HLT-NAACL 2003 Workshop: Building and Using Parallel Texts
Data Driven Machine Translation and Beyond , pp. 11-14
Edmonton, May-June 2003

ProAlign: Shared Task System Description

Dekang Lin and **Colin Cherry**
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
{lindek,colinc}@cs.ualberta.ca

Abstract

ProAlign combines several different approaches in order to produce high quality word word alignments. Like competitive linking, ProAlign uses a constrained search to find high scoring alignments. Like EM-based methods, a probability model is used to rank possible alignments. The goal of this paper is to give a bird's eye view of the ProAlign system to encourage discussion and comparison.

1 Alignment Algorithm at a Glance

We have submitted the ProAlign alignment system to the WPT'03 shared task. It received a 5.71% AER on the English-French task and 29.36% on the Romanian-English task. These results are with the no-null data; our output was not formatted to work with explicit nulls.

ProAlign works by iteratively improving an alignment. The algorithm creates an initial alignment using search, constraints, and summed ϕ^2 correlation-based scores (Gale and Church, 1991). This is similar to the competitive linking process (Melamed, 2000). It then learns a probability model from the current alignment, and conducts a constrained search again, this time scoring alignments according to the probability model. The process continues until results on a validation set begin to indicate over-fitting.

For the purposes of our algorithm, we view an alignment as a set of links between the words in a sentence pair. Before describing the algorithm, we will define the following notation. Let E be an English sentence $e_1, e_2, \ldots, e_m$ and let F be a French sentence $f_1, f_2, \ldots, f_n$. We define a **link** $l(e_i, f_j)$ to exist if e_i and f_j are a translation (or part of a translation) of one another. We define the **null link** $l(e_i, f_0)$ to exist if e_i does not correspond to a translation for any French word in F. The null link $l(e_0, f_j)$ is defined similarly. An **alignment** A for two sentences E and F is a set of links such that every word in E and F participates in at least one link, and a word linked to e_0 or f_0 participates in no other links. If e occurs in E x times and f occurs in F y times, we say that e and f **co-occur** xy times in this sentence pair.

ProAlign conducts a best-first search (with constant beam and agenda size) to search a constrained space of possible alignments. A state in this space is a partial alignment, and a transition is defined as the addition of a single link to the current state. Any link which would create a state that does not violate any constraint is considered to be a valid transition. Our start state is the empty alignment, where all words in E and F are implicitly linked to null. A terminal state is a state in which no more links can be added without violating a constraint. Our goal is to find the terminal state with the highest probability.

To complete this algorithm, one requires a set of constraints and a method for determining which alignment is most likely. These are presented in the next two sections. The algorithm takes as input a set of English-French sentence pairs, along with dependency trees for the English sentences. The presence of the English dependency tree allows us to incorporate linguistic features into our model and linguistic intuitions into our constraints.

2 Constraints

The model used for scoring alignments has no mechanism to prevent certain types of undesirable alignments, such as having all French words align to the same English word. To guide the search to correct alignments, we employ two constraints to limit our search for the most probable alignment. The first constraint is the **one-to-one constraint** (Melamed, 2000): every word (except the null words e_0 and f_0) participates in exactly one link.

The second constraint, known as the **cohesion constraint** (Fox, 2002), uses the dependency tree (Mel'čuk, 1987) of the English sentence to restrict possible link combinations. Given the dependency tree T_E and a (partial) alignment A, the cohesion constraint requires that phrasal cohesion is maintained in the French sentence. If two phrases are disjoint in the English sentence, the alignment must not map them to overlapping intervals in the French sentence. This notion of phrasal constraints on alignments need not be restricted to phrases determined from a dependency structure. However, the experiments conducted in (Fox, 2002) indicate that dependency trees demonstrate a higher degree of phrasal cohesion during translation than other structures.

Consider the partial alignment in Figure 1. The most probable lexical match for the English word *to* is the French word *à*. When the system attempts to link *to* and *à*, the distinct English phrases [*the reboot*] and [*the host to discover all the devices*] will be mapped to intervals in the French sentence, creating the induced phrasal intervals [*à* . . . [*réinitialisation*] . . . *périphériques*]. Regardless of what these French phrases will be after the alignment is completed, we know now that their intervals will overlap. Therefore, this link will not be added to the partial alignment.

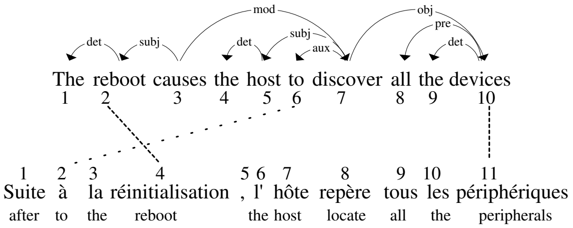
mod
obj
pre
subj
det
subj
det
aux
det
The reboot causes the host to discover all the devices
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10 11
Suite à la réinitialisation , l' hôte repère tous les périphériques
after to the reboot the host locate all the peripherals

Figure 1: An Example of Cohesion Constraint

To define this notion more formally, let $T_E(e_i)$ be the subtree of T_E rooted at e_i. The **phrase span** of e_i, *spanP*(e_i, T_E, A), is the image of the English phrase headed by e_i in F given a (partial) alignment A. More precisely, *spanP*$(e_i, T_E, A) = [k_1, k_2]$, where

$$\begin{aligned} k_1 &= \min\{j | l(u, j) \in A, e_u \in T_E(e_i)\} \\ k_2 &= \max\{j | l(u, j) \in A, e_u \in T_E(e_i)\} \end{aligned}$$

The **head span** is the image of e_i itself. We define *spanH*$(e_i, T_E, A) = [k_1, k_2]$, where

$$\begin{aligned} k_1 &= \min\{j | l(i, j) \in A\} \\ k_2 &= \max\{j | l(i, j) \in A\} \end{aligned}$$

In Figure 1, for the node *reboot*, the phrase span is [4,4] and the head span is also [4,4]; for the node *discover* (with the link between *to* and *à* in place), the phrase span is [2,11] and the head span is the empty set $\emptyset$.

With these definitions of phrase and head spans, we define two notions of overlap, originally introduced in (Fox, 2002) as **crossings**. Given a head node e_h and its modifier e_m, a **head-modifier overlap** occurs when:

$$\textit{spanH}(e_h, T_E, A) \cap \textit{spanP}(e_m, T_E, A) \neq \emptyset$$

Given two nodes e_{m_1} and e_{m_2} which both modify the same head node, a **modifier-modifier overlap** occurs when:

$$\textit{spanP}(e_{m_1}, T_E, A) \cap \textit{spanP}(e_{m_2}, T_E, A) \neq \emptyset$$

Following (Fox, 2002), we say an alignment is cohesive with respect to T_E if it does not introduce any head-modifier or modifier-modifier overlaps. For example, the alignment A in Figure 1 is not cohesive because $spanP(reboot, T_E, A) = [4, 4]$ intersects $spanP(discover, T_E, A) = [2, 11]$. Since both *reboot* and *discover* modify *causes*, this creates a modifier-modifier overlap. One can check for constraint violations inexpensively by incrementally updating the various spans as new links are added to the partial alignment, and checking for overlap after each modification. More details on the cohesion constraint can be found in (Lin and Cherry, 2003).

3 Probability Model

We define the word alignment problem as finding the alignment A that maximizes $P(A|E, F)$. ProAlign models $P(A|E, F)$ directly, using a different decomposition of terms than the model used by IBM (Brown et al., 1993). In the IBM models of translation, alignments exist as artifacts of a stochastic process, where the words in the English sentence generate the words in the French sentence. Our model does not assume that one sentence generates the other. Instead it takes both sentences as given, and uses the sentences to determine an alignment. An alignment A consists of t links $\{l_1, l_2, \ldots, l_t\}$, where each $l_k = l(e_{i_k}, f_{j_k})$ for some i_k and j_k. We will refer to consecutive subsets of A as $l_i^j = \{l_i, l_{i+1}, \ldots, l_j\}$. Given this notation, $P(A|E, F)$ can be decomposed as follows:

$$P(A|E, F) = P(l_1^t|E, F) = \prod_{k=1}^{t} P(l_k|E, F, l_1^{k-1})$$

At this point, we factor $P(l_k|E, F, l_1^{k-1})$ to make computation feasible. Let $C_k = \{E, F, l_1^{k-1}\}$ represent the context of l_k. Note that both the context C_k and the link l_k imply the occurrence of e_{i_k} and f_{j_k}. We can rewrite $P(l_k|C_k)$ as:

$$P(l_k|C_k) = \frac{P(l_k, C_k)}{P(C_k)} = \frac{P(C_k|l_k)P(l_k)}{P(C_k, e_{i_k}, f_{j_k})}$$

$$= P(l_k|e_{i_k}, f_{j_k}) \times \frac{P(C_k|l_k)}{P(C_k|e_{i_k}, f_{j_k})}$$

Here $P(l_k|e_{i_k}, f_{j_k})$ is link probability given a co-occurrence of the two words, which is similar in spirit to Melamed's explicit noise model (Melamed, 2000). This term depends only on the words involved directly in the link. The ratio $\frac{P(C_k|l_k)}{P(C_k|e_{i_k}, f_{j_k})}$ modifies the link probability, providing context-sensitive information.

C_k remains too broad to deal with in practical systems. We will consider only a subset FT_k of relevant features of C_k. We will make the Naïve Bayes-style assumption that these features $ft \in FT_k$ are conditionally independent given either l_k or (e_{i_k}, f_{j_k}). This produces a tractable formulation for $P(A|E, F)$:

$$\prod_{k=1}^{t} \left(P(l_k|e_{i_k}, f_{j_k}) \times \prod_{ft \in FT_k} \frac{P(ft|l_k)}{P(ft|e_{i_k}, f_{j_k})} \right)$$

More details on the probability model used by ProAlign are available in (Cherry and Lin, 2003).

3.1 Features used in the shared task

For the purposes of the shared task, we use two feature types. Each type could have any number of instantiations for any number of contexts. Note that each feature type is described in terms of the context surrounding a word pair.

The first feature type ft_a concerns surrounding links. It has been observed that words close to each other in the source language tend to remain close to each other in the translation (S. Vogel and Tillmann, 1996). To capture this notion, for any word pair (e_i, f_j), if a link $l(e_{i'}, f_{j'})$ exists within a window of two words (where $i-2 \leq i' \leq i+2$ and $j-2 \leq j' \leq j+2$), then we say that the feature $ft_a(i-i', j-j', e_{i'})$ is active for this context. We refer to these as **adjacency features**.

The second feature type ft_d uses the English parse tree to capture regularities among grammatical relations between languages. For example, when dealing with French and English, the location of the determiner with respect to its governor is never swapped during translation, while the location of adjectives is swapped frequently. For any word pair (e_i, f_j), let $e_{i'}$ be the governor of e_i, and let rel be the relationship between them. If a link $l(e_{i'}, f_{j'})$ exists, then we say that the feature $ft_d(j-j', rel)$ is active for this context. We refer to these as **dependency features**.

Take for example Figure 2 which shows a partial alignment with all links completed except for those involving the. Given this sentence pair and English parse tree, we can extract features of both types to assist in the alignment of the_1. The word pair (the_1, l') will have an active adjacency feature $ft_a(+1, +1, host)$ as well as a dependency feature $ft_d(-1, det)$. These two features will work together to increase the probability of this correct link.

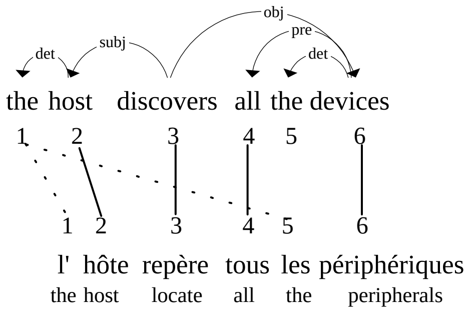
obj
pre
subj
det
det
the host discovers all the devices
1 2 3 4 5 6
1 2 3 4 5 6
l' hôte repère tous les périphériques
the host locate all the peripherals

Figure 2: Feature Extraction Example

In contrast, the incorrect link (the_1, les) will have only $ft_d(+3, det)$, which will work to lower the link probability, since most determiners are located before their governors.

3.2 Training the model

Since we always work from a current alignment, training the model is a simple matter of counting events in the current alignment. Link probability is the number of time two words are linked, divided by the number of times they co-occur. The various feature probabilities can be calculated by also counting the number of times a feature occurs in the context of a linked pair of words, and the number of times the feature is active for co-occurrences of the same word pair.

Considering only a single, potentially noisy alignment for a given sentence pair can result in reinforcing errors present in the current alignment during training. To avoid this problem, we sample from a space of probable alignments, as is done in IBM models 3 and above (Brown et al., 1993), and weight counts based on the likelihood of each alignment sampled under the current probability model. To further reduce the impact of rare, and potentially incorrect events, we also smooth our probabilities using m-estimate smoothing (Mitchell, 1997).

4 Multiple Alignments

The result of the constrained alignment search is a high-precision, word-to-word alignment. We then relax the word-to-word constraint, and use statistics regarding collocations with unaligned words in order to make many-to-one alignments. We also employ a further relaxed linking process to catch some cases where the cohesion constraint ruled out otherwise good alignments. These auxiliary methods are currently not integrated into our search or our probability model, although that is certainly a direction for future work.

5 Conclusions

We have presented a brief overview of the major ideas behind our entry to the WPT'03 Shared Task. Primary among these ideas are the use of a cohesion constraint in search, and our novel probability model.

Acknowledgments

This project is funded by and jointly undertaken with Sun Microsystems, Inc. We wish to thank Finola Brady, Bob Kuhns and Michael McHugh for their help. We also wish to thank the WPT'03 reviewers for their helpful comments.

References

P. F. Brown, V. S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312.

Colin Cherry and Dekang Lin. 2003. A probability model to improve word alignment. Submitted.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 304–311.

W.A. Gale and K.W. Church. 1991. Identifying word correspondences in parallel texts. In *4th Speech and Natural Language Workshop*, pages 152–157. DARPA, Morgan Kaufmann.

Dekang Lin and Colin Cherry. 2003. Word alignment with cohesion constraint. Submitted.

I. Dan Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, June.

Igor A. Mel'čuk. 1987. *Dependency syntax: theory and practice*. State University of New York Press, Albany.

Tom Mitchell. 1997. *Machine Learning*. McGraw Hill.

H. Ney S. Vogel and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *16th International Conference on Computational Linguistics*, pages 836–841, Copenhagen, Denmark, August.