A Path-based Transfer Model for Machine Translation

Dekang Lin
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
lindek@cs.ualberta.ca

Abstract

We propose a path-based transfer model for machine translation. The model is trained with a word-aligned parallel corpus where the source language sentences are parsed. The training algorithm extracts a set of transfer rules and their probabilities from the training corpus. A rule translates a path in the source language dependency tree into a fragment in the target dependency tree. The problem of finding the most probable translation becomes a graph-theoretic problem of finding the minimum path covering of the source language dependency tree.

1 Introduction

Given a source language sentence S, a statistical machine translation (SMT) model translates it by finding the target language sentence T such that the probability $P(T|S)$ is maximized. In word-based models, such as IBM Model 1-5 (Brown *et al* 1993), the probability $P(T|S)$ is decomposed into statistical parameters involving words. There have been many recent proposals to improve translation quality by decomposing $P(T|S)$ into probabilities involving phrases.

Phrase-based SMT approaches can be classified into two categories. One type of approach works with parse trees. In (Yamada&Knight 2001), for example, the translation model applies three operations (re-order, insert, and translate) to an English parse tree to produce its Chinese translation. A parallel corpus of English parse trees and Chinese sentences are used to obtain the probabilities of the operations.

In the second type of phrase-based SMT models, phrases are defined as a block in a word aligned corpus such that words within the block are aligned with words inside the block (Och *et al* 1999, Marcu&Wong 2002). This definition will treat as phrases many word sequences that are not constituents in parse trees. This may look linguistically counter-intuitive. However, (Koehn *et al* 2003) found that it is actually harmful to restrict phrases to constituents in parse trees, because the restriction would cause the system to miss many reliable translations, such as the correspondence between "there is" in English and "es gibt" ("it gives") in German.

In this paper, we present a path-based transfer model for machine translation. The model is trained with a word-aligned parallel corpus where the source language side consists of dependency trees. The training algorithm extracts a set of paths from the dependency trees and determines the translations of the paths using the word alignments. The result of the training process is a set of rules for translating paths in the source language into tree fragments in the target language with certain probabilities. To translate a sentence, we first parse it and extract a set of paths from its dependency tree S. We then find a set of transfer rules that cover S and produce a set of tree fragments obtained to form a tree T^* such that $T^*=\text{argmax}_T P(T|S)$. The output sentence can then simply be read off T^*.

In the remainder of the paper, we first define paths in dependency trees. We then describe an algorithm for learning transfer rules and their probabilities. The translation algorithm is presented in Section 4. Experimental result is presented in Section 5. We then discuss related work in Section 6.

2 Paths in Dependency Trees

The dependency tree of a sentence consists of a set of nodes, each of which corresponds to a word in the sentence. A link in the tree represents a dependency relationship between a pair of words. The links are directed from the head towards the modifier. Except the root of tree, every node has exactly one incoming link. An example dependency tree is shown in Fig. 1.

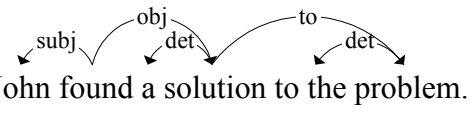

Figure 1. An example dependency tree

A sequence of nodes n_1, …, n_k, … n_m and the dependency links between them form a **path** if the following conditions hold:

a. $\forall\ i\ (1 \leq i < k)$, there is a link from n_{i+1} to n_i.

b. $\forall\ i\ (k \leq i < m)$, there is a link from n_i to n_{i+1}.

A set of paths is said to **cover** a dependency tree if the union of the nodes and links in the set of paths include all of the nodes and links in the dependency tree.

3 Acquisition of Transfer Rules

A transfer rule specifies how a path in the source language dependency tree is translated. We extract transfer rules automatically from a word-aligned corpus. For example, Fig. 2(b-g) are some of the rules extracted from the word aligned sentence in Fig. 2(a). The left hand side of a rule is a path in the source dependency tree. The right hand side of a rule is a fragment of a dependency tree in the target language. It encodes not only the dependency relations, but also the *relative* linear order among the nodes in the fragment. For example, the rule in Fig. 2(e) specifies that when the path *Connect→ to→controller* is translated into French *Branchez* precedes (but not necessarily adjacent to) *sur*, and *sur* precedes (but not necessarily adjacent to) *contrôleur*.

Note that the transfer rules also contain word-to-word mapping between the nodes in the source and the target (obtained from word alignments). These mappings are not shown in order not to clutter the diagrams.

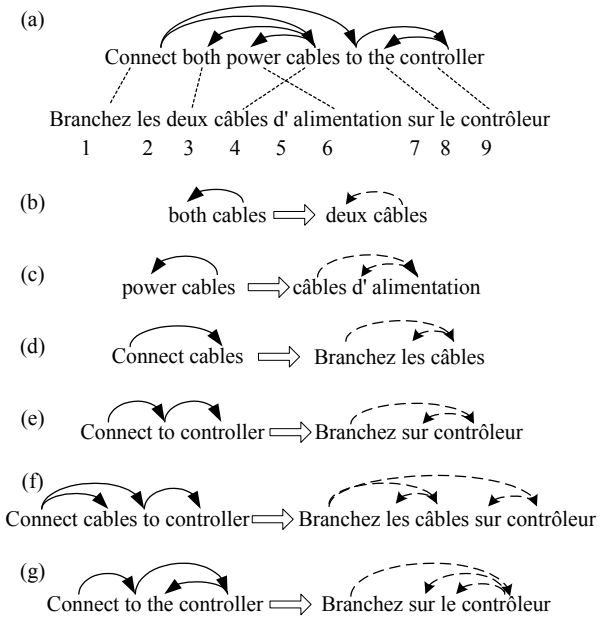

Figure 2. Examples of transfer rules extracted from a word-aligned corpus

3.1 Spans

The rule extraction algorithm makes use of the notion of spans (Fox 2002, Lin&Cherry 2003). Given a word alignment and a node *n* in the source dependency tree, the spans of *n* induced by the word alignment are consecutive sequences of words in the target sentence. We define two types of spans:

Head span: the word sequence aligned with the node *n*.

Phrase span: the word sequence from the lower bound of the head spans of all nodes in the subtree rooted at *n* to the upper bound of the same set of spans.

For example, the spans of the nodes in Fig. 2(a) are listed in Table 1. We used the word-alignment algorithm in (Lin&Cherry 2003a), which enforces a cohesion constraint that guarantees that if two spans overlap one must be fully contained in the other.

Table 1. Spans of nodes in Figure 2(a)

Node	Head Span	Phrase Span
Connect	[1,1]	[1,9]
both	[3,3]	[3,3]
power	[6,6]	[6,6]
cables	[4,4]	[3,6]
to		[8,9]
the	[8,8]	[8,8]
controller	[9,9]	[8,9]

3.2 Rule-Extraction Algorithm

For each word-aligned dependency tree in the training corpus, we extract all the paths where all the nodes are aligned with words in the target language sentence, except that a preposition in the middle of a path is allowed to be unaligned. In the dependency tree in Fig. 2(a), we can extract 21 such paths, 6 of which are single nodes (degenerated paths).

We first consider the translation of **simple paths** which are either a single link or a chain of two links with the middle node being an unaligned preposition. An example of the latter case is the path *Connect→to→controller* in Fig. 2(a). In such cases, we treat the two dependency link as if it is a single link (e.g., we call "Connect" the parent of "controller").

Suppose S_i is a simple path from node *h* to node *m*. Let *h*' and *m*' be target language words aligned with *h* and *m* respectively. Let *s* be the phrase span of a sibling of *m* that is located in between *h'* and *m'* and is the closest to *m'* among all such phrase spans. If *m* does not have such a sibling, let *s* be the head span of *h*.

The translation T_i of S_i consists of the following nodes and links:

- Two nodes labeled *h*' and *m*', and a link from *h*' to *m*'.
- A node corresponding to each word between *s* and the phrase span of *m* and a link from each of these nodes to *m'*.

Fig. 2(b-e) are example translations constructed this way. The following table lists the words *h'* and *m'* and the span *s* in these instances:

Table 2. Example spans

Example	*h'*	*m'*	*s*
Figure 2(b)	câbles	deux	[4,4]
Figure 2(c)	câbles	alimention	[4,4]
Figure 2(d)	Branchez	câbles	[1,1]
Figure 2(e)	Branchez	contrôleur	[4,6]

In general, a path is either a single node, or a simple path, or a chain of simple paths. The translations of single nodes are determined by the word alignments. The translation of a chain of simple paths can be obtained by chaining the translations of the simple paths. Fig. 2(f) provides an example.

Note that even though the target of a rule is typically a path, it is not necessarily the case (e.g., Fig. 2(g)). Our rule extraction algorithm guarantees the following property of target tree fragments: if a node in a target tree fragment is not aligned with a node in the source path, it must be a leaf node in the tree fragment.

3.3 Generalization of Rules

In addition to the rules discussed the in the previous subsection, we also generalize the rules by replacing one of the end nodes in the path with a wild card and the part of speech of the word. For example the rule in Fig. 2(b) can be generalized in two ways. The generalized versions of the rule apply to any determiner modifying *cable* and *both* modifying any noun, respectively.

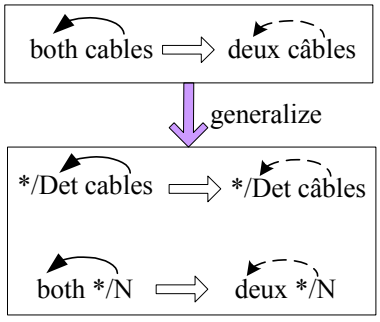

Figure 3. Generalization of Transfer rule

3.4 Translation Probability

Let S_i be a path in the source language dependency tree and T_i be a tree fragment in the target language. The **translation probability** $P(T_i|S_i)$ can be computed as

$$P(T_i \mid S_i) = \frac{c(T_i, S_i)}{c(S_i) + M}$$

where $c(S_i)$ is the count of S_i in the training corpus, $c(T_i,S_i)$ is the number of times T_i is the translation of S_i, and M is a smoothing constant.

4 Path-based Translation

Given a source language sentence, it is translated into the target language in the following steps:

Step 1: Parse the sentence to obtain its dependency structure.
Step 2: Extract all the paths in the dependency tree and retrieve the translations of all the paths.
Step 3: Find a set of transfer rules such that
  a) They cover the whole dependency tree.
  b) The tree fragments in the rules can be consistently merged into a target language dependency tree.
  c) The merged tree has the highest probability among all the trees satisfying the above conditions.
Step 4: Output the linear sequence of words in the dependency tree.

4.1 Merging Tree Fragments

In Step 3 of our algorithm, we need to merge the tree fragments obtained from a set of transfer rules into a single dependency tree. For example, the mergers of target tree fragments in Fig. 4(b-d) result in the tree in Fig. 4(e). Since the paths in these rules cover the dependency tree in Fig. 4(a), Fig. 4(e) is a translation of Fig. 4(a). The merger of target tree fragments is constrained by the fact that if two target nodes in different fragments are mapped to the same source node, they must be merged into a single node.

Proposition 1: The merger of two target tree fragments does not contain a loop.

Proof: The unaligned nodes in each tree fragment will not be merged with another node. They have degree 1 in the original tree fragment and will still have degree 1 after the merger. If there is a loop in the merged graph, the degree of a node on the loop is at least 2. Therefore, all of the nodes on the loop are aligned nodes. This implies that there is a loop in the source dependency tree, which is clearly false.

Proposition 2: If the condition parts of a set of transfer rules cover the input dependency tree, the merger of the right hand side of the rules is a tree.

Proof: To prove it is a tree, we only need to prove that it is connected since Proposition 1 guarantees that there is no loop. Consider the condition part of a rule, which is a path *A* in the source dependency

tree. Let *r* be the node in the path that is closest to the root node of the tree. If r is not the root node of the tree, there must exist another path *B* that covers the link between *r* and its parent. The paths *A* and *B* map *r* to the same target language node. Therefore, the target language tree fragments for *A* and *B* are connected. Using mathematical induction, we can establish that all the tree fragments are connected.

The above two propositions establish the fact that the merge the tree fragments form a tree structure.

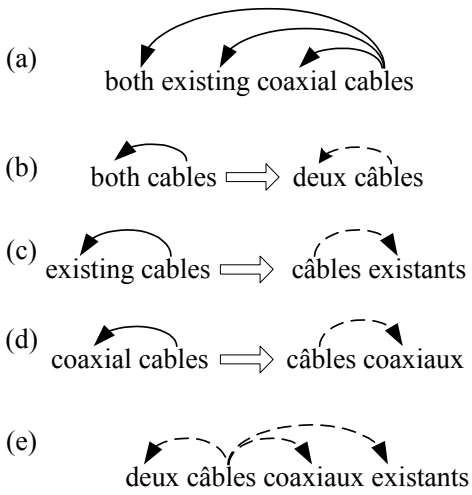

Figure 4. Examples of word ordering

4.2 Node Ordering

For each node in the merged structure, we must also determine the ordering of among it and its children. If a node is present in only one of the original tree fragments, the ordering between it and its children will be the same as the tree fragment. Suppose a node *h* is found in two tree fragments. For the children of *h* that come from the same fragment, their order is already specified. If two children m_1 and m_2 come from different fragments, we determine their order as follows:

- If m_1 and m_2 are on different sides of *h* in their original fragments, their order can be inferred from their positions relative to *h*. For example, the combination of the rules in Fig. 4(b) and Fig. 4(c) translate *both existing cables* into *deux câbles existants*.
- If m_1 and m_2 are on the same side of *h* and their source language counterparts are also on the same side of *h*, we maintain their relative closeness to the parent nodes: whichever word was closer to the parent in the source remains to be closer to the parent in the target. For example, the combination of the rules in Fig. 4(c) and Fig. 4(d) translates *existing coaxial cables* into *câbles coaxiaux existants.*
- If m_1 and m_2 are on the same side of *h* but their source language counterpart are on different sides of *h*, we will use the word order of their original in the source language.

4.3 Conflicts in Merger

Conflicts may arise when we merge tree fragments. Consider the two rules in Fig. 5. The rule in Fig. 5(a) states that when the word *same* is used to modify a noun, it is translated as *même* and appears after the noun. The rule in Fig. 5(b) states that *same physical geometry* is translated into *géométrie physique identique*. When translating the sentence in Fig. 5(c), both of these rules can be applied to parts of the tree. However, they cannot be used at the same time as they translate *same* to different words and place them on different location.

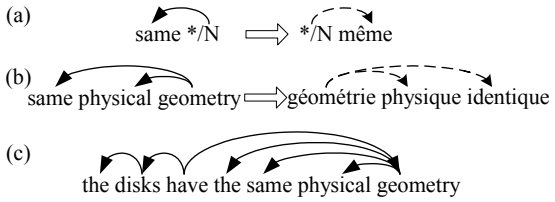

Figure 5. Example Conflicts

4.4 Probabilistic Model

Our translation model is a direct translation model as opposed to a noisy channel model which is commonly employed in statistical machine translation. Given the dependency tree *S* of a source language sentence, the probability of the target dependency tree *T*, $P(T|S)$, is computed by decomposing it into a set of path translations:

$$P(T \mid S) = \max_{C} \prod_{S_i \in C} P(T_i \mid S_i)$$

where *C* is a set of paths covering *S*; S_i's are paths in *C*; T_i's are possible translations for the corresponding S_i's and *T* is the merger of all T_i's. Note that the paths in C are allowed to overlap. However, no path should be totally contained in another, as we can always remove the shorter path to increase the probability without compromising the total coverage of C.

4.5 Graph-theoretic Formulation

If we ignore the conflict in merging tree fragments and assign the weight -log $P(T_i|S_i)$ to the path S_i, the problem of finding the most probable translation can be formulated as the following graph theory problem:

> Given a tree and a collection of paths in the tree where each path is assigned a weight. Find a subset of the paths such that they cover all the nodes and edges in the tree and have the minimum total weight.

We call this problem the **Minimum Path Covering of Trees**. A closely related problem is the Minimum Set Covering Problem:

> Given a collection F of subset set of a given set X, find a minimum-cardinality subcollection C of F such that the union of the subsets in C is X.

Somewhat surprisingly, while the Minimum Set Covering Problem is a very well-known NP-Complete problem, the problem of Minimum Path Covering of Trees has not previously been studied. It is still an open problem whether this problem is NP-Complete or has a polynomial solution.

If we assume that the number of paths covering any particular node is bounded by a constant, there exists a dynamic programming algorithm with O(n) complexity where n is the size of the tree (Lin&Lin, 2004). In the machine translation, this seems to be a reasonable assumption.

5 Experimental Results

We implemented a path-based English-to-French MT system. The training corpus consists of the English-French portion of the 1999 European Parliament Proceedings[1] (Koehn 2002). It consists of 116,889 pairs of sentences (3.4 million words). As in (Koehn, *et. al.* 2003), 1755 sentences of length 5-15 were used for testing. We parsed the English side of the corpus with Minipar[2] (Lin 2002). We then performed word-align on the parsed corpus with the ProAlign system (Cherry&Lin 2003, Lin&Cherry 2003b).

From the training corpus, we extracted 2,040,565 distinct paths with one or more translations. The BLEU score of our system on the test data is 0.2612. Compared with the English to French results in (Koehn *et. al.* 2003), this is higher than the IBM Model 4 (0.2555), but lower than the phrasal model (0.3149).

6 Related Work and Discussions

6.1 Transfer-based MT

Both our system and transfer-based MT systems take a parse tree in the source language and translate it into a parse tree in the target language with transfer rules. There have been many recent proposals to acquire transfer rules automatically from word-aligned corpus (Carbonell *et al* 2002, Lavoie *et al* 2002, Richardson *et al* 2001). There are two main differences between our system and previous transfer-based approach: the unit of transfer and the generation module.

The units of transfer in previous transfer based approach are usually subtrees in the source language parse tree. While the number of subtrees of a tree is exponential in the size of the tree, the number of paths in a tree is quadratic. The reduced number of possible transfer units makes the data less sparse.

The target parse tree in a transfer-based system typically does not include word order information. A separate generation module, which often involves some target language grammar rules, is used to linearize the words in the target parse tree. In contrast, our transfer rules specify linear order among nodes in the rule. The ordering among nodes in different rules is determined with a couple of simply heuristics. There is no separate generation module and we do not need a target language grammar.

6.2 Translational Divergence

The Direct Correspondence Assumption (DCA) states that the dependency tree in source and target language have isomorphic structures (Hwa *et. al.* 2002). DCA is often violated in the presence of translational divergence. It has been shown in (Habash&Dorr 2002) that translational divergences are quite common (as much as 35% between English and Spanish). For example, Fig. 6(a) is a Head Swapping Divergence.

Even though we map the dependency tree in the source language into a dependency tree in the target language, we are using a weaker assumption than DCA. We induce a target language structure using a source language structure and the word alignment. There is no guarantee that this target language dependency tree is what a target language linguist would construct. For example, derived dependency tree for "X cruzar Y nadando" is shown in Fig. 6(b). Even though it is not a correct dependency tree for Spanish, it does generate the correct word order.

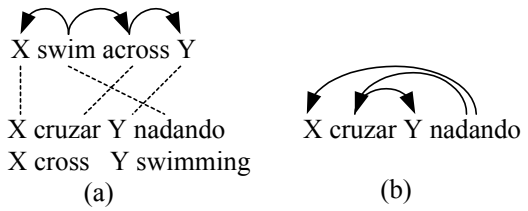

Figure 6. Translational Divergence

7 Conclusion and Future Work

We proposed a path-based transfer model for machine translation, where the transfer rules are automatically acquired from a word-aligned parallel corpus. The problem of finding the most probable translation is formulated as a graph-theoretic problem of finding the minimum path covering of the source language dependency tree.

[1] http://www.isi.edu/~koehn/europarl/

[2] http://www.cs.ualberta.ca/~lindek/minipar.htm

8 Acknowledgements

This research is supported by NSERC and Sun Microsystems, Inc.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Colin Cherry and Dekang Lin, 2003. A Probability Model to Improve Word Alignment. In *Proceedings of ACL-03*. pp.88-95. Sapporo, Japan.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of EMNLP-02*, pages 304-311. Philadelphia, PA.

Habash, Nizar and Bonnie J. Dorr, 2002. Handling Translation Divergences: Combining Statistical and Symbolic Techniques in Generation-Heavy Machine Translation, In *Proceedings of the Fifth Conference of the Association for Machine Translation in the Americas, AMTA-2002*,Tiburon, CA.

R. Hwa, P. Resnik, A. Weinberg, and O. Kolak. 2002. Evaluating Translational Correspondence using Annotation Projection. In *the Proceedings of the 40th Annual Meeting of the ACL*, Philadelphia, PA.

Philipp Koehn. 2002. Europarl: A Multilingual Corpus for Evaluation of Machine Translation. Unpublished Draft. http://www.isi.edu/~koehn/publications/europarl.ps

Philipp Koehn, Franz Josef Och, and Daniel Marcu, 2003. Statistical Phrase-Based Translation, In *Proceedings of HLT/NAACL 2003* pp. 127-133, Edmonton, Canada.

Lavoie, Benoit; White, Michael; and Korelsky, Tanya 2002. Learning Domain-Specific Transfer Rules: An Experiment with Korean to English Translation. In *Proceedings of the COLING 2002 Workshop on Machine Translation in Asia*, Taipei, Taiwan, pp. 60-66.

Dekang Lin and Colin Cherry, 2003a. Word Alignment with Cohesion Constraint. In *Proceedings of HLT/NAACL 2003*. Companion Volume, pp. 49-51, Edmonton, Canada.

Dekang Lin and Colin Cherry, 2003b. ProAlign: Shared Task System Description. In *Proceedings of the Workshop on Building and Using Parallel Texts*, pp. 11-14. Edmonton, Canada.

Guohui Lin and Dekang Lin. 2004. Minimum Path Covering of Trees. Submitted to *Information Processing Letters*.

Daniel Marcu and William Wong. 2002. A Phrase-Based, Joint Probability Model for Statistical Machine Translation. *Proceedings of the Conference on EMNLP-2002*, pp.133-139. Philadelphia, PA.

Franz Josef Och, Christoph Tillmann, Hermann Ney, 1999. Improved Alignment Models for Statistical Machine Translation. pp. 20-28; In *Proceedings of EMNLP-99*. University of Maryland, College Park, MD.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (ACL), pages 311–318.

Steve Richardson, W. Dolan, A. Menezes, and J. Pinkham. 2001. Achieving commercial-quality translation with example-based methods. In *Proceedings of MT Summit VIII*, Santiago De Compostela, Spain, pp. 293-298.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics* (ACL'01), Toulouse, France.