Experiments in discriminating phrase-based translations on the basis of syntactic coupling features

Vassilina Nikoulina and Marc Dymetman
Xerox Research Centre Europe
Grenoble, France
{nikoulina,dymetman}@xrce.xerox.com

Abstract

We describe experiments on discriminating English to French phrase-based translations through the use of syntactic "coupling" features. Using a robust rule-based dependency parser, we parse both the English source and the French translation candidates from the n-best list returned by our phrase-based system; we compute for each candidate a number of coupling features, that is, values that depend on the amount of alignment between edges in the source and target structures, and discriminatively train the weights of these coupling features. We compare different feature combinations. Although the improvements in terms of automatic measures such as Bleu and Nist are inconclusive, an initial human assessment of the results appears to show certain qualitative improvements.

1 Introduction

1.1 Motivation

When we use the phrase-based SMT system MATRAX (Simard et al., 2005) to translate the sentence *Our declaration of rights is the first of this millenium* from English to French, the result returned by the system is the erroneous translation *Notre déclaration des droits de la première est de ce millénaire*, while somewhere down the n-best list of lesser-scored candidates we find a correct translation: *Notre déclaration des droits est la première de ce millénaire*.

On closer inspection, the difference of scores between the two candidates is the following. In the second (correct) case, the phrase *of rights* was translated into the phrase *des droits*, while in the first (incorrect) case, the phrase *of* was translated into the phrase *de* and the phrase *rights* into the phrase *des droits*. However, while the two bi-phrases *of/de* and *rights/des droits* independently make perfect sense, the sequence *de des droits* in French is not possible, a situation which is easily detected by a standard ngram language model; the language model has then a tendency to try to place the (in fact superfluous) *de* at a further place in the target (just before *la première*), where it is more acceptable to it. The overall consequence is a translation that while formally possible from the viewpoint of a simple language model, is not an adequate representation of the meaning of the source.

Now suppose that we parse both the source and the two candidates with a dependency parser. If we compare the parses of the source and of the correct translation, we find a close (in the current example, very close) isomorphism between dependency edges connecting pairs of aligned words (s_1, s_2) and (t_1, t_2), where s_i is aligned to t_i: the presence of an edge between s_1 and s_2 often implies that of an edge between t_1 and t_2. This is less the case if we compare the parses of the source and of the incorrect translation; in this case, the word *première* is now linked to *droits*, while the word *first* was linked to *millenium*.

While this is of course just one example, it does help to motivate the approach we have taken: we compute different measures of association strength between edges in the source and target dependency trees, and use these measures as features for rerank-

Proceedings of the Second ACL Workshop on Syntax and Structure in Statistical Translation (SSST-2), pages 55–60,
ACL-08: HLT, Columbus, Ohio, USA, June 2008. ©2008 Association for Computational Linguistics

ing the n-best candidates of a baseline phrase-based system. The hope is that by doing so, we will increase the adequacy of translations, and possibly to some extent, their fluency (at least their "semantic" fluency, which is influenced by their adequacy, as opposed to their "grammatical" fluency, which would be better addressed by target-specific syntactic features than by coupling syntactic features).

1.2 Related Work

There is a growing body of work on the use of syntax for improving statistical machine translation, from approaches such as (Chiang, 2007) that use "formal syntax", that is syntactic structures for the source and target that are discovered on the basis of a bilingual corpus, but without resort to an externally motivated parser, to approaches such as (Yamada and Knight, 2001) and (Marcu et al., 2006) that use an external parser on the target only, or such as (Quirk et al., 2005) on the source only, or such as (Cowan et al., 2006) that use external parsers both on the source and on the target.

Our approach is in this last category, but is distinguished from all the cited approaches by the fact that it does not try to build a target structure (or string) directly, but rather by using a baseline *phrase-based system* as a generator of candidates, and then selecting between these candidates through a discriminative procedure. Some other researchers have taken a similar line, for example (Hasan et al., 2006), which only uses a parser on the target, and attempts to improve the fluency of the translation produced, and especially (Och et al., 2003) that reports experiments using a large number of syntactic features. In one of the experiments briefly reported, a dependency parser is used both for the source and for the target and a few features are introduced for counting the number of edges that project from the source to the target. This experiment, which as far as we know was not followed up by deeper investigations, is very similar to what we do. However we introduce and compare results for a wider variety of coupling features, taking into account different combinations involving normalization of the counts, symmetrized features between the source and target, labelled dependencies, and also consider several ways for computing the word alignment on the basis of which edge couplings are determined.

2 The approach

2.1 Background

Matrax. The phrase-based SMT system Matrax (Simard et al., 2005), developed at Xerox, was used in the experiments. Matrax is based on a fairly standard log-linear model, but one original aspect of the system is the use of non-contiguous biphrases. Most existing phrase-based models depend on phrases that are sequences of contiguous words on either the source or the target side (e.g. *prendre feu* / *catch fire*). By contrast, Matrax considers pairs of non-contiguous phrases, such as *ne ... plus* / *not ... anymore*, where words in the source and target phrases may be separated by gaps, to be filled at translation time by lexical material provided by some other such pairs. One motivation behind this approach is that, basically, the fact that the source expression *ne ... plus* is a good predictor of *not ... anymore* does not depend on the lexical material appearing inside the source expression, an insight which is generally unexploitable by models based on contiguous phrases.[1]

XIP. For parsing, we used the *Xerox Incremental Parser* XIP (Aït-Mokhtar et al., 2002), which is a robust dependency parser developed at the Xerox Research Centre Europe. XIP is fast (around 2000 words per second for English) and is well adapted to a situation, like the one we have here, were we need to parse on the order of a few hundred target candidates on the fly. Also of interest to us is the fact that XIP produces labelled dependencies, a feature that we use in some of our experiments.

2.2 Decoding and Training

Coupling features such as the ones we use require access to the parses of candidate translations, and these parses, at least for a parser such as XIP (and for many similar parsers), can only be obtained once the complete candidate translation is known. This is why it is difficult to introduce them internally in the Matrax stack-based decoder, which would require to provide partial parses for *prefixes* of the target candidates and also associated heuristics to estimate the syntactic structure of completions of these prefixes.

[1] The Hiero system (Chiang, 2007) is a well-known instance of a structure-oriented system that also has a notion of gapped phrases, but contrary to Hiero, Matrax is based on non-hierarchical phrases.

Instead, we resort to a standard reranking approach in which we produce an n-best list of Matrax candidate translations (with n = 100 in our experiments), and then rerank this list with a linear combination of our parse-dependent features. In order to train the feature weights, we use an averaged structured perceptron approach à la Collins, where we try to learn weights such that the first candidate to emerge is equal to the "oracle" candidate, that is, the candidate that is closest to the reference in terms of NIST score.

2.3 Coupling Features

Our general approach to computing coupling features between the dependency structure of the source and that of a candidate translation produced by Matrax is the following: we start by aligning the words between the source and the candidate translation, we parse both sides, and we count (possibly according to a weighting scheme) the number of configurations ("rectangles") that are of the following type: $((s_1, s_{12}, s_2), (t_1, t_{12}, t_2))$, where s_{12} is an edge between s_1 and s_2, t_{12} is an edge between t_1 and t_2, s_1 is aligned with t_1 and s_2 is aligned with t_2. We implemented several variants of this basic scheme.

We start by describing different "generic" coupling functions derived from the basic scheme, assuming that word alignments have been already determined, then we describe the option of taking into account specific dependency labels when counting rectangles, and finally we describe two options for computing the word alignments.

2.3.1 Generic features

The first measure of coupling is based on simple, non-weighted, word alignments. Here we simply consider that a word of the source and a word of the target are aligned or not aligned, without any intermediary degree, and consider that a rectangle exists on the quadruple of words s_1, s_2, t_1, t_2 iff s_i is aligned to t_i, s_1 and s_2 have a dependency link between them (in whatever direction) and similarly for t_1 and t_2. The first feature that we introduce, *Coupling-Count*, is simply the count of all such rectangles between the source and the target.

We note that the value of this feature tends to be correlated with the size of the source and target dependency trees. We therefore introduce some normalized variants of the feature:

- *Coupling-Recall*. We compute the number of source edges for which there exists a projection in the target. More formally, the number of edges between two words s_1, s_2 such that there exist two words t_1, t_2 with s_i aligned to t_i and such that t_1, t_2 have an edge between them. We then divide this number by the total number of edges in the source.
- *Coupling-Precision*. We do the same thing this time starting from the target.
- *Coupling-F-measure*. In the case of perfectly isomorphic dependency trees (a situation that of course rarely occurs because of the linguistic divergences between languages), we would have precision and recall both equal to 1. In order to measure divergence from this ideal case, we introduce a feature that we call *Coupling-F-measure*, which is defined as the harmonic mean of the two previous features.

One deficiency of the previous measures is that they rely a lot on "hard" word alignments, but do not take into account the probability of aligning a source and a target word. We introduce another feature *Coupling-Lex* that exploits lexical translation probabilities: each rectangle found between the source and target trees is weighted according to the product of the translation probabilities associated with (s_1, t_1) and (s_2, t_2).

2.3.2 Label-specific features

The features previously defined do not take into account the labels associated with edges in the dependency trees. However, while rectangles of the form $((s_1, \text{subj}, s_2), (t_1, \text{subj}, t_2))$ may be rather systematic between such languages as English and French, other rectangles may be much less so, due on the one hand to actual linguistic divergences between the two languages, but also, as importantly in practice, to different representational conventions used by different grammar developers for the two languages.[2]

In order to control this problem, we introduce a collection of *Label-Specific-Coupling* features, each for a specific pair of source label and target label.

[2] Although the XIP formalism is shared between grammar developers of French and English, the grammars do sometimes follow slightly different conventions.

The values of a label-specific feature are the number of occurrences for this specific label pair. We use only label pairs that have been observed to be aligned in the training corpus (that is, that participate in observed rectangles). In one version of that approach, we use all such pairs found in the corpus, in another version only the pairs above a certain frequency threshold in the corpus.

2.3.3 Giza-based alignment

In order to compute the features described above, a prerequisite is to be able to determine a word alignment between the source and a candidate translation. Our first approach is to use GIZA++ to create these alignments, by producing for a given source and a given candidate translation n-best alignment lists in both directions and applying standard techniques of symmetrization to produce a bidirectional alignment.

2.3.4 Phrase-based alignment

Another way to find word alignments is to use the information provided by our baseline system. Since Matrax is a phrase-based system, it has access to the bi-phrases (aligned by definition) that are used in order to generate a candidate translation. However note that if we use the bi-phrases directly we are not able to establish the alignments on a word level (since Matrax does not provide any information about word alignments inside the bi-phrases), but only on a phrase level, and we need to adapt the coupling features accordingly.

To overcome this problem, we will transform the dependencies between words into dependencies between phrases. Thus, two phrases c_1, c_2 will have a dependency edge between them if there exists a dependency edge between a word $w_1 \in c_1$ and a word $w_2 \in c_2$. Once this transformation is done both for the source and the target, we get dependency graphs having phrases as nodes. We also know the alignments between these phrases, implicit in the bi-phrases used by Matrax. So, we can consider the phrases as *super-words*, and introduce coupling features of the same type as before, but operating on a higher level (super-words) this time.

3 Experiments

3.1 Description

For all our experiments we use the training, development and test sets provided for the English-French News Commentary corpus in WMT-08. The number of sentences in these sets are respectively 55039, 1057 and 1064, and the average sentence length is 21 words (English) and 24.5 words (French).

We take Matrax as the baseline system. With this system we generate 100-best lists of candidate translations for all source sentences of the test set, we rerank these candidates using our features, and we output the top candidate. We present our results in Table 1, distinguished according to the actual combination of features used in each experiment.

- The *Baseline* entry in the table corresponds to Matrax results on the test set, without the use of any of the coupling features.

- We distinguish two sub-tables, according to whether Giza-based alignments or phrase-based alignments were used.

- The *Generic* keyword corresponds to the coupling features introduced in section 2.3.1, based on rectangle counts, independent of the labels of the edges.

- The *Matrax* keyword corresponds to using Matrax “internal” features as reranking features, along with the coupling features. These Matrax features are pretty standard phrase-based features, apart from some features dealing explicitly with gapped phrases, and are described in detail in (Simard et al., 2005).

- The *Labels* and *Frequent Labels* keywords corresponds to using label-specific features. In the first case (Labels) we extracted all of the aligned label pairs (label pair associated with a coupling rectangle) found in a training set of 1000 source sentences along with their 100-best Matrax translations (this set was chosen to be different from the development set in order to avoid overfitting effects when reranking on the development set); we then obtained 2053 features of this kind. In the second case

	NIST	**BLEU**	**-**	**+**	**Diff**
Baseline	6.4093	0.2034	0	0	0
Giza-based alignments					
Generic	6.3383	0.2043	15	17	2
Generic, Matrax	*6.3782*	*0.2083*	*4*	*18*	*14*
Labels	6.3483	0.1963	12	18	6
Labels, Generic	6.3514	0.2010	3	18	15
Labels, Generic, Matrax	*6.4016*	*0.2075*	*3*	*20*	*17*
Frequent Labels	6.3815	0.2054	7	11	4
Frequent Labels, Generic	6.3826	0.2044	6	18	12
Frequent Labels, Generic, Matrax	*6.4177*	*0.2100*	*2*	*16*	*14*
Phrase-based alignments					
Generic	6.2869	0.1964	12	14	2
Generic, Matrax	*6.3972*	*0.2031*	*4*	*11*	*7*
Labels	6.3677	0.1995	16	15	-1
Labels, Generic	6.3567	0.1977	8	15	7
Labels, Generic, Matrax	*6.4269*	*0.2049*	*4*	*17*	*13*
Frequent Labels	6.3701	0.1998	3	15	12
Frequent Labels, Generic	6.3846	0.2013	7	16	9
Frequent Labels, Generic, Matrax	*6.4160*	*0.2049*	*4*	*16*	*12*
Giza Generic, Phrase Generic, Giza Labels, Matrax	*6.4351*	*0.2060*	*7*	*22*	*15*

Table 1: Reranking results.

(Frequent Labels), we only kept the most frequently observed among these label pairs, retaining only 137 such features.

- When several keywords appear on a line, we used the union of the corresponding features, and in the last line of the table, we show a combination involving at the same time some features computed on the basis of Giza-based alignments and of phrase-based alignments.

- Along with the NIST and BLEU scores of each combination[3], we also conducted an informal manual assessment of the quality of the results relative to the Matrax baseline. We took a random sample of 100 source sentences from the test set and for each sentence, assessed whether the first candidate produced by reranking was better, worse, or indistinguishable in terms of quality relative to the baseline translation. We report the number of improvements (+) and deteriorations (-) among these 100 samples as well as their difference.

3.2 Discussion of the results

While the overall results in terms of Bleu and Nist do not show major improvements relative to the baseline, there are several interesting observations to make. First of all, if we focus on feature combinations in which Matrax features are included (shown in italics in the table), we see that there is a general tendency for the results, both in terms of automatic and human evaluations, to be better than for the same combination without the Matrax features; the explanation seems to be that if we do not use the Matrax features during reranking, but consider the 100 candidates in the n-best list to be equally valuable from the viewpoint of Matrax features, we lose essential information that cannot be recovered simply by appeal to the syntactic coupling features.[4]

If we now concentrate on the lines which do include Matrax features and compare their results with the baseline, we see a trend for these results to be better than the baseline, both in terms of automatic measures as (more strongly) in terms of human eval-

[3] These scores were computed on the basis of only one reference.

[4] This is not very surprising and probably on the basis of this observation it would be useful in further experiments to introduce as an additional feature the log-linear score given by the Matrax baseline.

uation. Taken individually, perhaps the improvements are not very clear, but *collectively*, a trend does seem to appear in favor of syntactic coupling features generally, although we have not conducted formal statistical tests to validate this impression. A more detailed comparison between individual lines, inside the class of combinations that include Matrax features, appears however difficult to make on the basis of the current experiments.

4 Conclusion and Perspectives

Although there is some consensus that the future of statistical machine translation lies in the use of structural information, it is generally admitted that it is currently difficult to significantly improve over phrase-base systems in this way, at least in terms of automatic evaluation measures. Our results do not contradict that impression, although they are more encouraging in terms of preliminary human assesments than in terms of the automatic measures.

The reranking approach to using syntactic features on top of a phrase-based system is attractive because on the one hand it is easier to implement than a full new syntax-aware decoder, and on the other hand it guarantees at least as good performance as the baseline phrase-based system, if some precautions are taken. On the other hand, its main limitations concern the size of the n-best list of candidates that is realistic in terms of decoding time.[5] At least two approaches seem promising in order to alleviate this problem: (1) find a way to capitalize on the factorization of translation candidates in the internal lattice used by the phrase-based decoder, in order to produce factorized parses that would permit comparison between more candidates than can be seen through a final n-best list; (2) allow the reranker to perform local transformations of the n-best candidates, in the spirit of (Langlais et al., 2007), in order to be able to explore a larger space of promising candidates than is provided by the static list.

Another interesting direction would be to learn the feature weights by reranking towards another type of oracle than the one we used, which is defined as the closest candidate in the list in terms of NIST score relative to the reference; instead it might be worthwhile to use as an oracle the candidate in the list which receives the best human assessment in terms of fluency and adequacy, giving a better chance to the syntactic features to show their worth; but this would probably also require that these systems be mostly evaluated in terms of human assessment, a trend which is more and more noticeable in the SMT community.

[5] It should be noted however that we could increase this size from 100 to 1000 without incurring too much penalty, given the speed of the XIP parser we use.

References

Salah Aït-Mokhtar, Jean-Pierre Chanod, and Claude Roux. 2002. Robustness beyond shallowness: incremental deep parsing. *Natural Language Engineering*, 8(3):121–144.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Brooke Cowan, Ivona Kucerova, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. In *Proceedings EMNLP*.

Saša Hasan, Oliver Bender, and Hermann Ney. 2006. Reranking translation hypotheses using structural properties. In *Proceedings of the EACL Workshop on Learning Structured Information in Natural Language Applications*.

Philippe Langlais, Alexandre Patry, and Fabrizio Gotti. 2007. A greedy decoder for phrase-based statistical machine translation. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 104–113, Skvde, Sweden, Sept.

D. Marcu, W. Wang, A. Echihabi, and K. Knight. 2006. SPMT: Statistical Machine Translation with Syntactified Target Language Phrases. In *Proceedings EMNLP*, pages 44–52.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2003. Syntax for statistical machine translation: Final report of john hopkins 2003 summer workshop. Technical report, John Hopkins University.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *ACL05*.

Michel Simard, Nicola Cancedda, Bruno Cavestro, Marc Dymetman, Éric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlais, and Arne Mauser. 2005. Translating with non-contiguous phrases. In *HLT/EMNLP*.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings ACL*, pages 531–538.