Microsoft Research Treelet Translation System: NAACL 2006 Europarl Evaluation

Arul Menezes, Kristina Toutanova and Chris Quirk

Microsoft Research
One Microsoft Way
Redmond, WA 98052
{arulm,kristout,chrisq}@microsoft.com

Abstract

The Microsoft Research translation system is a syntactically informed phrasal SMT system that uses a phrase translation model based on dependency treelets and a global reordering model based on the source dependency tree. These models are combined with several other knowledge sources in a log-linear manner. The weights of the individual components in the log-linear model are set by an automatic parameter-tuning method. We give a brief overview of the components of the system and discuss our experience with the Europarl data translating from English to Spanish.

1. Introduction

The dependency treelet translation system developed at MSR is a statistical MT system that takes advantage of linguistic tools, namely a source language dependency parser, as well as a word alignment component. [1]

To train a translation system, we require a sentence-aligned parallel corpus. First the source side is parsed to obtain dependency trees. Next the corpus is word-aligned, and the source dependencies are projected onto the target sentences using the word alignments. From the aligned dependency corpus we extract all treelet translation pairs, and train an order model and a bi-lexical dependency model.

To translate, we parse the input sentence, and employ a decoder to find a combination and ordering of treelet translation pairs that cover the source tree and are optimal according to a set of models. In a now-common generalization of the classic noisy-channel framework, we use a log-linear combination of models [2], as in below:

\[ \text{translation}(S, F, \Lambda) = \arg\max_T \left[ \sum_{f \in F} \lambda f(S, T) \right] \]

Such an approach toward translation scoring has proven very effective in practice, as it allows a translation system to incorporate information from a variety of probabilistic or non-probabilistic sources. The weights \( \Lambda = \{ \lambda_f \} \) are selected by discriminatively training against held out data.

2. System Details

A brief word on notation: \( s \) and \( t \) represent source and target lexical nodes; \( S \) and \( T \) represent source and target trees; \( s \) and \( t \) represent source and target treelets (connected subgraphs of the dependency tree). The expression \( \forall f \in T \) refers to all the lexical items in the target language tree \( T \) and \( |T| \) refers to the count of lexical items in \( T \). We use subscripts to indicate selected words: \( T_n \) represents the \( n \) lexical item in an in-order traversal of \( T \).

2.1. Training

We use the broad coverage dependency parser NLPIPE [3] to obtain source language dependency trees, and we use GIZA++ [4] to produce word alignments. The GIZA++ training regimen and parameters are tuned to optimize BLEU [5] scores on held-out data. Using the word alignments, we follow a set of dependency tree projection heuristics [1] to construct target dependency trees, producing a word-aligned parallel dependency tree corpus. Treelet translation pairs are extracted by enumerating all source treelets (to a maximum size) aligned to a target treelet.

2.2. Decoding

We use a tree-based decoder, inspired by dynamic programming. It searches for an approximation of
the n-best translations of each subtree of the input
dependency tree. Translation candidates are
composed from treelet translation pairs extracted
from the training corpus. This process is described
in more detail in [1].

2.3. Models

2.3.1. Channel models

We employ several channel models: a direct
maximum likelihood estimate of the probability of
target given source, as well as an estimate of
source given target and target given source using
the word-based IBM Model 1 [6]. For MLE, we
use absolute discounting to smooth the
probabilities:

\[
P_{\text{MLE}}(t|s) = \frac{c(s,t) - \lambda}{c(s,*)}
\]

Here, \( c \) represents the count of instances of the
treelet pair \((s, t)\) in the training corpus, and \( \lambda \)
determined empirically.

For Model 1 probabilities we compute the sum
over all possible alignments of the treelet without
normalizing for length. The calculation of source
given target is presented below; target given
source is calculated symmetrically.

\[
P_{\text{MI}}(t|s) = \prod_{t \in T} \sum_{s \in S} P(t|s)
\]

2.3.2. Bilingual n-gram channel models

Traditional phrasal SMT systems are beset by a
number of theoretical problems, such as the ad hoc
estimation of phrasal probability, the failure to
model the partition probability, and the tenuous
connection between the phrases and the
underlying word-based alignment model. In
string-based SMT systems, these problems are
outweighed by the key role played by phrases in
capturing “local” order. In the absence of good
global ordering models, this has led to an
inexorable push towards longer and longer
phrases, resulting in serious practical problems of
scale, without, in the end, obviating the need for a
real global ordering story.

In [13] we discuss these issues in greater detail
and also present our approach to this problem.
Briefly, we take as our basic unit the Minimal
Translation Unit (MTU) which we define as a set
of source and target word pairs such that there are
no word alignment links between distinct MTUs,
and no smaller MTUs can be extracted without
violating the previous constraint. In other words,
these are the minimal non-compositional phrases.
We then build models based on n-grams of MTUs
in source string, target string and source
dependency tree order. These bilingual n-gram
models in combination with our global ordering
model allow us to use shorter phrases without any
loss in quality, or alternately to improve quality
while keeping phrase size constant.

As an example, consider the aligned sentence
pair in Figure 1. There are seven MTUs:

\[
m_1 = \text{<we should / hemos>}
m_2 = \text{<NULL / de>}
m_3 = \text{<follow / cumplir>}
m_4 = \text{<the / el>}
m_5 = \text{<Rio / Rio>}
m_6 = \text{<agenda / programa>}
m_7 = \text{<NULL / de>}
\]

We can then predict the probability of each MTU
in the context of (a) the previous MTUs in source
order, (b) the previous MTUs in target order, or
(c) the ancestor MTUs in the tree. We consider all
of these traversal orders, each acting as a separate
feature function in the log linear combination. For
source and target traversal order we use a trigram
model, and a bigram model for tree order.

2.3.3. Target language models

We use both a surface level trigram language
model and a dependency-based bigram language
model [7], similar to the bilexical dependency
modes used in some English Treebank parsers
(e.g. [8]).

\[
P_{\text{surf}}(T) = \prod_{i=1}^{|T|} P_{\text{trisurf}}(T_i|T_{i-2},T_{i-1})
\]

\[
P_{\text{bilex}}(T) = \prod_{i=1}^{|T|} P_{\text{bidep}}(T_i|\text{parent}(T_i))
\]

\( P_{\text{trisurf}} \) is a Kneser-Ney smoothed trigram language
model trained on the target side of the training
corpus, and \( P_{\text{bilex}} \) is a Kneser-Ney smoothed
bigram language model trained on target language dependencies extracted from the aligned parallel dependency tree corpus.

2.3.4. Order model

The order model assigns a probability to the position \( \text{pos}(t) \) of each target node relative to its head based on information in both the source and target trees:

\[
P_{\text{order}}(\text{order}(T)|S,T) = \prod_{t \in T} P(\text{pos}(t, \text{parent}(t))|S,T)
\]

Here, position is modeled in terms of closeness to the head in the dependency tree. The closest pre-modifier of a given head has position -1; the closest post-modifier has a position 1. Figure 1 shows an example dependency tree pair annotated with head-relative positions.

We use a small set of features reflecting local information in the dependency tree to model \( P(\text{pos}(t, \text{parent}(t))|S,T) \):

- Lexical items of \( t \) and \( \text{parent}(t) \), the parent of \( t \) in the dependency tree.
- Lexical items of the source nodes aligned to \( t \) and \( \text{head}(t) \).
- Part-of-speech ("cat") of the source nodes aligned to the head and modifier.
- Head-relative position of the source node aligned to the source modifier.

These features along with the target feature are gathered from the word-aligned parallel dependency tree corpus and used to train a statistical model. In previous versions of the system, we trained a decision tree model [9]. In the current version, we explored log-linear models. In addition to providing a different way of combining information from multiple features, log-linear models allow us to model the similarity among different classes (target positions), which is advantageous for our task.

We implemented a method for automatic selection of features and feature conjunctions in the log-linear model. The method greedily selects feature conjunction templates that maximize the accuracy on a development set. Our feature selection study showed that the part-of-speech labels of the source nodes aligned to the head and the modifier and the head-relative position of the source node corresponding to the modifier were the most important features. It was useful to concatenate the part-of-speech of the source head with every feature. This effectively achieves learning of separate movement models for each source head category. Lexical information on the pairs of head and dependent in the source and target was also very useful.

To model the similarity among different target classes and to achieve pooling of data across similar classes, we added multiple features of the target position. These features let our model know, for example, that position -5 looks more like position -6 than like position 3. We added a feature “positive”/“negative” which is shared by all positive/negative positions. We also added a feature looking at the displacement of a position in the target from the corresponding position in the source and features which group the target positions into bins. These features of the target position are combined with features of the input.

This model was trained on the provided parallel corpus. As described in Section 2.1 we parsed the source sentences, and projected target dependencies. Each head-modifier pair in the resulting target trees constituted a training instance for the order model.

The score computed by the log-linear order model is used as a single feature in the overall log-linear combination of models (see Section 1), whose parameters were optimized using MaxBLEU [2]. This order model replaced the decision tree-based model described in [1].

We compared the decision tree model to the log-linear model on predicting the position of a modifier using reference parallel sentences, independent of the full MT system. The decision tree achieved per decision accuracy of 69% whereas the log-linear model achieved per decision accuracy of 79%. In the context of the full MT system, however, the new order model provided a more modest improvement in the BLEU score of 0.39%.

2.3.5. Other models

We include two pseudo-models that help balance certain biases inherent in our other models.

- **Treelet count**. This feature is a count of treelets used to construct the candidate. It acts as a bias toward translations that use a smaller number of treelets; hence toward larger sized treelets incorporating more context.

- **Word count**. We also include a count of the words in the target sentence. This feature

\[\text{1 The per-decision accuracy numbers were obtained on different (random) splits of training and test data.}\]
helps to offset the bias of the target language model toward shorter sentences.

3. Discussion

We participated in the English to Spanish track, using the supplied bilingual data only. We used only the target side of the bilingual corpus for the target language model, rather than the larger supplied language model. We did find that increasing the target language order from 3 to 4 had a noticeable impact on translation quality. It is likely that a larger target language corpus would have an impact, but we did not explore this.

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline treetop system</td>
<td>27.60</td>
</tr>
<tr>
<td>Add bilingual MTU models</td>
<td>28.42</td>
</tr>
<tr>
<td>Replace DT order model with log-linear model</td>
<td>28.81</td>
</tr>
</tbody>
</table>

Table 1: Results on development set

We found that the addition of bilingual n-gram based models had a substantial impact on translation quality. Adding these models raised BLEU scores about 0.8%, but anecdotal evidence suggests that human-evaluated quality rose by much more than the BLEU score difference would suggest. In general, we felt that in this corpus, due to the great diversity in translations for the same source language words and phrases, and given just one reference translation, BLEU score correlated rather poorly with human judgments. This was borne out in the human evaluation of the final test results. Humans ranked our system first and second, in-domain and out-of-domain respectively, even though it was in the middle of a field of ten systems by BLEU score. Furthermore, n-gram channel models may provide greater robustness. While our BLEU score dropped 3.61% on out-of-domain data, the average BLEU score of the other nine competing systems dropped 5.11%.

4. References


