Recent Trends in Statistical Machine Translation

Sourish Chaudhuri and Juan Pino
Language Technologies Institute
Carnegie Mellon University
{sourishc, jmpino}@cs.cmu.edu

November 24, 2008
1 Introduction

The idea of using machines to translate human languages can be traced back to as early as 1949, when Warren Weaver of the Rockefeller Foundation suggested using cryptographic techniques and universals of language to automatically translate text (Weaver 1955). The earliest attempts involved word to word translation; this approach led to a period of disillusionment among researchers as they realized the enormity of the challenges facing them due to the difference in language structure, morphological complexities, semantic and idiomatic barriers and varying levels of ambiguity. The ALPAC report (Pierce and Carroll 1966) concluded that automatic translation seemed to have no immediate prospects of making machine translation useful and suggested developing tools to assist human translation.

Nonetheless, the idea of using machines to automatically translate remained ingrained as part of researchers’ dreams, albeit latent. Interlingua based processes were human-intensive, requiring parsing, transfer rules and generation (Carbonell et al. 1981; Carbonell et al. 1992). Most initial techniques suffered from limitations of being domain specific and not scalable. Over the last two decades, statistical machine translation methods led to considerable improvements; moving from word-based methods (discussed in Section 3) to phrase based methods and then to syntactic models (both are further discussed in Section 3) has improved machine translation (MT, henceforth) methods even further. In this survey we discuss a few papers which have contributed to recent advances in MT:

• David Chiang’s article on Hierarchical Phrase-Based Translation (2007) which was the first syntax-based model to perform significantly better than phrase-based models.

• A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model by Shen et al. (2008) where they propose using a target dependency language model during decoding to exploit long-distance word dependencies.

• Mi et al.’s paper (2008) on Forest-Based Translation where they use a packed forest of exponentially many parses instead of a $k$-best parse on the source side while generating gains in performance and time.

• Blunsom et al.’s (2008) paper on modeling derivations as a latent variable in statistical machine translation to overcome the problems of spurious ambiguity and machine translation.

• Finally, we look at a two theoretical papers by Shieber (2007) that attempts to outline a framework for using probabilistic synchronous tree adjoining grammars for MT using bilingual dictionaries, and describes a set of properties desirable in a good model for MT.
In Section 2 of this survey, we discuss the challenges to MT, discuss ideal properties of an MT model and define the statistical framework for MT; Section 3 contains a description of MT paradigms, and is followed by a description of grammar formalisms for parallel texts in Section 4. In Section 5, we discuss the first four papers mentioned in the list above and their contributions. Finally, Section 6 presents models that have not been tested empirically but model translation equivalence in an attractive way using stronger formalisms.

2 Background and Context of Statistical Machine Translation

Statistical Machine Translation (SMT, henceforth) treats translation as a machine learning problem. This means that a learning algorithm is applied to a large body of previously translated text, known as a parallel corpus. Translations may be learned at different levels of granularity (word-based or phrase based) and at different levels of language and syntax analysis (mapping words and phrases or mapping language rules). After the learning phase, the system is able to translate previously unseen sentences. In this section, we present the challenges in building a system that will translate text in one language into another language. We follow by discussing a set of properties that a model used for performing SMT should ideally possess, and a formal description of the SMT problem and a discussion of the standard models used for SMT.

2.1 Challenges facing MT

MT systems need to deal with fundamental differences between the source and the target languages. Morphologically, languages can go from isolating languages which have one morpheme per word to polysynthetic languages that have many morphemes per word—these morpheme differences can influence word alignment, and also determine fertility (fertility is a measure of how many words in the target language one source language word might map to). While English sentences are typically ordered as subject-verb-object (SVO), Japanese sentences are subject-object-verb (SOV) and a number of other languages such as Hindi and Russian are free word ordered, meaning that more than one permutation of S-V-O are grammatically acceptable.

Other differences include head-marking languages and dependent-marking languages (Jurafsky and Martin, 2007). For instance, in French (head-marking), gender marks in possessive constructions agree with the possessee (e.g. son frère) while they agree with the possessor in (dependent-marking) English (e.g. her brother). Powerful models are needed to model translation between verb-framed and satellite-framed languages. For instance, Italian verbs of direction with modifiers of motion are translated with English verbs of motion with directional modifiers (e.g. attraversare qualcuno di corsa is translated into run across somebody) (Shieber 2007). MT models should be expressive enough to deal with
these differences. Shieber (2007) argues that the nature of the translation relation is such that an appropriate formalism for realizing it should have the following three desirable properties: expressivity, trainability and efficiency.

2.2 Desirable properties of an SMT model

According to Shieber (2007), a formalism should have the following properties, if it were to be powerful enough to capture the translation relations between languages.

• Expressivity

A formalism for describing the translation relation should be able to capture the relations between words in source and target language, while distinguishing between different usages of the same word. Word-based (and phrase-based) translation systems (which we shall discuss in the next section) allow for re-ordering of words (or phrases) (also known as the distortion model) and ranking of the alternatives (monolingual language model). Shieber introduces the Construction principle, a property of the translation relation implicit in the structure of bilingual dictionaries and one which underlies the use of synchronous grammars for MT – “Words and phrases translate differently in construction with other words”. An illustrative example would be idiomatic usage against standard usage – to pull someone’s leg is different from pulling someone’s (say) table, and may translate differently. While the verb take in English translates to prendere in Italian, the phrase take a bath translates to fare un bagno in Italian. The author uses an analysis of a small random sample of data from the Harper-Collins Italian College Dictionary (Love and Clari 1995) to show that as many as 52% of the entries were multi-word expressions, albeit with the caveat that almost half of them could also be interpreted as being simple word-to-word translations. The author refers to this as the property of expressivity.

• Trainability

In order to have a robust model that may be seriously considered as a basis for machine translation, it should be trainable based on statistical evidence found in the corpora. The word and phrase based models use an underlying probabilistic formalism, typically structuring the parameters based on a universal parametric normal form (n-gram probabilities) and apply an efficient training algorithm to set values for the parameters. Shieber’s synchronous tree-adjoining grammar system (described in Section 4.2 and later in Section 6.1) would be able to express the above as well as the detailed bilingual constructional relationships as found in a bilingual dictionary. The normal form would serve to smooth the brittle construction-specific part, while the construction-specific part would relieve the burden of allocating parameters for rare constructions on the universal learned part. Nesson et al. (Nesson, Shieber, and Rush 2006) show that a probabilistic variant of S-TAGs (Shieber 1992) perform at
a level comparable to standard word and phrase based systems, at least on small training sets.

- Efficiency

  The most important question to be answered before employing a formalism is whether it is practical to use it, given the time complexity involved. Shieber points out that it is important not to preclude a formalism merely based on impracticality of its current use given that processing speed is ever-increasing.

  Shieber points out, however, that making a formalism expressive enough to successfully model translation relations might result in some loss in the efficiency system (as we shall see later in Section 6.1).

2.3 Formal Description of SMT Models

Given a sentence \( S \) in the source language, which is a sequence of tokens \( s_1, ..., s_n \), our task is to convert it to another sequence of tokens \( t_1, ..., t_m \) that constitute the target sentence \( T \). While the machine translation problem had earlier been modeled in a noisy channel framework (Brown et al. 1990), log-linear models have been predominant (Foster 2000; Och and Ney 2002) recently.

In this section, we describe the noisy channel and the log-linear model for MT, and conclude with a summary of desirable properties of a model that attempts to capture translation relations between languages.

2.3.1 Noisy Channel Model

Given a sentence \( S \) in the source language, we seek the sentence \( \hat{T} \) that maximizes the probability \( p(T|S) \). We are solving the following equation:

\[
\hat{T} = \arg \max_T p(T|S)
\]

Bayes’ rule and the fact that \( S \) is given while \( T \) varies allows us to rewrite this equation as:

\[
\hat{T} = \arg \max_T p(T) \ p(S|T)
\]

Thus, the decoder has 2 components – one to model the fluency of the target string \( p(T) \), and the other \( p(S|T) \) to model adequacy or “faithfulness” of the translation. The modeling of the latter is where most systems differ, in the manner of allowing probabilities to be assigned for pairs that were not encountered in the training stage. In this regard, we can notice that Shen et al.’s paper distinguish itself by its original approach to language modeling (2008). So far, word and phrase based models have been the dominant approaches.
to factorization of the probability distributions. Two arguments were made to justify breaking down the translation model into a fluency model and a translation model. First, it allows to work separately on two independent models, with the hope that an improvement in each of them will lead to an improvement of the translation overall. Furthermore, the fluency model can take advantage of large amounts of monolingual data that are available unlike bilingual data.

2.3.2 Log-linear Model

Och and Ney (2002) argue that individual improvement in fluency and adequacy models lead to overall improvements only when estimated probability distributions are the true probability distributions. Furthermore, the noisy-channel model is not flexible enough to add dependencies. Finally, they notice that the following erroneous model

$$\hat{T} = \arg \max_T p(T) p(T|S)$$

performs comparably to the theoretically sound noisy channel model. This means that the term $p(T|S)$ is also useful. Log-linear models are well suited in order to incorporate various non independent features. In this model, $p(T|S)$ is defined as:

$$p(T|S) = \frac{\exp(\sum_{m=1}^M \lambda_m h_m(T, S))}{\sum_{T'} \exp(\sum_{m=1}^M \lambda_m h_m(T', S))}$$

where $M$ is the number of features, $h_m$ are feature functions. Usually, features include the language model and translation models in both directions.

3 Machine Translation Paradigms

Translation models define some correspondence between the source and target strings, while operating on different levels of granularity, such as words or phrases. These approaches determine how the source and target sentences were aligned with each other. In this section, we look at the prevalent MT paradigms: word-based methods, phrase-based methods and syntax-based methods.

3.1 Word-Based Methods

In the early 1990s, IBM introduced a number of approaches that used word-based models and which came to represent the first generation of SMT models. These models differed in two respects; first, cardinality of the relation between source and target words, which could be one-to-one, one-to-many, many-to-many or even one-to-zero (in case of deletion). The second point of difference was the dependency assumptions involved in mapping source to target words. As per this approach, words in the target language can move into the source
string either at a random position (IBM model 1), or depending on their absolute position (IBM model 2), or conditioned on where the previous word moves (IBM models 3, 4 and 5). Lexical translation probabilities are based on single words in both the source and target languages.

### 3.2 Phrase-Based Methods

To remedy the fact that word-based models were designed to model the lexical dependencies between single words, phrase based models were introduced (Vogel et al. 2000; Marcu and Wong 2002; Och and Ney 2004). Translation using the phrase-based models proceeds as follows. First, the sentence is broken into phrases, that are independently translated as blocks to target phrases. Then these translated phrases are re-ordered to produce the final order 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 and is very similar to the method used in word-based methods.

Phrase-based models produced better translations than word-based models and are widely used. They successfully model many local re-orderings and individual passages are often fluent. However, they cannot easily model long-distance reordering without having arbitrary permutations, leading to what is known as a phrase salad (Lopez 2008). Incorporation of syntax into phrase-based models was not, however, unambiguously successful, and Koehn, Och and Marcu (2003) argued that syntax is detrimental, and actually decreases the accuracy. In general, Lopez (Lopez 2008) concluded that phrase based models seemed to be a poor fit for syntax, which usually depended on hierarchical models of string generation.

### 3.3 Syntax-Based Methods

An argument against phrase-based SMT systems has been that they do not employ sufficient syntactical information in order to enable grammatical coherence in the target language (Och et al. 2003), and the implicitly captured elements of syntax using n-gram language models had limitations such as not being able to model long-distance dependencies.

Syntax-based machine translation techniques (see Section 3.3) attempt to support statistical systems with explicit representations of syntax to produce high-quality outputs while not requiring human-intensive endeavors. Using various models of translational equivalence and different granularities, researchers have tried to build more syntax-aware models for SMT (Brown et al. 1993; Marcu and Wong 2002; Och and Ney 2004). The hypothesis investigated by syntax-based translation was that reorderings would respect linguistic syntax in translation, an assumption that was not completely borne out by the empirical evidence. There are a handful of ways of viewing the syntactic MT problem, es-
sentially delineated by which side(s) of the translation the tree appears on\(^1\), and whether the tree is induced entirely from data or whether it is induced from a treebank.

- **String-to-tree:** Here, we have a string on the input and a tree on the output. In Chinese-to-English (C2E, henceforth) translation, this would mean that at translation time, we are essentially trying to parse Chinese into an English tree by means of local reordering. Yamada and Knight (Yamada and Knight 2001) exploited rich resources on the English side and found that it worked well with a syntactic language model. In order to train such a model, we need (a) parallel data and (b) a parser on the target language (Yamada and Knight used the Collins parser). Translation is typically done by some variant of CKY.

- **Tree-to-string:** Here, we map an input tree to an output string (Liu et al. 2006). In C2E, this means that at translation time, we first parse the Chinese, then reorder and flatten it out into an English string. In order to train, we need (a) parallel data and (b) a parser on the source side. Once parsing is done, translation is usually a lot easier than running CYK.

- **Tree-to-tree:** Here, we map an input tree to an output tree (Cowan et al. 2006). In C2E, this means that at translation time, we first parse the Chinese, then reorder the tree and translate subtrees/leaves into English. In order to train, we need (a) parallel data, (b) a parser on the source side and (c) a parser on the target side. Translation is similar to Tree-to-string.

- **String-to-string:** The key idea here is to translate without having a parser on either side. At translation time, we essentially parse and translate simultaneously (rather like string-to-tree). But at training time, we do not have access to source trees: we have to induce them from just (a) parallel data (Wu 1997).

## 4 Grammar Formalisms

We describe here several grammar formalisms used in MT. These formalisms were introduced in the field of SMT in order to account for more syntactic coherence of the target sentence and to model long distance dependencies.

### 4.1 Probabilistic Synchronous Context Free Grammars

Probabilistic synchronous context free grammars PSCFG are like probabilistic context free grammars except that the rules specify two outputs. A PSCFG rule has the following form:

\[ X \rightarrow \langle \gamma, \alpha, \sim \rangle \]

\(^{1}\text{http://nlpers.blogspot.com/search?q=tree+to+string}\)
where $X$ is a nonterminal, $\gamma$ and $\alpha$ are strings of terminals and nonterminals, and $\sim$ is a mapping between nonterminal in $\gamma$ and nonterminals in $\alpha$. We use here Chiang’s notation (Chiang 2007).

Syntax-based translation and hierarchical phrase-based translation are two approaches that use PSCFG as framework. We will describe examples of both later on.

### 4.2 Synchronous Tree Adjoining Grammars

In tree-adjoining grammar (TAG), the grammar consists of a certain number of elementary trees 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 (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 it depends on. An elementary tree lexicalized for a given verb, for example, can include substitution nodes for each of its required arguments.

In 1990, Shieber and Schabes introduced synchronous TAG, and Shieber (Shieber 2007) in his paper, outlined later, provides a basis for conceptualizing machine translation using PSTAG (probabilistic S-TAG) supported by evidence from the structures of bilingual dictionaries of the last several millennia. 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 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.

### 5 Moving From Phrase-based to Syntax-based MT

In Section 3.2, we saw how phrase-based models improved over the word-based models. Attempts to augment phrase-based models with syntax met with some successes (Collins et al. 2005) and some failures (Och et al., 2004). In this section, we discuss four papers using syntax-based models to gain significant improvements over state-of-the-art.
Chiang’s paper was the first to combine fundamental ideas from both syntax-based and phrase-based translation methods successfully, thus convincing people that substantial gains could be made by incorporating syntax into translation models.

The paper on forest-based translation tackles the problem of having to deal with some loss of information at the input step where only the 1-best or k-best trees can be used as input for a tree-to-string system. The novel method of using a forest of parses allows this work to have a translation forest on the target side and obtain the best target string from the translation forest.

Blunsom et al. use derivations as a hidden variable pointing out that the best translation string may be reached by a number of derivations and that derivations are not equivalent to translations.

Shen et al. use a novel string to dependency MT algorithm with a target dependency language model that allows the system to exploit long-distance word relations that cannot be modeled with a traditional n-gram language model.

5.1 Hierarchical Phrase Based Translation

Chiang’s system (Chiang 2007) was the first syntax-based system, or at least the first system using syntax, to perform better than phrase-based models, rekindling interest in syntax for MT. The system can be described as a combination of syntax-based and phrase-based translation.

One of the problems encountered with phrase-based models is that they can reorder groups of words but they cannot reorder the groups themselves. For example, the following Chinese sentence:

\[
\text{Aozhou (Australia) shi (is) yu (with) Beihan (North Korea) you (have) bangjiao (diplomatic relations) de (that) shaoshu (few) guojia (countries) zhiyi (one of).}
\]

is translated as following with a phrase-based system:

\[
\text{Australia has diplomatic relations with North Korea is one of the few countries.}
\]

This phrase-based system is unable to reorder the phrases “is one of the few countries” and “has diplomatic relations with North Korea” which are themselves made of reordered sub-phrases. With hierarchical models, this becomes possible, as shown in the following example of a rule right hand side of a Chinese-English PSCFG:

\[
\langle X_1 \text{ de } X_2, \text{ the } X_2 \text{ that } X_1 \rangle
\]

In order to extract these rules, phrase pairs are extracted first. Given a word-aligned sentence \((f, e)\), all sub-sequences \((f', e')\) of \((f, e)\) extracted verify these conditions:

- The length \(|f'|\) and \(|e'|\) are bounded by a threshold, typically 10 (although Zollmann et al. (2008) use 12)
• $f'$ and $e'$ have at least one pair of words aligned.
• words in $f'$ align to words in $e'$ only and conversely.

The set of phrase pairs gives a set of initial rules $X \rightarrow \langle f', e' \rangle$. The set of initial rules is completed with the following induction: if $X \rightarrow \langle \gamma, \alpha \rangle$ is a rule and $(f', e')$ is a phrase pair included in $(\gamma, \alpha)$, then $X \rightarrow \langle \gamma', \alpha' \rangle$ is a rule ($\gamma'$ is $\gamma$ where the part corresponding to $f'$ has been replaced by a non terminal $X$ and similarly for $\alpha'$).

During training, the author generates $k$-best lists used for minimum error rate training (Och 2003) and language model rescoring. In order to do so, he first generates a parse forest, then selects the $k$-best scoring parses. This list is used for language model (LM) rescoring, which is one of the three methods used for adding the LM into the model. The other two methods incorporate the LM directly into the grammar or use a technique called cube pruning. The problem of rescoring is that the best translation might be found very far in the list. This issue is addressed in (Mi et al. 2008) by using parse forests. Since an LM can be viewed as a weighted finite state automaton, it can be intersected with the English side of the SCFG. The third method, called cube pruning, is also used in (Mi et al. 2008).

Chiang's system outperforms a state-of-the-art phrase-based model, showing that a combination of probabilistic models with syntax do help. This trend is confirmed by a thorough study (Zollmann et al. 2008) comparing phrase-based, hierarchical and syntax-based models.

5.2 Overcoming the 1-best source tree problem in tree-to-string translations

We discussed syntax-based approaches in Section 3. As we saw there, syntax-based models can be divided into two categories: the string-based systems whose input is a string to be simultaneously parsed and translated by a synchronous grammar (Wu 1997), and the tree-based systems whose input is already a parse tree to be directly converted into a target tree or string (Lin 2004). Compared with their string-based counterparts, tree-based systems offer some attractive features: they are much faster in decoding (Huang, Knight, and Joshi 2006), do not require a binary-branching grammar as in string-based models (Zhang, Huang, Gildea, and Knight 2006) and can have separate grammars for parsing and translation, say, a context-free grammar for the former and a tree substitution grammar for the latter (Huang, Knight, and Joshi 2006).

5.2.1 From tree-based to forest-based translation

Tree-based translation models, which take the parse tree of the source sentence as input, are faster and simpler than their string-based counterparts. Current tree-based systems, which use the 1-best parse of the source sentence to direct the translation, potentially introduce
translation mistakes due to parsing errors on the source side (Quirk and Corston-Oliver 2006). One obvious solution to the problem is use the k-best parses as input - this postpones some disambiguation to the decoder, which may recover from parsing errors by obtaining the best translation from a parse that is not necessarily the 1-best parse. As Mi et al (Mi, Huang, and Liu 2008) point out, a k-best list has too few variations and too many redundancies, since k=50 would only take care of between 5 to 6 binary ambiguities. Decoding separately with each of these very similar trees is inefficient, and longer sentences - where the number of parses grows exponentially with sentence length aggravates this problem. The authors suggest a novel forest-based approach for the decoding, which show improvements of 1.7 BLEU points over the 1 best baseline and 0.8 points higher than decoding with 30-best parses, while taking even less time.

5.2.2 Forests

A forest is a compact representation of all the derivations for a given sentence under a context free grammar. Consider the Chinese sentence shown in Fig 1 which can have 2 readings depending on the part of speech of the words. In the example, the 2 parse trees can be represented by a single forest by sharing common subtrees such as $NP_{0,1}$ and $VP_{3,6}$. Such a forest has a hypergraph structure, where an item like $NP_{0,3}$ is called a node, and derivation steps correspond to hyperedges. Formally, a forest is a pair $<V, E>$, where $V$ is the set of nodes and $E$ is the set of hyperedges. Each node contains the non-terminal and the indices of the substring that it spans. Each hyperedge $e$ is a pair $<\text{tails}(e), \text{head}(e)>$, where head($e$) is the consequent node in the derivation step, and tails($e$) contains the list of antecedent nodes. The forest structure will also contain a root node TOP denoting the goal item in parsing, which is $S_{0,l}$, where S is the start symbol and l is the sentence length.

- **Translation forest**

Given a parse forest and a translation rule set, we can generate a translation forest with a similar hypergraph structure. Each node in the parse forest is visited in top-down order, and an attempt is made to pattern-match each translation rule $r$ against the local sub-forest under a node $v$. For example, in Figure 1, at node $VP_{1,6}$, two rules $r_3$ and $r_7$ both match the local subforest, and will thus generate two translation hyperedges $e_3$ and $e_4$ (see Figure 1). Formally, a function match $(r, v)$ attempts to pattern-match rule $r$ at node $v$ in the parse forest, and returns a list of descendant nodes that are matched to variables in $r$, or returns an empty list if the match fails. A translation hyperedge from match($r, v$) to $v$ for each node $v$ and rule $r$ is constructed, while keeping track of the target string specified by rule $r$. A translation hyperedge is, thus, a triple $<\text{tails}(e), \text{heads}(e), s>$ where $s$ is the target string as a result of application of the rule. A summary of this procedure as pseudo code is given in Fig 2.

- **Decoding**
Figure 1: Forest Based Translation: (a) An example parse forest (b) Corresponding translation forest (c) Correspondence between translation hyperedges and translation rules

**Pseudocode 1** The conversion algorithm.

1. **Input**: parse forest $H_p$ and rule set $\mathcal{R}$
2. **Output**: translation forest $H_t$
3. **for** each node $v \in V_p$ in top-down order **do**
4. **for** each translation rule $r \in \mathcal{R}$ **do**
5. $\text{vars} \leftarrow \text{match}(r, v)$ \hspace{1cm} \Comment{variables}
6. **if** $\text{vars}$ is not empty **then**
7. $e \leftarrow \langle \text{vars}, v, s(r) \rangle$
8. add translation hyperedge $e$ to $H_t$

Figure 2: Pseudo-code for creating a translation forest from a parse forest
Once the translation forest is obtained, the decoder performs 1-best search with integrated language model- done using the cube-pruning technique (Chiang, 2007) and k-best search with LM to be used in minimum error rate training- done using lazy Algorithm 3 of Huang and Chiang (2005).

- Forest Pruning

Using the pruning algorithm of Huang (2008) (Huang 2008), this paper uses an Inside-Outside algorithm to compute the Viterbi inside cost $\beta(v)$, and the Viterbi outside cost $\alpha(v)$ and then compute the merit $\alpha \beta(e)$ for each hyperedge $e$:

$$
\alpha \beta(e) = \alpha(\text{head}(e)) + \sum_{u_i \in \text{tails}(e)} \beta(u_i)
$$

This merit is the cost of the best derivation that traverses $e$, and the difference $\delta(e) = \alpha \beta(\text{TOP})$ can be seen as the distance away from the globally best derivation. A hyperedge $e$ is pruned away if $\alpha \beta(e) - \beta(\text{TOP}) > p$ for a threshold $p$. Nodes with all incoming hyperedges pruned are also pruned.

5.3 Discriminative Latent Variable Model

Improvements in SMT in recent years have been driven by a move to phrase-based and syntax-inspired approaches. Progress within these approaches, however, has been less dramatic because frequency count based models (approaches using MERT, which re-scale a handful of generative features estimated from frequency counts and do not support large sets of non-independent features) cannot easily incorporate non-independent and overlapping features, which are extremely useful in modeling the translation process. Discriminative models (Ittycheriah and Roukos, 2007; Liang et al., 2006) held much promise, in this context, since they didn’t make assumptions of independence between features. However, while discriminative models promise much, they have not proved to be significantly better when compared to simpler models.

5.3.1 Modeling derivations as a latent variable

Blunsom et al (2008) argue that in order to perform better, discriminative models for SMT must address a number of challenges- primary among which is the problem of spurious ambiguity and degenerate solutions. These occur when there are many ways to translate a source sentence to the same target sentence by applying a sequence of steps of phrase translations and grammar rules. Such a sequence of steps is known as a derivation. Existing discriminative models require a reference derivation to optimize against, although no parallel corpora annotated for derivations exist. Marginalizing out the derivations would allow the model to predict the best translation rather than the best derivation. However,
doing so exactly is NP-complete, and therefore, discriminative problems have chosen to
side-step the problem by choosing simple model and feature structures, or by treating
derivations as translations.

In their work, Blunsom et al model the derivation \( d \) as a latent variable \( p(e, d \rightarrow f) \),
which is marginalized out in training and decoding (Suppose \( e \) and \( f \) are the English and
foreign sentences respectively). They develop a log-linear model of translation which is
globally trained on a significant number of parallel sentences, and maximizes the condi-
tional likelihood of the data \( p(e \rightarrow f) \). Further, they present efficient methods for training
and prediction, demonstrating their scaling properties by training on more than a 100K
sentences. They contend that their main findings are general ones and could be applied
to other models, discriminative and generative, to further progress the state-of-the-art in
machine translation.

In the following subsections, we outline the global log-linear model that Blunsom et al
used, their training and decoding methods. We conclude with a short discussion on their
findings when evaluating the system.

5.3.2 Global log-linear model

The log-linear translation model defines a conditional probability distribution over the
target translations of a given source sentence. A particular sequence of SCFG rule appli-
cations which produces a translation from a source sentence is referred to as a derivation,
and each translation may be produced by a number of different derivations. With the
training data only providing the source and target sentences, the derivations are modeled
as a latent variable. The conditional probability of a derivation \( d \), for a target translation
\( e \), conditioned on the source \( f \), is given by:

\[
p_{\Lambda}(d, e|f) = \frac{\exp \sum_k \lambda_k H_k(d, e, f)}{Z_{\Lambda}(f)}
\]

where

\[
H_k(d, e, f) = \sum_{r \in d} (h_k(f, r))
\]

Here, \( k \) ranges over the model’s features, and \( \lambda = \lambda_k \) are the model parameters. The fea-
tures are defined over the source and target sentence pairs and can include non-independent
features of the data. The distribution is globally normalized by the partition function which
sums out the numerator in (1) for every derivation of \( f \). Thus, using the first equation, the
conditional probability of a target translation given the source is the sum over all of its
derivations:

\[
p_{\Lambda}(e|f) = \sum_{d \in \Delta(e, f)} p_{\Lambda}(d, e|f)
\]

16
Unlike previous work, which approximated the sum over derivations by choosing a single best derivation using a Viterbi or beam search algorithm, this work derives its advantage by accounting for the derivational ambiguity, which the authors show is tractable to compute, and their findings echo those observed for latent-variable models successfully used in monolingual parsing (Petrov et al., 2007).

5.3.3 Training
The model parameters are estimated using a maximum a posteriori (MAP) estimator, which maximizes the likelihood of the parallel training sentences. A zero-mean Gaussian prior with the probability density function $p_0(\lambda_k) \propto \exp\left(-\frac{\lambda_k^2}{2\sigma^2}\right)$ is used to penalize the likelihood, and results in the following objective function and gradient.

$$
\mathcal{L} = \sum_{(e,f) \in \mathcal{D}} \log(p_A(e|f)) + \sum_k \log(p_0(\lambda_k))
$$

$$
\frac{\partial \mathcal{L}}{\partial \lambda_k} = E_{P_A(d|e,f)}[h_k] - E_{p_A(e|f)}[h_k] - \frac{\lambda_k}{\sigma^2}
$$

The model is trained by maximizing (INSERT: eq 1 above) using L-BFGS (Sha and Pereira, 2003). Calculating the objective and gradient value for each iteration requires two separate packed charts- one being the full chart over the space of possible derivations given the source sentence, and the other containing the space of derivations which produce the reference translations from the source. The full derivation chart is produced using a CYK parser in the same manner as Chiang (2005), and has complexity $O(|e|^3)$.

5.3.4 Decoding
Marginalizing over derivations in decoding is NP-complete, and the standard solution is to estimate the maximum probability translation using a single derivation (Koehn et al, 2003). In this work, the authors approximate the sum over derivations directly using a beam search which produces a beam of high probability translation sub-strings for each cell in the parse chart. This method- very similar to decoding with an SCFG intersected with an n-gram language model (Chiang, 2005)- also creates a hypergraph structure which stores the entire target string. Additionally, instead of maximizing scores in each beam cell, the inside scores for each derivation that produces a given string for that cell is summed. When the beam search is complete, we have a list of translations in the top beam cell spanning the entire source sentence along with their approximated inside derivation scores. The space of translation substrings is exponential in each cell’s span, and this algorithm only sums over a small fraction of possible strings. While the resulting probabilities are merely estimates, results show that this is more effective than maximum derivation decoding.
5.3.5 Results and Discussion

Blunsom et al compare their results against their implementation of the hierarchical model (Hiero). Using the latent variable model that accounts for all derivations against Hiero, results obtained over the Europarl French-English parallel corpus indicate the value in optimizing translation rather than derivations show that the former model has a larger impact on the BLEU scores.

The results further indicate that an unregularised maximum likelihood model lags behind a regularized max-translation model, probably due to degenerate solutions in the former, which the regularized MAP model manages to avoid. The idea behind regularizing with a Gaussian prior with unit variance is as follows- dispreferred rules must receive large negative weights in order to find a degenerate solution; with the prior penalizing large weights, the best strategy for the regularized model is to evenly distribute probability mass.

With a minimalistic feature set, this max-translation system achieves comparable performance against Hiero. At training time, it is hard to achieve full coverage for which a grammar that generates all observed source-target pairs has to be induced. Instead, the system discards the unreachable portion of the training sample. Experiments with changing the size of the training corpus show that the system’s performance continues to improve as the training set size increases, and had been continuing to increase when the limit was reached. This shows that using the 40K sentences that the system discards due to unreachability could have further improved performance.

5.4 Using a target dependency language model

A regeneration of interest for diverse syntactic models, or at least for their inclusion in statistical models arose after Chiang’s success. However, powerful syntactic models that would take into account all language differences mentioned in section 2 are not practical. Such models are presented in section 6. In this section, we describe practical models that make use of syntax.

A limitation of SCFGs using monolingual parsing is that they do not allow alignments of non-constituents, even if these constituents make sense from a linguistic point of view. For example, “the red” is a noun modifier but it is not a constituent. As a consequence, syntax based system have a poor rule coverage compared to phrase-based systems (DeNeefe et al. 2007). Chiang solves this problem by combining phrase-based techniques with an SCFG that does not use monolingual parsing. Shen and colleagues (Shen et al. 2008) extend Chiang’s hierarchical model by introducing dependency parsing on the target side while at the same time avoiding the rule coverage issue. Indeed their system allows well-formed dependency structures on the target side that include non-constituents. Let us explain the concept of well-formedness through a schema rather than formally. Figure 3 shows and example of dependency parse for a target sentence. This parse can be assembled by substructures in various ways. Shen and colleagues constrain these substructures to be
either fixed or floating as in Figures 4 and 5 respectively. Fixed structures have a known head and their span is continuous. Floating structure have an unknown head, they are formed of complete siblings. When extracting rules, the authors use the same process as Chiang but the target side maintains well-formed dependency structures. Shen and colleagues’ system is based on a log-linear model; one of the features include a dependency language model that leads to substantial performance improvements.

6 Future Trends: Powerful Models for MT

Section 2 describes the differences between languages that translation models should cope with. After Chiang’s empirical proof that syntax does help SMT, other models have confirmed this success, sometimes outperforming Chiang’s hierarchical model (Shen, Xu, and Weischedel 2008). However, remember that we noted in Section 3.3 the hypothesis investigated by syntax-based translation: that reorderings would respect linguistic syntax in translation. Fox (2002) shows that reordering tends to respect the boundaries of syntactic phrases, but also describes some systematic exceptions. Additional discrepancies are described by Wellington et al. (2006). The rigidity of full isomorphism at the level of SCFG productions harms syntax-based translation. These difficulties have prompted investigation into more powerful formalisms for syntactic translation (Galley et al. 2004). We feel that in the future, SMT will turn to more powerful models and formalisms, whose perceived applications are (at best) limited at this point of time. While these models are not tractable yet, theoretical interest in them has flourished. In this section, we present two such models- one by Shieber, making use of probabilistic S-TAGs (S-TAGs were briefly explained in Section 4.2) and the second by Melamed (2004), making use of multi-text grammars.
6.1 S-TAGs for MT

In section 3, we described 3 properties that the author propounds as desirable in an MT system, and, when possible, elaborated on how the S-TAG system fulfills those properties. In addition, we will describe here Shieber’s (2007) work in his position paper where he puts forth a conceptual basis for thinking of machine translation in terms of synchronous grammars in general, and probabilistic synchronous tree-adjoining grammars in particular, an approach Shieber claims is corroborated by the evidence found in the structure of bilingual dictionaries over several millennia.

The directness with which the S-TAG formalism follows from the structure of bilingual dictionaries suggests a possibility of making direct use of bilingual dictionary material in an SMT system. Shieber points out that for syntax-based MT, the reordering step to reconstruct tree-alignments from data is more difficult than for phrase-based MT systems, and hence extracting them from a dictionary becomes much more appealing. He contends that the expressivity of this formalism matches the thought put in by lexicographers over the years to capture the translation relation. Representing MT in terms of S-TAGs may be seen as underlying what is known as syntax-aware or syntax-informed MT, and that probabilistic S-TAGs are the most appropriate formalism for realizing the translation relation- no other formalism displays the requisite properties of expressivity, trainability and efficiency (outlined in Section 2.2).

Perhaps the most important feature of this formalism is that it can express ramified bilingual constructions as found in lexicons as well as the universal assumption-free normal
forms underlying modern statistical MT at the same time, and can, therefore, be trained together based on bilingual corpora. While STAGs suffer from not being able to efficiently code all generalizations about the translation relation, it appears sufficient that it is not required to specify such missing generalizations separately from the other things that we come across.

The question of efficiency is one which the S-TAG formalism does not satisfactorily answer, at least at this point of time. STAG parsing can be done in $O(n^{12})$ time under suitable restrictions of binarizability, over a bilingual corpus—this is the square of the time required to perform monolingual TAG parsing. The authors contend, however, that methods such as those of Chiang (Chiang 2007) which reduce the complexity of SCFG parsing may be applicable to STAG parsing as well. With recent research beginning to unify synchronous grammar formalisms and tree transducers (Shieber, 2004), there may well be equally direct transducer formalisms that express construction-based translation relations.

6.2 Generalized Multitext Grammars

It is necessary to take discontinuous constituents in translation, simply because discontinuous constituents can be translated into continuous constituents. An example of French discontinuous is the negation “ne ... pas”. Phrasal verbs are another example of discontinuous constituents in English, as in the sentence “Look it up!”. Besides, elements of a language often disappear in translation. These two observation motivate Melamed and colleagues’ paper on generalized multitext grammars (GMTG). GMTG are an extension of multitext grammars with additional power. Apart from allowing independent rewriting and modeling discontinuous constituents, GMTG can model long distance movements such as clitic climbing and extraposition. For example, the following set of rules can produce the English-Russian sentence pair (I fed the cat, Ya kota kormil):

\[
\begin{align*}
[S, (S)] &\rightarrow [(PN^1 VP^2), (PN^1 VP^2)] \\
[PN], (PN) &\rightarrow [(I), (Ya)] \\
[VP], (VP) &\rightarrow [(V^1 NP^2), (NP^2 V^1)] \\
[VP], (VP) &\rightarrow [(fed), (kormil)] \\
[VP], (VP) &\rightarrow [(D^1 N^2), (N^2)] \\
[VP], (VP) &\rightarrow [(the), ()] \\
[NP], (NP) &\rightarrow [(cat), (kota)]
\end{align*}
\]

Rules 5 and 6 show how to use independent rewriting. Without independent rewriting, for example in SCFGs, there would be a $D$ in the target side until it is rewritten as a constituent. This generates spurious ambiguity, therefore wastes computation time. Independent rewriting also allows different ways of expressing the same meaning, which is very frequent in translation. We can take the example of the English sentence pair [(Tim got a pink slip), (Tim got laid off)].

While MTGs require that production components be contiguous, GMTG do not. For
example, the rule \([(NP),(NP,NP)] \rightarrow [(\text{his teeth}),(le, los dientes)]\) is valid in GMTG and allows to derive the sentence pair \([(\text{The doctor treats his teeth}), (El mdico le examino los dientes)]\).

The authors also design a CKY-style parsing algorithm to parse GMTG. Currently, this powerful formalism has not been empirically tested but its promising modeling properties should lead to subsequent experiments.

7 Conclusion

In this report, we have attempted to give an overview of a few different syntax-based translation models that achieved significant performance gain over accepted baselines. We also discuss papers that outline powerful formalisms that could potentially become usable if the computations could be made tractable or if it were to become possible to solve them quickly. For the purpose of this survey, we endeavored to pick out papers that focused on solving problems which provided some insight into overcoming limitations that had appeared to be binding on the MT community—degenerate solutions were avoided by modeling derivations as a latent variable; the problem of being limited to using k-best parses was overcome by using a parse forest at the source.

Even if it is more satisfactory to have a model that works for any given pair of languages, it is also important to consider which models work best for particular pairs of languages, depending on how similar these languages are syntactically and from a word order point of view. Is it better to use more constrained models when working with languages from the same family and more expressive formalisms when working with languages farther from each other or should we use the most expressive available model provided that it is tractable computationally in order to take into account the pathological cases that happen for languages close to each other?

References


Fourth International Conference on Theoretical and Methodological Issues in Machine Translation, pp. 25–27.


