREQ or GIZA++. We made evaluation of the individual alignments in both experimental settings: without a start-up bilingual lexicon and with an initial mid-sized bilingual lexicon. Surprisingly enough, we found that while the performance of YAWA increases a little bit (approx. 1% increase of the F-measure) MEBA is doing better without an additional lexicon. Therefore, in the evaluation presented in the previous section MEBA uses only the training data vocabulary.

YAWA is very sensitive to the quality of the bilingual lexicons it uses. We used automatically translation lexicons (with or without a seed lexi-

⁶ <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

con), and the noise inherently present might have had a bad influence on YAWA's precision. Replacing the TREQ-generated bilingual lexicons with validated (reference bilingual lexicons) would further improve the overall performance of this aligner. Yet, this might be a harder to meet condition for some pairs of languages than using parallel corpora.

MEBA is more versatile as it does not require a-priori bilingual lexicons but, on the other hand, it is very sensitive to the values of the parameters that control its behaviour. Currently they are set according to the developers' intuition and after the analysis of the results from several trials. Since this activity is pretty time consuming (human analysis plus re-training might take a couple of hours) we plan to extend MEBA with a supervised learning module, which would automatically determine the "optimal" parameters (thresholds and weights) values.

It is worth noticing that with the current versions of our basic aligners, significantly improved since the ACL shared word alignment task in June 2005, YAWA is now doing better than MEBA, and the COWAL F-measure increased with 9.4%. However, as mentioned before, these performances were measured on a different tokenization of the evaluation texts and on the partially corrected gold standard alignment (see footnote 1).

References

- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Robert J. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2): 263–311.
- Thomas G. Dietterich. 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10 (7) 1895-1924.
- Rong-en Fan, Pai-Hsuen Chen, Chij-Jen Lin. 2005. Working set selection using the second order information for training SVM. Technical report, Department of Computer Science, National Taiwan University (www.csie.ntu.edu.tw/~cjlin/papers/quadworkset.pdf).
- William A. Gale, Kenneth W. Church. 1991. Identifying word correspondences in parallel texts. In *Proceedings of the Fourth DARPA Workshop on Speech and Natural Language*. Asilomar, CA:152–157.
- Radu Ion. 2006. TTL: A portable framework for tokenization, tagging and lemmatization of large corpora. PhD thesis progress report. Research Institute for Artificial Intelligence, Romanian Academy, Bucharest (in Romanian), 22p.
- Dan Melamed. 2001. *Empirical Methods for Exploiting Parallel Texts*. Cambridge, MA, MIT Press.
- Rada Mihalcea, Ted Pedersen. 2003. An Evaluation Exercise for Word Alignment. *Proceedings of the HLT-NAACL 2003 Workshop: Building and Using Parallel Texts Data Driven Machine Translation and Beyond*. Edmonton, Canada: 1–10.
- 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
- Robert Moore. 2002. Fast and Accurate Sentence Alignment of Bilingual Corpora in Machine Translation: From Research to Real Users. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas*, Tiburon, California), Springer-Verlag, Heidelberg, Germany: 135-244.
- Franz J. Och, Herman Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, 29(1):19-51.
- Franz J. Och, Herman Ney. 2000. Improved Statistical Alignment Models. In *Proceedings of the 38th Conference of ACL*, Hong Kong: 440-447.
- Joerg Tiedemann. 2003. Combining clues for word alignment. In *Proceedings of the 10th EACL*, Budapest, Hungary: 339–346.
- Dan Tufiş. 2002. A cheap and fast way to build useful translation lexicons. In *Proceedings of COLING2002*, Taipei, China: 1030-1036.
- Dan Tufiş, Ana-Maria Barbu, Radu Ion. 2003. TREQ-AL: A word-alignment system with limited language resources. In *Proceedings of the NAACL 2003 Workshop on Building and Using Parallel Texts; Romanian-English Shared Task*, Edmonton, Canada: 36-39.
- Dan Tufiş, Radu Ion, Alexandru Ceaşu, Dan Stefănescu. 2005. Combined Aligners. 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, pp. 107-110.
- Dan Tufiş, Radu Ion. 2005. Multiple Sense Inventories and Test-Bed Corpora. In C. Burileanu (ed.) *Trends in Speech Technology*, Publishing House of the Romanian Academy, Bucharest: 49-58.