Syntax-Based Statistical Machine Translation: A review

Amr Ahmed
Language Technologies Institute
Carnegie Mellon University

Greg Hanneman
Language Technologies Institute
Carnegie Mellon University

1. Introduction

Ever since the incipient of computers and the very first introduction of artificial intelligence, machine translation has been a target goal — or better said, a dream that at some point in the past deemed impossible (ALPAC 1966). The problem that machine translation aims to solve is very simple: given a document/sentence in a source language, produce its equivalent in the target language. This problem is complicated because of the inherent ambiguity of languages: the same word can have different meaning based on the context, idioms plus many other computational factors. Moreover extra domain knowledge is needed for a high quality output.

Early techniques to solve this problem were human-intensive via parsing, transfer rules and generation with the help of an Interlingua (Hutchins 1995). This approach, while performing well in restricted domains, is not scalable and not suitable for languages that we do not have a syntactic theory/parser for. In the last decade, statistical techniques using the noisy channel model dominated the field and outperformed classical ones (Brown et al. 1993), however one problem with statistical methods is that they do not employ enough linguistic-theory to produce a grammatically coherent output(Och et al. 2003). This is because these methods incorporate little or no explicit syntactical theory and it only captures elements of syntax implicitly via the use of an n-gram language model in the noisy channel framework, which ca not model long dependencies.

The goal of syntax-based machine translation techniques is to incorporate an explicit representation of syntax into the statistical systems to get the best out of the two worlds: high quality output while not requiring intensive human efforts. In this report we will give an overview of various approaches for syntax-aware statistical machine translation systems developed,or proposed, in the lase two decades. In our survey, we will stress the tension between the expressivity of the model and the complexity of its associated training and decoding procedures.

The rest of this report is organized as follows: first, Section 2, gives a brief overview of the basic statistical machine translation model that serves as the basis of the subsequent discussions, and motivates the need for deploying syntax in the translation pipeline. In Section 3, we discuss various formal grammar formalisms which were proposed to model parallel texts. Then in section 4, we describe how these theoretical ideas have been used to augment the basic models in Section 2, and detail how the resulting models are trained from data, as well as assessing their complexity against the extra accuracy gained. Finally we conclude in Section 5

2. Statistical Machine Translation

In this section, we describe briefly the basic concept of statistical machine translation (SMT) and its components. Formally, given a sentence (sequence of words) f in the source language, the goal of a SMT system is to produce its equivalent sentence e in the target language. As shown in Figure 1, the translation problem is modeled in a noisy channel framework. $P(e)$ denotes the language model (*LM*) of the target that generates valid, fluent target sentences. These sentences are passed through the noisy channel to produce the source sentence f with probability $p(f|e)$. This probability is known as the translation model *TM*. Given an f sentence, the decoding problem is to find the target sentence e which with highest probability could have generated f. This best e is found using Bayes rule:

$$e^* = \text{argmax}_\text{e} \ \text{p(e|f)} \tag{1}$$

$$= \text{argmax}_\text{e} \ \text{p(e)p(f|e)} \tag{2}$$

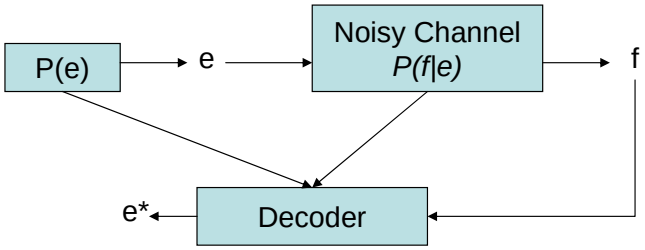


Figure 1
The noisy channel approach to machine translation

Using this decomposition, the decoder has two components: $p(e)$ which models how fluent is the target string, and $p(f|e)$ which models how faithful is the translation. Most SMT systems differ on the TM, that is how they model $p(f|e)$, or more formally how this probability distribution is factored to allow for assigning probabilities to unseen pairs. In the following two subsections we will briefly review the basic two dominant factorization approaches: word-based models and phrased-based models (for comprehensive survey, interested readers are refereed to Deng and Byrne (2006)). At the end of this section, we conclude by motivating the need for syntactical approaches and summarizing the rest of this report.

2.1 Word-based Methods

A key issue in the translation model is how to define correspondences between the words of the target sentence and those of the source one. During 1990s, IBM (Brown et al. 1993) introduced various approaches, that range in sophistication, which are also known as alignment models. These models differ in two dimensions: the first is the cardinality of the relation between source and target words. The second dimension is

the dependency assumptions involved in this mapping. Along the first dimension, the options are one-one, one-many, many-many as well as deletion (one-zero) which models different lexical realizations between the source and target language pairs. Figure 2.a illustrate these options, for instance, the word "eight" in the target side, is translated into "acht Uhr" in the source language (one-many), while most of the other alignments are one-one. Along the second dimension, words in the target language can move into the source string either: to a random position(IBM-1), conditioned on their absolute position (IBM-2), or conditioned on where the previous word moves (IBM-3, IBM-4, IBM-5). In all of these models, lexical translation probabilities are based on single word in each of the two languages.

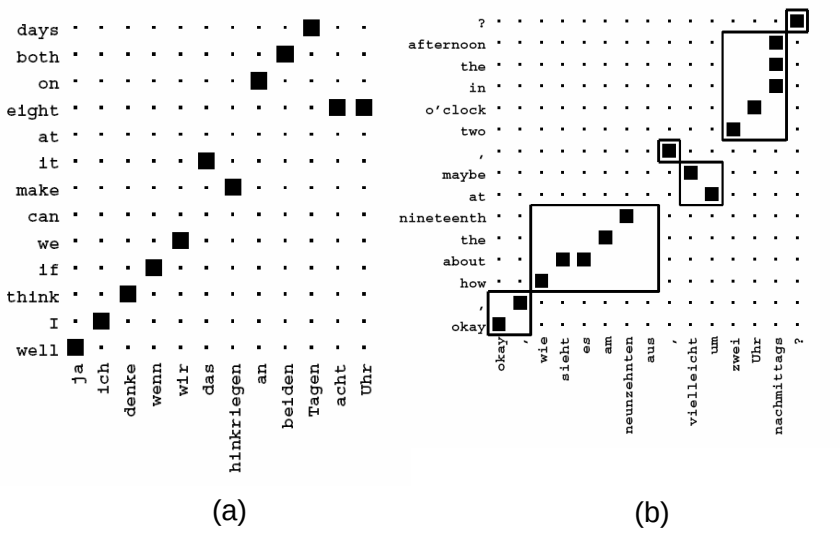


Figure 2
Alignment models. a) IBM word-based alignment and b) Phrase-based alignment

2.2 Phrase-based Methods

While the IBM word-based models were a breakthrough in pioneering the work in SMT, one of their general shortcoming is that they are mainly designed to model the lexical dependencies between single words. To remedy this , phrase based models were introduced (Vogel et al. (2000), Marcu and Wong (2002) and Och and Ney (2004)). As opposed to using a single word as the basis of the TM, phrase-based models add another layer of structure called "phrases" (a contiguous sequence of words) and use it as the basic translation unit. In other words, a whole group of adjacent words in the target language may be aligned with a whole group of adjacent words in the source language. As a result, the context of words tends to be explicitly taken into account, and the difference in local word orders between source and target languages can be learned explicitly. Figure 2.b gives an example of phrases extracted from a parallel

sentence pairs. In a nutshell, the TM model here proceed as follows. First, the target language string is broken into phrases, that are independently translated as a block to source phrases. Then these translated phrases are re-ordered to produce the source one according to a distortion probability. This distortion probability can be either position based (models the probability of target phrase i moving to source phrase j) or simple models that are designed to encourage a monotone ordering.

A key issue in phrase-based models is how to extract phrases from a parallel corpus. We identify two main methods here: an alignment-based method (conditional) and a joint-based one. As described in (Vogel et al. (2000) and Och and Ney (2004)), and detailed in Figure 2.b, an IBM-word word-based model is used to align the corpus, and then phrases up to a certain maximum are extracted subject to the constraints that words in the both sides are mapped into each other. Then, a probability of using this phrase to produce the source part is estimated from the corpus using relative frequency conditioned on the target part of the phrase. Marcu and Wong (2002) gave an alternative translation model that assumes that lexical correspondences can be established not only at the word level, but at the phrase level as well. To learn such correspondences, they introduced a phrase based joint probability model that simultaneously generates both the source and target phrases in a parrel corpus using a generative story. EM learning is then used to estimates both the joint phrase probability model (i.e. probability that phrase f is a translation of phrase e) and a distortion model which is the probability of a target phrase i moving to source phrase j.

Apart from the phrase extraction phase, decoding in phrase based methods proceeds using a beam-based search approach. The target string is built from left to right. At each step, a phrase whose source side match a subsequence in the source string is selected, and its target side is added to the target string being built. Each of the created partial target stings are scored based on the phrase transaction probability, the distortion model, and the target language model. To help scale up this huge search space, a beam search is used, which at each step keeps only the best N-partial hypothesis.

2.3 To Parse Or Not To Parse?

In a nutshell the translation models described in section 2.1 and **??** differ on how they structure the strings at the target and source side of the channel. In word-based models, a flat structure is used, that is the sentence is one level tree with words as leaves. While in phrase-based methods, each sentence is viewed as a two level tree: the sentence first derives a set of phrases, and then each phrase derives a set of lexical words. This representation allows phrase-based models to move words as a block in the TM which conforms to the linguistic study by Yarowsky et al. (2001) and Fox (2002). Indeed, experimental results showed that phrase-based models outperform word-based ones. Therefore one might hypothesis that adding structure to the TM would help, theodore moving from phrase-based models, which use a two-level tree structure, to a full tree structure induced via a syntactical theory might be beneficial.

However, Koehn, Och, and Marcu (2003) had another opinion: based on their experiments they concluded that syntax is detrimental and not only does not boast the performance, but also decreases the accuracy! To better understand this result, one should frame this conclusion in its proper context. In (Koehn, Och, and Marcu 2003) a systematic comparison of three phrased-based SMT methods were carried out. As detailed in Section 2.2, all the components of a phrase-based SMT system were fixed except the phrase extraction module. Three phrase extraction approaches were compared: the alignment based one (Och and Ney 2004), the joint-based approach (Marcu

and Wong 2002) and an approach were the phrases from the alignment approach were filtered to remove any phrases that do not correspond to a grammatical constituent. The grammatical constituents were found by *parsing* the corpus with a statistical parser. The results showed that the first two approaches give comparable results, while the syntax-motivated one gives the worst result. That is because many useful translation phrases like "there is", which does not correspond to a syntactical constituent, were removed. Therefore Syntax does not matter!

We believe that the results presented in this paper were convincing indeed but not to that conclusion, instead to the following one: *Syntax, when incorporated into the SMT model in this **specific way** results in a worse performance*, that is totally correct, and the take home message from this paper would be: *there is a need to develop other syntax-based models in which the fact that there are parallel corpus is **explicitly** represented in the model*. Moreover, we believe that the experiments were a little bit biased, for instance, the phrases in Marcu's model were discovered based on a generative story that jointly generates the parallel corpus. Similarly, the phrases in Och's model were collected from a word-based alignment matrix. Therefore, in both cases, the phrase generation module was *aware* of the task it is being contextualized in. In contrast, one can not ask a system to produce a result that it was not trained to give in the first place. The statistical parsers used to filter out phrases, were trained to maximize a certain objective function, for example the likelihood of the phrases, not the BLEU score. For instance, one could learn a shallow parser model or just a bracketing model jointly with the translation model to optimize the BLEU score or the likelihood of the parallel corpus, indeed this is what Marcu's system did. Moreover, the decoding approach used in their experiments, which was described in section 2.2, is a linear one, whereas the constituents discovered by parsing are due to hierarchal decoding.

Therefore, the word "syntax" was loosely defined in their settings, and we believe that one possible characterization of syntax in the context of machine translation, is a method used to generate *constituents* (phrases in SMT jargon) and model their movements across languages (Alignment model). The question now is what is the right syntactical theory for this task? In Section 3 we will detail many of such ones, and in section 4 we will describe how these models were used to build systems that outperformed state of the art phrase-based models.

3. Grammar Formalisms for Parallel Texts

Syntax-based statistical machine translation models may rely on some form of synchronous grammar as a way to simultaneously generate source and target sentences and the correspondence between them. In this section, we examine some published formalisms for such grammars and discuss their expressive powers. We begin by exploring string-based context-free formalisms in sections 3.1, 3.2, and 3.3. In sections 3.4 and 3.5, we extend the domain of locality beyond context-free rules and look at tree-based formalisms. After having introduced all the formalisms, the final section 3.6 provides a short overview of how the formalisms have been adapted to fit into a probabilistic framework.

3.1 Inversion Transduction Grammars

Building on earlier work in applying finite-state transducers to tasks in natural language processing, Wu (1997) introduced a context-free transduction formalism that allows for some reordering of the constituents at each level.

As a base, a simple transduction grammar has production rules similar to those of a regular context-free grammar, except the terminal symbols are marked as either occurring in one of two distinct output streams. The terminal symbol pair x/y on the right-hand side of a grammar rule indicates that x is produced in the first output stream and y is produced in the second. A symbol that appears in one stream without a corresponding symbol in the other can be expressed with a "singleton" pair x/ϵ or ϵ/x. Non-terminals are shared between the two streams. In this way, a rule $A \rightarrow B\ x\ C\ z$ for one output stream and a rule $A \rightarrow B\ y\ C$ for a second output stream can be combined into the transduction grammar rule $A \rightarrow B\ x/y\ C\ z/\epsilon$.

A severe limitation of transduction grammars, when applied to natural language, is that the constituents in each output stream must appear in exactly the same order. This means that the two languages being described must have the same grammatical structure according to the grammar. Wu's inversion transduction grammar (ITG), however, allows reordering at the rule level by allowing the constituents of the second output stream to be written out in reverse order. In ITG, putting square brackets around the right-hand side of a production rule indicates that the rule is to be interpreted as in a simple transduction grammar, while using angled brackets denotes that the constituents should be reversed in the second output stream. Thus, a rule $A \rightarrow \langle B\ x/y\ C\ z/\epsilon \rangle$ is equivalent to $A \rightarrow B\ x\ C\ z$ in the first stream, but $A \rightarrow C\ y\ B$ in the second.

Any ITG can be rewritten in a normal form, similar to Chomsky Normal Form for context-free grammars, in which every production rule can be expressed as a single non-terminal going to a single terminal symbol or a pair of other non-terminals. Since both singletons and pairs of terminals are allowed, and since the order of consituents can be inverted, there are six allowable rule types in the normal form:

$$\begin{array}{lll} S \rightarrow \epsilon/\epsilon & A \rightarrow x/\epsilon & A \rightarrow [B\ C] \\ A \rightarrow x/y & A \rightarrow \epsilon/y & A \rightarrow \langle B\ C \rangle \end{array}$$

3.2 Synchronous Context-Free Grammars

ITGs are a special case of synchronous context-free grammars (CFGs), which allow for arbitrary reordering of constituents on the right-hand side of a production rule. A synchronous CFG rule consists two regular CFG rules that share a left-hand side and whose right-hand constituents are mapped to each other in a one-to-one relationship (Chiang 2006). These mappings express the translational eqivalence between source-language and target-language constituents. In the following example grammar for English-Japanese, the mappings are indicated by matching numerical superscripts; the notation style is from Melamed, Satta, and Wellington (2004).

$$S \rightarrow [(NP^1\ VP^2), (NP^1\ VP^2)] \tag{3}$$

$$VP \rightarrow [(V^1\ NP^2), (NP^2\ V^1)] \tag{4}$$

$$NP \rightarrow [(\text{I}), (\text{watashi wa})] \tag{5}$$

$$NP \rightarrow [(\text{the box}), (\text{hako wo})] \tag{6}$$

$$V \rightarrow [(\text{open}), (\text{akemasu})] \tag{7}$$

When a constituent on one of the synchronous right-hand sides is rewritten using a grammar rule, the same rewriting is carried out on the corresponding coindexed constituent on the other right-hand side. In this way, applying a set of grammar rules

derives two strings at the same time:

$$[(S^{10}), (S^{10})] \tag{8}$$

$$\Rightarrow \quad [(NP^{11}\ VP^{12}), (NP^{11}\ VP^{12})] \tag{9}$$

$$\Rightarrow \quad [(NP^{11}\ V^{13}\ NP^{14}), (NP^{11}\ NP^{14}\ V^{13})] \tag{10}$$

$$\Rightarrow \quad [(\text{I}\ V^{13}\ NP^{14}), (\text{watashi wa}\ NP^{14}\ V^{13})] \tag{11}$$

$$\Rightarrow \quad [(\text{I open}\ NP^{14}), (\text{watashi wa}\ NP^{14}\ \text{akemasu})] \tag{12}$$

$$\Rightarrow [(\text{I open the box}), (\text{watashi wa hako wo akemasu})] \tag{13}$$

One pitfall of synchronous CFGs is that they are not guaranteed to be convertible to Chomsky Normal Form if they contain production rules with right-hand sides of length four or more. The usual conversion technique of rewriting adjacent elements in the right-hand side as new non-terminals fails when none of the right-hand side elements have numerically adjacent coindexes in both languages.

At the same time, even in Chiang's restricted form — where the mapping between non-terminals is exactly one-to-one and non-terminals of type X may only map to others of type X — the synchronous CFG formalism can be cleaner than an ITG for the same data thanks to its more expressive reordering possibilities. For example, any reordering of three or more constituents can be expressed in one synchronous CFG rule:

$$X \rightarrow [(A^1\ B^2\ C^3), (A^1\ C^3\ B^2)] \tag{14}$$

An ITG, however, in this case requires two

$$X \rightarrow [A\ Y] \tag{15}$$

$$Y \rightarrow \langle B\ C \rangle \tag{16}$$

and for some reorderings of length four and higher, an ITG is incapable of expressing them at all (Wu 1997).

3.3 Multitext Grammars

There are also linguistic situations that synchronous CFGs under Chiang's formation are unable to handle, requiring a more permissive formalism. Multitext grammars (MTGs) were introduced by Melamed (2003) to provide a formalism that is both expressive and computationally tractable. Notably, MTGs are able to handle discontinuous constituents and independent (monolingual) rewriting where both ITGs and synchronous CFGs cannot.

MTG is a generalization of synchronous CFG to an arbitrary number of synchronous grammars; synchronous CFGs can thus be denoted as the subset 2-MTG. An MTG, like a CFG, has disjoint sets of terminal and non-terminal symbols. The set of non-terminals includes additional categories for the start symbol and the empty string; lexemes for both of these are added to the terminals set as well. Translational equivalences are represented in vectors of symbols called links. An MTG also has a number of production rules, such as the following unlexicalized rules for English-Russian adapted

from Melamed (2003):

$$\begin{pmatrix} S \\ S \end{pmatrix} \Rightarrow\bowtie \begin{matrix}[1,2,3] \\ [1,3,2]\end{matrix} \begin{pmatrix} Pro\ V\ NP \\ Pro\ V\ NP \end{pmatrix} \tag{17}$$

$$\begin{pmatrix} Pro \\ Pro \end{pmatrix} \Rightarrow \begin{pmatrix} \text{I} \\ \text{ya} \end{pmatrix} \tag{18}$$

Yield rules, such as (18) above, have the form $\mathbf{X} \Rightarrow \mathbf{t}$, where $\mathbf{X}$ is a link of non-terminals and $\mathbf{t}$ is a link of terminals. Depend production rules, such as (17) above, have the form $\mathbf{X} \Rightarrow\bowtie \mathbf{PM}$, where $\mathbf{P}$ is a vector of permutations and $\mathbf{M}$ is a link of non-terminals. The $\bowtie$ operator (called "join") reorders the elements in one row of a link according to the permutation given for that row. In the example above, at the end of the derivation the $\bowtie$ operator would rearrange the English sentence to $Pro\ V\ NP$ and the Russian to $Pro\ NP\ V$.

The rule notation was revised by Melamed, Satta, and Wellington (2004) to a more compact form more closely resembling CFG notation. Following the later convention, (17) and (18) would be written as

$$[(S),(S)] \rightarrow [(Pro^1\ V^2\ NP^3),(Pro^1\ NP^3\ V^2)] \tag{19}$$

$$[(Pro),(Pro)] \rightarrow [(\text{I}),(\text{ya})] \tag{20}$$

This matches the notation style used in Section 3.2 of this paper, and is the convention we will use in the remainder of this section as well.

Independent rewriting removes the restriction that the right-hand sides of MTG grammar rules must share a left-hand side. It leads to a more elegant representation of constituents that do not share grammatical substructure across languages:

$$[(NP),(NP)] \rightarrow [(D^1\ N^2),(N^2)] \tag{21}$$

$$[(D),()] \rightarrow [(\text{the}),()] \tag{22}$$

$$[(N),(N)] \rightarrow [(\text{cat}),(\text{kota})] \tag{23}$$

Without independent rewriting, rule (21) would be required to include a determiner on the target side, and then (22) would have to posit $D \rightarrow \epsilon$ since the determiner does not appear in the Russian text. Allowing rewriting reduces the overall size of the grammar and can also help avoid ambiguity (Melamed, Satta, and Wellington 2004).

Generalized MTG can also handle discontinuous constituents by relaxing the constraint of one-to-one mapping between non-terminals on the right-hand sides of rules.

$$[(S),(S)] \rightarrow [(NP^1\ VP^2),(NP^1\ VP^2)] \tag{24}$$

$$[(VP),(VP)] \rightarrow [(V^1\ NP^2),(NP^2\ V^1\ NP^2)] \tag{25}$$

$$[(NP),(NP)] \rightarrow [(\text{the doctor}),(\text{el médico})] \tag{26}$$

$$[(V),(V)] \rightarrow [(\text{treats}),(\text{examina})] \tag{27}$$

$$[(NP),(NP,NP)] \rightarrow [(\text{his teeth}),(\text{le},\text{los dientes})] \tag{28}$$

The discontinuous constituent *le ... los dientes* in Spanish is marked by a comma on both the left- and right-hand sides of rule (28). When (28) is applied to the yield of (25), the first NP chunk (*le*) fills the first NP^2, and the second chunk (*los dientes*) takes the second. The single English NP (*his teeth*) from (28) takes the single instance of NP^2 on the English side of rule (25), as usual.

Despite the additions to CFG that are found in MTG and generalized MTG, the scope of the operations being applied to the source and target languages is still bounded by the scope of a single context-free production rule. Under these formalisms, there is no method to enforce reordering or synchronization constraints involving constituents that are not siblings of each other, and in many natural languages a rule-sized domain of locality is not large enough.

3.4 Synchronous Tree-Adjoining Grammars

In tree-adjoining grammar (TAG), the grammar consists of a certain number of elementary trees — portions of syntactic structure — that can incrementally combine with each other to produce a final parse tree. Given a partially complete parse tree, a new elementary tree can be merged into it via two operations: it can be "plugged in" at an existing fringe node by substitution, or it can be spliced into the tree by adjunction. Among other sources, Joshi and Schabes (1997) give more details on the TAG formalism and its operations.

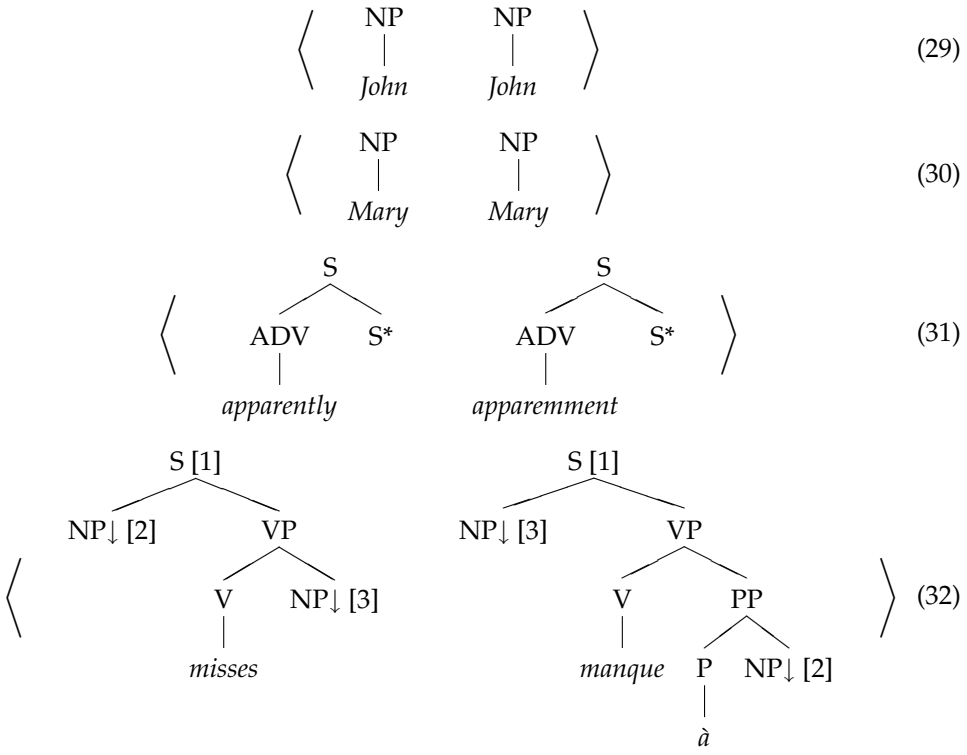
One of TAG's main strengths is the ability to easily encode long-distance dependencies, such as filler-gap constructions, by placing the dependency in the same elementary tree as the thing itdepends on. An elementary tree lexicalized for a given verb, for example, can include substitution nodes for each of its required arguments. Idioms or other non-compositional structures can also be specified in single elementary trees.

A synchronous version of TAGs was developed by Shieber and Schabes (1990). In the introductory paper, synchronous TAGs are applied to derive a sentence concurrently with its representation in logical form, so the topics discussed are more related to semantic analysis than machine translation. The authors note, however, that a human language can easily be substituted for the logical form, and it follows that the same advantages of synchronous TAGs apply. A companion paper (Abeillé, Schabes, and Joshi 1990) specifically applies the use of lexicalized synchronous TAGs to the task of machine translation.

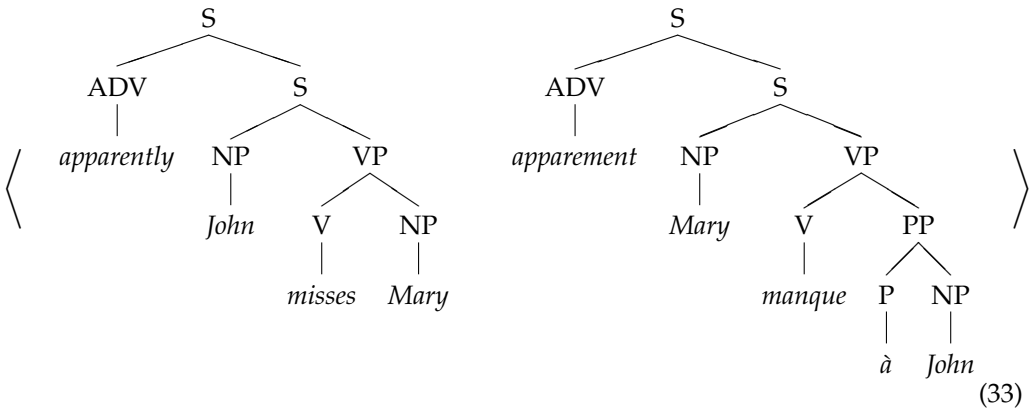
A lexicalized synchronous TAG can be built from two monolingual lexicalized TAGs, one for the source language and one for the target language. Each element in the new lexicon is a pair of elementary trees from the two original lexicons that represents a translational equivalence. A node from the source tree may be linked to a node in the target tree; these links mark the places where other pairs of trees may be synchronously substituted or adjoined in.

The following derivation example is taken from Abeillé, Schabes, and Joshi (1990). In the synchronous TAG below, links between source and target nodes are indicated by

the coindexes [1], [2], and [3].



In the first step of the derivation, the tree (29) is synchronously substituted into (32) at the nodes marked [2]. Then (30) is substituted at [3]. Finally, tree (31) is adjoined into (32) at [1]. The final result is the pair

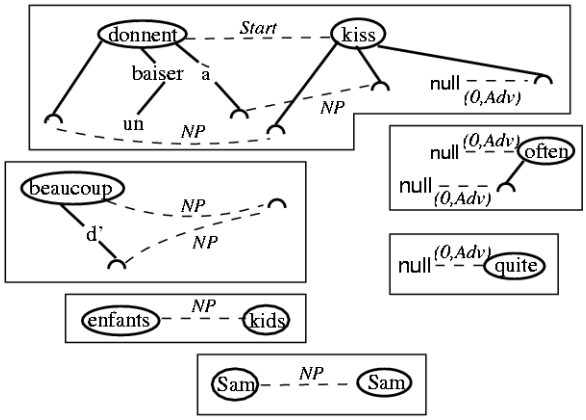


The translation can be carried out in three steps: first, the source sentence is parsed according to the source grammar, producing a derivation tree. Then, for each step in the derivation, an equivalent target tree can be produced by following the same series of substitutions and adjunctions at coindexed nodes, but this time using the target trees from the target grammar. Finally, the target sentence is collected by reading off the leaves of the final target tree.

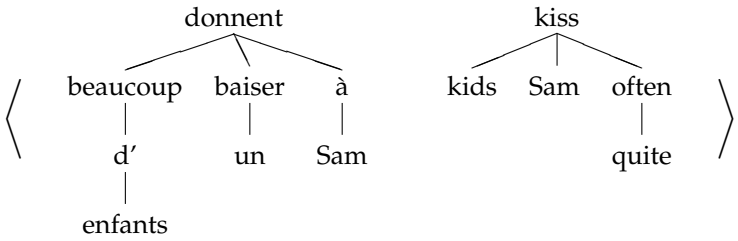
3.5 Synchronous Tree-Substitution Grammars

Eisner's (2003) synchronous tree-substitution grammar (TSG) is a new version of synchronous TAG without adjunction operations. Like in synchronous TAG, elements of a synchronous TSG are pairs of trees with nodes linked between them to show where further trees may be synchonously substituted, though the substitutions are restricted to taking place at leaf nodes. Links between nodes are labeled, and only tree pairs whose roots are connected with a link bearing the same label may be substituted. (The label thus controls what kind of tree may be substituted at the nodes.)

Eisner gives examples using dependency trees, as below, though he states that the formalism is general enough to apply to syntax trees as well.



The boxed trees are joined together as shown visually above to produce the final derivation. This example shows a rather free translation in which the French *beaucoup d'* ("many") has been translated as *quite often* in English. The translation is carried out by elementary tree pairs where one side is null: *beaucoup d'* disappears in translation, and then *quite often* is asynchronously added in a separate step.



(34)

3.6 Probabilistic Interpretations

Many of the grammar formalisms discussed in the preceeding sections have probabilistic or stochastic interpretations. We review them in this section.

In a weighted (or "stochastic") CFG-based grammar, probabilities are assigned to grammar rules subject to the constraint that, for each non-terminal A in the grammar

and its possible multilingual right-hand sides α,

$$\sum_{\alpha} P(A \rightarrow \alpha) = 1 \tag{35}$$

The probability of a derivation in the grammar is the product of the probabilities of all the grammar rules that were used to derive it. Recursively, for a rule $X \rightarrow A\ B$,

$$P(X_{i,k}) = P(A_{i,j})P(B_{j+1,k}) \tag{36}$$

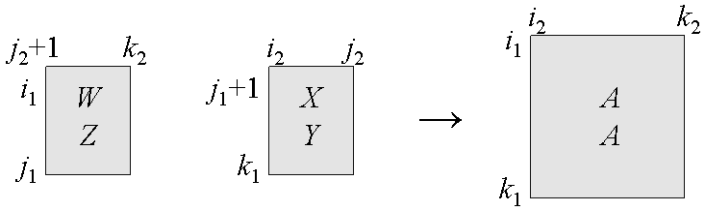
where $A_{i,j}$ represents a constituent of type A in the input from position i to position j. In the synchronous formalisms for translation treated in this paper, the positions are more properly expressed as two-dimensional vectors representing positions in both the source and target input.

As the statistical revolution affected machine translation more and more strongly during the mid-1990s, Wu (1997) was the first to describe how his formalism could be adapted to work in a statistical system. Wu described a dynamic programming parsing algorithm for weighted ITGs that computed the maximum-likelihood bilingual parse given a string pair. Since any ITG can be rewritten in an adapted Chomsky Normal Form, the bilingual parsing algorithm is similar to the monolingual CKY algorithm. Its complexity, however, is doubled to $O(n^6)$ for a source and target sentence each of length n. Full details are given in Wu (1997).

The synchronous CFGs described by Chiang (2006) have a similar interpretation, with probabilties being assigned to rules and derivations in the same way as above. Synchronous CFGs that contain production rules with right-hand sides longer than length three, however, are not guaranteed to be translatable into a normal form, so the parsing algorithm considered by Chiang is a probabilistic Earley algorithm that runs in $O(n^3)$. It considers only the source-language side of each production rule, produces a monolingual derivation, and then constructs the equivalent derivation in the target language by applying the previously ignored target-side rules. If there are two grammar rules that have the same source side but different target sides, multiple target translations will be produced. Since each rule has its own weight, however, the most likely of the various translations can still be found.

Probabilistic MTGs were addressed by Melamed (2004), as have a variety of optimized and general parsers (Melamed 2003). The left-hand sides of MTG grammar rules does not have to be shared by the right-hand sides, so the parse is explicitly carried out in both dimensions rather than only one. For a binarized MTG in normal form, parsed constituents are built up from existing adjacent ones in both dimensions. First, the chart is initialized with "seeds," one-dimensional items representing the words in each input:

Two-dimensional items are called hyperedges (or "hedges" for short); they are the constituents built from adjacent chart items according to the grammar rules. Each grammar rule leads to one inference rule for the parser. As an example, the grammar rule $[(A), (A)] \rightarrow [(W^1\ X^2), (Y^3\ Z^4)]$ gives



For an unlexicalized binary grammar, this parsing algorithm runs in $O(n^6)$. Binarization procedures for MTGs are discussed by both Melamed (2003) and Melamed, Satta, and Wellington (2004).

Eisner (2003) gave a probabilistic formulation for synchronous TSGs. In it, the root and each leaf node of a tree t is given a state q, and the function $P(t' \mid q)$ specifies a probability distribution over all trees t' whose root state is also q. This gives a distribution over trees that can be chosen for substitution at each node.

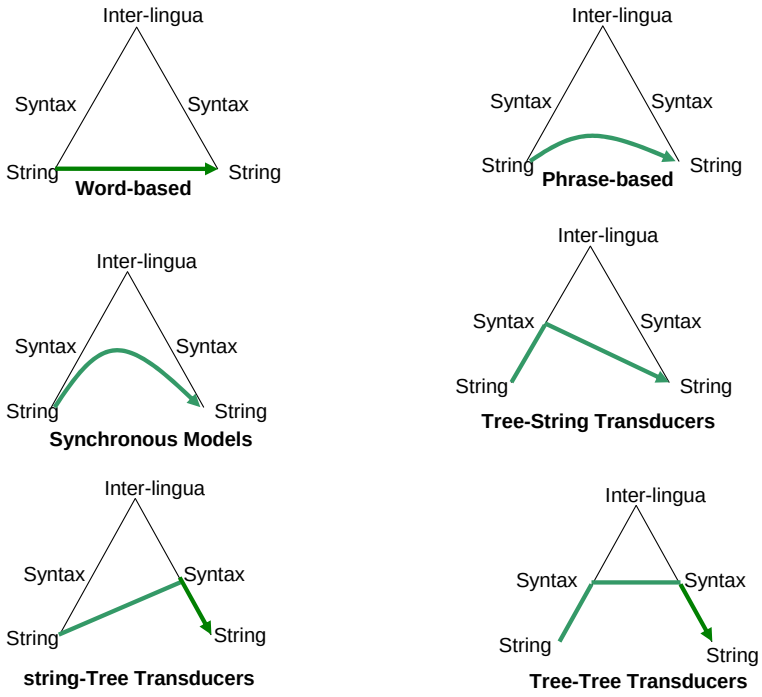


Figure 3
SMT Models. The arrows indicate the direction of the translation model (Decoding goes in the other direction).

4. Syntax-based Statistical Machine Translation Models

In this section we detail how the statistical translation models from section 2 have been enriched with syntactical theory from section 3. Figure 2 gives a unifying representation of all the models discussed in this paper in terms of a translation pyramid. The direction of the arrow indicates the translation model; therefore the left part of the triangle indicates input to the noisy channel (target or e string), while the right part is the output (source or f string). The word-based model remains at the bottom of the pyramid without exploiting any structure in the languages being modeled. Phrase-based models rise up from this bottom level sightly by adding a hidden structural phrase layer. This layer allows the model to move words as a block,resulting in a better translation of noun and propositional phrases as they tend to cohere between most languages (Yarowsky et al. (2001),Fox (2002)). Unfortunately, the output of phrase-based models fails to capture long-range movement at a deeper level like modifier movements between English and French (Chiang 2005). To help remedy this problem, and produce fluent output, syntax-based models aim at modeling deeper level of structure at the two sides of the noisy channel. As shown in Figure 3, these models can be characterized according to the representation at the input and output of the channel, as well as according to the monolingual resources required. In the following subsection we will detail each of these models.

4.1 Learning Synchronous Grammar

In section 3, we gave various synchronous formalisms to model bilingual text, and cast the translation process as parsing using these synchronous grammars. However, most of the techniques discussed in Section 3 do not have a practical implementation yet because of scalability issues. To learn these synchronous grammars, one needs to model the joint probability of the source and target languages—that is $p(e, f)$— using hidden variables to account for the missing bitree. An EM algorithm is then used to estimate the required parameters. However this involves a costly E-Step during which the bilingual text is parsed in (n^6) in most formalisms using a variant of the inside-outside algorithm used in mono-lingual statistical parsers. Therefore, to scale these systems one needs to impose certain restrictions on the grammar expressiveness to avoid the costly E-Step.

Among the formalism discussed in section 3, and until recently, only ITG (section 3.1) has a practical learning algorithm using EM. However, the grammar discussed there uses binary rules which severely limits the amount of phrasal movement that can be induced using this formalism. Therefore, its usages are confined to analysis of parallel text: inducing an initial alignment for more sophisticated models, segmentation tasks, etc. (Wu 1997). Recently the power of ITG has been improved in two ways. First, Zhang and Gildea (2005) gave a lexicalized version of ITG called (LITG), in which the grammar production probabilities are conditioned on lexical information throughout the tree. This model is reminiscent of lexicalization as used in Modern statistical parsers, in that a unique head word is chosen for each constituent in the tree. However it differs from monolingual approaches as the head word is chosen through the EM-algorithm rather than via deterministic rules. By introducing lexicalization, the grammar can learn lexicalized re-ordering rules in a purely unsupervised fashion. Testing the model on an alignment task and using the alignment error rate as a measure, the model outperformed IBM model 4 and slightly outperformed unlexicalized ITG on a Chinese-English corpus. In a follow up study, Zhang and Gildea (2006) extended the ITG with head-modifier lexicalization that allows the system to model, for instance, verb-subject

agreement and other monolingual dependency structures. The new model is still efficient to be trained using a variant of the inside-outside algorithm, however, this variant did not show any improvement in alignment over LITG. However it improved recovery of dependencies which can be extracted from the synchronous tree by following the head word.

Nevertheless, apart from all of these extensions, ITG and its variants, are still a limited formalism for many language pairs. Therefore, recently researchers focused in scaling up other synchronous grammar formalisms.

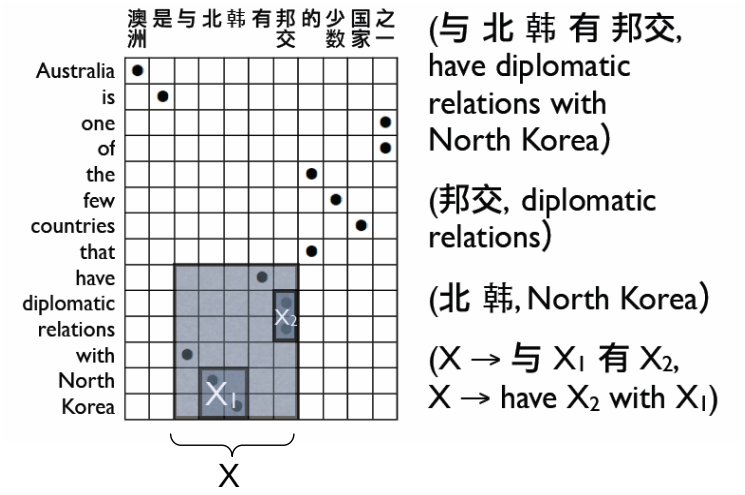


Figure 4
Illustrating phrase-extraction in Chiang (2005)

Chiang (2005) and Chiang et al. (2005) gave a heuristic approach to learning synchronous CFG on top of the output of a phrase-based model, without any linguistic commitment to certain non-terminal naming conventions such as VP, NP, etc. One can view Chiang's approach as a generalization of phrase-based TM that endows phrases with the ability to have sub-phrases. The method is described with reference to Figure 4. As we noted before, phrase-based models has an excellent ability of modeling NP block translation, yet it has a limited ability in modeling long-rang movements. To alleviate that, the extracted phrases from a phrase-based systems were analyzed to induce hierarchial phrases. As evident from Figure 4, any phrase-based system can extract the phrase-pairs: $X1, X2$ and X, yet it fails to note their topological relationships and the reordering rule it induces. Indeed, combining these three phrase-pairs, one can learn that the string "with X1 have X2" in Chinese, is translated to "have X2 with X1" in English, where X1 and X2 are phrase-pairs in turn. Note here that these phrase-pairs are not syntactically-decorated to impose some grammatical constraints on what should fill X1 or X2 like (NP or PP, etc). The hope here is that the language model component would discourage meaningless combinations. The extracted phrases define a weighted synchronous CFG as the one described in section 3.2 with one start symbol S and one non-terminal symbol X. The probabilities associated with each rule are estimated using relative frequency from the output of phrase-based models (no EM learning was done here, hence the name heuristics). Decoding, that is given a source string f, get the target string e, is modeled as a parsing process. To help scale and slightly determinize the

parse, some restrictions were enforced on the right-hand side of the SCFG grammar, the notable of which are: 1) No too adjacent non-terminal symbols are allowed and 2) There should be at most two non-terminal symbols in the right hand side of any rule. Those constrains has two ramifications: first the grammar can be parsed using CYK, and second, the extracted rules has the intuitive interpretation that it models phrase movement guided by functional words like with,have, etc. On a Chinese-English translation task, the system achieves a relative improvement of 7.5% over state of the art phrase-based models. Nevertheless, no deep analysis has been made to quantify what we loose by abstracting away the linguistic constraints on the phrases to be inserted in a given rule: like X1 must be NP or PP, etc. and for which language pairs this might not be a problem.

Apart from a restricted SCFG learnt heuristically as in Chiang (2005), recently Nesson, Shieber, and Rush (2006) gave an EM-like algorithm for unsupervised learning of synchronous tree-insertion grammar (which is similar to the synchronous tree-substitution grammar introduced in section 3.5)[1].

4.2 Learning Tree-String Transducers

In this section we describe approaches that leverage a monolingual parse tree at the target side (input to the noisy channel), therefore their TM learns to map a target tree T_e to a source string f, that is $P(f|T_e)$. It should be noted here that since the tree at the target side is not gold-standard, but the result of statistical parser, one needs to marginalize over all possible such trees. However, for scalability issues, this marginalization step is replaced with a max operator: that is the best tree of a statistical parser is taken as the gold-standard one.

Yamada and Knight (2001) and Yamada and Knight (2002)) gave a tree-string TM in the context of a Japanese-English translation task. The model is described by detailing how the English-tree (input to the channel) is stochastically transformed to the Japanese string. Three stochastic channel operations are employed as depicted in Figure 5: first the children of a given node are re-ordered stochastically, then for each node in the re-ordered tree, with a specific probability, a Japanese word *may* be inserted to its left or right. Finally the leaves of the resulting tree are translated to Japanese *independently* based on a probabilistic lexical translation table. The intuition here is that these steps would model mapping from SVO languages to SOV ones. To estimate the probabilities of these stochastic operations, an EM algorithm is used with an efficient dynamic programming implementation for the costly E-Step. Decoding is then modeled as parsing the source side (Japanese here) to get the target English tree, and then read the English sentence from its leaves. This is possible since in a nutshell we have constructed the Japanese parse tree for training Japanese string using the stochastic channel operations. The PCFG rules used to parse the Japanese string into the English tree are collected as follows. First, all the PCFG rules found in the English training corpus are collected and augmented with re-ordering, insertion and lexical translation rules. For instance if the English corpus has a rule of the form $V \rightarrow NP\ PP\ VB$, which can be re-ordered stochastically into $NP\ VB\ VP$ for the Japanese parse tree, then a new rule is added as follows: $V \rightarrow NP\ VB\ VP$ and its probability is set according to the re-ordering probability. Similar rules are added to model right and left insertions of the form $V \rightarrow XV$ and

1 we discovered this paper very late, so we did not have time to go over it, however we added it for completeness

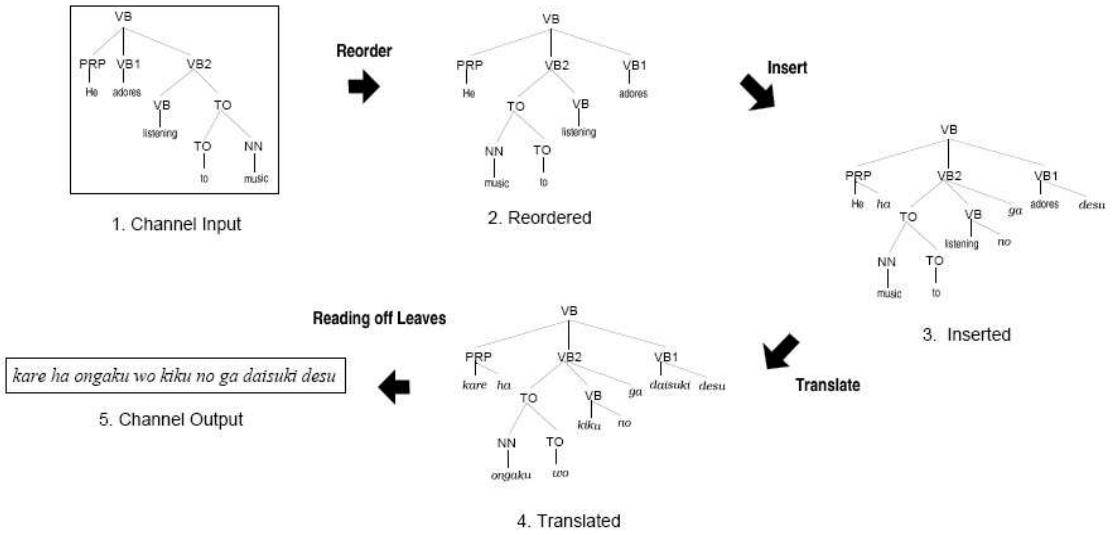


Figure 5
Illustrating tree-string channel operations, Yamada and Knight (2001)

$V \rightarrow VX$ along with $X \rightarrow foriegn_word$ with appropriate channel probabilities. Finally, lexical translation rules of the form ($englishWord \rightarrow JapaneseWord$) are added with their corresponding probabilities. The Japanese string is then parsed using these rule to get the English prase-tree. If a reordering rule is used during the parsing, then the resulting tree is reversed. For instance, in Figure 6, the top level production is labeled with the correct English order (remember that we added these rules to account for channel operations, so we can keep track of this order), therefore, to get the correct English tree, the second and third child are swapped.

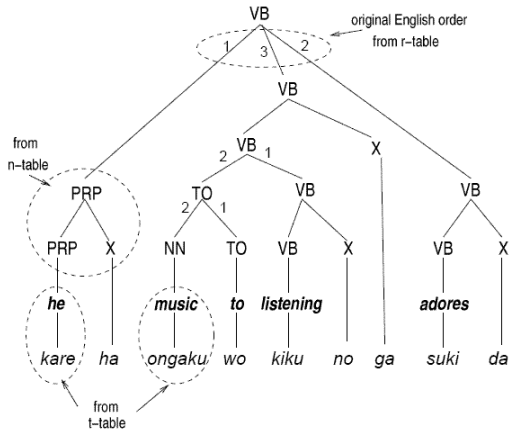


Figure 6
Illustrating decoding for the tree-string TM, Yamada and Knight (2002)

As noted in Fox (2002), this approach can account for language pairs that are different up to a child-reordering. However, it fails to capture translation patterns that fall outside this scope. This is true even for languages as similar as English and French. For example, English adverbs tend to move outside the local parent/children in environment, and the English word *not* is translated to the discontinuous pair *ne ... pas*. Moreover, as noted in Galley et al. (2004), English parsing errors also cause troubles, as a normally well-behaved re-ordering environment can be disrupted by wrong phrase attachment. For other distant language pairs, the divergence is expected to be greater.

It is interesting here to note that due to these factors, Zhang and Gildea (2004) found that ITG (unsupervised model)produces better alignment in terms of alignment error rate over the tree-string model described above when tested on a Chinese-English corpus. In another study Gildea (2003) extends the channel operations in (Yamada and Knight 2001) to allow for a richer re-ordering patterns that could cross brackets. For instance, the Yamada and Knight (2001)'s model can not model movements from SVO languages represented using the tree-bank convention [Subject [Verb Object]] to VSO represented as [Verb Subject Object]. To allow for this pattern, Gildea added a cloning operation in which a subtree can be cloned (copied) into another part of the tree stochastically. For example, in the tree fragment [Subject [Verb Object]], a cloning operation might copy the subtree rooted at Subject into the subtree rooted at Verb to have a flatter structure [Subject Verb Object] that could be then re-ordered using other channel operations. This approach results in a reduction of the alignment error-rate from 0.42 using Yamada and Knight (2002) to 0.3 on Korean-English corpus.

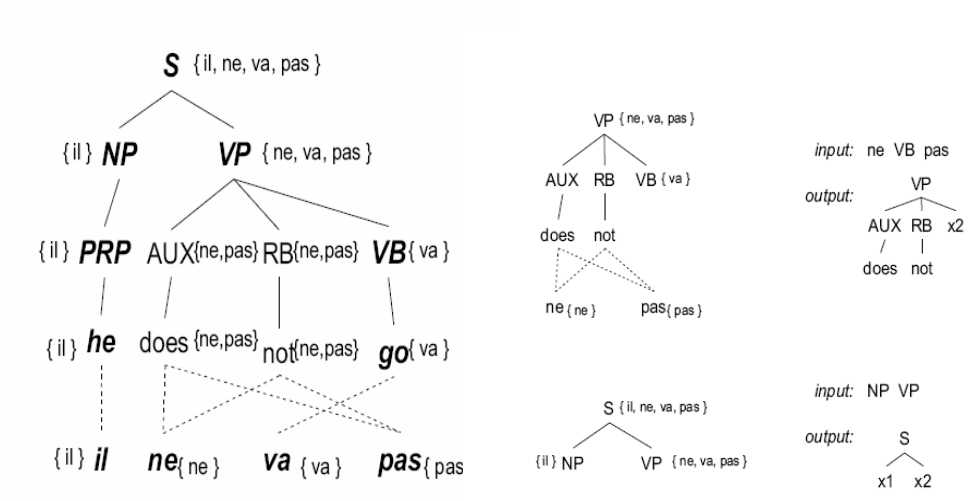


Figure 7
Illustrating rule extraction in Galley et. al. (2004). Left: the tree-string aligned graph. Right, example tree-frontier fragments and the rules extracted from them. The rule has the format that the input (left side) is rewritten as the output(right hand)

Another more principled solution has been proposed in (Galley et al. 2004). Their basic insight is that instead of using a noisy channel model and then extracting the parsing rules from it, one could learn a ***direct*** translation model that maps source strings to target trees. This new TM uses rules that condition on a larger tree-fragment that goes

beyond the parent-children one. The main contribution of their work is the introduction of a clever extraction algorithm for such rules. The extraction algorithm is explained with reference to Figure 7. A set of minimally, lexicalized rules are extracted as follows. First, the parallel corpus is aligned using Giza++ and the target (English) side of it is parsed using any statistical parser. This gives rise the alighted tree-string shown in Figure 7.a. Then the foreign (source) string is projected upward into the internal nodes of the tree, and a set of frontier nodes are identified. Those frontier nodes has the property that their span in the foreign side is exclusive (i.e. does not intersect with spans from nodes outside their family: ancestors and descendants). These frontier nodes are interesting as the system can learn to extract re-ordering pattern over them. This is valid since their spans can not overlap. After that, a set of minimal rules are extracted that have lexical and/or non-terminal nodes at one side, and tree fragment at the other side, as show in the figure. These rules are minimal in the sense that any other frontier fragment can be obtained as a combination of them. For instance, one rules that solved the movement of *not* to *ne ..pas* is as follows: $neVBpas \rightarrow (VP(Aux(does))(RB(not))x2)$ where $x2$ unifies with the second argument VB in the string. As in Yamada and Knight (2002), decoding is done via parsing the source string and reading the target one from the leaves of the resulting tree. However, the rule extracted here are not associated with probabilities, instead, the system is evaluated based on coverage: that is how much of the parallel English-French corpus can be explained using these rules. In other words, how many English parse trees can be derived from French strings. The authors reported a 100% coverage using at most 25 rule expansions on both English-French and English-Chinese corpus.

To give the above extracted rules a probabilistic semantic, two extensions have been proposed. In a follow up study, Galley et al. (2006) used EM to learn rule probabilities using a variant of the generic tee-transducers learning framework described in (Graehl and Knight 2004). The costly E-Step that sums over all possible derivations is efficiently computed using a derivation-forest. They also extended the rule extraction to deal with the case when there are unaligned words in the foreign side. However, the resulting system was not able to beat state of the art phrase-based alignment-template (AlTemp) approach on Arabic-English task. However, the authors noted that this result was expected as their decoder (which was work in progress) is not as mature as that of the AlTemp which can use more features and has a better parameter tuning algorithm. In another follow-up study Huang, Knight, and Joshi (2006) gave another parametrization of the translation system via a direct model, that is modeling $P(T_e|f)$ directly instead of passing through the noisy channel. They used a log-linear model to model rule weights that can incorporate other features such as a language model score and a sentence length penalty. The decoding algorithm presented there can give not-only the best target translation, but also the the non-duplicate k-best ones. Those K-best translations could be later used with n-gram re-scoring (a feature that was lacking in (Graehl and Knight 2004)). The authors reported a BLEU increase from 23.5 using Pharaoh (state of the art phrase-based model) to 26.69 on a Chinese-English translation task.

One can safely conclude now that syntax-base models do really help over phrase-based ones. However, there are two questions to be addressed here. First, is the resulting improvement a result of using a non-contagious phrases, or due to using a hierarchal decoder via parsing as opposed to a linear decoders in phrase-based models? Second, most of the above systems were trained on a relatively smaller number of sentences as opposed to operational phrases based models. Therefor, one tempting question is what would happen in large-scale training situations? would the methods above scale?

would it still beat phrase-based models, or phrase-based models can now learn longer phrase thus reducing the importance of hierarchical re-ordering?

To answer the first question, Zollmann and Venugopal (2006) used the extracted phrases from a phrase based system and decorated it with syntactic constituents obtained from a statistical parser. If a phrase does not represent a syntactic unit, it is assigned a default one named X. Then using Chiang (2005) heuristic, each phrase-pair is generalize with recursive partitioning (here the recursive phrase has a specific syntactic category). Decoding is achieved via chart-parsing the source side. The authors reported a BLEU score improvement from 30.61 to 31.76 over the Pharaoh phrase-based model. Thus empirically structuring phrases via annotation and generalization is the main factor that contributes to the superiority of syntactic approaches.

To answer the second question, Marcu et al. (2006) developed a large scale syntax-based system. The structure of the transfer rules is very similar to that of Galley et al. (2006), with the exception that rules are restricted to be phrase-based compatible, that is *only* maps contiguous pairs of words. For instance, the "ne ..pas" rule in figure 7 is not allowed in thier system. Moreover, they go around phrases that do not correspond to syntactic constituents by introducing pseudo-internal nods in the parse tree that span the non-constituent phrase. The reason for being phrase-based compatible is to be able to scale the system via using features from a phrase-based model. No EM-training was done here due to scalability issues, instead, the rule probabilities were estimated using relative frequency, and combined with a language model, and other phrase-based features using a log-linear model. The feature weights of this log-linear model were tuned using Och's (2003) Maximum BLEU training approach. The resulting rules were then binarized using (Zhang et al. 2006) and a bottom up CYK decoder was used to parse the source string. The authors reported a significant 1-BLEU score improvement (with 95% confidence) over the phrase-based baseline system on a Chinese to English task. That results proves that syntax-based model can be made scalable, however at the cost of loosing some expressive power. It is quite interesting to see how this approach will perform on languages that require non-contagious rules like English-French.

4.3 Learning String-Tree Transducers

In this section we describe approaches that leverage a monolingual parse tree at the source side (output of the noisy channel). We will describe two of such approaches here, both of them departs from the noisy channel approach, and learn a direct translation model—that is $p(e|T_f)$ — and both of them are also heavily inspired by phrase-based approaches.

in Langlais and Gotti (2006), the authors extended phrase-based models with the ability to model non-contagious phrases. The source side of the parallel corpus was parsed to produce a dependency-based parse trees, and then the two strings were aligned using Gize++. Phrases were extracted as in a normal phrase-based models, yet augmented with a new kind of phrase-pairs called Treelet-phrase pair (TP), which align a one-level source dependency tree (a treelet) with a target phrase. To extract those TPs, the source dependency trees were broken into treelets of depth one (head and its modifiers), see Figure 8. The part of the target string that align with lexical items in this treelet, is attached to form the TP pair. This TP pair can have gabs on both the source and target sides. TP probabilities are calculated using relative frequencies. A typical phrase-based decoder is then used in a left-right fashion by adding to the target string one phrase at a time. A complication arises if the added phrase-pair is a TP one, as it might contains gaps or shared lexical symbols. To deal with that, a constraint is added

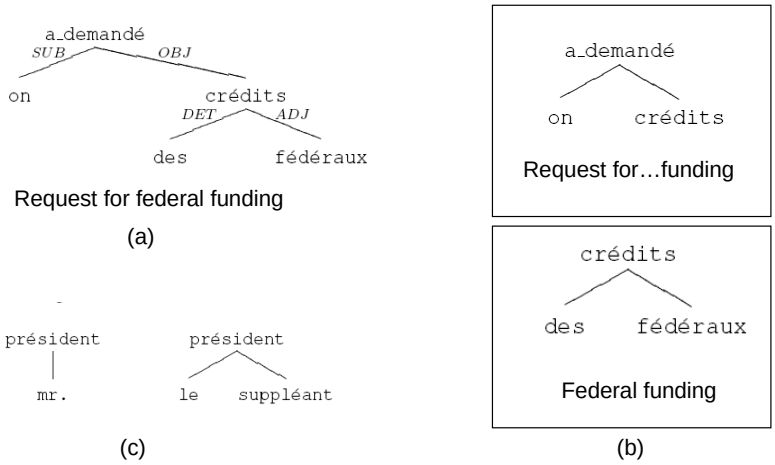


Figure 8
Illustrating the Treelet-Phrase translation approach. a) Source dependency tree and target string. b) extracted TP pairs. c) Example of incompatible treelets

that enforces that if two TP are added back to back, and they share a common source word, then this word must be a head word in one treelet and a modifier in the other one (i.e. the treelets can be linked together to form a bigger one, and therefore the gap on the first TP-pair can be filled with the second one). For example, in Figure 8.c, the two treelets are incomputable, while those in 8.b are compatible. The authors reported a slight improvement over phrase-based models that only use Phrase-Phrase pairs. While gaining slight improvement over purely phrase-based models, the authors looses one of the key strengthes of syntactic approaches, namely hieratical decoding via parsing as we reported in section 4.2.

In fact the system described above is a simplification of the one described in Quirk, Menezes, and Cherry (2005). Quirk, Menezes, and Cherry (2005) use the same approach, that is building a direct translation model $P(e|T_f)$. However, Quirk, Menezes, and Cherry (2005) use more sophisticated decoding and phrase extraction techniques. During training, the source side of the parallel corpus is parsed to get its best dependency tree. Then the corpus is alighted at the string level, and the source dependency tree is projected onto the target one. Several heuristics were employed to deal with one-one and many-many word alignment when projecting dependencies. Then each node is indexed with its order among its sibling (see Figure 9 for an example). Then treelet-pairs are extracted up to a given depth. Those treelet-pairs are the analogous to Phrase-phrase pairs in phrase-based models. To learn an ordering model between children of a given head on the target side (when stitching treelets together), a set of independence assumptions are made, and a probability distribution, parameterized as a log-linear model, is learnt over each children relative sibling index . This log-linear model depends on features of the child, head, and alighted source child its head. During decoding, the source dependency tree is constructed from the source string, and decoding proceeds then bottom up (as opposed to left-right in phrase-based models), where at each step, a treelet-pairs that match a source head is added to the partial parse (see for example

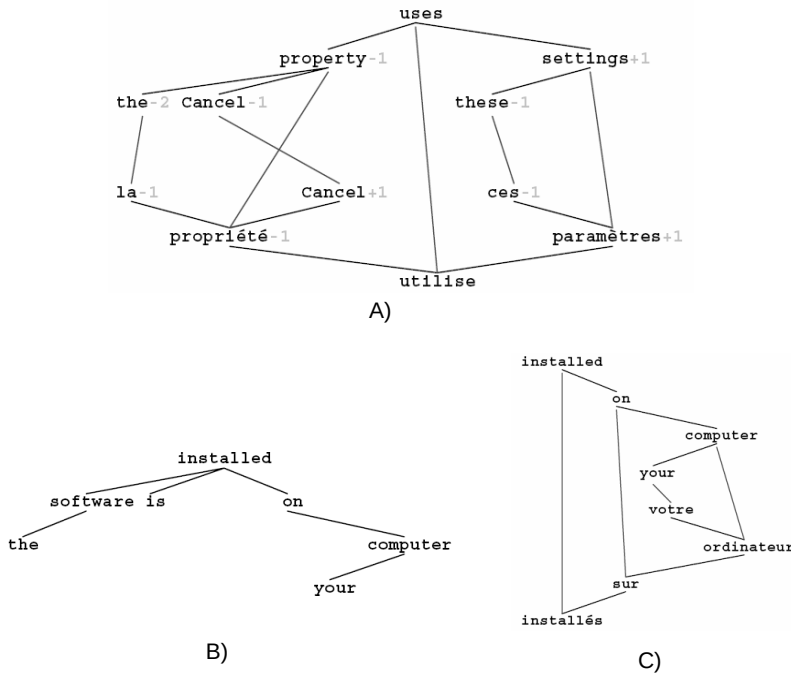


Figure 9
Illustrating dependency-treelet translation approach. a) Aligned, projected dependency trees. b) Translation input example. c) Example treelet translation pairs

Figure 9.c). When a treelet pair is used, the relative order of target nodes to their head is read from the treelet. The order model comes into play when stitching another treelets to the same head, for instance, when other treelets that match "is" and "the software" are aligned and added to the "installes" head,a relative order need to be predicted for French subtrees rooted at "is" and "the software". An exhaustive approach is used to try all combinations and each one is scored using the factored order model (a greedy approximation to this step is used as a fall back for large number of children). The authors reported a significant improvement over phrase-based models from 38.8 to 40.6 on an English-to-French translation task.

4.4 Learning Tree-Tree Transducers

In this section we describe approaches that leverage a monolingual parse tree at both target and source sides. Therefore, their TM learns to map a target tree T_e to a source tree T_f, that is $P(T_f|T_e)$. As we noted in section 4.2, no marginalization is done over all possible such trees, instead the best tree from a statistical parser is taken as the gold-standard at each side.

Tree-transducers have many applications that goes beyond machine translation (see Knight and Graehl (2005) for an overview). One interesting observation about this setting is that there are no hidden variables (except the tree-tree alignment), therefore, these models can be trained discriminatively as well as generatively. Approaches here differ on two dimensions: the first is the training procedure and the second is the syntactical representation employed at both sides of the noisy channel.

For the first dimension, as we mention above, these models can be trained generatively or discriminatively. Graehl and Knight (2004) gave a generic tree-transducer learning algorithm that utilizes an EM-Style algorithm augmented with a modified inside-outside dynamic programming scheme to scale the E-Step. The input to such system is the tree-pairs and the set of allowed transformation rules. These rules are language-pairs specific and are extracted from the corpus (in a similar way to that of **?**) or provided to the system based on linguistic knowledge about syntactics divergence between the language pairs under consideration (similar to the approach of Yamada and Knight (2001)). It should be noted that tree-transducers encompass as a subcase tree-string transducers. In fact in (Graehl and Knight 2004), the authors used their system to trains the earlier tree-string transducer of Yamada and Knight (2001). Generic and general as it may sounds, most authors prefer to train their systems using their own approach. One reason is due to the fact that the generality of the training algorithm in (Graehl and Knight 2004) might preclude case-specific optimizations.

Apart from this generative training scheme, discriminative approaches have been also used to train tree-tree transducers. Turian, Wellington, and Melamed (2006) describe a large scale discriminative training approach utilizing regularized decision tree ensembles. The expressiveness of the system was severely limited to isomorphic mapping between nodes and child-reordering, which reduces their system to the tree-string approach of Yamada and Knight (2001). However, this was made on purpose as the goal was to compare the discriminative training with the earlier generative one, and indeed the discriminative training outperform the generative one. However, does this mean that discriminative training should be the method of choice here? In fact it is too early to answer this question. For a more sophisticated tree-tree transducers, one need to employ a structure-prediction formalism (see for reference Klein and Taskar (2005)) with inference rules that allow the parts to be combined in a legitimate way. This might be not a non-trivial task to consider.

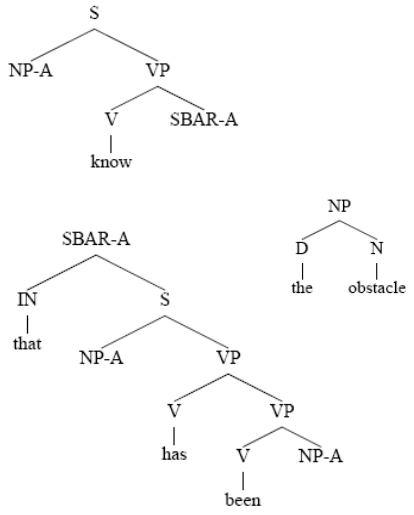


Figure 10
Illustrating the extended projection representation (EP) used in Cowan, Kucerova, and Collins (2006).

Cowan, Kucerova, and Collins (2006) described a more sophisticated tree-tree transducer that was trained discriminatively. The system translates from German-English. To solve the above mentioned inference problem, some simplifying assumptions were made. In fact the system learns to map a German clause, that has no embedded clauss, and only on verb, to an English tree-fragment called EP (extended projection). EP is an elementary tree with one verb, lexical functional word and holes for modifiers like NPs and PPs, see Figure 10 for illustration. To acquire training pairs, the English and German strings were parsed and then broken into clauses using deterministic linguistic rules. The corpus were also aligned using Giza++ and NPs and PPs on both side are mapped to each other. That induce a mapping between verb sub-categorization on clause-pairs. These clause-pairs act as the training instances, and the system learns to map German clauses into English EPs. To solve the inference problem, the mapping is modeled as a sequential decision problem, each of which is trained discriminatively and each one's candidate list depends on the decision taken on the most recent steps. For instance, the first step is to predict a spine for the EP part, followed by main verb, followed by mapping from German modifiers to the EP ones, etc. This approach makes the inference problem scalable. In the decoding phase, predicted English clauses are just combined together to produce the final translation. The authors reported a comparable BLEU score to phrase-based models but the surprising result is that human annotators did not favor their results over phrase based ones. We believe that this is because no hieratical organization were employed in their system, which as we mentioned before, the most powerful weapon syntax brings to the problem.

Apart from the training framework, along the other dimension systems differ on the type of representation used at each side. Here the candidates are phrase-structure and dependency trees. As reported in Fox (2002), the most coherent syntactical representation across languages is dependency trees, that is because its nodes tend to move with little crossing (either head-parent) or children crossing. However, Gildea (2004) presents a surprising results in which he fixed the transfer rules of a tree-tree transducer system, and compared the performance resulting from using both phrasal and dependency structures. He found that phrasal structures provide better translation accuracy on English-Chinese corpus. He reported that the result might be due to the inability of the dependency tree transducer to recover from errors made during parsing. We might add here that it might be also not fair to use the same transformation rules on both models. However, this result is interesting in two ways. First, it reinforces the well-known fact that modeling parser's error is quite important in the translation process. Second, language-pairs might make a huge difference in the conclusion made, as Fox's results were based on analyzing English-French corpus.

Apart from this negative result, most of the tree-tree transducers still use dependency trees. systems here differ on the form of the Translation model and transfer rules employed.

Ding and Palmer (2005) gave an approach based on recursively splitting dependency trees. As shown in Fig 11, a score is computed for each word-word alignment that takes into consideration some lexical and neighborhood features. The method goes in phases where at each phases only words with a given POS are allowed to align (this models Fox's result that NPs tend to cohere more than other structures). Once a word pair is declared alighted with high probability, the two trees are split at this pair and the method is repeated recursively. Figure 11 shows the resulting final tree-tree pairs. An order model (distortion model in Phrase-based terminology) is estimated based on the head and parent word. For decoding, the system uses a top down Veterbi-like algorithm to simultaneously find the best segmentation of the source dependency tree into treelets

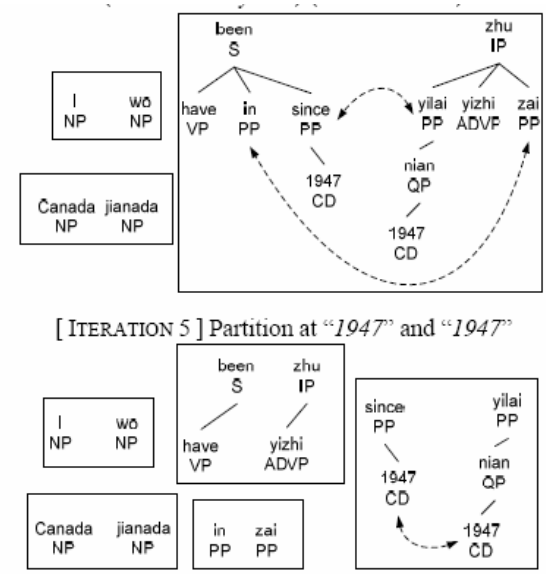


Figure 11
Illustrating hierarchical dependency tree partitioning.

as well combing it with the distortion model and a language model based on child-parent pairs. The system was compared to IBM-model 4 and outperforms it in both speed and quality, however, it was not clear why an IBM model was chosen as opposed to a phrase-based one?

Smith and Eisner (2006) also used dependency trees on both sides, but allowed for a more "sloppy" transfer rules that could capture a wider range of syntactic movements, which could address the result reported in Ding and Palmer (2005). The transfer rules used in Smith and Eisner (2006) are depicted in Figure 12. The system was trained using an EM algorithm where the E-step is handled via the inside outside algorithm. The system is still work in progress and early results show that using the full transfer rules gave the best alignment results on an English-German translation task.

4.5 Modeless Approaches: Post-Processing And Discriminative Re-Ranking

Here the translation model is not aware of syntax at all, however, syntax is applied as a post processing step. The method relies on discriminatively re-ranking the N-best output of any statistical-based machine translation system. In Shen, Sarkar, and Och (2004) and Och et al. (2003), a discriminative (mostly log-linear) classifier is trained using features that capture long-range syntactical dependencies. Nevertheless, negative results were obtained using this approach as this re-ranking step does not help improve the performance. However there is a caveat to this conclusion, namely, that if the BLEU score of the top N-list is comparable, then syntax can not help a lot. Putting it equivalently, if the reference translation is not recovered in the top N-list, then again the re-ranking step can not recover it.

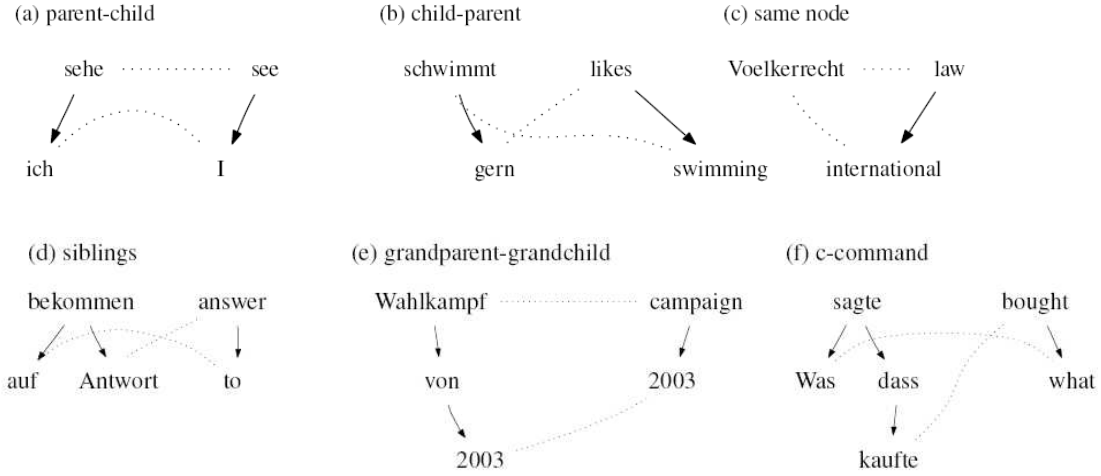


Figure 12
Illustrating transfer-rules (head-head alignment-models) in Smith and Eisner (2006).

4.6 Research Inspired by Syntax-based MT Models

Introducing Syntax into the machine translation systems, and its need for various monolingual components, have inspired researchers to improve state of the art methods in parsing and language modeling. While the transducers that were discussed above make the commitment to the best parse tree, many studies show that errors in parse trees can not be recovered from by the translation models (Ding and Palmer 2005). Therefore, many researchers started to investigate the use of the N-best list output of a parser. Moreover, in many syntax-based decoders, it is quite important to have an efficient representation of the top K-list parses to intersect them with a finite state language model. Huang and Chiang (2005) gave a recent efficient methods rooted in the framework of hypergraph parsing and used it successfully to decode his hierarchial phrase-based machine transaction system (Chiang 2005). Another important problem with syntax-based machine translation is the cost of decoding, which is mostly modeled as a parsing problem. Therefore many researcher either bias their synchronous grammar to be CYK-compatible to allows for efficient parsing. However many other systems like (Galley et al. 2006) uses expressive translation models that can not be made CYK-compatible. To help solve this problem Zhang et al. (2006) gave a binarizaton algorithm for synchronous grammar that maps int into a CYK-compatible representation.

As mentioned above, current (if not most) syntactic approaches make a commitment to the best parse tree output by a statistical parses, therefor one tempting question might be how parse quality affect translation quality? does the langue pairs make a difference? Quirk and Corston-Oliver (2006) examined the impact of dependency-tree parse accuracy in the context of their treelet-translation system (Quirk, Menezes, and Cherry 2005) discussed in section 4.3. The research was conducted on English-German as well as English-Japanese corpus. Different parsers' quality were simulated by changing the amount of data they train on. They found that their system is sensitive

to parser accuracy: with less accurate parsers, their system is comparable to phrase-based models, yet as the parse quality increases, their system significantly outperforms phrase-based models. They also reported that the quality of the translation for English-Japanese is much more sensitive to English parse quality than English-German. While this result is useful, the question is still not fully answered, what is the interplay between language-pairs, translation models,syntactical representation, and parser accuracy?

Another interesting direction for research that was inspired by introducing syntax into the SMT world, is reviving the question of the adequacy of the BLEU score as a translation measure. C.Callison-Burch and Koehn (2006) gave counterexamples to BLUE's correlation with human judgments. Recently, Liu and Gildea (2005) proposed to use syntactic features like head-modifiers and constituent labels, which was shown to correlate better with human judgment than BLEU scores. Along the same line Popovic et al. (2006) proposed to use morpho-syntactical features to automatically analyze the error of machine translation systems. They showed that the output of their system correlate well with human error analysis. We believe that this is an important area that deserve further attention, since the BLUE score is the measure of success, therefore if it doesn't capture fluency criteria in its evaluation, then hard research work can deem by BLUE to be not useful while it is really significant.

5. Discussion and Conclusions

In this report, we gave a survey to one of the most exciting innovation in the area of statistical machine translation over the last decades, namely syntax-based statistical machine translation. While statistical approaches to transaction have resulted in a breakthrough in the transaction quality over 1990s, they still produce quite serious grammatical errors that plague their usability and perception by humans. Adding syntax back to the translation process is not an innovation, rather it is just fall back to an old standard practice in human-intensive machine translation systems. However, the main breakthrough here is that no human labor is required beyond those effort already investigated to build monolingual syntactical parsers. In this report, we gave an overview that we hope will give the reader an appreciation of the importance of this problem as well as its significance. We hope also that this report will provide a sufficient background for new researchers interested in exploring this area, and arm them with the necessary knowledge in their endeavor. No claim made either explicit or implicit that this survey gave a comprehensive coverage of all the work in the field, a task which we should leave to the expert of the field. However, we did our best in trying to give a broad coverage to vast amount of work, while varying the level of details in each category based on how much knowledge we have and how much mature we think the direction is. However, with such an evolving field, a static survey will be of no avail, therefore, we hope to continue working and monitoring the work done in this interesting area, and continuously revising this report to both reflect our current understanding of the field, and the current state of the art methods.

Syntax is indeed, as we reported in this servery more than once, is a useful weapon only when used in the right way at the right level of abstraction. Therefore, one important direction which we believe is lacking is a systematic comparison of different methods over the same langue pairs as well as across different language pairs. Is there any correlations that can be reported? what constitute an easy language-pairs? and is this notion an absolute criteria or contingent of a certain syntactic theory?

Another area which we believe lacks deeper analysis is the link between synchronous formalism and practical systems rooted in the noisy channel framework. How

can we make this missing link? and is it useful? would it drive further practical models? Or just deepen our understating to why and when models will work and when and why not?. Also one noticeable observation is the dichotomy in the field between researchers working in formal theory and those working in building translation systems. We believe that breaking this dichotomy would results in many innovations. Moreover, also many of the systems described in this survey would benefit from recent progress in statistical machine learning, and therefor a close tie between researchers in both areas might be useful in accelerating progress in both sides.

6. Acknowledgments

We would like to thank the instructor, Prof. Noah Smith, for suggesting this interesting topic as well as for proposing an excellent initial set of references that helped us get started on this survey.

References

Abeillé, Anne, Yves Schabes, and Aravind Joshi. 1990. Using lexicalized tags for machine translation. *Proceedings of the 13th International Conference on Computational Linguistics*.

ALPAC. 1966. *Language and machines: Computers in translation and linguistics*. National Academy of Sciences, National Research Council.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computatinal Linguistics*, pages 263–311.

C.Callison-Burch, M. Osborne and P. Koehn. 2006. Reevaluating the role of bleu in machine translation research. *The European Associant for Computational Linguistic*.

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. *Proceedings of the 43rd Annual Meeting of the ACL*, pages 263–270.

Chiang, David. 2006. An introduction to synchronous grammars. Tutorial notes from a talk at ACL 2006.

Chiang, David, Adam Lopez, Nitin Madnani, Christof Monz, Philip Resnik, and Michael Subotin. 2005. The hiero machine translation system: Extensions, evaluation, and analysis. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Cowan, Brooke, Ivona Kucerova, and Michael Collins. 2006. A discriminative model for tree-to-tree translation. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Deng, Yonggang and William Byrne. 2006. Hmm word and phrase alignment for statistical machine translation. *Submitted to IEE transaction of Accustics, Speech and Language Processing*.

Ding, Yuan and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. *The 43rd Annual Meeting of the Association of Computational Linguistics*.

Eisner, Jason. 2003. Learning non-isomorphic tree mappings for machine translation. *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

Fox, H. 2002. Phrasal cohesion and statistical machine translation. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language*.

Galley, M., M. Hopkins, K. Knight, and D. Marcu. 2004. What's in a translation rule? *Proceedings of the Human Language Technology Conference – North American Chapter of the Association for Computational Linguistics annual meeting*.

Galley, Michel, Jonathan Graehl Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–996.

Gildea, Daniel. 2003. Loosely tree-based alignment for machine translation. *Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics*.

Gildea, Daniel. 2004. Dependencies vs. constituents for tree-based alignment. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Graehl, Jonathan and Kevin Knight. 2004. Training tree transducers. *Proceedings of the Human Language Technology conference – North American chapter of the Association for Computational Linguistics annual meeting*.

Huang, Liang and David Chiang. 2005. Better k-best parsing. *the 9th International Workshop on Parsing Technologies*.

Huang, Liang, Kevin Knight, and Aravind Joshi. 2006. Statistical syntax-directed translation with extended domain of locality. *Proceedings of HLT-NAACL Workshop on Computationally Hard Problems.*

Hutchins, W. John. 1995. Machine translation: A brief history. *Concise history of the language sciences: from the Sumerians to the cognitivists*, pages 431–445.

Joshi, Aravind and Yves Schabes. 1997. Tree-adjoining grammars. *Handbook of Formal Languages, Vol. 3*, pages 69–124.

Klein, Dan and Ben Taskar. 2005. Max-margin methods for nlp: Estimation, structure, and applications. *Proceedings of the 43rd Annual Conference of the Association for Computational Linguistics*.

Knight, Kevin and Jonathan Graehl. 2005. An overview of probabilisitc tree tranceducers for natural language processing. *Proceedings of the Sixth International Conference on Intellignet Text Processing and Computional Linguistics*.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. *Proceedings of the Human Language Technology Conference (HLT-NAACL).*

Langlais, Philippe and Fabrizio Gotti. 2006. Phrase-based smt with shallow tree-phrases. *In HLT-NAACL Workshop on statistical machine translation.*

Liu, Ding and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. *ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Marcu, Daniel, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. Spmt: Statistical machine translation with syntactified target language phrases. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Marcu, Daniel and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Melamed, I. Dan. 2003. Multitext grammars and synchronous parsers. *Proceedings of HLT-NAACL 2003*, pages 79–86.

Melamed, I. Dan. 2004. Algorithms for syntax-aware statistical machine translation. *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*.

Melamed, I. Dan, Giorgio Satta, and Benjamin Wellington. 2004. Generalized multitext grammars. *Proceedings of the 42nd Annual Conference of the Association for Computational Linguistics*.

Nesson, Rebecca, Stuart M. Shieber, and Alexander Rush. 2006. Induction of probabilistic synchronous tree-insertion grammars for machine translation. *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA 2006)*.

Och, Franz Josef, 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 Johns Hopkins 2003 Summer Workshop*.

Och, Franz Josef and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computatinal Linguistics*.

Popovic, Maja, Adrià de Gispert, Deepa Gupta, Patrik Lambert, Hermann Ney, José B. Mari no, and Rafael Banchs. 2006. Morpho-syntactic information for automatic error analysis of statistical machine translation output. *Proceedings of the HLT/NACCL Workshop on Statistical Machine Translation*, pages 1–6.

Quirk, Chris and Simon Corston-Oliver. 2006. The impact of parse quality on syntactically-informed statistical machine translation. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Quirk, Chris, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal smt. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 271–279.

Shen, L., A. Sarkar, and F. Och. 2004. Discriminative reranking for machine translation. *Proceedings of HLT-NAACL*.

Shieber, Stuart and Yves Schabes. 1990. Synchronous tree-adjoining grammars. *Proceedings of the 13th International Conference on Computational Linguistics*.
Smith, David and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. *Proceedings of the Workshop on Statistical Machine Translation*.
Turian, Joseph, Benjamin Wellington, and I. Dan Melamed. 2006. Scalable discriminative learning for natural language parsing and translation. *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS), to appear*.
Vogel, S., F. Och, C. Tillmann, S. Nieen, H. Sawaf, and H. Ney. 2000. Statistical methods for machine translation. *In Verbmobil: Foundations of Speech-to-Speech Translation, Wolfgang Wahlster (ed.)*, pages 377–393.
Wu, Dekai. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23:377–404.
Yamada, Kenji and Kevin Knight. 2001. A syntax-based statistical translation model. *Meeting of the Association for Computational Linguistics*, pages 523–530.
Yamada, Kenji and Kevin Knight. 2002. A decoder for syntax-based statistical mt. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.
Yarowsky, David, Grace Ngai, , and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. *Proceedings of the ARPA Human Language Technology Workshop*, pages 109–116.
Zhang, Hao and Daniel Gildea. 2004. Syntax-based alignment: Supervised or unsupervised? *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*.
Zhang, Hao and Daniel Gildea. 2005. Stochastic lexicalized inversion transduction grammar for alignment. *Proceedings of the 43rd Annual Conference of the Association for Computational Linguistics*.
Zhang, Hao and Daniel Gildea. 2006. Inducing word alignments with bilexical synchronous trees. *Proceedings of the International Conference on Computational Linguistics/Association for Computational Linguistics (COLING/ACL)*.
Zhang, Hao, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. *Proceeding of the Conference of Human Language Technology (HLT-NAACL)*.
Zollmann, Andreas and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. *In HLT-NAACL Workshop on statistical Machine Translation*.