Syntactic Models for Structural Word Insertion and Deletion

Arul Menezes and **Chris Quirk**
Microsoft Research
One Microsoft Way, Redmond, WA 98052, USA
{arulm, chrisq}@microsoft.com

Abstract

An important problem in translation neglected by most recent statistical machine translation systems is insertion and deletion of words, such as function words, motivated by linguistic structure rather than adjacent lexical context. Phrasal and hierarchical systems can only insert or delete words in the context of a larger phrase or rule. While this may suffice when translating in-domain, it performs poorly when trying to translate broad domains such as web text. Various syntactic approaches have been proposed that begin to address this problem by learning lexicalized and unlexicalized rules. Among these, the treelet approach uses unlexicalized order templates to model ordering separately from lexical choice. We introduce an extension to the latter that allows for structural word insertion and deletion, without requiring a lexical anchor, and show that it produces gains of more than 1.0% BLEU over both phrasal and baseline treelet systems on broad domain text.

1 Introduction

Among the phenomena that are modeled poorly by modern SMT systems is the insertion and deletion of words, such as function words, that are motivated by the divergent linguistic structure between source and target language. To take the simplest of examples, the English noun compound "*file name*" would typically be translated into Spanish as "*nombre de archivo*", which requires the insertion of the preposition "*de*". Conversely, when translating from Spanish to English, the "*de*" must be deleted. At first glance, the problem may seem trivial, yet the presence and position of these function words can have crucial impact on the adequacy and fluency of translation.

In particular, function words are often used to denote key semantic information. They may be used to denote case information, in languages such as Japanese. Failing to insert the proper case marker may render a sentence unreadable or significantly change its meaning. Learning these operations can be tricky for MT models best suited to contiguous word sequences. From a fluency standpoint, proper insertion of determiners and prepositions can often make the difference between laughably awkward output and natural sounding translations; consider the output "*it's a cake piece*" as opposed to "*it's a piece of cake*".

Furthermore, since missing or spurious function words can confuse the target language model, handling these words properly can have an impact beyond the words themselves.

This paper focuses on methods of inserting and deleting words based on syntactic cues, to be used in the context of a syntax-informed translation system. While the models we build are relatively simple and the underlying templates are easy to extract, they add significant generalization ability to the base translation system, and result in significant gains.

2 Background

As a motivating example, let us return to the English/Spanish pair "*file name*" and "*nombre de archivo*". In principle, we would want a machine translation system to be capable of learning the following general transformation:

$$\text{“ NOUN}_1\ \text{NOUN}_2\text{”} \rightarrow \text{“ NOUN}_2\ \textit{de}\ \text{NOUN}_1\text{”} \quad (1)$$

Yet even this simple example is beyond the capabilities of many common approaches.

The heavily lexicalized approaches of phrasal systems (Koehn et al., 2003), are inherently incapable of this generalization. As a proxy, they

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 735–744,
Honolulu, October 2008.

acquire phrase pairs such as "*nombre de archivo*" → "*file name*", "*nombre de*" → "*name*" and "*de archivo*" → "*file*". Note that the inserted word is attached to adjacent context word(s). When the test set vocabulary has significant overlap with the training vocabulary, the correct translation can often be assembled based on the head or the modifying noun. However, as we show in this paper, this is woefully inadequate when translating truly out-of-domain input.

In principle, phrase-based translation systems may employ insertion phrase pairs such as

"[NULL]" → "*de*" (2)

but the ungrounded nature of this transformation makes its use during decoding difficult. Since there are no constraints on where such a rule may apply and the rule does not consume any input words, the decoder must attempt these rules at every point in the search.

The reverse operation

"*de*" → "[NULL]" (3)

is more feasible to implement, though again, there is great ambiguity – a source word may be deleted at any point during the search, with identical target results. Few systems allow this operation in practice. Estimating the likelihood of this operation and correctly identifying the contexts in which it should occur remain challenging problems.

Hierarchical systems, such as (Chiang, 2005) in principle have the capacity to learn insertions and deletions grounded by minimal lexical cues. However, the extracted rules use a single non-terminal. Hence, to avoid explosive ambiguity, they are constrained to contain at least one aligned pair of words. This restriction successfully limits computational complexity at a cost of generalization power.

Syntax-based approaches provide fertile context for grounding insertions and deletions. Often we may draw a strong correspondence between function words in one language and syntactic constructions in another. For instance, the syntactic approach of Marcu et al. (2006) can learn unlexicalized rules that insert function words in isolation, such as:

NP(NN:x0 NN:x1) → x1 *de* x0 (4)

However, as discussed in (Wang, Knight & Marcu, 2007), joint modeling of structure and lexical choice can exacerbate data sparsity, a problem that they attempt to address by tree binarization. Nevertheless, as we show below, unlexicalized structural transformation rules such as (1) and (4) that allow for insertion of isolated function words, are essential for good quality translation of truly out-of-domain test data.

In the treelet translation approach (Menezes & Quirk, 2007), lexical choice and syntactic re-ordering are modeled separately using lexicalized treelets and unlexicalized order templates. We discuss this approach in more detail in Section 4. In Section 5, we describe how we extend this approach to allow for structural insertion and deletion, without the need for content word anchors.

3 Related Work

There is surprisingly little prior work in this area. We previously (Menezes & Quirk, 2005) explored the use of deletion operations such as (3) above, but these were not grounded in any syntactic context, and the estimation was somewhat heuristic[1].

The tuple translation model of Crego et al. (2005), a joint model over source and target translations, also provides a means of deleting words. In training, sentence pairs such as "*nombre de archivo*" / "*file name*" are first word aligned, then minimal bilingual tuples are identified, such as "*nombre* / *name*", "*de* / NULL" and "*archivo* / *file*". The tuples may involve deletion of words by allowing an empty target side, but do not allow insertion tuples with an empty source side. These inserted words are bound to an adjacent neighbor. An n-gram model is trained over the tuple sequences. As a result, deletion probabilities have the desirable property of being conditioned on adjacent context, yet this context is heavily lexicalized, therefore unlikely to generalize well.

More recently, Li et. al. (2008) describe three models for handling "single word deletion" (they discuss, but do not address, word insertion). The first model uses a fixed probability of deletion

[1] We assigned channel probabilities based on the sum of the Model1 probability of the source word being aligned to NULL or one of a list of "garbage collector" words. This exploits the property of Model1 that certain high-frequency words tend to act as "garbage collectors" for words that should remain unaligned.

P(NULL), independent of the source word, estimated by counting null alignments in the training corpus. The second model estimates a deletion probability per-word, P(NULL|w), also directly from the aligned corpus, and the third model trains an SVM to predict the probability of deletion given source language context (neighboring and dependency tree-adjacent words and parts-of-speech). All three models give large gains of 1.5% BLEU or more on Chinese-English translation. It is interesting to note that the more sophisticated models provide a relatively small improvement over the simplest model in-domain, and no benefit out-of-domain.

4 Dependency treelet translation

As a baseline, we use the treelet translation approach (which we previously described in Menezes & Quirk, 2007), a linguistically syntax-based system leveraging a source parser. It first unifies lexicalized treelets and unlexicalized templates to construct a sentence-specific set of synchronous rewrite rules. It then finds the highest scoring derivation according to a linear combination of models. We briefly review this system before describing our current extension.

4.1 The treelet translation model

Sentence-specific rewrite rules are constructed by unifying information from three sources: a dependency parse of the input sentence, a set of treelet translation pairs, and a set of unlexicalized order templates. Dependency parses are represented as trees: each node has a lexical label and a part of speech, as well as ordered lists of pre- and post-modifiers.

A *treelet* represents a connected subgraph of a dependency tree; *treelet translation pairs* consist of source and target treelets and a node alignment. This alignment is represented by indices: each node is annotated with an integer alignment index. A source node and a target node are aligned *iff* they have the same alignment index. For instance:

$$((old_1/\text{JJ})\ man_2/\text{NN}) \rightarrow (hombre_2\ (viejo_1)) \quad (5)$$
$$(man_1/\text{NN}) \rightarrow (hombre_1) \quad (6)$$

Order templates are unlexicalized transduction rules that describe the reorderings, insertions and deletions associated with a single group of nodes that are aligned together. For instance:

$$((\text{x0}{:}\star/\text{DT})\ (\text{x1}{:}\star/\text{JJ})\ \star_1/\text{NN}) \rightarrow ((\text{x0})\ \star_1\ (\text{x1})) \quad (7)$$
$$((\text{x0}{:}\star/\text{DT})\ (\text{x1}{:}\star/\text{JJ})\ \star_1/\text{NN}) \rightarrow ((\text{x0})\ (\text{x1})\ \star_1) \quad (8)$$
$$((\text{x0}{:}\star/\text{DT})\ \star_1/\text{NN}) \rightarrow ((\text{x0})\ \star_1) \quad (9)$$
$$((\text{x0}{:}\star/\text{RB})\ \star_1/\text{JJ}) \rightarrow ((\text{x0})\ \star_1) \quad (10)$$

Each node is either a placeholder or a variable. Placeholders, such as $\star_1$/NN on the source side or $\star_1$ on the target side, have alignment indices and constraints on their parts-of-speech on the source side, but are unconstrained lexically (represented by the $\star$). These unify at translation time with lexicalized treelet nodes with matching parts-of-speech and alignment.

Variables, such as x0:$\star$/DT on the source side and x0:$\star$ on the target side, also have parts-of-speech constraints on the source side. Variables are used to indicate where rewrite rules are recursively applied to translate subtrees. Thus each variable label such as x0, must occur exactly once on each side.

In effect, a template specifies how all the children of a given source node are reordered during translation. If translation were a word-replacement task, then templates would be just simple, single-level tree transducers. However, in the presence of one-to-many and many-to-one translations and unaligned words, templates may span multiple levels in the tree.

As an example, order template (7) indicates that an NN with two pre-modifying subtrees headed by DT and JJ may be translated by using a single word translation of the NN, placing the translation of the DT subtree as a pre-modifier, and placing the translation of the JJ subtree as a post-modifier. As discussed below, this template can unify with the treelet (6) to produce the following rewrite rule:

$$((\text{x0:DT})\ (\text{x1:JJ})\ man/\text{NN}) \rightarrow ((\text{x0})\ hombre\ (\text{x1})) \quad (11)$$

Matching: A treelet translation pair matches an input parse *iff* there is a unique correspondence between the source side of the treelet pair and a connected subgraph of the input parse.

An order template matches an input parse *iff* there is a unique correspondence between the source side of the template and the input parse, with the additional restriction that all children of input nodes that correspond to placeholder

template nodes must be included in the correspondence. For instance, order template (7) matches the parse

((*the*/DT) (*young*/JJ) *colt*/NN) (12)

but not the parse

((*the*/DT) (*old*/JJ) (*grey*/JJ) *mare*/NN) (13)

Finally, an order template matches a treelet translation pair at a given node *iff*, on both source and target sides, there is a correspondence between the treelet translation nodes and template nodes that is consistent with their tree structure and alignments. Furthermore, all placeholder nodes in the template must correspond to some treelet node.

Constructing a sentence-specific rewrite rule is then a process of unifying each treelet with a matching combination of order templates with respect to an input parse. Each treelet node must be unified with one and only one order template placeholder node. Unifying under these constraints produces a rewrite rule that has a one-to-one correspondence between variables in source and target. For instance, given the input parse:

((*the*/DT) ((very/RB) *old*/JJ) *man*/NN) (14)

we can create a rewrite rule from the treelet translation pair (5) by unifying it with the order template (7), which matches at the node *man* and its descendents, and template (10), which matches at the node *old*, to produce the following sentence-specific rewrite rule:

((*the*/DT) ((x1: ⋆/RB) *old*/JJ) *man*/NN) →
((*el*) *hombre* ((x1) *viejo*)) (15)

Note that by using different combinations of order templates, a single treelet can produce multiple rewrite rules. Also, note how treelet translation pairs capture contextual lexical translations but are underspecified with respect to ordering, while order templates separately capture arbitrary reordering phenomena yet are underspecified lexically. Keeping lexical and ordering information orthogonal until runtime allows for the production of novel transduction rules never actually seen in the training corpus, leading to improved generalization power.

***Decoding*:** Given a set of sentence-specific rewrite rules, a standard beam search algorithm is used to find the highest scoring derivation. Derivations are scored according to a linear combination of models.

4.2 Training

The process of extracting treelet translation pairs and order templates begins with parallel sentences. First, the sentence pairs are word segmented on both sides, and the source language sentences are parsed. Next, the sentence pairs are word aligned and the alignments are used to project a target language dependency tree.

Treelet extraction: From each sentence pair S, T with the alignment relation $\sim$, a treelet translation pair consisting of the source treelet $\mathbf{s} \subseteq S$ and the target treelet $\mathbf{t} \subseteq T$ is extracted *iff*:

(1) There exist $s \in \mathbf{s}$ and $t \in \mathbf{t}$ such that $s \sim t$.
(2) For all $s \in S$, and $t \in T$ such that $s \sim t$, $s \in \mathbf{s}$ *iff* $t \in \mathbf{t}$.

Order template extraction is attempted starting from each node S_{root} in the source whose parent is not also aligned to the same target word(s). We identify T_{root}, the highest target node aligned to S_{root}. We initialize the sets S_0 as $\{S_{root}\}$ and T_0 as $\{T_{root}\}$. We expand S_0 to include all nodes adjacent to some element of S_0 that are (a) unaligned, or (b) aligned to some node in T_0. The converse is applied to T_0. This expansion is repeated until we reach a fixed point. Together, S_0 and T_0 make up the placeholder nodes in the extracted order template. We then create one variable in the order template for each direct child of nodes in S_0 and T_0 that is not already included in the order template. *Iff* there is a one-to-one word alignment correspondence between source and target variables, then a template is extracted. This restriction leads to clean templates, at the cost of excluding all templates involving extraposition.

5 Insertion/deletion order templates

In this paper, we extend our previous work to allow for insertion and deletion of words, by allowing unaligned lexical items as part of the otherwise unlexicalized order templates. Grounding insertions and deletions in templates rather than treelets has two major benefits. First, insertion and deletion can be performed even in the absence of specific lexical context, leading to greater generalization power. Secondly, this increased power is tempered by linguistically

informative unlexicalized context. Rather than proposing insertions and deletions in any arbitrary setting, we are guided by specific syntactic phenomena. For instance, when translating English noun compounds into Spanish, we often must include a preposition; this generalization is naturally captured using just parts-of-speech.

The inclusion of lexical items in order templates affects the translation system in only a few places: dependency tree projection, order template extraction, and rewrite rule construction at runtime.

Dependency tree projection: During this step of the baseline treelet system, unaligned words are by default attached low, to the lowest aligned neighbor. Although this worked well in conjunction with the discriminative order model, it prevents unaligned nodes from conditioning on relevant context in order templates. Therefore, we change the default attachment of unaligned nodes to be to the highest aligned neighbor; informal experiments showed that this did not noticeably impact translation quality in the baseline system. For example, consider the source parse and aligned target sentence:

$$((\textit{calibrated}_1/\text{JJ})\ (\textit{camera}_2/\text{NN})\ \textit{file}_3/\text{NN})$$
$$\textit{archivo}_3\ \textit{de}_4\ \textit{cámara}_2\ \textit{calibrado}_1 \qquad (16)$$

Using the baseline projection algorithm would produce this target dependency tree:

$$(\textit{archivo}_3\ ((\textit{de}_4)\ \textit{cámara}_2)\ (\textit{calibrado}_1)) \qquad (17)$$

Instead, we attach unaligned words high:

$$(\textit{archivo}_3\ (\textit{de}_4)\ (\textit{cámara}_2)\ (\textit{calibrado}_1)) \qquad (18)$$

Order template extraction: In addition to the purely unlexicalized templates extracted from each training sentence, we also allow templates that include lexical items for each unaligned token. For each point in the original extraction procedure, where S_0 or T_0 contain unaligned nodes, we now extract two templates: The original unlexicalized template, and a new template in which only the unaligned node(s) contain the specific lexical item(s). From the example sentence pair (16), using the projected parse (18) we would extract the following two templates:

$$((\text{x0}{:}\star/\text{JJ})\ (\text{x1}{:}\star/\text{NN})\ \star_1/\text{NN}) \rightarrow$$
$$(\star_1\ (\star_2)\ (\text{x1})\ (\text{x0})) \qquad (19)$$
$$((\text{x0}{:}\star/\text{JJ})\ (\text{x1}{:}\star/\text{NN})\ \star_1/\text{NN}) \rightarrow$$
$$(\star_1\ (\textit{de}_2)\ (\text{x1})\ (\text{x0})) \qquad (20)$$

Template matching and unification: We extend the template matching against the input parse to require that any lexicalized source template nodes match the input exactly. When matching templates to treelet translation pairs, any unaligned treelet nodes must be consistent with the corresponding template node (i.e. the template node must be unlexicalized, or the lexical items must match). On the other hand, lexicalized template nodes do not need to match any treelet nodes -- insertions or deletions may now come from the template alone.

Consider the following example input parse:

$$((\textit{digital}/\text{JJ})\ (\textit{camera}/\text{NN})$$
$$(\textit{file}/\text{NN})\ \textit{extension}/\text{NN}) \qquad (21)$$

The following treelet translation pair provides a contextual translation for some of the children, including the insertion of one necessary preposition:

$$((\textit{file}_1/\text{NN})\ \textit{extension}_2/\text{NN}) \rightarrow$$
$$(\textit{extension}_2\ (\textit{de}_3)\ (\textit{archivo}_1)) \qquad (22)$$

The following order template can provide relative ordering information between nodes as well as insert the remaining prepositions:

$$((\text{x0}{:}\star/\text{JJ})\ (\text{x1}{:}\star/\text{NN})\ (\text{x2}{:}\star/\text{NN})\ \star_1/\text{NN}) \rightarrow$$
$$(\star_1\ (\textit{de}_2)\ (\text{x2})\ (\textit{de}_3)\ (\text{x0})\ (\text{x1})) \qquad (23)$$

The unification of this template and treelet is somewhat complex: the first inserted *de* is agreed upon by both template and treelet, whereas the second is inserted by the template alone. This results in the following novel rewrite rule:

$$((\text{x0}{:}\star/\text{JJ})\ (\text{x1}{:}\ \star/\text{NN})\ (\textit{file})\ \textit{extension}) \rightarrow$$
$$(\textit{extension}\ (\textit{de})\ (\textit{archivo})\ (\textit{de})\ (\text{x0})\ (\text{x1})) \qquad (24)$$

These relatively minimal changes produce a powerful contextualized model of insertion and deletion.

Parameter estimation: The underlying treelet system includes a template probability estimated by relative frequency. We estimate our lexicalized templates in the same way. However early experiments showed that this feature alone was not enough to allow even common insertions, since the probability of even the most common insertion templates is much lower than that of unlexicalized templates. To improve the modeling capability, we included two additional feature functions: a count of structurally inserted words, and a count of structurally deleted words.

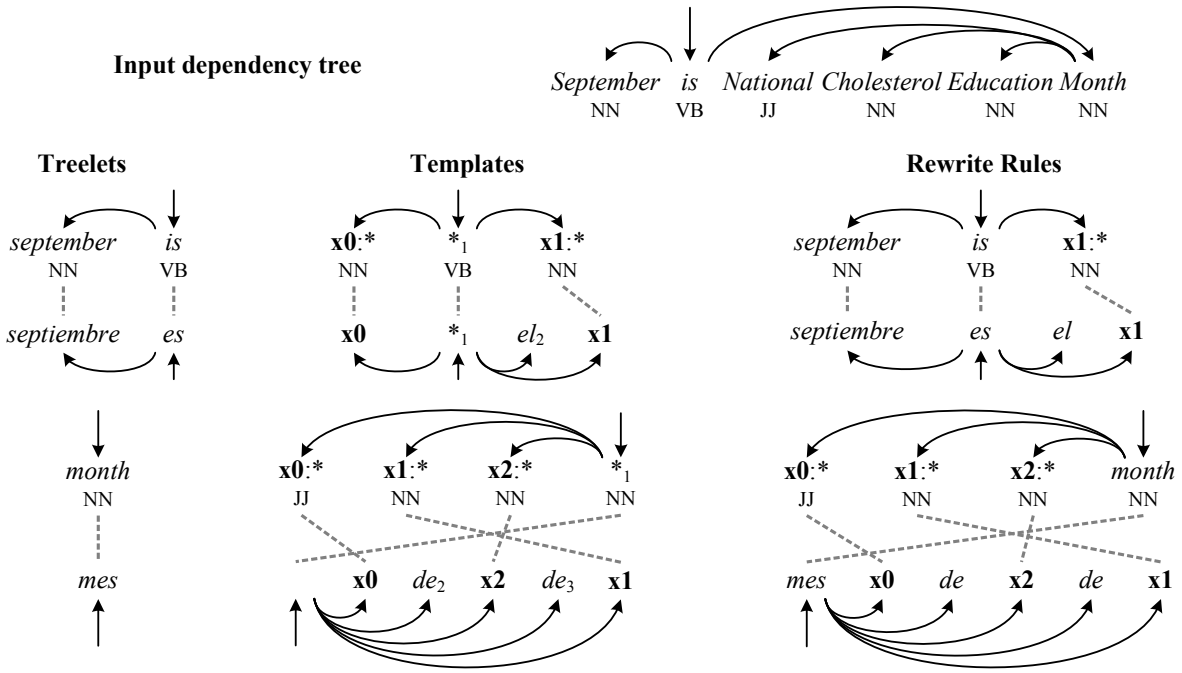

Figure 6.1: Example sentence, matching treelets, structural insertion templates and unified rewrite rules

6 Example

Consider the following English test sentence and corresponding Spanish human translation:

September is National Cholesterol Education Month

Septiembre es el Mes Nacional para la Educación sobre el Colesterol

The baseline treelet system without structural insertions translates this sentence as:

Septiembre es Nacional Colesterol Educación Mes

Not only is the translation missing the appropriate articles and prepositions, but also in their absence, it fails to reorder the content words correctly. Without the missing prepositions, the language model does not show a strong preference among various orderings of "*nacional*" "*colesterol*" "*educación*" and "*mes*".

Using structural insertion templates, the highest scoring translation of the sentence is now:

Septiembre es el Mes Nacional de Educación de colesterol

Although the choice of prepositions is not the same as the reference, the fluency is much improved and the translation is quite understandable. Figure 6.1, lists the structural insertion templates that are used to produce this translation, and shows how they are unified with treelet translation pairs to produce sentence-specific rewrite rules, which are in turn composed during decoding to produce this translation.

7 Experiments

We evaluated the translation quality of the system using the BLEU metric (Papineni et al., 2002). We compared three systems: (a) a standard phrasal system using a decoder based on Pharaoh, (Koehn et al., 2003), (b) A baseline treelet system using unlexicalized order templates and (c) The present work, which adds structural insertion and deletion templates.

7.1 Data

We report results for two language pairs, English-Spanish and English- Japanese. For English-Spanish we use two training sets: (a) the Europarl corpus provided by the NAACL 2006 Statistical Machine Translation workshop (b) a "general-domain" data set that includes a broad spectrum of data such as governmental data, general web data and technical corpora.

For English-Japanese we use only the "general-domain" data set.

	Sentence pairs	Tokens	Phr size	MERT data
Europarl E-S	730K	15M	7	Europarl
General E-S	3.7M	41M	4	Web
General E-J	2.6M	16M	4	Web

Table 7.1 Training data

For English-Spanish we report results using the four test sets listed in Table 7.2. For English-Japanese we use only the web test set. The first two tests are from the 2006 SMT workshop and the newswire test is from the 2008 workshop. The web test sets were selected from a random sampling of English web sites, with target language translations provided by professional translation vendors. All test sets have one reference translation.

	Domain	Sentence pairs
eu-test	Europarl	2000
nc-test	News commentary	1064
News	News wire	2051
Web	General web text	5000

Table 7.2 Test data

7.2 Models

The baseline treelet translation system uses all the models described in Menezes & Quirk (2007), namely:

- Treelet log probabilities, maximum likelihood estimates with absolute discounting.
- Forward and backward lexical weighting, using Model-1 translation log probabilities.
- Trigram language model using modified Kneser-Ney smoothing.
- Word and phrase count feature functions.
- Order template log probabilities, maximum likelihood estimates, absolute discounting.
- Count of artificial *source order templates.*[2]
- Discriminative tree-based order model.

The present work does not use the discriminative tree-based order model[3] but adds:

- Count of structural insertions: This counts only words inserted via templates, not lexical insertions via treelets.
- Count of structural deletions: This counts only words deleted via templates, not lexical deletions via treelets.

The comparison phrasal system was constructed using the same alignments and the heuristic combination described in (Koehn et al., 2003). This system used a standard set of models:

- Direct and inverse log probabilities, both relative frequency and lexical weighting.
- Word count, phrase count.
- Trigram language model log probability.
- Length based distortion model.
- Lexicalized reordering model.

7.3 Training

We parsed the source (English) side of the corpus using NLPWIN, a broad-coverage rule-based parser able to produce syntactic analyses at varying levels of depth (Heidorn, 2000). For the purposes of these experiments, we used a dependency tree output with part-of-speech tags and unstemmed, case-normalized surface words. For word alignment we used a training regimen of five iterations of Model 1, followed by five iterations of a word-dependent HMM model (He, 2007) in both directions. The forward and backward alignments were combined using a dependency tree-based heuristic combination. The word alignments and English dependency tree were used to project a target tree. From the aligned tree pairs we extracted treelet and order template tables.

For the Europarl systems, we use a phrase/treelet size of 7 and train model weights using 2000 sentences of Europarl data. For the "general-domain" systems, we use a phrase/treelet size of 4, and train model weights using 2000 sentences of web data.

For any given corpus, all systems used the same treelet or phrase size (see Table 7.1) and the same trigram language model. Model weights were trained separately for each system, data set and experimental condition, using minimum error rate training to maximize BLEU (Och, 2003).

[2] When no template is compatible with a treelet, the decoder creates an artificial template that preserves source order. This count feature allows MERT to deprecate the use of such templates. This is analogous to the glue rules of Chiang (2005).

[3] In our experiments, we find that the impact of this model is small in the presence of order templates; also, it degrades the overall speed of the decoder.

	% BLEU
Phrasal	13.41
Baseline treelet	15.89
+Deletion only	16.00
+Insertion only	16.16
+Deletion and Insertion	**17.01**

Table 8.1: English-Japanese system comparisons

8 Results and Discussion

Tables 8.1 and 8.4 compare baseline phrasal and treelet systems with systems that use various types of insertion and deletion templates.

English-Japanese: As one might expect, the use of structural insertion and deletion has the greatest impact when translating between languages such as English and Japanese that show significant structural divergence. In this language pair, both insertions and deletions have an impact, for a total gain of 1.1% BLEU over the baseline treelet system, and 3.6% over the phrasal system. To aid our understanding of the system, we tabulated the most commonly inserted and deleted words when translating from English into Japanese in Tables 8.2 and 8.3 respectively. Satisfyingly, most of the insertions and deletions correspond to well-known structural differences between the languages. For instance, in English the thematic role of a noun phrase, such as subject or object, is typically indicated by word order, whereas Japanese uses case markers to express this information. Hence, case markers such as "を" and "は" need to be inserted. Also, when noun compounds are translated, an intervening postposition such as "の" is usually needed. Among the most common deletions are "*the*" and "*a*". This is because Japanese does not have a notion of definiteness. Similarly, pronouns are often dropped in Japanese.

English-Spanish: We note, in Table 8.4 that even between such closely related languages, structural insertions give us noticeable improvements over the baseline treelet system. On the smaller Europarl training corpus the improvements range from 0.5% to 1.1% BLEU. On the larger training corpus we find that for the more in-domain governmental[4] and news test sets, the effect is smaller or even slightly negative, but on the very broad web test set we still see an improvement of about 0.7% BLEU.

Word	Count	%age	Type
の	2844	42%	Postposition
を	1637	24%	Postposition/case marker
は	630	9.3%	Postposition/case marker
、	517	7.6%	Punctuation
に	476	7.0%	Postposition
する	266	3.9%	Light verb
で	101	1.5%	Postposition
が	68	1.0%	Postposition
して	27	0.40%	Light verb
。	26	0.38%	Punctuation
か	19	0.28%	Question marker

Table 8.2: E-J: Most commonly inserted words

Word	Count	%age	Type
the	875	59%	Definite article
-	159	11%	Punctuation
a	113	7.7%	Indefinite article
you	53	3.6%	Pronoun
it	53	3.6%	Pronoun
that	26	1.8%	Conjunction, Pronoun
"	23	1.6%	Punctuation
in	16	1.1%	Preposition
.	10	0.68%	Punctuation
's	10	0.68%	Possessive
I	9	0.61%	Pronoun

Table 8.3: E-J: Most commonly deleted words

As one might expect, as the training data size increases, the generalization power of structural insertion and deletions becomes less important when translating *in-domain* text, as more insertions and deletions can be handled lexically. Nevertheless, the web test results indicate that if one hopes to handle truly general input the need for structural generalizations remains.

Unlike in English-Japanese, when translating from English to Spanish, structural deletions are less helpful. Used in isolation or in combination with insertion templates they have a slightly negative and/or insignificant impact in all cases. We hypothesize that when translating *from* English *into* Spanish, more words need to be inserted than deleted. Conversely, when translating in the reverse direction, deletion templates may play a bigger role. We were unable to test the reverse direction because our syntax-based systems depend on a source language parser. In future work we hope to address this.

[4] The "general domain" training corpus is a superset of the Europarl training set, therefore, the Europarl tests sets are "in-domain" in both cases.

		EU-devtest	EU-test	NC-test	Newswire	Web test
EUROPARL E-S						
	Phrasal	27.9	28.5	24.7	17.7	17.0
	Baseline treelet	27.65	28.38	27.00	18.46	18.71
	+Deletion only	27.66	28.39	26.97	18.46	18.64
	+Insertion only	28.23	28.93	**28.10**	**19.08**	**19.43**
	+Deletion and Insertion	**28.27**	**29.08**	27.82	18.98	19.19
GENERAL E-S						
	Phrasal	28.79	29.19	29.45	21.12	27.91
	Baseline treelet	28.67	29.33	32.49	**21.90**	27.42
	+Deletion only	28.67	29.27	32.25	21.69	27.47
	+Insertion only	**28.90**	**29.70**	32.53	21.84	**28.30**
	+Deletion and Insertion	28.34	29.41	**32.66**	21.70	27.95

Table 8.4: English-Spanish system comparisons, %BLEU

In table 8.5 and 8.6, we list the words most commonly inserted and deleted when translating the web test using the general English-Spanish system. As in English-Japanese, we find that the insertions are what one would expect on linguistic grounds. However, deletions are used much less frequently than insertions and also much less frequently than they are in English-Japanese. Only 53 words are structurally deleted in the 5000 sentence test set, as opposed to 4728 structural insertions. Furthermore, the most common deletion is of quotation marks, which is incorrect in most cases, even though such deletion is evidenced in the training corpus[5].

On the other hand, the next most common deletions "*I*" and "*it*" are linguistically well grounded, since Spanish often drops pronouns.

de	3509	74%	Preposition
la	555	12%	Determiner
el	250	5.3%	Determiner
se	77	1.6%	Reflexive pronoun
que	63	1.3%	Relative pronoun
los	63	1.3%	Determiner
del	57	1.2%	Preposition+Determiner
,	42	0.89%	Punctuation
a	30	0.63%	Preposition
en	21	0.44%	Preposition
lo	9	0.19%	Pronoun
las	6	0.13%	Determiner

Table 8.5: E-S: Most commonly inserted words

"	38	72%	Punctuation
I	5	9.4%	Pronoun
it	2	3.8%	Pronoun
,	2	3.8%	Punctuation
-	2	3.8%	Punctuation

Table 8.6: E-S: Most commonly deleted words

9 Conclusions and Future Work

We have presented an extension of the treelet translation method to include order templates with structural insertion and deletion, which improves translation quality under a variety of scenarios, particularly between structurally divergent languages. Even between closely related languages, these operations significantly improve the generalizability of the system, providing benefit when handling out-of-domain test data.

Our experiments shed light on a little-studied area of MT, but one that is nonetheless crucial for high quality broad domain translation. Our results affirm the importance of structural insertions, in particular, when translating from English into other languages, and the importance of both insertions and deletions when translating between divergent languages. In future, we hope to study translations from other languages into English to study the role of deletions in such cases.

[5] In many parallel corpora, quotes are not consistently preserved between source and target languages.

References

Chiang, David. A hierarchical phrase-based model for statistical machine translation. ACL 2005.

Crego, Josep, José Mariño and Adrià de Gispert. Reordered search and tuple unfolding for Ngram-based SMT. MT Summit 2005.

He, Xiaodong. Using Word Dependent Transition Models in HMM based Word Alignment for Statistical Machine Translation. Workshop on Statistical Machine Translation, 2007

Heidorn, George. “Intelligent writing assistance”. In Dale et al. Handbook of Natural Language Processing, Marcel Dekker. 2000

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. Statistical phrase based translation. NAACL 2003.

Chi-Ho Li, Dongdong Zhang, Mu Li, Ming Zhou, Hailei Zhang. An Empirical Study in SourceWord Deletion for Phrase-based Statistical Machine Translation. Workshop on Statistical Machine Translation, 2008

Marcu, Daniel, Wei Wang, Abdessamad Echihabi, and Kevin Knight. SPMT: Statistical Machine Translation with Syntactified Target Language Phrases. EMNLP-2006.

Menezes, Arul, and Chris Quirk. Microsoft Research Treelet translation system: IWSLT evaluation. International Workshop on Spoken Language Translation, 2005

Menezes, Arul, and Chris Quirk. Using Dependency Order Templates to Improve Generality in Translation. Workshop on Statistical Machine Translation, 2007

Och, Franz Josef. Minimum error rate training in statistical machine translation. ACL 2003.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. ACL 2002.

Wang, Wei, Kevin Knight and Daniel Marcu. Binarizing Syntax Trees to Improve Syntax-Based Machine Translation Accuracy. EMNLP-CoNLL, 2007