Dependency Structure Trees in Syntax Based Machine Translation

Vamshi Ambati*
Language Technologies Institute,
Carnegie Mellon University

1. Introduction

Machine Translation (MT) has long been an unsolved problem and more so an interesting and engaging problem. MT deals with the translation of a sentence in a source language into a sentence in the target language preserving the meaning in its full detail. This requires the computer to encode the knowledge of both languages in a representation that can be used at runtime to translate a give input. In earlier years translation problem was approached in three main ways. One, memorizing all the source and target sentences ahead of time called Translation Memory (TM) and reproducing the translation at runtime by a simple lookup process (Hutchins and Somers 1992). Second, use complete source language knowledge for analysis and generate the translation according to the syntax of the target language (Nirenburg et al. 1992). Third, a more semantically motivated and simultaneous multiple language oriented approach that projects the task of translation into a common space, with a uniform representation of knowledge called Interlingua (Hutchins and Somers 1992). Although these approaches have been experimented in great detail in the last two to three decades, a more promising approach called Statistical Machine Translation (SMT) with firm support from statistical and mathematical grounds has taken prominence in the last decade (Brown et al. 1993) (Koehn, Och, and Marcu 2003).

Statistical Machine Translation (SMT) approaches use massive amounts of corpus to learn translation models at sub-sentential level which can generalize well to unseen data, unlike the TM. SMT addresses the problem of translation as a noisy channel paradigm where the channel model is usually called the 'Translation Model' and the source model is called the 'Language Model'. The translation model is estimated under a generative story for word correspondences. The language model estimated as ngram sequence models with markov assumptions. This formulation of the translation problem is language agnostic and does not assume any kind of syntax information from either the source or the target. Translation output produced under the SMT framework tend to usually be fragmented and context insensitive. With rigorous estimation techniques and heuristics for incorporation of context (Koehn, Och, and Marcu 2003), we have seen improved performance over the past few years. But, the quality is still far from human consistent. This is primarily to do with the fact that these models are learnt under no syntax scenarios and so are ill-informed about the phenomena of divergences that occur across languages. A recent body of approaches have looked into the incorporation of syntax at various phases of translation process with reasonable success (Yamada and Knight 2001; Chiang 2005). With more researchers looking into intelligent ways

* Adv MT Seminar Course Report, Spring 2008

of syntax motivated approaches to MT, the future seems promising for MT in particular and NLP approaches to syntax as such.

Syntax can be of any form, ranging from part-of-speech annotations to complete parse trees from different grammar formalisms. In this particular report, we survey approaches that look in particular at the incorporation of dependency structure parse trees into the translation process. In particular we look at work related to syntactic translation models and their estimation. We also discuss approaches to decode using these new translation models.

The rest of the report is organized as follows. In the Section 2 we discuss syntax based machine translation in general and point to the juncture of importance for this report. In Section 3 we discuss the Dependency formalism that we will be using for our syntax part in MT. In Section 4 we survey work in improved translation models using dependency structure trees. In particular we survey two kinds of approaches - one, where parse trees are used from both the language and two, where only parse tree for source side is used. In Section 7 we discuss some of the approaches for decoding using these improved translation models. We highlight some of the problems that make decoding difficult and some approaches used to address them and produce better translations. Finally we conclude by proposing some of the immediate and future research problems related to this area, in hope to share the insight aggregated in our survey.

2. Syntax in Statistical Machine Translation

2.1 Statistical MT

The IBM Models (Brown et al. 1993) were introduced to model the translation problem in a statistical framework. Also called the alignment models, these models ranging from 1 to 5 were proposed with increasing sophistication to explain the various divergences that occur between languages. This helped improve the word level correspondences which are important for the overall translation. However there were a few problems that the word level translation models were unable to model like local context information. This led to research in the area of Phrase based SMT (PBSMT), which moves away from the fundamental limitation of word-based models. PBSMT makes phrases as the first class entities in translation, thus by passing the need to synthesize translations word-word. Sometimes even though a phrase can be translated word-word into a perfectly grammatical translation (French: the cabbage - le chou, chou), it may not be a valid one for that language. Even when a phrase appears compositional the incorporation of context information helps (Quirk and Menezes 2006). This not only allows to construct fluent translations for sub-sentential fragments, but also inherently captures the word reordering within the phrases - also called 'local reordering'.

SMT approaches are good with exact substring matches, that are contiguous in nature, but discontiguous phrases like the 'ne pas' construction from French, can not be handled in a generalized manner, unless all the possible variations are encoded in the phrase table. The Hiero MT system (Chiang 2005), addresses this with some limitations in order not to explode the phrase table sizes.

Finally, global reodering which happens betweens chunks or phrases separated by a distance although quite common across languages, is difficult to handle in Phrase based SMT. The distortion models that guide the reordering of phrases in SMT, are difficult to model precisely over long distances. They also do not generalize well as they are currently modeled based on lexical items. Complete reordering of all phrases in a sentence becomes difficult even with small sized phrases as the possible combinations

already are large assuming an average length of phrase as 2, which is common in SMT. Recent approaches to incorporating syntax apply a lot of restrictions to limit the space of phrasal reorderings. Popular among them are the ITG (Wu 1997) and (Chiang 2005). Since they are not linguistically motivated phrases, applying SVO,SOV sort of constraints becomes difficult.

2.2 Syntax in MT

Syntax can address the global reordering problems discussed in section above. There has been an increasing interest in recent years in methods that incorporate syntactic knowledge into MT. Syntax-based reordering rules can be used as a pre-processing step for PBSMT (and other approaches), to decrease the word-order and syntactic distortion between the source and target languages (Xia and McCord 2004). There has also been approaches to re-rank the final n-best list of hypotheses produced by standard MT systems and pick the linguistically motivated hypothesis, which is more likely to correspond to a fluent translation of the source. Many linguistic features have been applied over an n-best list of the order of a few thousand and improvements have been noted (Och et al. 2004),(Shen, Sarkar, and Och 2004). More interestingly some approaches have tied up syntax with the translation model more closely. A variety of hierarchical and syntax-based models, which are applied during decoding, have also been developed (Yamada and Knight 2001). Many of these approaches involve automatic learning and extraction of the underlying syntax-based rules from data. The underlying formalisms used has been quite broad and include simple formalisms such as ITGs (Wu 1997), hierarchical synchronous rules (Chiang 2005), string to tree models by (Galley et al. 2004) and (Galley et al. 2006), synchronous CFG models such (Xia and McCord 2004) (Yamada and Knight 2001), synchronous Lexical Functional Grammar inspired approaches (Probst et al. 2002) and others.

In this report, we will not further discuss about these approaches in any detail, and will only concentrate on approaches that use dependency formalism inside a Syntax based MT system.

3. Dependency Structure Trees

Dependency structures represent the grammatical relations that hold between constituents. They are more abstract when compared to syntactic trees, in the sense that they do not restrict or prescribe a particular word order, nor do they have an explicit notion of 'constituentness'. They are more specific in terms of semantics and the notion of relations across words is explicit.

A dependency tree for a sentence is a directed acyclic graph with words as nodes and relations as edges. Each word in the sentence either modifies another word or is modified by a word. The root of the tree is the only entry that is modified but does not modify anything else. The relation between any two words in the tree can be given as a 'parent-child' or 'modifier-modified' relation or 'head-modified' relation or 'governer-governed' relation. The more specific relation that the two words participate in is given as a name on the edge connecting the nodes. The direction of the arrows are usually from the parent to the child , but the opposite is also valid, given the notation is agreed upon and is consistent for the entire tree. More formally, the dependency structure tree can be expressed as follows: Given a sentence $S\{w0....wn\}$, a set of edges or dependencies $E\{e1...en\}$ are defined such that each ei connects two words in the sentence, and w0 is a root word that only connects a word to another word.

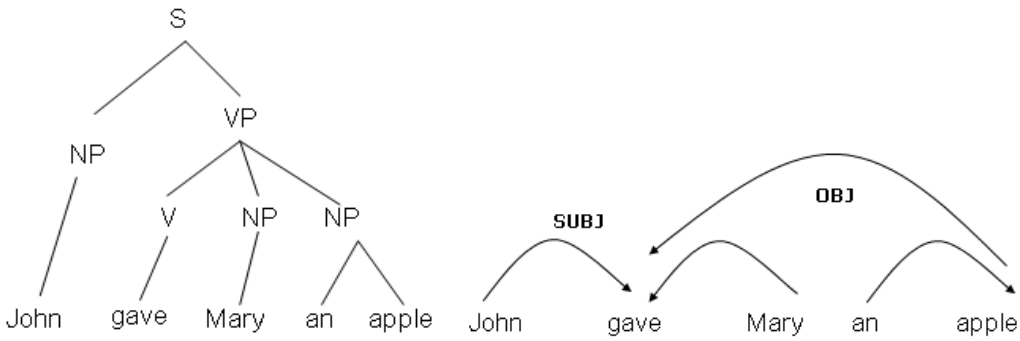

Figure 1
Phrase structure tree and the corresponding Dependency structure tree

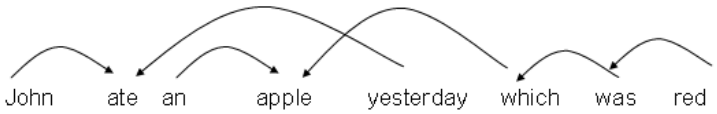

Figure 2
An example of a Non-projective depdenency tree

The dependency tree in figure 1 are labelled,that is, edges are annotated with labeled relation names. A labeled dependency tree is useful to understand the grammatical functions of words and the roles they play in a sentence. For example, 'John' is the subject of the sentence that is headed by 'ate' and apple is the object of the same sentence. Labeled dependencies have proven to be useful in downstream NLP application tasks like question answering and discourse analysis. Although labeled dependencies have been used for improving MT evaluation task (Owczarzak, van Genabith, and Way 2007), no significant work has been put into the translation improvement as such. In this report all the work surveyed looks only at unlabeled dependency trees, unless otherwise specified.

In figure 1 though we have not represented the dependency structure as a branched tree that corresponds in structure with the phrase structure tree, it is a common way to represent it in a linear order preserving the sequence of words positions in the string. In this representation it is also much easier to see the long distance interaction of words in a sentence. It is also easy to explain one of the fine distinctions of the variations of dependency structures called projectivity. All sentences in natural language can be explained by a dichotomy of the dependency tree structures called - projective trees or non-projective trees. A projective tree is one that when written out with words in a predefined linear order, do not have any crossings between any of the edges. The dependency tree in figure 2 also happens to be a projective tree. We can also say that a tree is projective if and only if an edge from word w_i to word at w_j implies that for any word w_k in between, w_i is a direct or indirect ancestor.

For the English language, most of the parse trees are projective. However, there are certain examples in which a nonprojective tree is preferable. Consider the sentence, John ate an apple yesterday which was red. Here the relative clause 'which was red' and

the object it modifies 'an apple' are separated by a temporal modifier of the main verb. There is no way to draw the dependency tree for this sentence with no crossing edges, this is illustrated in Figure 2. Although this distinction is very important for the research in parsing, as the algorithms change based on the kind of tree (McDonald, Crammer, and Pereira 2005), it is not very particularly taken care of in Machine Translation. As we will see the translation models being built are agnostic of the type of the parse tree.

4. Dependency based Translation Models

Machine Translation between a majority language on source side and a minority language for the target or vice versa, is a more common scenario. In such cases where we have only a parse tree for one side of the language pair requires us to use some sort of projection technique based on the intuition of the languages to do an annotation transfer.

Given a parallel corpus the projection technique is applied in the following way to induce structure for the target sentences. The source language text can be manually annotated or a dependency parser can be used to annotate the text. The usage of some sort of correspondences between the words in the parallel sentence pairs is pre-identified. Usually this is the viterbi-best alignment, but n-best alignments could be applied. This information is then used along with certain heuristics to do the annotation transfer. The quality of the correspondences decide the accuracy of the transfer. No two languages are completely identical in syntax. There are bound to be divergences and therefore the annotation transfer can not be guaranteed to produce perfectly legitimate syntax on the target side. This calls for some amount of post-processing depending on the type of annotation and the kind of language.

Direct correspondence assumption (DCA) is first introduced in (Hwa et al. 2005). It is originally used for dependency relation projection. The assumption states that a certain kind of dependency relation is preserved under some condition through direct projection. DCAs usually come from empirical studies of phenomena in bilingual corpora (Fox 2002). They are the basis and start for most of the syntax projection problems. However, DCA also tends to create very noisy annotations for target language because it is too simple and deterministic when considered the complexity of real languages. Thus, probability models are usually used on top of or instead of DCA assumption for projection robustness (Smith and Eisner 2006).

The projection techniques using DCA approach, result in creating syntax for the target language which is isomorphic in structure to the source language tree. Translation models built from such corpus can be classified in isomorphic translation models. We can immediately notice that such an isomorphism can not always exists between two languages. Even if there is a great degree of isomorphism, it is often difficult to notice it in parallel corpus, due to the free nature of translation or the noisiness in translation quality. For example the construction in example Figure 5 shows non-isomorphism between two sentences. Learning translation models in such a scenario is a challenging problem as identifying the alignment across subtree units is extremely difficult. We classify models with deal with this scenario as 'nonisomorphism based translation models' and discuss in detail in Section 6. In the next few sections we will survey the building and decoding in translation using dependency based translation models.

5. Isomorphic Translation Models

In this section we primarily survey research in incorporating dependency trees into the task of translation, where isomorphism is explicitly assumed between the two language pairs.

5.1 Minimum Tree cover based translation

(Lin 2004) discusses a path based transfer model for machine translation. The model is trained with word-aligned parallel corpus where the source side consists of dependency trees. The training algorithm extracts a set of paths on the source dependency trees and determines the corresponding translations of the paths using word alignments. The outcome of training is a set of transfer rules that given a certain path in the source, provide the equivalent translation fragment in the target. Rules are extracted where all the words in the path have translation links. A preposition is allowed to be unaligned. Each rule not only encodes the dependency relations for the target side, but also the relative linear order among the nodes in the fragment. The algorithm also extracts two kinds of spans for the node in the source tree. A 'head span' which is the word sequence aligned with the node, and the 'phrase span' which is the maximal closure span of all the subtrees below the node are also extracted These spans are used at rule extraction to prevent ill-formed rules. In order to add power of generality to the translation model, some of the rules thus extracted from each sentence pair are generalized. The generalization is very constrained so as to not explode the number of rules extracted. Currently it is only the end nodes of each tree fragment that are generalized to its part-of-speech. In order to assign probabilities to each of the rules extracted, they compute the $P(T_i/S_i)$ by relative likelihood scoring with a smoothing constant to reduce noise.

Translation with a model extracted above is defined as , given a source sentence , parse it to produce a set of paths from its dependency tree. Then find a set of transfer rules that 'cover' the entire dependency tree and produce a set of tree fragments on the target. 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 tree. The translation is read off this target tree. Main challenge here is to be able to merge the tree fragments obtained for different paths into a single tree that has highest probability. The tree fragments combine together to form a tree $T*$ = $argmax P(T_i/S_i)$. Merging is usually done on the target nodes that align with the same source node, and that do not introduce a cycle in the target tree. Regarding ordering of the words in the tree fragments - incase the 'transfer rules' come up with tree fragments that are unique or from same tree its not a problem, but if they are from different tree examples, then some relative closeness estimates are used.

5.2 Treelet pair approach to translation

(Quirk, Menezes, and Cherry 2005) employs a source language dependency parser and projects the dependencies based on word level alignment. After projection, they perform a re-attachment phase where all words that rupture the linear sequence of the target sentence in the dependency tree are re-attached to the lowest possible node, where target order can be preserved with respect to the siblings. While one-many alignment come free by this, in many-one alignments they make the right most word the head and others the depdendents. As the source side language here is always English, the assumption stated here applies, as English is a head-final language. However, one

$((men_1)\ and_2) \mapsto ((hommes_1)\ et_2)$

$(and_1\ (dogs_2)) \mapsto (et_1\ (chiens_2))$

$((men_1)\ and_2\ (dogs_3)) \mapsto ((hommes_1)\ et_2\ (chiens_3))$

$(((tired_1)\ men_2)\ and_3\ (dogs_4)) \mapsto ((hommes_2)\ et_3\ (chiens_4)\ (fatigues_1))$

$((men_1)\ *\ (dogs_2)) \mapsto ((hommes_1)\ *\ (chiens_2))$

Figure 3
Sample treelets extracted from English Spanish parallel sentences

has to make sure the assumption carries well before applying to a different language. In case of unaligned words they find the set of indices that overarch the unaligned word in any direction and make it depend on the closer index. After this, the corpus is now ready to extract translation rules, which are also called as treelet pairs in this work. In treelet extraction phase all possible treelets of the source are extracted and only those treelets are kept where the projected target treelet corresponding to it is also a connected graph. Counts are kept track of for maximum likelihood estimation (MLE). Special case of treelets with wild cards are allowed (Figure 3), and treelet pairs without head (single words etc) are also extracted.

(Quirk, Menezes, and Cherry 2005) generalize beyond a simple noisy channel model to a log-linear framework, to incorporate a variety of models along with the channel model and the target model. While channel model is a simple MLE estimate of target treelet given source treelet, it puts them at disadvantage due to data sparsity. So they also include a word-word model which here is an unnormalized version of Model1, probabilities in both directions. The target language model is a simple ngram LM, which works on the target language string from the dependency tree. However, one has to make sure that the ordering of the dependency tree for target is appropriate given that we are synthesizing it from tree-let pairs. Therefore they concentrate exclusively on designing on "Order Model" which scores all possible reorderings of the target language tree to give an appropriate ordering for the tree. This was the most comprehensive work done in the area of dependency trees for SMT thus far. The decoding aspect of their translation system is more detailed in insight when compared with other systems and it is currently on par with the state-of-art baseline systems for MT. They apply a dynamic programming motivated search technique to exhaustively search the hypothesis space for the right translation. We will give a more thorough detail of the approach when we discuss more about decoding using such translation models in Section 7.

5.3 Generalized treelet pair approach

Existing SMT approaches that are trained on a particular data do not generalize well to a new domain. There is a precipitous drop in quality, as the phrases which provide the reordering and contextual translation simply don't match out-of-domain text. (Menezes and Quirk 2007) introduces a new reordering model based on 'dependency order templates' that generalizes well in such cases. This work is done in the context of the treelet pair approach discussed above.

Treelets are allowed to match more loosely, and unmatched children are placed by exploring all possible orderings and scoring them with order model and language model. Although this decouples 'reordering' and 'word choice' and is exhaustive and

guarantees optimality, the tricks to curb complexity make this approach susceptible to search error. A dependency order template is an unlexicalised part-of-speech transduction rule mapping dependency tree on source side to unlexicalized target tree. These templates are not used right away in decoding. While exhaustively exploring all possible reorderings in the treelet approach, they only consider those which atleast match some 'ordered dependency template' thus cutting search space and speeding decoding.

6. Non-Isomorphism based Translation Models

In this section we survey approaches where dependency translation models are built using trees provided for both the sides of the language pair. Given this information, it is should be easy to see that isomorphism assumption no longer exists, but at the same time it is a challenging problem to identify the correspondences between the nodes of the tree pairs to build relevant translation models. We will first look at an approach that induces trees synchronously for both sides when no extra syntax is provided, and then uses this dependency translation model in a transducer framework. We will then look at other approaches where trees are provided for both sides, but no subtree alignment information is provided.

6.1 Head Transducers based Translation: Alshwai

(Alshawi, Douglas, and Bangalore 2000) propose a dependency transduction model for translation in terms of a set of weighted head transducers. A head transducer unlike a finite state transducer which consumes input from left-to-right, consumes it "middle-out". The output from such transducer is also built middle-out at positions in the output string. Therefore the formal definition of a head transducer, in addition to a regular finite state transducer contains position information for consuming input symbol and position information in output string for producing the symbol.

When head transducers are applied to the translation task, we call them 'dependency transduction models' which can be thought of in this manner. Each of the source strings with a 'head' word and left and right dependent words get applied on by a transducer to produce the corresponding 'head' and left and right counterparts in the target language. A collection of such transducers recursively decompose the source and target strings, to explain the dependency structure between the two languages. The model produces synchronized dependency trees where each local tree is produced by a particular transducer. In Figure 4 each pair of source and target trees are generated in this manner. The cost of a derivation in such a framework is the sum of the individual costs of each of the transducer and so one chooses to pick the lowest cost tree that can be constructed using the dependency model. The learning of costs or weights for each of the individual transducer is done on an unannotated corpus of source and target strings. The training approach first computes cooccurence statistics from the sentences and searches for an optimal hierarchical alignment using it. Hierarchical alignment is performed using a dynamic programming algorithm that optimizes a cost function that involves, word translation probabilities as given by the co-occurences and also relative distances between head and dependents in both source and target. They use this alignment to construct head transducers that can explain the sentences, with a maximum likelihood estimation technique.

One drawback of the approach is that the training algorithm does not necessarily learn dependency structures that are linguistically motivated, but rather those that try

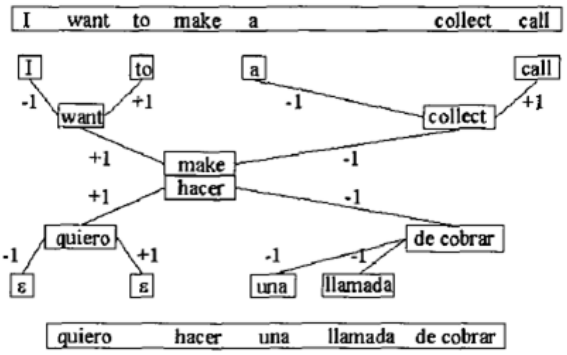

Figure 4
Head Transducers for Dependency translation models

to explain the synchronous phenomenon between the two languages. However, the authors observe that in most cases it corresponds very closely to individual dependency structure trees for both the languages. Also, since it was a very early work the authors do not compare it with traditional state-of-art approaches like IBM Models. The approach also does not scale for longer sentences, and so they apply their system on small sentences less than 20 words in length.

6.2 Synchronous Dependency Insertion Grammars: Marcu

(Ding and Palmer 2005) propose a version of synchronous grammar version of the Dependency Insertion Grammars (DIG) called Synchronous Dependency Insertion Grammars (SDIG). A DIG is a generative grammar formalism that captures word reorder phenomena within the dependency formalism. An SDIG therefore is a generative way of explaining the derviation process of two trees for both languages. The basic units of their grammar, elementary trees (ET) are sub-sentential structures containing one or more lexical items. The derivation process is proposed being 'isomorphic' at the cross-lingual level and any non-isomorphism is encapsulated within the elementary tree fragments. In Figure 5 we notice that although the ET are non-isomorphic, there is an overall thread of isomorphism between the trees. This is the underlying assumption of the generative process of the grammar.

(Ding and Palmer 2005) also propose an approach to induce grammar rules for SDIG from parallel dependency trees. One important limitation to note here is that the SDIG does not explain crossing dependencies or other divergence phenomenon like head-switching. These divergences get implicitly handled if they exist within an ET, but can not be represented explicitly in rules. The induction algorithm extends a hierarchical tree partitioning algorithm proposed in (Ding and Palmer 2004). The authors consider the partitioning of trees conditioned upon the label of the word. The intuition is to start with 'Noun phrases' which are generally found to be cohesive in nature and iteratively decompose the tree based on heuristics and features like word translation probabilities as given by IBM Model1, part-of-speech of likely matching words, inside-outside probabilities of trees etc. They use a graphical model to combine the multiple features and make a decision on whether to decompose synchronously at a node pair or

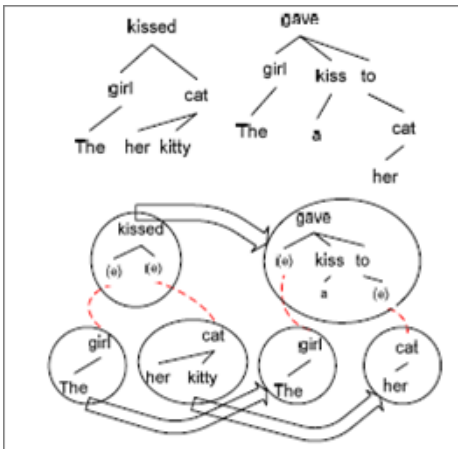

Figure 5
Tree to Tree transduction in SDIG

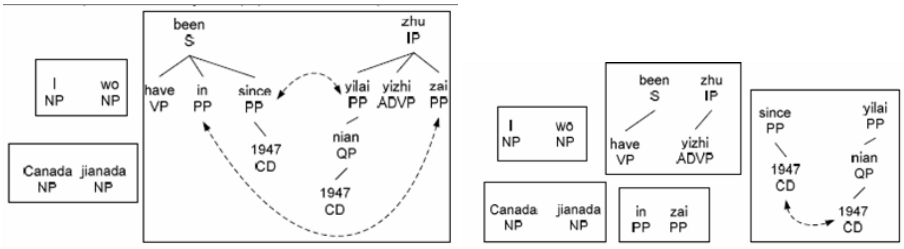

Figure 6
Sample example of the iterative induction process of SDIG

not. In Figure 6, one can notice the iterative decomposition of the trees into synchronous subtree fragments.

The MT system built for such a model, takes as input a sentence and first parses it to obtain a dependency tree. The tree is then decomposed into all possible elementary trees, according to the pipeline discussed above. The elementary trees as transferred automatically to obtain the target elementary trees which are then combined together to obtain the translation. One can look at the tree-to-tree transduction process for MT as an optimization process that, given a foreign sentence obtains the best translation for it.

6.3 Synchronous Tree Substitution Grammar

(Eisner 2003) makes a proposal for a Synchronous Tree Substitution Grammar (STSG) for Machine Translation. The aim is to learn good translation models from non-isomorphic trees, which could be caused by loose translation, divergences or noise in corpus. STSG is a Synchronous TAG (Shieber and Schabes 1990) without the adjunction operation. The authors note that dropping the adjunction operation does not lose expressive power in modeling string pairs, but only makes parsing faster as TSG can be parsed as fast as CFG. They propose methods to learn and score these decompositions using dynamic programming algorithms. This is work in progress and it still awaits to be seen how these algorithms for reestimating the tree pairs and parsing them

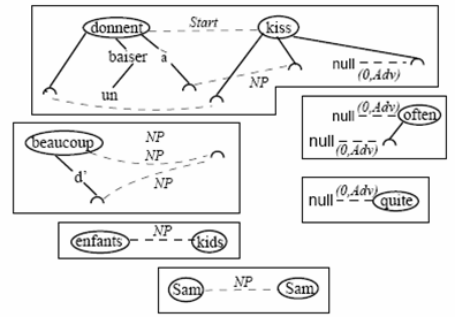

Figure 7
Synchronous Tree Substitution Grammar

effectively using STSG improves the translation quality when compared to state-of-art syntax based systems. The algorithms are generic and can also be applied to phrase structure trees without modification.

An example in Figure 7 shows the non-isomorphic nature of trees along with their possible decompositions into elementary trees that are non-isomorphic too Elementary trees represent idiomatic translation pieces with frontier nodes that can have unfilled roles in them.

6.4 Quasi Synchronous Grammars for Nonisomorphic trees

We have seen so far that approaches for learning translation models, require an alignment stage for reliably decomposing the trees at appropriate decomposition points. The alignment algorithms play a cruicial role in the accuracy of the translation models built. Clearly the direct correspondence assumption type of techniques (Hwa et al. 2005) do not work with non-isomorphic tree structures, where the corpus consists of diverging translations or free translations that only correspond in meaning, but not choice or structure of lexical items. (Gildea 2003) propose cloning operations to perform alignment in a tree-to-string and tree-to-tree scenario. Their algorithms produce good results in alignment, but it is still to be tested out on an end-to-end syntax based MT system for translation quality. (Smith and Eisner 2006) proposes a sloppy syntactic alignment model for dealing with the parallel corpus loose translation problem. They do not require a node in a target tree to be a well-behaved translation of its node in source tree. This means that any node can align to any other node among the trees, allowing for the data to empirically decide the alignment, rather than enforcing any linguistically motivated biases. They pose the problem as a joint conditional modeling problem of the target tree and the alignment given the source tree. This notion of looseness in attachment is modeled as a selection preference in a synchronous grammar called 'Quasi Synchronous Grammars'. A quasi synchronous grammar is a monolingual grammar that generates translations of a source language sentence. Each state of this monolingual grammar is annotated with a possibility or 'sense' which here is a set of zero or more nodes from the source tree. To overcome the exponential nature of the alignment problem , they pose constraints based on type of nodes and length etc, to keep the space tractable. They parameterize the model and train it on bitext under the EM framework. To start with, the initial lexical probabilities are taken from IBM Model 4. They do not demonstrate the use of this in a MT system, but apply Quasi Grammars

to unseen text and show that a better fit to bilingual data is achieved by allowing greater syntactic divergence.

7. Decoding Approaches

Decoding in a statistical translation system is the task of finding the best possible translation for an input sentence based on the translation models. The accuracy of this task depends largely upon the discriminative power of the models and the algorithm that guides the search in the right directions, without spending too much time exploring the irrelevant search space. Decoding while using syntactic translation models can take place in two possible ways. When a tree is not provided during run time for the input, the problem becomes similar to a left to right decoding without syntax, or approaches try to create both source and target parse trees in lock-step process. However, the most common case in literature is the scenario where a dependency tree is available for the input during run time. Advantage of using syntax from a monolingual parser or tools of the like is that we can cut down on those parts of the search space which are less likely to produce a good translation. The assumption is that, the space of translations explored with the source syntax in perspective is likely to produce a good translation, given the translation models that are built from similar syntax. The flip side of these approaches, that use a syntactically analyzed input, is that it limits the usability of these approaches to a few scenarios where the source language always has a parser.

When translating starting with a dependency tree or any syntactic parse, the standard left to right decoding as in the string case does not work. The task is to start with the tree either top-down or bottom-up and produce target language trees correspondingly, which can be produced by various transformations and orderings on the tree. The main challenges here are to handle incomplete sub-tree orderings, overlapping subtrees, discontinuous subtrees and the combinatorially explosive search space. We will look at how the work so far addresses decoding to address these finer challenges.

7.1 Search Based Techniques

In (Quirk, Menezes, and Cherry 2005) several approaches were explored for decoding using input dependency tree parse. An initial attempt was to exhaustively search through the source input tree and translate bottom-up finding best translations matches for each subtree. A match in this case means that there is a treelet whose source side has close isomorphism with the subtree, where the lexical items, the part-of-speech tags, the ordering of the children and the head all match exactly. The matching is performed by a bottom-up traversal of the tree finding translations at every subtree. The translations of a subtree are reused in translating a higher level node. Since (Quirk, Menezes, and Cherry 2005) work with treelet pairs, it is quite effective when the translation model is rich enough to contain all the subtrees that an input decomposes into. But we notice that in many cases we find partial matches between the treelet pairs and the input, thus containing missing children nodes. These missing nodes pose challenges of ordering during decoding. The reason is that although we can individually find a translation for these missing pieces, it then becomes difficult to fit in the ordering of the whole sentence as it no longer agrees with treelet pair ordering. To address this, while some approaches rely on the estimates from a language model (Ding and Palmer 2005),(Fox 2005), it is more effective to consider a monolingual reordering model that scores the likeliness of a particular sequence of subtree ordering.

As seen earlier, (Quirk, Menezes, and Cherry 2005) learn such an ordering model called the 'order model' as part of the translation models. Given a target node, and the source tree, the order model wishes to model the probability of seeing the pre-modifiers and post-modifiers in a certain order for that node. This means it models the likelihood of seeing the ordering of a subtree (node and children). It could be modeled as relative position to the head or as a binary decision of to swap or not to swap. It can be easy to see that the swap or no swap binary decision case may not have the sparseness problem during estimation when compared to the relative position to the head case. The ordering is predicted individually for each of the heads in the tree and final score for tree is the product (assuming independence) of scores for subtrees. Factoring the decoding into a problem of ordering and translation separately, has the advantage of exploring a larger search space without necessarily relying too much on the language model, which often is a bi-gram or n-gram model and cannot model long-distance reorderings to begin with.

Finally, whenever one has several models to be used in decoding, it is a standard tradition to combine them in some effective way with appropriate weights associated with each of the models. (Quirk, Menezes, and Cherry 2005) combine the models in a log-linear framework, and use a 'MaxBleu' (Och 2003) approach for minimum error rate training. The idea is to find a right combination of the parameters that weight the different models while tuning over a development set. The assumption here is that the parameters that work well on a development set, also tend to show better results on an unseen test set.

7.2 Decoding tricks

Although the reordering models help prune down a major portion of the search space, still a lot of optimizations are required for getting the 'dynamic programming' solution to work. One such technique is to maintain an n-best list of subtree translations at each node. This is a common technique used in other systems (Koehn, Och, and Marcu 2003) and has the disadvantage of pruning out probable hypotheses early on, due to the decision making based on only local information available. Selection of good features that discriminate good from bad candidate hypothesis even with local information is the key to success of an n-best list approach. Since ordering decisions based on exploring all possible attachment slots of a subtree in the higher level tree is the expensive operation, decoding approaches usually look at pruning down search space even before scoring using a reordering model. The less number of possibilities that have to pass through an order model scoring, the faster the decoding.

Greedy techniques also play a role in decoding, where one can imagine picking the best translation from a subtree and going forward. Although this is a very stringent version of the decoder which can not guarantee an optimal solution as per the translation model, a variation of it which uses greedy techniques for a 'ordering model' decision seems to provide fast decoding times at expense of some loss in accuracy. In practice channel model scores are good predictors of high quality translation and so pruning away the low scoring treelet pairs is also a good place to start optimization. Usually a standard threshold is used in pruning or a relative cut-off from the best treelet pair is used. Channel probabilities here can be simple MLE estimates of the treelet pair, or even the IBM Model 1 lexical scores for the treelets.

8. Discussion and Future Work

We have seen various approaches for incorporating syntax and methods to handle isomorphism and non-isomorphism in syntax. Recent approaches like (Quirk, Menezes, and Cherry 2005) have already started to produce state-of-art results on standard data sets. When one takes a closer look at the MT literature in recent years, it can be safely assumed that research in the improvement of phrase structure parsers (Charniak 2000) (Collins 1997) has triggered the beginning of syntax based machine translation models (Yamada and Knight 2001). Currently this trend can be seen also in dependency structures and machine translation. There is parallel research in building dependency structure parsers for various languages with high accuracy (McDonald, Crammer, and Pereira 2005) . Therefore the time is appropriate for increased efforts in machine translation based on dependency trees.

In this section, we share our insight from this survey by discussing what we feel are some of the interesting problems in this area of dependency based syntax machine translation. We first discuss some problems which we think need immediate attention and to solve which, the community already has the tools and techniques. We then also discuss other interesting avenues for research in the future.

8.1 Down the block

We have seen a rise of interest in the word alignment using syntax as well as other discriminative techniques (Taskar, Simon, and Dan 2005). Work has been done to observe how this affects the statistical machine translation end-to-end systems. This effect needs to be tested out on syntactic machine translation models, which rely heavily on underlying word alignments. Similarly another dimension for an effective syntactic model is the accuracy of the monolingual parsers which provide the analysis for building the translation models. We need to empirically try out and observe how the accuracy of an external parser affects the end-to-end translation systems. How much of improvement in the field of monolingual dependency parsing is required before it reflects in a dependency based syntactic machine translation system? Such questions are to be posed in the community and answers have to be researched.

Although we have seen some generalization in the models discussed above and how they help while dealing with out of domain data (Menezes and Quirk 2007), there were restrictions imposed to keep the search space within bounds. Also generalizations of a dependency based translation model at various levels of granularity - morphological, part-of-speech, word classes etc, is still to be explored. Effective decoding algorithms that deal with very large generalized translation models needs to evolve , along with robust estimation techniques to score these models. Finally, most of the approaches work with a dependency tree for the input and perform sub-tree matching with the source side of the translation models. We need to put in place quick algorithms that do this in polynomial time, and decoders that are in the open source domain will help the growth of research in this direction. One can notice at this point that one of the reasons for the recent interest and growth of research in Phrase based SMT (PBSMT) is the availability of tools like Moses (Koehn et al. 2007) and GIZA++(Och and Ney 2003) for rapid prototyping of these systems.

Novel decoding techniques based on Artificial Intelligence algorithms that have worked quite well for PBSMT approaches should be applied to Syntax Based MT. In particular A* decoding applied in PBSMT (Och, Ueffing, and Ney 2001) can be easily applied here by characterizing the appropriate cost functions. Work in building efficient

decoding approaches continues to be explored in other flavors of MT, and can be ported to dependency based syntax MT.

8.2 Down the road

A major problem in the dependency based syntax models is the 'ordering' problem where all the matching subtree translations are to be organized appropriately in the target language. It is to be observed that decoding using a dependency tree model has a wider degree of freedom for reorderings than when a phrase structure model is used. While this freedom is useful to generalize well, it poses a problem of reordering as the sub-pieces no longer have the contiguity constraint that is inherent in the phrase based models. 'Reordering' has been given special treatment in (Quirk, Menezes, and Cherry 2005), where during training a reordering model is trained to score good reorderings over bad ones. While this is a promising direction, since it decouples translation and reordering ,giving more flexibility, there is room for substantial improvement here. One can imagine using information from other sources like Language Models, Phrase structure trees also for better reordering models.

Recently discriminative techniques have proven effective for various Natural Language Processing tasks like Parsing (Collins 1997), Word Alignment (Taskar, Simon, and Dan 2005), Depedency Parsing (McDonald, Crammer, and Pereira 2005). Discrminative frameworks make it easy for incorporating diverse and informative features into the training for the task at hand. Machine Translation is yet to see benefit from these approaches and syntax based machine translation is definitely a good platform for experimenting linguistically motivated features along with statistical evidence. This area is definitely promising and if tuned correctly to the task, yields good translation accuracies.

One benefit of having syntactic translation models is that during decoding time, we will also have access to a syntactic form of the target language hypothesis , in most cases as a tree. Such syntactic information can immediately be used in conjunction with traditional n-gram language models that have often been used for scoring the competing hypotheses. Syntax based Language Models (Charniak, Knight, and Yamada 2003) have been proposed in context of syntax based machine translation, and have shown some promise. It would be interesting to see how they fare in the context of dependency models.

Dependency trees are seen as a first step towards semantics. It is quite intuitive that Machine Translation requires not only syntax but also semantics to provide meaningful translations. Although it is too early for translation systems to incorporate semantics, it is not difficult to see that dependency trees already come with information related to semantics and hence can act as a door-step in this direction. The labels on dependency trees, which encode semantic information of 'who did what to whom' , have not been used so far in the syntactic translation models. However, we have seen that these labels do help in the MT Evaluation task (Owczarzak, van Genabith, and Way 2007), which is an orthogonal task to the translation problem. Therefore it waits to be seen, if depedency based statistical translation models are the windows to successful incorporation of semantics.

Joint modeling approaches (Eisner 2003) have been proposed as a natural solution to non-isomorphism in parallel corpora available. Anyone working in corpus based translation approaches, knows how important the problem of divergences in translation is to MT. Also, we are aware of the freeness in translation in the corpus, which poses a challenge to building effective translation models. (Eisner 2003) proposes approaches to

handle this problem, but results are still to be seen from this direction. Nevertheless, it is an interesting direction and the necessary step towards handling the low quality corpus problems, and errors in monolingual syntax analysis.

9. Conclusion

In this report we have surveyed syntax based machine translation in general and incorporation of dependency structure syntax in statistical machine translation systems. We also discussed the Dependency formalism and the formal definition and variants like projective and non-projective cases. In particular we surveyed two kinds of approaches of incorporating dependencies - one, where parse trees are used from both the sides of the language and two, where only parse tree for source side is used. We then discussed the main challenges of decoding using dependency based translation models, and some approaches to address them. Finally we have highlighted some of the areas that we think need more attention and open for reasearch in the interesection of Machine Translation and Dependency Trees.

10. Acknowledgements

I would like to thank Prof.Alon Lavie and Prof.Stephan Vogel for the opportunity and their valuable comments and discussions throughout the Spring 2008, Advanced MT Seminar course. I would like to thank Prof.Lori Levin for her discussion regarding depedency formalism and pointing to related resources. Thanks to Mr.Amr Ahmed for discussion related to the choice of papers. Thanks to Ms.Rohini U for editorial comments.

References

Alshawi, Hiyan, Shona Douglas, and Srinivas Bangalore. 2000. Learning dependency translation models as collections of finite-state head transducers. *Comput. Linguist.*, 26(1):45–60.

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

Charniak, E., K. Knight, and K. Yamada. 2003. Syntax-based language models for statistical machine translation.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 132–139, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine translation. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270, Morristown, NJ, USA. Association for Computational Linguistics.

Collins, Michael. 1997. Three generative, lexicalized models for statistical parsing. In Philip R. Cohen and Wolfgang Wahlster, editors, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 16–23, Somerset, New Jersey. Association for Computational Linguistics.

Ding, Yuan and Martha Palmer. 2004. Synchronous dependency insertion grammars: A grammar formalism for syntax based statistical mt. In Geert-Jan M. Kruijff and Denys Duchier, editors, *COLING 2004 Recent Advances in Dependency Grammar*, pages 90–97, Geneva, Switzerland, August 28. COLING.

Ding, Yuan and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 541–548, Morristown, NJ, USA. Association for Computational Linguistics.

Eisner, Jason. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), Companion Volume*, pages 205–208, Sapporo, July.

Fox, Heidi. 2005. Dependency-based statistical machine translation. In *Proceedings of the ACL Student Research Workshop*, pages 91–96, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Fox, Heidi J. 2002. Phrasal cohesion and statistical machine translation. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 304–3111, Morristown, NJ, USA. Association for Computational Linguistics.

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. In *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 961–968, Morristown, NJ, USA. Association for Computational Linguistics.

Galley, Michel, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In Susan Dumais; Daniel Marcu and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Gildea, Dan. 2003. Loosely tree-based alignment for machine translation.

Hutchins, W. John and Harold L. Somers. 1992. *An Introduction to Machine Translation*. Academic Press.

Hwa, Rebecca, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng.*, 11(3):311–325, September.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edomonton, Canada, May 27-June 1.

Lin, Dekang. 2004. A path-based transfer model for machine translation. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 625, Morristown, NJ, USA. Association for Computational Linguistics.

McDonald, Ryan, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 91–98, Morristown, NJ, USA. Association for Computational Linguistics.

Menezes, Arul and Chris Quirk. 2007. Using dependency order templates to improve generality in translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 1–8, Prague, Czech Republic, June. Association for Computational Linguistics.

Nirenburg, Sergei, Jaime Carbonnell, Masaru Tomita, and Kenneth Goodman. 1992. *Machine Translation: A Knowledge-based Approach*. Morgan Kaufmann Publishers, Los Altos, CA.

Och, F., D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. 2004. A smorgasbord of features for statistical machine translation.

Och, Franz Josef. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Och, Franz Josef, Nicola Ueffing, and Hermann Ney. 2001. An efficient a* search algorithm for statistical machine translation. In *Proceedings of the workshop on Data-driven methods in machine translation*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Owczarzak, Karolina, Josef van Genabith, and Andy Way. 2007. Labelled dependencies in machine translation evaluation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 104–111, Prague, Czech Republic, June. Association for Computational Linguistics.

Probst, Katharina, Lori Levin, Erik Peterson, Alon Lavie, and Jaime Carbonell. 2002. Mt for minority languages usingelicitation-based learning of syntactictransfer rules. *Machine Translation*, 17(4):245–270.

Quirk, Chris and Arul Menezes. 2006. Do we need phrases?: challenging the conventional wisdom in statistical machine translation. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 9–16, Morristown, NJ, USA. Association for Computational Linguistics.

Quirk, Chris, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: syntactically informed phrasal smt. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 271–279, Morristown, NJ, USA. Association for Computational Linguistics.

Shen, L., A. Sarkar, and F. Och. 2004. Discriminative reranking for machine translation.

Shieber, Stuart M. and Yves Schabes. 1990. Synchronous tree-adjoining grammars. In *Proceedings of the 13th Conference on Computational Linguistics*, pages 253–258, Morristown, NJ, USA. Association for Computational Linguistics.

Smith, David A. and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings of the HLT-NAACL Workshop on Statistical Machine Translation*, pages 23–30, New York, June.

Taskar, Ben, Lacoste-Julien Simon, and Klein Dan. 2005. A discriminative matching approach to word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 73–80, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Wu, Dekai. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist.*, 23(3):377–403.

Xia, Fei and Michael McCord. 2004. Improving a statistical machine translation system with automatically learned rewrite patterns. In *COLING '04: Proceedings of the 20th International Conference on Computational Linguistics*, page 508, Morristown, NJ, USA. Association for Computational Linguistics.

Yamada, Kenji and Kevin Knight. 2001. A syntax-based statistical translation model. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530, Morristown, NJ, USA. Association for Computational Linguistics.