Doctoral Thesis Proposal

Multilingual Translation Methods

Author:
Lane Schwartz

Advisor:
Dr. William Schuler

University of Minnesota
Institute of Technologoy
Department of Computer Science and Engineering

August 28, 2008

Contents

Chapter 1

Introduction

To date, the vast majority of research in machine translation has focused on the task of translating from a single source language into a single target language. Yet governments, companies, and other international organizations commonly translate documents into many languages. In general, documents translated into more than one language will likely be translated into many more languages (Kay, 1980).

The recent development of large *multi-parallel* corpora has made research into multilingual translation practical. A multi-parallel corpus contains the same texts in more than two languages. The Europarl (Koehn, 2005), Acquis Communautaire (Steinberger et al., 2006), and News Commentary (Callison-Burch et al., 2007) corpora are freely available multi-parallel corpora that together include most European languages. Other multi-parallel corpora available in machine-readable format include many United Nations documents (UN, 1994), the Bible (Resnik et al., 1999), and George Orwell's novel *1984* (Erjavec, 2004).

Multi-parallel texts provide a rich source of information which could be exploited to reduce ambiguity and improve translation choices. This proposal surveys the current state of the art in techniques to exploit multi-parallel corpora and techniques for using multiple source languages in statistical machine translation, presents preliminary experiments which show the limitations of existing hypothesis ranking methods, and proposes novel techniques for exploiting multi-parallel corpora in the context of statistical machine translation.

The remainder of this proposal is structured as follows. Chapter 2 presents basic background material in statistical natural language processing and statistical machine translation. Chapter 3 reviews existing related research which uses multilingual resources and multi-parallel corpora in statistical machine translation. Chapter 4 presents our work to date on multilingual translation. Finally, our proposals for further research in multilingual translation are presented in chapter 5.

1.1 Terminology

A *source* language is the original language of a text. A *target* language is a language into which a text is translated.

A *corpus* is a body of text. In many natural language processing tasks, the available corpora are typically divided into two or three disjoint sections. Machine learning algorithms are commonly used to analyze a *training corpus*. The goal of this process is to derive a useful model of the data in the training corpus. The training corpus is also referred to as the *training data*. A *development corpus* may be used if parameter tuning is performed. Development corpora are typically much smaller in size than the training corpus. A *test corpus* is used to test the performance of the trained model on unseen text.

1.2 Notation

In the context of a sentence, the notation w_m^n represents the contiguous sequence of words beginning with the m^{th} word of the sentence and continuing up to and including the n^{th} word of the sentence.

The notation f refers to the (foreign) source language. The notation e refers to the (English) target language.

Chapter 2

Background

The horse raced past the barn fell.

The man gave the girl a ring impressed a watch.

Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo.

Three arguably fluent English sentences.

2.1 History

Early work in machine translation attempted to directly encode linguistic insights into databases of translation rules. The process of machine translation transformed an input sentence in the source language into an output sentence in the target language using these linguistically-motivated rules . *Rule-based machine translation* thus has the major advantage that expert knowledge from linguists can directly inform the translation process.

However, rule-based translation suffers from several major downsides. The process of constructing a database of rules requires expert linguists familiar with the source and target languages. Constructing a rule-based system for a new pair of languages is an expensive and time-consuming proposition.

With good system design, this process can be streamlined somewhat. It is sometimes possible to reduce the duplication of effort required for a new language pair by separating the translation process into separate source analysis and target generation steps. In such systems, there is a common intermediate representation. The system designers build an analysis module for each source language which transforms the source into the common intermediate representation. Likewise, a generation module is built for each target language which transforms the common intermediate representation into the target language.

Nevertheless, once built, the resulting systems tend to be relatively static — adapting a system for a specific domain or new vocabulary may involve a significant amount of work by experts familiar with the languages and the system architecture. For many large European language pairs, general-domain rule-based systems are commercially available.

In recent years, the research in machine translation has moved sharply away from rule-based systems to focus on statistical methods. *Statistical machine translation* attempts to use statistics gathered from existing translated corpora to guide the translation process. Sections 2.2 and 2.3 present brief overviews of relevant topics from statistical natural language processing; these topics represent important mathematical and linguistic ideas which underlie current statistical machine translation techniques. Sections 2.4.4 and 2.4.8 present the two dominant techniques used in current statistical machine translation research.

2.2 N-gram language models

In many natural language processing problems, including statistical machine translation, it is useful to know whether a given sequence of words is likely to occur in a given language. The task of calculating the probability of a sequence of words in a language is known as *language modeling*.

A good language model should assign high probability to text which is common in the language, and low probability to text which is uncommon. In English, *the* is far more common than *seismogenic*. We would therefore expect a good language model to say that $P(the) > P(seismogenic)$.

2.2.1 Unigrams

Statistical natural language processing attempts create useful mathematical models of human language, trained from data using machine learning techniques. Given a corpus of English text, how can we calculate the above probabilities? We start simply by counting. In a corpus[1] of approximately 18.2 million words, *the* occurs 479,708 times while *seismogenic* occurs only twice.

A *unigram* language model can be calculated by dividing the number of times each word occurs in the corpus by the total number of words in the corpus.

$$P(word) = \frac{count(word)}{sizeOf(corpus)} \tag{2.1}$$

[1] Proceedings of the European Union Parliament (Koehn, 2005)

Applying this equation to *the* and *seismogenic* produces the unigram language model probabilities for those words:

$$\begin{aligned} P(the) &= \frac{479,708}{18,203,089} = 0.026 \\ P(seismogenic) &= \frac{2}{18,203,089} = 1.1e-07 \end{aligned} \tag{2.2}$$

These unigram language model probabilities match our intuition that the probability of encountering the extremely common word *the* in an English text is far greater than the probability of encountering the extremely rare word *seismogenic*. In machine learning terms, the unigram language model defined by equation 2.1 represents a *maximum likelihood estimate* of the training data. That is, given the available training data, the probability distribution defined by equation 2.1 will explain the training data better than any other probability distribution for a unigram language model.

2.2.1.1 Markov Assumptions

Unigram language model probabilities provide the probability of a single word in a given language. But what if the language model probability of a phrase or a sentence is needed? The words in a sentence are dependent on each other. To illustrate this point, consider the sentences in figure 2.1.

I object to the report, Madam ____.
In London, you should try and visit Madam ____ .

Figure 2.1: In each sentence, the most likely word to follow *Madam* depends on the other words in the sentence.

The first sentence appears to be from a parliamentary report. Reasonable guesses for the word following *Madam* might be *President* or *Speaker*. On the other hand, the second sentence appears to be related to tourism in England. Given the other words in the sentence, we might expect *Tussauds* to be the most likely next word. Put another way, the word following *Madam* is *conditionally dependent* on all of the words preceding it in the sentence. The actual probability of a sentence with N words, $P(w_1^N)$, is then defined in equation 2.3.

$$P(w_1^N) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2)...P(w_N|w_1...w_{N-1}) \tag{2.3}$$

Equation 2.3 states that the probability of a sentence is the probability of the first word $P(w_1)$, times the probability of the second word given the first word

$P(w_2|w_1)$, times the probability of the third word given the first and second words $P(w_3|w_1w_2)$, and so on. Unfortunately, there is no easy way calculate good estimates for language model probabilities which take long histories into account (such as $P(w_N|w_1...w_{N-1})$. As a result, real language models must make simplifying assumptions.

A unigram language model makes severe *independence assumptions*. The model assumes, incorrectly, that the probability of a particular word in a sentence is independent of the other words in the sentence. This assumption of complete statistical independence from all history is called a *zero-order Markov assumption*. This assumption leads to the following formula to calculate the unigram language model probability of a sequence of words:

$$\begin{aligned} P(w_1^N) &= P(w_1)P(w_2)P(w_3)...P(w_N) \\ &= \prod_{n=1}^{N} P(w_n) \end{aligned} \tag{2.4}$$

The conditional probabilities from equation 2.3 are changed in equation 2.4 to unconditional prior probabilities. Consider what word a unigram language model might predict in figure 2.1. Since the unigram model makes a zero-order Markov assumption, all history is ignored. The model will assign the highest probability to the most common words in the corpus (see equations 2.1 and 2.2). This means that common words like *the* or *and* would likely be incorrectly predicted as the next word for the sentences in figure 2.1.

2.2.2 Higher Order N-grams

An important and common use of n-gram language models in statistical machine translation involves using the language model to help choose fluent translation over disfluent translations. Sequences of words with high language model probabilities are interpreted to be more fluent than sequences with low language model probabilities. However, unigram language model probabilities for a sequence of words do not take into account any other words in the sentence (equation 2.4).

Figure 2.2 illustrates a critical problem with n-gram language models. A language model which does not sufficiently account for context will not consistently assign higher probabilities to fluent sequences of words than it does to disfluent sequences. Here, the unigram language model, which uses no context, assigns a much lower probability to an actual English sentence (figure 2.2c) than it does to a repetition of a common word (figure 2.2a) or a jumble of words (figure 2.2b).

To counter this problem, language models which take more context into account are used. *Higher order* language models attempt to provide a better approximation

a) the the the the the the the the
b) she the it it the and a an
c) the session will resume after a short recess

Figure 2.2: Of the three sequences above, a unigram language model ranked sequence **a** as significantly more probable than sequence **b** or **c**. Sequence **c** was assigned a much lower probability than either sequence **a** or **b**.

to equation 2.3 by making less severe independence assumptions than unigram language models. The *order* of an n-gram language model is equal to the number of words in the history that are considered in probability calculations. A unigram language model has order 0. A *bigram* language model has order 1; bigram probabilities are conditionally dependent on one word of history (equation 2.5).

$$\begin{aligned} P(w_1^N) &= P(w_1)P(w_2|w_1)P(w_3|w_2)...P(w_N|w_{N-1}) \\ &= \prod_{n=1}^{N} P(w_n|w_{n-1}) \end{aligned} \tag{2.5}$$

In a *trigram* language model, with order 2, probabilities are dependent on two words of history (equation 2.6).

$$\begin{aligned} P(w_1^N) &= P(w_1)P(w_2|w_1)P(w_3|w_1^2)...P(w_N|w_{N-1}^{N-2}) \\ &= \prod_{n=1}^{N} P(w_n|w_{n-1}) \end{aligned} \tag{2.6}$$

In general, an n-gram language model with order o can be defined by equation 2.7 for all $o > 0$.

$$P(w_1^N) = \prod_{n=1}^{N} P(w_n|w_{n-o}^{n-1}) \tag{2.7}$$

Given identical training data, higher order n-gram language models are better able to correctly model fluency than lower order n-gram language models. Shannon (1951) examines how higher-order n-gram language models can correctly predict the next word in a sentence. Trigram and bigram language models each correctly assign the fluent sequence in figure 2.2c a higher score than either figure 2.2a or figure 2.2b.

If higher order n-grams provide better models of fluent language, why do real systems not simply set the n-gram order to infinity? In other words, why make any Markov assumptions if the true probability of a given word appearing might depend on all other words in the sentence?

The answer is data sparsity. No matter how large the training set, nearly all problems will require the language model to assign probabilities to sequences of words which were never seen in the training set. When this happens, the language model will assign a probability of zero to the word sequence. The higher the n-gram order, the more frequently unseen sequences will be encountered. If no Markov assumptions are made, the vast majority of sentences in any test set will contain word sequences which are assigned zero probability.

The use of smoothing and backoff techniques can prevent zero probabilities from occurring when scoring unseen sequences. An examination of these techniques is beyond the scope of this proposal. Even when smoothing and backoff techniques are applied, the problem of data sparsity results in unreliable probability distributions for high order n-grams. It is necessary to find a balance between the benefits in better fluency modeling that higher order models provide and the problems introduced by unreliable probability distributions for high order n-grams.

In practice, many real natural language processing applications use trigram language models. When very large training corpora are available, n-grams of up to order 4 are sometimes used in statistical machine translation.

2.3 Context-free Grammars

N-gram language models are strictly linear models. Such models assign probabilities to sequences of words based only on how likely each word is given the words which immediately precede it. Consider the sample sentence in figure 2.3.

DT	N	V	DT	N
the	man	took	the	books

Figure 2.3: A simple sentence. Each word in the sentence is annotated with a corresponding grammatical category.

The words in human language are generally accepted to have associated linguistically-motivated grammatical categories. The sample sentence in figure 2.3 shows each word annotated with its corresponding grammatical category. The sequence of categories is linear; as such this information can be incorporated into a richer n-gram language model (Brown et al., 1992).

One important defining feature of human language is its hierarchical nature. Beginning with the words, a sentence can be described in terms of hierarchical levels of grammatical constituents. A common way to illustrate this hierarchical structure is by using trees. Figure 2.4 shows the grammatical constituents that comprise the sample sentence drawn as a tree.

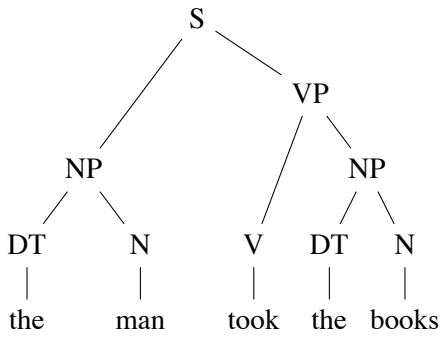

Figure 2.4: Tree representation of a simple sentence.

This tree shows that the sentence (S) is composed of a noun phrase (NP) and a verb phrase (VP). The verb phrase encompasses a verb (V) and a noun phrase that represents the object of the verb. Each noun phrase is composes of a determiner (DT) and a noun (N). At the terminal leaf level of the tree, each word of the sentence is attached to a corresponding pre-terminal grammatical category label.

Each node in the tree has one or more children, except for the terminal nodes, which have no children. These parent-child relationships can be re-written as *context-free rules* of the form $A \rightarrow \beta$ where A is a nonterminal grammatical category and β is a sequence of one or more terminals or nonterminal symbols. The rules are context-free because they represent a parent-child relationship which is independent of the context of the rest of the tree. As an example, the rule NP $\rightarrow$ DT N expresses a valid parent-child relationship that is not dependent on other context in the tree. The rules corresponding to the tree in figure 2.4 are listed in figure 2.5.

S	$\rightarrow$	NP VP	DT	$\rightarrow$	the
NP	$\rightarrow$	DT N	N	$\rightarrow$	man
VP	$\rightarrow$	V NP	N	$\rightarrow$	books
			V	$\rightarrow$	took

Figure 2.5: Context-free rules

A context free grammar is formally defined by a finite set of terminal symbols

$\mathcal{T}$, a finite set of nonterminal symbols $\mathcal{N}$, a set of rules $\mathcal{R}$, and a unique start symbol $\mathcal{S} \in \mathcal{N}$.

The set of terminal symbols $\mathcal{T}$ represents all valid words in the language represented by the grammar. The nonterminal set $\mathcal{N}$ represents all grammatical categories in the grammar. The rule set $\mathcal{R}$ includes all context-free rules for the grammar. The start symbol $\mathcal{S}$ is the only nonterminal allowed at the root of valid trees covered by this grammar; in natural language applications, the nonterminal representing an entire sentence (S) is commonly used as the start symbol. A simple but complete context-free grammar capable of describing the sentence and tree in figure 2.4 is shown in section 2.3.

$$
\begin{array}{rcl}
\mathcal{S} & = & \text{S} \\
\mathcal{N} & = & \{\text{S, NP, VP, N, DT, V}\} \\
\mathcal{T} & = & \{\text{the, man, took, book}\} \\
\mathcal{R} & = & \{\ \text{S} \rightarrow \text{NP VP} \\
 & & \text{NP} \rightarrow \text{DT N} \\
 & & \text{VP} \rightarrow \text{V NP} \\
 & & \text{DT} \rightarrow \text{the} \\
 & & \text{N} \rightarrow \text{man} \\
 & & \text{N} \rightarrow \text{books} \\
 & & \text{V} \rightarrow \text{took}\ \}
\end{array}
$$

Figure 2.6: Sample context-free grammar

Context-free grammars for real human languages can be constructed by hand by linguists. In modern natural language processing, this is rarely done. Instead, a common approach for constructing a context-free grammar is to extract a large set of context-free rules from a *treebank*. A treebank is a collection of trees that represent sentences in a corpus. Because context-free rules describe parent-child relationships in trees, a set of context-free rules that describe all parent-child relationships in that tree can be directly extracted from a tree that represents a sentence. Given a sufficiently large corpus, a set of rules can be extracted which provide large coverage of the language.

Context-free grammars were introduced by Chomsky (1956). While there is some disagreement over the formal expressive power of human languages, context-free grammars provide an intuitive model for describing the hierarchical nature of human languages. Context-free grammars are used in many natural language processing applications. They provide the part of the foundation for hierarchical phrase-based machine translation (section 2.4.8).

2.3.1 CKY Parsing

Rules for a context-free grammars can be automatically extracted from a treebank. However, such a grammar is only useful in real natural language processing tasks if it can be used to answer certain questions about novel sentences which were not part of the training corpus. A useful grammar should help determine whether a new sentence is well-formed. If so, we may wish to extract a tree that shows how the grammar was used to derive the sentence. These two tasks, using a grammar to analyze the well-formedness of a sentence and deriving a tree representation, are collectively known as *parsing*.

Numerous algorithms exist for parsing natural language. This section introduces one of the most common algorithms for parsing context-free grammars, the CKY parsing algorithm (Kasami, 1965; Younger, 1967; Cocke and Schwartz, 1970).

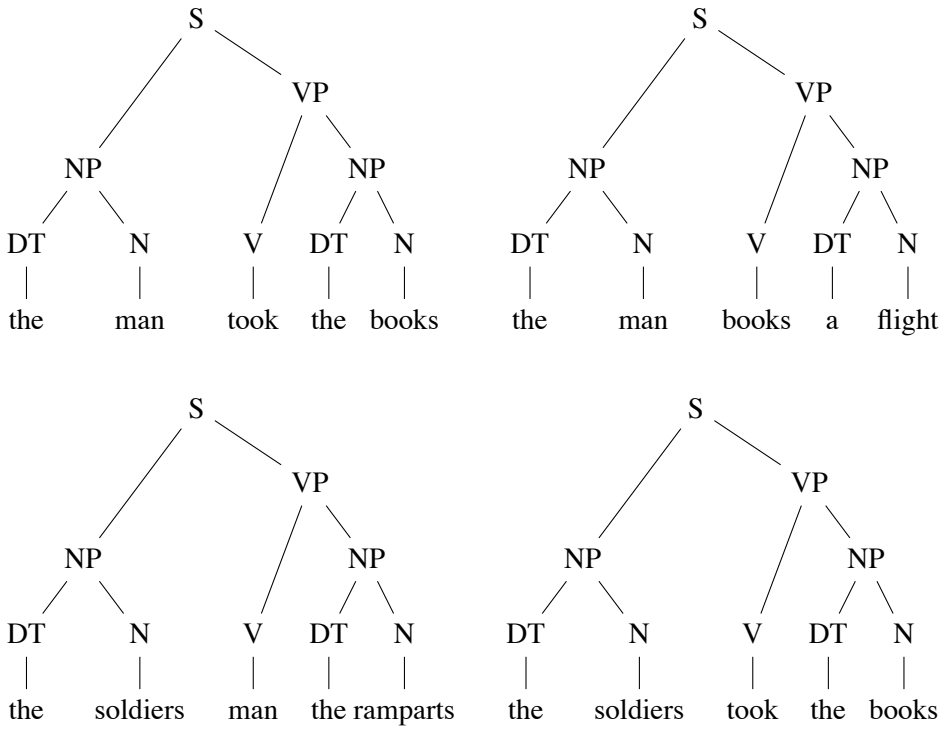

Figure 2.7: Small treebank of four sentences.

Figure 2.7 presents a small treebank representing four sentences and their corresponding trees. The remainder of this section presents the CKY parsing algorithm, walking through a simple parsing example. The example uses a context-free gram-

mar constructed from this treebank.

```
   input  : A sequence in of terminal symbols t; Grammar grammar of rules
   output: A parse chart out
 1 initialize out as an empty chart such that each entry in the chart is an
   initially empty set of nodes.
   // Look up unary rules and insert the left-hand side
      nonterminal into the chart
 2 for i=0; i<n-1; i=i+1 do
 3     foreach unaryRule in grammar.lookup(in[i]) do
 4         entry = new node containing left-hand side of unaryRule;
 5         add entry to out[i][i + 1]
 6     end
 7 end
   // Look up binary rules and insert the left-hand side
      nonterminal into the chart
 8 for span=2; span<=n; span=span+1 do
 9     for i=0; i<n-1; i=i+1 do
10         j = i + span;
11         for k=i+1; k<j; k=k+1 do
12             foreach rule in grammar.lookup(out[i][k],out[k][j]) do
13                 entry = new node containing left-hand side of rule;
14                 add entry to out[i][j]
15             end
16         end
17     end
18 end
   // Return the chart.
19 return out
```

Algorithm 1: CKY parsing algorithm

CKY is a bottom-up parsing algorithm; pseudo-code is presented in algorithm 1. The algorithm takes the terminal symbols of the input sentence and a context-free grammar as input. The algorithm maintains a data structure called the chart (line 0). The chart consists of a two-dimensional array of cells. Each cell is an initially empty set of nodes. A node represents a nonterminal in a parse tree. Figure 2.8 shows an empty chart with a sentence to be parsed.

The algorithm begins by seeding the chart with pre-terminal symbols (lines 1-

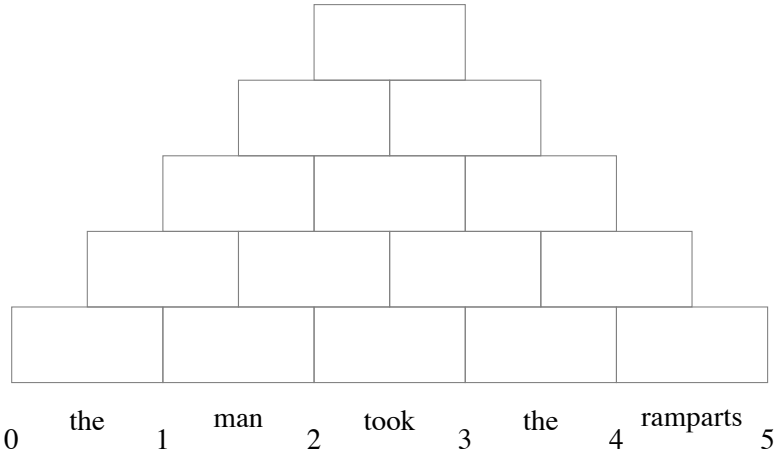

Figure 2.8: CKY chart is empty before parsing begins.

6). For each terminal in the input sentence, the algorithm queries the grammar to find all context-free rules where the right-hand side of the rule exactly matches the terminal symbol. For each rule that matches, a new node is initialized with the left-hand side nonterminal of the rule. These new nodes are stored at the bottom of the chart, in the cell that corresponds with the current terminal span (line 4). Figure 2.9 shows the chart after the pre-terminal symbols have been added.

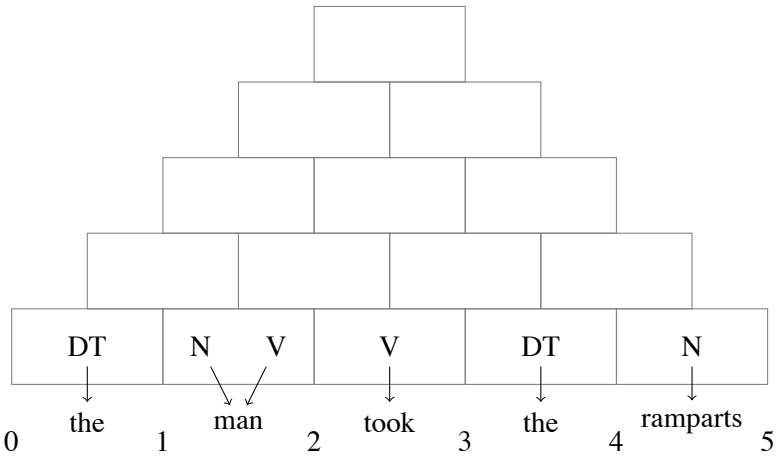

Figure 2.9: CKY chart after pre-terminals have been added.

Pre-terminal symbols correspond to the left-hand side nonterminals of unary

rules. Once all applicable unary rules have been applied (lines 1-6), the bottom level of the chart has been filled in. The algorithm continues filling in the chart, one level at a time (lines 7-17). Each level in the chart corresponds with a span width over the input. The algorithm iterates over each span width (line 7). Within this loop, the algorithm iterates over each potential span starting point (line 8). Since each span under consideration is size 2 or greater, there will always be at least one index between the starting point i and the ending point j. This index, k, represents a possible split point. The algorithm iterates over each possible split point (line 10).

The start, end, and split point indices uniquely identify two chart cells. Because the algorithm is bottom-up, it is guaranteed that these two cells will have already been processed. Each cell may contain one or more nodes; each node represents a nonterminal from the grammar. The algorithm takes the nodes from two cells, and queries the grammar for rules of the form $\alpha \rightarrow \beta\gamma$, where β is a nonterminal from cell[i][k] and γ is a nonterminal from cell[k][j] (line 11). For each rule that matches, the nonterminal associated with that rule is stored in cell[i][k] (lines 12-13).

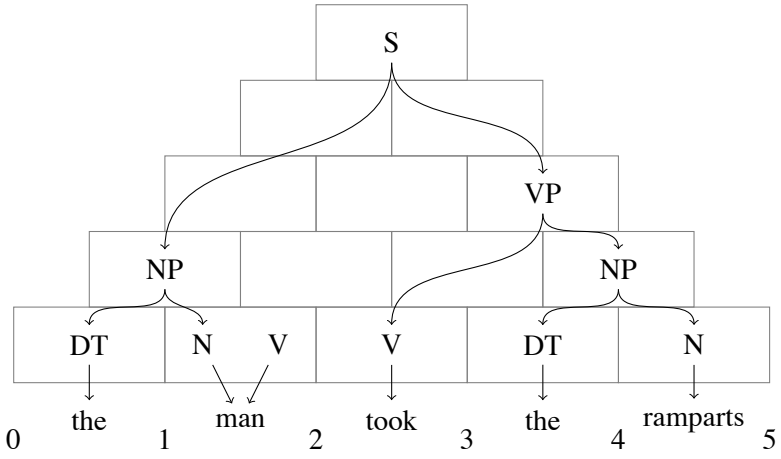

Figure 2.10: CKY chart after all nonterminals have been added.

As an example, consider the point in the algorithm where $i = 0, k = 1, j = 2$. The algorithm queries the grammar for relevant rules using cell[0][1] and cell[1][2]. The grammar finds a rule NP $\rightarrow$ DT N, but no rule of the form $\alpha \rightarrow$ DT V. The result, NP, is stored in cell[0][2]. Figure 2.10 shows the chart after the parsing algorithm has been completed.

If the completed chart contains the start symbol $\mathcal{S}$ in the top cell of the chart (cell[0][$in.size$]), then the sentence is well-formed according to the grammar. A parse tree for the sentence can be obtained by simply tracing the derivation from

the start symbol to all terminal symbols at the bottom of the chart. The parse tree extracted from figure 2.10 is shown in figure 2.11.

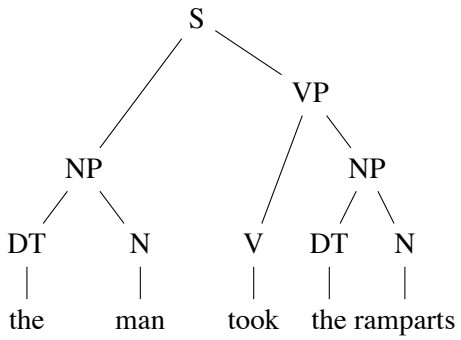

Figure 2.11: Tree extracted from completed parse chart.

2.3.2 Probabilistic Context-free Grammars

In the previous example, note that not all nodes in the chart will necessarily participate in the derivation which corresponds to the final parse tree. The V node in cell[1][2] of figure 2.10 is not used in the parse tree in figure 2.11. In general, it is extremely common for many nodes to not participate in the final parse tree. This poses no problem, since such extra are simply discarded.

A more problematic case arises when tracing the final derivation to extract a parse tree. In section 2.3 we simply said that a tree is extracted by tracing a path from the start symbol at the root of the tree to the terminal symbols. What happens, if more than one such path exists? Natural language is ambiguous. It is very common that multiple possible parse trees could each explain the hierarchical structure that underlies a sentence.

A classic sentence with more than one valid derivation is "Time flies like an arrow." A parse chart for this sentence is shown in figure 2.12. Because this sentence has (at least) two valid derivations, there are two valid trees that can be traced from the root. The two trees encoding in the parse chart for this sentence are shown in figure 2.13. The first interpretation is straightforward and metaphorical — time is flying in a manner resembling an arrow. The second interpretation treats the first two words as a compound noun — there is a particular arrow which "time flies" are fond of.

When large grammars are used to parse sentences in real texts, it is common for sentences to have hundreds or even thousands of possible derivations. In order to

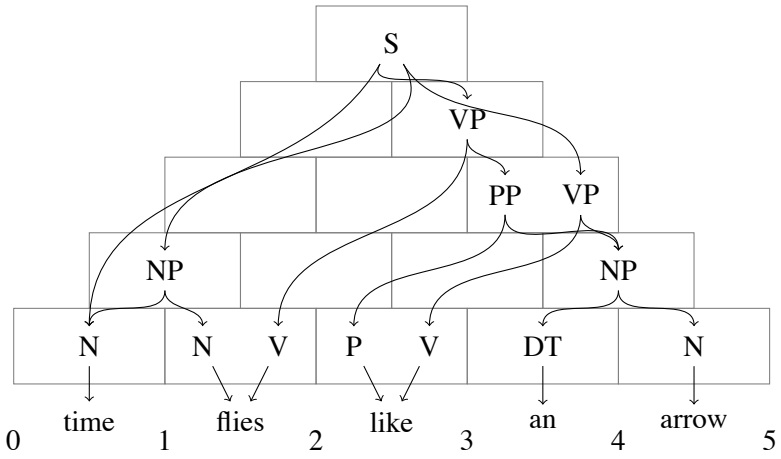

Figure 2.12: CKY chart for an ambiguous sentence.

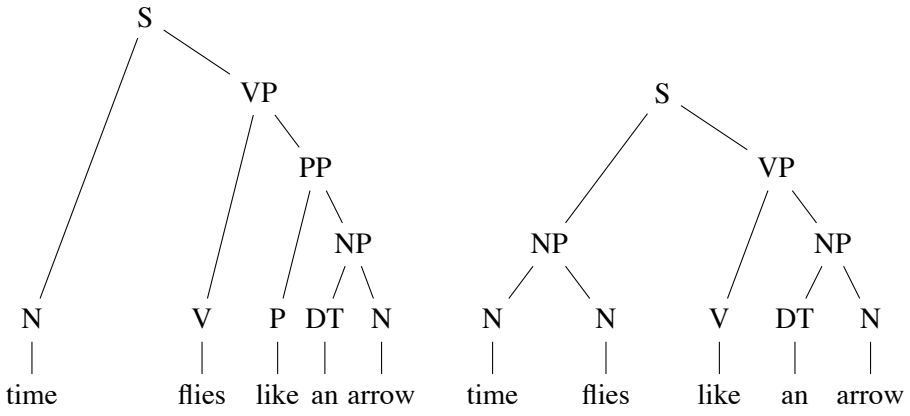

Figure 2.13: Two possible trees derived from an ambiguous sentence.

choose the best derivation from a shared forest, a *probabilistic context-free grammar* (PCFG) can be used. A PCFG is a context-free grammar where each rule has an associated probability.

Recall that each context-free rule is of the form $A \rightarrow \beta$, where β is a sequence of one or more terminal or nonterminal symbols. In a PCFG each rule has a corresponding probability $P(A|\beta)$. Section 2.2 showed that n-gram language model probabilities can be directly estimated from a training corpus of sentences using maximum likelihood estimation. Similarly, PCFG rule probabilities can be estimated from a training treebank corpus.

Figure 2.7 on page 14 presents a small treebank of four sample sentences. Each sentence in the treebank can be broken down into a collection of context-free rule applications; each parent-child relationship in the tree corresponds to a single tree fragment, and a single rule application.

$$P(A|\beta) = \frac{count(A \rightarrow \beta)}{\sum_{\alpha} count(\alpha \rightarrow \beta)} \tag{2.8}$$

The probability of a context-free rule is defined in equation 2.8 to be the number of times the rule was applied in the treebank divided by the number of times any rule with that right-hand side was applied in the treebank. Figure 2.14 shows the context-free rules and associated probabilities for a probabilistic context-free grammar associated with the treebank in figure 2.7.

S → NP VP	P(S | NP VP) = 4/4	= 1.0
NP → DT N	P(NP | DT N) = 8/8	= 1.0
VP → V NP	P(VP | V NP) = 4/4	= 1.0
DT → a	P(DT | a) = 1/1	= 1.0
DT → the	P(DT | the) = 1/1	= 1.0
N → man	P(N | man) = 2/3	= 0.666
V → man	P(V | man) = 1/3	= 0.333
N → books	P(N | books) = 2/3	= 0.666
V → books	P(V | books) = 1/3	= 0.333
N → soldiers	P(N | soldiers) = 2/2	= 1.0
N → ramparts	P(N | ramparts) = 1/1	= 1.0
V → took	P(V | took) = 2/2	= 1.0

Figure 2.14: Context-free rules

Once probabilities are associated with rules in the grammar, those probabilities can be used to help choose a parse tree from the shared forest encoded in the parse chart. The task of selecting the most probable tree from the shared forest is

equivalent to choosing the most probable path through a weighted directed acyclic hypergraph.

$$\hat{T} = \arg\max_{T} \prod_{\alpha \rightarrow \beta \in T} P(\alpha|\beta) \tag{2.9}$$

The most probable tree $\hat{T}$ in the shared forest is the tree where the product of rule probabilities for that tree provides a higher probability than any other tree in the forest (equation 2.9).

2.3.3 Generalized Chart Parsing

The CKY algorithm allows sentences to be parsed using a context-free grammar in time $O(n^3)$ where n is the number of words in the input sentence. However, the algorithm is well-defined only for grammars in *Chomsky Normal Form* (CNF). A grammar is in CNF if the following conditions are met:

- All unary rules are of the form $N \rightarrow t$, where N is a nonterminal and t is a terminal.
- All binary rules are of the form $N \rightarrow PQ$ where N, P, and Q are all nonterminals.
- No rule has more than two symbols on the right-hand side.

Generalized chart parsing (Kay, 1986) is a generalization of CKY parsing and Earley parsing (Earley, 1970) that overcomes this restriction, while maintaining $O(n^3)$ time complexity.

Chart parsing follows the same essential insights as CKY parsing — constituent phrases are constructed in an incremental manner, reusing as much structure as possible during dynamic programming. Chart parsing allows rules with arbitrary numbers of terminals and nonterminals on the right-hand side. In order to maintain cubic runtime, a secondary data structure, called an *active* chart, is maintained.

The standard CKY algorithm takes two nodes from the chart and looks to see if they form the complete right-hand side of any rules in the grammar. Chart parsing performs this step, but also queries the grammar to determine if the two nodes form a strict prefix for any rule's right-hand side. If any prefix matches are found, they are stored in the active chart at the current span indices. Entries in the active chart represent partially completed rule matches.

The final modification to the standard CKY algorithm allows entries in the active chart to be extended. A partial match in the active chart represents a rule, and an index (often called the *dot*) which records how much of the rule has been

matched. Two nodes in the main chart can be combined to create a new node in the main chart if they represent the complete right-hand side of a rule. Similarly, an incomplete match from the active chart can be combined with a node from the main chart if the node from the main chart represents the symbol to the right of the dot in the partially matched rule.

Some of the machine translation models which will be discussed later in this proposal will use context-free grammars which are not in Chomsky Normal Form. While any context-free grammar can be converted into an equivalent grammar which is in CNF, this conversion introduces huge numbers of additional nonterminals and results in a grammar which is far messier than the original. When the grammars involved are *synchronous* context-free grammars (see section 2.4.8), the process of conversion is far more difficult. For the remainder of this proposal, all references to context-free parsing will refer to generalized chart parsing unless otherwise specified.

2.4 Statistical Machine Translation

2.4.1 Noisy Channel Model

The idea that computers could be used to translate human language in a statistical framework has been around at least since Weaver (1949) proposed that translation could be viewed a statistical problem similar to decryption.

However, for the following forty years, nearly all work in machine translation was strictly symbolic, based on encoding linguistic constraints into databases of dictionaries and rules. It was not until Brown et al. (1990) that any published research attempted to use statistical methods to perform translation. The work by Brown et al. at IBM formalized the intuitions of Weaver by modeling translation as a *noisy channel*.

The original statistical machine translation work at IBM used the Canadian Hansards corpus of parliamentary proceedings. Their work involved translating from French f into English e. Following their work, machine translation literature conventionally refers to the source language as f and the target language as e, even when French and English are not the languages under consideration.

According to the noisy channel model, source language sentence f actually began as target language sentence e which, when transmitted over a noisy channel, was transformed into f. The process of translation can then be cast as an attempt to recover the "original" version of the sentence in language e.

$$\begin{aligned}\hat{e} = \arg\max_e P(e|f) &= \arg\max_e P(e,f) \\ &= \arg\max_e \frac{P(e)P(f|e)}{P(f)}\end{aligned} \tag{2.10}$$

Given a source sentence f, the noisy channel model selects the target sentence e which maximizes the product of translation model probability $P(f|e)$ and language model probability $P(e)$. Because the normalization factor $P(f)$ in equation 2.10 is constant, the selection of $\hat{e}$ is not dependent on $P(f)$. This leads to the simplified noisy channel equation 2.11.

$$\hat{e} = \arg\max_e P(e)P(f|e) \tag{2.11}$$

2.4.2 Word Based Translation

Brown et al. (1993) introduced five closely related statistical models for word-based translation that built on the work of Brown et al. (1990). These models are known as IBM models 1-5.

The IBM models are based on the noisy channel model. The translation model probability $P(f|e)$ in the IBM models is defined in terms of three components:

- *Lexical translation model* $P(f_w|e_w)$ models the probability of source word f_w given target word e_w)
- *Fertility model* ϕ models the number of words in the source language sentence that a given target language word will translate as.
- *Reordering model*, also called the *distortion model*, models the probability distribution over non-monotonic word order between source sentence f and target sentence e.

The IBM models vary in how these components are defined. In order to use the models to translate text, a *decoder* is used. Brown et al. (1990) used a variant of *stack decoding* (Jelinek, 1969) to apply the noisy channel model to translate text in a tractable manner. A stack decoder can be considered a finite state transducer.

Germann et al. (2001) provide the only freely available implementation of a word based decoder; their decoder implements word based decoding for IBM Model 4. The GIZA++ toolkit (Och and Ney, 2003) provides a freely available open source implementation to train the IBM models parameters from a parallel corpus.

2.4.3 Word Alignments

In order to estimate the lexical translation probabilities for a word based translation model, a word-aligned training corpus is needed. Word alignments provide a mapping between words in the source language and words in the target language.

Brown et al. (1993) define how word alignments can be trained from a sentence-aligned parallel corpus. This training procedure is implemented as part of GIZA++.

While word based translation is not used as the basis of any modern translation system, the use of word alignments is critical to nearly every modern statistical translation system. The GIZA++ toolkit is very commonly used to get word alignments for a parallel corpus. However, because the IBM models only define one-to-many alignments from target words to source words, the alignments produced by GIZA++ are not many-to-many alignments.

Real translations often have a many-to-many correspondence between source and target words. To work around the one-to-many restriction of the IBM models, Och et al. (1999) propose running GIZA++ in both directions. This results in two alignments: one-to-many and many-to-one. Using a heuristic, the intersection of the two alignment sets is augmented by some of the points from the union of the sets. Koehn et al. (2003) introduce a variant heuristic for augmenting the intersection of the sets, which they call "final-and." In each case, the result is a many-to-many word alignment for each sentence which shows which source words translate as which target words. This process, including the "final-and" heuristic, is commonly used to obtain many-to-many word alignments for use in other parts of the translation process.

2.4.4 Phrase Based Translation

The fertility model in word based translation serves to allow one target word to correspond to multiple source words in translation, but does not allow multiple target words to correspond to one source word. For each target language word, the fertility model selects the number n of source language words that the target language word will translate as; n copies of the target language word are created. This allows the lexical translation model to model simple one-to-one translation of words.

In real language, it is common for an entire sequence of words in a source sentence to correspond to a sequence of words in the target language. In order to allow many-to-many translation, Och et al. (1999) proposed the *alignment template model*; this was the first statistical translation system to model translation as a correspondence from a sequence of words in one language to a sequence of words in another language.

This type of model is known as *phrase based translation*. It is important to note that a *phrase* here is simply a contiguous sequence of words, which do not necessarily comprise a linguistic constituent.

Further seminal work in phrase based translation was introduced by Koehn et al. (2003); their approach has become known as the standard phrase based model. In the standard phrase based model, a stack-based decoder segments in the input into phrases. Rather than choosing a single segmentation, a distribution over possible segmentations is maintained. The translation model maintains a probability distribution for translating a phrase in one language into a phrase in the other language. This contrasts with the lexical translation model in word based translation, which maintained only translations over single words. Phrase based translation, like word based translation, requires a reordering model to allow for non-monotone translation order.

The translation model probability in the standard phrase based approach a combination of three factors:

- *Lexical translation model* $P_w(f|e)$ models the lexical translation probability of the words in source phrase f given the words in target phrase e.
- *Phrase translation model* $P(f|e)$ models the lexical translation probability of the words in source phrase f given the words in target phrase e.
- *Reordering model* d, also called the *distortion model*, models the probability distribution over non-monotonic phrase order between source sentence f and target sentence e.

The lexical translation model here is very similar to the lexical translation model in word based approaches, and is trained with maximum likelihood estimation using word alignment data. The word alignment data allows parallel phrases to be extracted from the training corpus. Once phrases have been extracted, maximum likelihood estimation is used to calculate the phrase translation probability distribution. The training procedures of Marcu and Wong (2002) can be used to extract a joint probability model for distortion using expectation maximization; the joint probability can be marginalized to produce a conditional distortion model d.

2.4.5 Log-linear Model

The word based and phrase based approaches examined thus far all define complex translation models composed of several dependent models (lexical translation model, fertility model, phrase translation model, distortion model). As additional

features are incorporated, the simple noisy channel model (equation 2.11) fails to provides a straightforward mechanism to naturally incorporate additional features.

Additionally, certain features may provide more information than other features. When that happens, it may be valuable to assign a higher weight to features which are more important to translation quality. Koehn et al. (2003), for example, found that weighting the lexical translation model was useful.

One technique for resolving this problem is to move from a noisy channel approach to a more general log-linear framework motivated by maximum entropy techniques (Berger et al., 1996). To replace the noisy channel approach (equation 2.11), Och and Ney (2004) present the log-linear model in equation 2.12 to define the probability of a target sentence e given a source sentence f.

$$P(e|f) = \frac{exp(\sum_{m=1}^{M} \lambda_m h_m(e, f))}{\sum_{e'} exp(\sum_{m=1}^{M} \lambda_m h_m(e', f))} \tag{2.12}$$

In equation 2.12, each h_m is a feature function that provides a score for a given e and f. Each feature function h_m is weighted by λ_m. Given equation 2.12, the most probable translation $\hat{e}$ of source sentence f is the translation e that maximizes $P(e|f)$ (equation 2.13).

$$\hat{e} = \arg\max_{e} \frac{exp(\sum_{m=1}^{M} \lambda_m h_m(e, f))}{\sum_{e'} exp(\sum_{m=1}^{M} \lambda_m h_m(e', f))} \tag{2.13}$$

Note that for a given source sentence f, the denominator in equation 2.13 is constant. Because the desired result is the most probable translation, and not a probability, equation 2.13 can be simplified to equation 2.14.

$$\hat{e} = \arg\max_{e} exp(\sum_{m=1}^{M} \lambda_m h_m(e, f)) \tag{2.14}$$

The result of the arg max operation is not dependent on the exponentiation operation, so equation 2.14 can be further simplified to equation 2.15.

$$\hat{e} = \arg\max_{e} \sum_{m=1}^{M} \lambda_m h_m(e, f) \tag{2.15}$$

Equation 2.15 provides a flexible and extensible mechanism for incorporating arbitrary features into the translation process. Some features which have been found to be useful in phrase based translation include phrase translation probabilities $P(e|f)$ and $P(f|e)$, lexical translation probabilities $P_w(e|f)$ and $P_w(f|e)$, distortion model d, and language model probability $P(e)$.

2.4.6 Minimum Error Rate Training

By modeling the translation decision using a linear combination of features (equation 2.15) additional features that are empirically found to be useful can be incorporated into the model. The model assigns a weight λ to each feature function; this weight determines the impact that a particular feature will have on the translation process. The question then becomes, what values should be assigned to the various λ weights?

Berger et al. (1996) present general techniques, including Improved Iterative Scaling, for determining feature weights in a maximum entropy framework. These techniques would tune feature weights to maximize a translation model probability (equation 2.16).

$$\hat{e} = \arg\max_{e} P(e|f) \tag{2.16}$$

But in translation, the ultimate goal is not to find the translation with the highest score or highest probability. Rather, the goal is to find the best translation e of source sentence f. Och and Ney (2003) introduce a technique called Minimum Error Rate Training (MERT). The intuition behind MERT is that feature function weights should be tuned such that the sentences produced by the decoder are good translations. Given an objective function capable of scoring the quality of a candidate translation against a reference translation, MERT attempts to optimize the feature function weights for the objective function.

MERT is not necessarily guaranteed to converge to a global optimum. When used with the BLEU objective function (see section 2.4.7 below), there are many local optima. In order to deal with local optima, random restarts are required. If the entire MERT process is run twice with the same data and objective function, each run will produce different feature weight values. This can make it difficult to compare published results between two systems, even when the systems use the same underlying model, because the parameter weights will be different.

2.4.7 Translation Objective Functions

If the objective function does a good job of judging the quality of a candidate translation, then the feature weights produced by MERT should be optimized to produce high quality translations. The question then arises, what is a good objective function to measure translation quality? This is a major question in machine translation which has not been completely solved. No existing metric is able to replicate human judgements of translation quality.

Nevertheless, some objective functions have gained widespread use. Many early experiments in statistical machine translation used word error rate (WER)

as an evaluation criteria. WER measures the number of insertions, deletions, and substitutions required to transform a candidate translation hypothesis into a reference translation. When a candidate translation contains most or all of the same words as the reference, but in an order, WER penalizes the candidate substantially. Position-independent word error rate (PER) addresses this issue by treating candidate and reference as unordered bags of words. BLEU (Papineni et al., 2001) is an n-gram precision metric, augmented by a brevity penalty to discourage unduly short sentences. METEOR (Banerjee and Lavie, 2005) was designed to improve on BLEU by incorporating recall and taking synonymy into account. Translation edit rate (TER) (Snover et al., 2006) is similar to WER, but allows arbitrary movement of phrases with no penalty.

Since BLEU was introduced, it has been the most widely used automatic evaluation metric in statistical machine translation. Most published research which uses MERT tunes with BLEU as the objective function.

2.4.8 Hierarchical Phrase Based Translation

Standard phrase based translation does well at modeling local reordering. Local changes to word order between source language and target language are effectively captured by the relatively short phrases that phrase based translation extracts and uses during decoding. An example of this can be seen in the local difference in noun-adjective order between Romance languages and English. A phrase based system trained to translate from Spanish to English would likely contain the phrase pair $\langle$ *el hombre alto* , *the tall man* $\rangle$. Even though the phrase based system contains no formal model of syntax, the local phrases used by such a system effectively capture local reordering. The n-gram language model used in such systems also tends to encourage correct local reordering; a well-trained trigram language model for English will assign a much higher score to *the tall man* than to *the man tall*.

Phrase based translation systems are typically far weaker at dealing with long-distance reordering. The phrases stored by such systems are typically short, often five or fewer words on the source side. Because of the Markov assumptions used in n-gram language models, the language model provides little to no guidance to the decoder in making long-distance word order decisions. Phrase pairs and n-gram language models both fail to model the hierarchical nature of human language. To capture this hierarchical structure, a more powerful formalism is required: context-free grammars (section 2.3).

Context-free grammars provide a good model for the hierarchical structure of human language. A probabilistic context-free grammar is defined by a set of terminals, a set of nonterminals, a start symbol from the nonterminal set, and a set of rules of the form $X \rightarrow \gamma$, where γ is a sequence of terminals and nonterminals, and each

rule has an associated probability. A sentence in the language can be recognized by performing chart parsing using the grammar.

If a context-free grammar can be created for the source language, perhaps the process of translation can simply be treated as a parsing problem. Such a CFG would contain only information about the source language. Any rules for translation must know about both source and target. Aho and Ullman (1969) introduce *synchronous context-free grammars* (SCFG). Rules in synchronous context-free grammars are extended to include information about the target language in addition to the source language.

A synchronous context free grammar is formally defined by a finite set of nonterminal symbols $\mathcal{N}$, a unique start symbol $\mathcal{S} \in \mathcal{N}$, a finite set of terminal symbols in the source language $\mathcal{T}_f$, a finite set of terminal symbols in the target language $\mathcal{T}_e$, and a set of rules $\mathcal{R}$. Each rule is of the form $\mathrm{X} \rightarrow \langle \gamma,\ \alpha,\ \sim \rangle$, where γ is a string of nonterminals and source language terminals, α is a string of nonterminals and target language terminals, and $\sim$ is a one-to-one mapping between nonterminals in γ and nonterminals in α.

Some research (Wu and Wong, 1998; Yamada and Knight, 2001; Galley et al., 2006) has examined the use of SCFGs and related formalisms as a way to incorporate linguistically motivated syntax into statistical machine translation. These attempts have found some success, but in general have not clearly outperformed phrase based methods, which have little or no linguistic foundation.

Chiang (2005) introduces *hierarchical phrase based translation* as a way to take advantage of the hierarchical structure provided by SCFGs, which generally working within the framework of phrase based translation. Chiang's implementation of hierarchical phrase based translation is called Hiero.

The model is formally a SCFG; however, the rules are not linguistically motivated. The training procedure for a monolingual PCFG requires a parsed treebank as a training corpus. No treebank is required to train the SCFG for hierarchical phrase based translation. Instead, as in phrase based translation, the rules for hierarchical phrase based translation are automatically extracted from an aligned parallel corpus.

The nonterminal set for the Hiero grammar is extremely limited. Instead of linguistic constituent categories, such as NP (noun phrase) and VP (verb phrase), the only nonterminals are X (wildcard) and S (start symbol).

Chiang (2005, 2007) report state-of-the-art performance using this hierarchical model. Li and Khudanpur (2008) present an open-source implementation of the hierarchical phrase-based model, called Joshua. Several of the experiments in chapter 5 will be carried out using extensions to Joshua.

Chapter 3

Related Work

$$23^2 - 23 = 506$$

Number of language pairs within the official languages of the European Union

3.1 Scope

This proposal will explore novel techniques that exploit multi-parallel to generate higher quality machine translations. While the field of machine translation is decades old, and research in statistical machine translation dates to the early 1990s, very little research has considered how multilingual resources could be used to enhance translation quality.

However, some existing work has examined techniques indirectly related to this topic. Specifically, related research has considered how multilingual resources and multi-parallel corpora can be used to enhance bilingual translation resources, such as word alignments, phrase tables, and even the decoding process itself.

The remainder of this chapter is structured as follows. Section 3.2 reviews techniques which use multilingual resources to extract bilingual lexicons. Section 3.3 considers how higher quality bilingual word alignments can be extracted when the training corpus is multi-parallel. Section 3.4 presents a bootstrapping technique to improve bilingual translation phrase tables. Techniques which use multiple bilingual phrase tables are reviewed in section 3.5. Section 3.6 examines a technique for combining the outputs of multiple systems to create higher quality overall translations. Finally, the only existing techniques designed specifically to use multiple source languages are reviewed in section 3.7.

3.2 Lexicon induction

Bilingual lexicons can serve as a valuable starting point for translation, especially for low-resource languages and less common language pairs. Mann and Yarowsky (2001) show that existing bilingual lexicons can be used in conjunction with probabilistic models for cognate words to create new lexicons for related languages. English-Czech and English-Polish lexicons can, for example, serve as a starting point for the creation of bilingual lexicons between English and other Slavic languages.

Dyvik (2002) presents a method for automatically extracting a sense-distinguished bilingual lexicon from a bilingual parallel corpus. A related technique by Sammer and Soderland (2007) allows the creation of a multilingual sense-distinguished lexicon from bilingual lexicons and monolingual corpora.

3.3 Improving Word Alignments

Nearly every statistical machine translation technique begins with the assumption of a parallel corpus of aligned sentences. Gale and Church (1991) present algorithms which use expectation maximization and mutual information calculations to align sentences in an unaligned bilingual parallel corpus.

Simard (1999) compares the problem of alignment in a multi-parallel corpus to various multiple sequence alignment problems in computational biology, observing that a multi-parallel corpus contains extra information that should allow for better better bilingual sentence alignments. Simard presents a technique for aligning sentences in a multi-parallel corpus. This technique results in higher quality bilingual alignments.

Kumar et al. (2007) describe a technique for word alignment in a multi-parallel sentence-aligned corpus and show that this technique can be used to obtain higher quality bilingual word alignments than traditional bilingual word alignment techniques.

3.4 Improving Phrase Tables

Multilingual resources can also be used to directly improve the quality of a bilingual translation phrase table. Callison-Burch (2002) presents a technique called co-training in which the best output of several translation systems is used as additional training data, leading to an improvement in translation quality.

Co-training begins with a multi-parallel corpus and several bilingual translation systems. Each bilingual translation system translates the training corpus into the

target language. The target hypothesis for each sentence with the highest language model score is then treated as additional training data, and the training process is repeated.

3.5 Pivot translation

Eisele (2006) proposes that existing bilingual translation systems which share one or more common *pivot* languages can be coupled to build translation systems for language pairs for which no parallel corpus exists; using this approach, for example, existing Arabic-English, Arabic-Spanish, Spanish-Chinese, and English-Chinese systems could together be used to effect an Arabic-Chinese translation system. Wu and Wang (2007) report positive results using a similar technique with a single pivot language in conjunction with a small bilingual training corpus. Utiyama and Isahara (2007) show that in addition to sentence-based pivot methods, phrase translation tables can be built directly from phrase tables that share a pivot language.

Cohn and Lapata (2007) present another pivot approach centered on phrase tables, which they call triangulation. This technique maintains separate phrase tables for each language pair; during decoding, source phrases are translated into multiple intermediate language phrases, which are finally translated into target language phrases.

3.6 Consensus Decoding

In contrast to pivot-based techniques, consensus network decoding (Mangu et al., 2000) attempts to improve translation quality by finding a novel, higher quality hypothesis based on the hypotheses produced by multiple translation systems. Much recent research (Frederking and Nirenburg, 1994; Bangalore et al., 2001; Jayaraman and Lavie, 2005; Rosti et al., 2007) has explored consensus decoding where all systems translate the same language pair. Matusov et al. (2006) adapts this approach to a multilingual setting, performing consensus decoding when translating Japanese and Chinese into English; gains of 4.8 BLEU higher than the single best system are reported. Callison-Burch et al. (2008) report preliminary results that indicate promising results when applying system combination techniques on the multi-source News Commentary corpus.

3.7 Hypothesis Ranking

Alternatively, hypothesis ranking techniques attempt to select the single best hypothesis from a list of output hypotheses produced by different translation systems. Most existing work in hypothesis ranking assumes several bilingual translations, all translating the same language pair. Kaki et al. (1999) and Callison-Burch and Flourney (2001) use only the target language model to rank the hypotheses. This approach follows the intuition that the hypothesis with the highest language model score will be the most fluent. Nomoto (2004) take this one step further by using multiple language models which vote on candidate hypotheses.

Och and Ney (2001) present two techniques, called MAX and PROD, designed specifically for multi-source translation. These two techniques assume multiple bilingual translation engines, each translating from a different source language into a common target language. The translation model scores from each system are used to identify the best translation of a sentence from any of the translation systems. Och and Ney report positive results for combining two or three systems using MAX and even better results for combining three or more systems using PROD. These two technique are examined and critiqued in detail in chapter 4.

Chapter 4

Work to Date

ויהי כל־הארץ שפה אחת ודברים אחדים׃

και ην πασα η γη χειλος εν και φωνη μια πασιν

Erat autem terra labii unius, et sermonum eorumdem.

Es hatte aber alle Welt einerlei Zunge und Sprache.

And the whole earth was of one language, and of one speech.

– Genesis 11:1

4.1 Oracle Hypothesis Ranking

The two techniques that have been used successfully for multi-source translation are sentence-level hypothesis ranking (Och and Ney, 2001) and consensus decoding (Matusov et al., 2006). In this work we are interested in determining whether the techniques presented in Och and Ney (2001) can be replicated using current multi-parallel corpora with long sentences and modern phrase-based decoders, and measuring the translation quality of these techniques against current metrics.

In order to provide a context for the possible gains from the hypothesis ranking methods of Och and Ney (2001), it is worth examining the maximum possible gains in translation quality which these methods can achieve.

To begin, ten bilingual translation systems were trained on the Europarl corpus. The standard phrase-based Moses decoder (Koehn et al., 2007) was used for all ten systems. The system parameters were tuned using minimum error rate training (Och, 2003) to optimize BLEU (Papineni et al., 2001) on the dev2006 development set. The target language for all systems was English. Table 4.1 shows results for these ten systems on the in-domain Europarl test05 data. Scores are listed for the

languages	BLEU	TER	METEOR
da-en	28.4	57.5	52.9
de-en	27.3	58.9	52.4
el-en	29.3	56.4	53.6
es-en	**32.5**	**52.8**	**56.3**
fi-en	24.6	62.1	50.4
fr-en	31.9	53.1	55.8
it-en	29.2	57.1	53.7
nl-en	25.7	62.7	50.4
pt-en	31.8	53.7	56.0
sv-en	**32.7**	**52.3**	**56.6**

Table 4.1: Results of ten bilingual phrase based decoders into English. All systems were trained on Europarl v3. Test set is Europarl test05. Best results are bold.

TER (Snover et al., 2006) and METEOR (Banerjee and Lavie, 2005) metrics in addition to BLEU. We observe that the Swedish and Spanish systems perform the best according to all three metrics. The Dutch and Finnish systems perform the worst. In all experiments, each test sentence had only one reference translation.

Two oracle experiments were conducted to estimate the maximum possible gains in translation quality achievable by hypothesis ranking techniques. All hypothesis ranking methods by definition simply choose one target sentence from a list of two or more possible hypotheses. The best such method is one that always chooses the target sentence which represents the best translation from the available options. In each experiment, an oracle selected the best target sentence from the available hypotheses by selecting the one with the lowest word error rate (WER) when compared with the reference.

languages	BLEU	TER	METEOR
oracle-all	40.8	40.5	62.5

Table 4.2: Scores after combining results of ten bilingual phrase based decoders into English, using a WER-based oracle to choose which system output to select.

The first oracle experiment examined the possible gains when all ten bilingual systems are combined using sentence-level hypothesis ranking. For each test sentence, the oracle selected the hypothesis from the list of system output hypotheses with the lowest WER against the reference. Table 4.2 lists the results. The oracle BLEU score achieved here is 8.3 BLEU higher than the best individual system.

This indicates that the combined translation systems together provide substantial additional information to positively influence translation quality.

system	% selected
da-en	14.1
de-en	9.6
el-en	10.3
es-en	14.0
fi-en	4.0
fr-en	12.9
it-en	7.2
nl-en	5.5
pt-en	9.8
sv-en	12.9

Table 4.3: Percentage of time that sentences from each system were selected in an All-English oracle WER experiment. Score for overall oracle output was 43.8 WER and 40.8 BLEU.

This oracle experiment also tracked for each system the number of times its hypothesis was selected as the best overall hypothesis. Table 4.3 lists these percentages. This distribution is flatter than we anticipated. It is not surprising that the systems which performed the best individually (sv-en, fr-en, and es-en) were chosen a large percentage of the time. However, the da-en system, which ranked seventh individually, was chosen by the oracle more often than any other. Even fi-en and nl-en, systems which performed substantially worse than the others individually, were selected a reasonable number of times. This data suggests that additional research is warranted to investigate the types of sentences for which bilingual systems with different source languages systems perform well.

The second experiment calculated oracle hypotheses for each pair of systems. In this experiment each of the 45 pairs of systems were combined by the WER oracle to simulate ideal sentence-level hypothesis ranking. Table 4.4 lists the absolute increase in BLEU score achieved by the oracle on the test set compared with the best BLEU score achieved by either system individually.

The difference between the best BLEU score for each pair and the oracle score was substantial for nearly all pairs of systems. The lowest absolute increase in BLEU scores (0.6) is seen when combining the worst individual system, Finnish, with the best individual system, Swedish. The mean and median increase in BLEU score for the 46 system pairs is 2.4 BLEU. The average difference between oracle BLEU and the score achieved by the MAX method for the same system pair is 3.3

	da	de	el	es	fi	fr	it	nl	pt	sv
da	—	3.2	**3.7**	2.4	1.9	2.6	**4.0**	2.4	2.4	1.7
de		—	2.7	2.0	1.9	2.0	3.3	2.7	2.1	1.6
el			—	2.1	1.8	2.3	**3.7**	2.6	2.5	2.5
es				—	1.2	3.1	2.4	1.7	3.1	**3.7**
fi					—	1.0	1.9	2.7	1.1	0.6
fr						—	2.4	1.6	3.5	**3.7**
it							—	2.4	2.5	2.7
nl								—	1.8	1.3
pt									—	3.5
sv										—

Table 4.4: Absolute change in BLEU after combining two languages using oracle compared with the best BLEU of either language individually. The largest increases come from combining da & el, el & it, es & sv, fr & sv (each +3.7) and da & it (+4.0). The smallest increases come from combining fr & fi (+1.0) and fi & sv (+0.6). Best results in bold.

BLEU (median 3.4 BLEU). This data shows that substantial gains are achievable from sentence-level hypothesis ranking methods, even when only two systems are combined.

4.2 Hypothesis Ranking using MAX and PROD

Given that significant gains in translation quality are possible through sentence-level hypothesis ranking, this section considers the MAX and PROD ranking methods proposed by Och and Ney (2001).

The original work is limited in the scope of its experiments. At the time, no large multi-parallel corpus was available. The authors assembled a training corpus from the *Bulletin of the European Union* with 117k-139k sentences per language for eleven European languages. Their test set was restricted to sentences 10 to 14 words in length. The metrics in common use today, including BLEU, METEOR, and TER, had not been developed. Results were reported in terms of word error rate (WER) and position-independent word error rate (PER). Their decoder used the alignment template system (Och et al., 1999).

In the following sections, we attempt to reproduce the techniques and results presented in Och and Ney (2001). We do this to answer three important questions:

- Can the techniques for multi-source translation presented in Och and Ney

(2001) be replicated using current phrase-based decoders?

- Can the positive results they report be replicated on a larger data set which includes longer sentences?
- And finally, do the results presented for WER correlate with current automatic evaluation metrics?

4.3 MAX Ranking

Och and Ney (2001) propose that the best output translation from distinct bilingual translation systems can be chosen by taking the hypothesis with the highest score according to a noisy channel model. Given n source languages, the best translation $\hat{\mathbf{e}}$ is defined using both the language model and a translation model as

$$\hat{e} = \arg\max_{e} \{p(e) \cdot \max_{n} p(f_n|e)\} \quad (4.1)$$

$$= \arg\max_{e,n} \{p(e) \cdot p(f_n|e)\} \quad (4.2)$$

languages	BLEU	WER	PER	TER	METEOR
sv	32.7	60.2	53.6	52.3	56.6
sv+es	**33.1**	59.2	52.6	51.2	**56.9**
sv+es+fr	33.0	**58.8**	**52.1**	**50.9**	56.8
sv+es+fr+el	32.6	58.9	52.3	51.0	56.3

Table 4.5: Combination using MAX ranking method.

This method is straightforward, and has the advantage that no modifications to the bilingual decoders are needed. The decoders must simply be capable of reporting language model and translation model probabilities along with each hypothesis. We note that the translation model probability reported by the decoder is in fact an approximation of $p(f|e)$ as $p(f|e,d)$ where d is the derivation selected by the decoder.

Because the translation model probabilities from various systems are not necessarily comparable, it might be valuable to train weights for each system. Och and Ney (2001) report that such weights did not diverge much from one in their experiments. Due to time constraints, we did not perform system weighting.

	da	de	el	es	fi	fr	it	nl	pt	sv
da	—	**0.4**	**0.1**	-0.8	-1.3	-1.3	**0.3**	-1.4	-0.7	-1.6
de		—	-0.2	-0.6	-0.8	-2.0	-0.1	-0.8	-0.8	-1.1
el			—	-0.2	-1.8	-1.0	**0.6**	-1.9	-0.3	-0.5
es				—	-1.5	**0.5**	-0.9	-2.6	**0.1**	**0.3**
fi					—	-2.9	-1.3	-0.3	-1.9	-2.3
fr						—	-1.6	-3.7	**0.2**	**0.2**
it							—	-1.5	-1.0	-1.0
nl								—	-2.4	-2.9
pt									—	-0.1
sv										—

Table 4.6: Absolute change in BLEU after combining two languages using MAX ranking method compared with the best BLEU of either language individually. Best results come from combining es & sv (+0.4), es & fr (+0.5), and el & it (+0.6). Worst results come from combining fi & fr (-2.9), nl & sv (-2.9), and fr & nl (-3.7). Only 20% of MAX pairwise combinations led to an improvement in BLEU. Results which indicate an improvement in BLEU are bold.

	da	de	el	es	fi	fr	it	nl	pt	sv
da	—	**-0.8**	**-1.1**	**-0.1**	2.1	0.6	**-2.3**	1.6	**-0.7**	1.1
de		—	0.6	0.0	0.7	3.1	**-1.0**	0.1	0.2	1.2
el			—	**-1.0**	3.4	0.2	**-1.5**	2.9	**-0.9**	**-0.2**
es				—	2.0	**-2.0**	**-0.2**	2.6	**-1.6**	**-1.0**
fi					—	5.1	1.2	**-2.3**	2.4	3.5
fr						—	1.5	5.0	**-1.0**	**-0.9**
it							—	0.9	**-0.2**	0.3
nl								—	2.4	3.6
pt									—	**-0.6**
sv										—

Table 4.7: Absolute change in WER after combining two languages using MAX ranking method compared with the best WER of either language individually. Best results come from combining da & it (-2.0) or from fi & nl (-2.0). Worst results come from combining fr with nl (+5.0) or with fi (+5.1). Only 44% of MAX pairwise combinations led to an improvement in WER. Results which indicate an improvement in WER are bold.

	da	de	el	es	fi	fr	it	nl	pt	sv
da	—	**0.9**	-0.4	-2.6	-1.8	-2.8	**0.3**	-1.5	-2.0	-1.4
de		—	-1.4	-3.1	-1.3	-3.5	**0.3**	-1.1	-1.9	-0.5
el			—	0.0	-2.9	-1.0	**0.2**	-3.1	-0.1	-0.2
es				—	-4.4	**0.3**	-2.8	-5.7	-0.2	**0.4**
fi					—	-4.3	-0.8	-0.1	-4.1	-4.0
fr						—	-2.3	-5.4	**0.2**	**0.1**
it							—	-0.8	-2.2	-2.1
nl								—	-4.7	-3.9
pt									—	**0.4**
sv										—

Table 4.8: Absolute change in BLEU after combining two languages using MAXLL ranking method compared with the best BLEU of either language individually.

4.3.1 Experiments using MAX

To examine how well MAX performs, we can take the output hypotheses produced by the bilingual translation systems, and apply the method using the translation model probabilities and language model probabilities reported by the decoder. We begin by selecting the translation system which produced the highest BLEU score, then added the system which gave the highest incremental gain. Table 4.5 reports results for the MAX method when applied with an increasing number of source languages. Och and Ney (2001) report the highest gains from MAX by combining three languages, French, then Swedish, then Spanish. We see the same three languages in our results, but with Swedish placed before French.

Because Och and Ney (2001) report only WER and PER, we report those metrics in addition to current metrics so that our results can be more directly compared with theirs. We see that our best results using MAX (58.8 WER for sv+es+fr) are significantly worse than their best result (52.0 WER for fr+sv+es). Most of this discrepancy is likely due to our use of a different corpus with longer sentences.

We also performed MAX ranking on all 45 system pairs, using every pair of foreign languages to translate into English. Table 4.6 shows the absolute change in BLEU score for the MAX ranking compared with the best BLEU score for either input system. For comparison with the results in Och and Ney (2001), table 4.7 presents this data in terms of WER. Our experiment reports results for all available languages, including German and Finnish; results for those two languages were not included in Och and Ney (2001).

Och and Ney (2001) show positive results using the MAX method for all 21 language pairs on which they report results. They report positive results showing absolute decreases in WER ranging from -0.5 (fr & it) to -4.3 (da & nl). Even if Finnish and German are excluded, we observe a wide range of mostly negative results (+5.0 for fr & nl, -2.0 for da & it). In total, 56% of combinations (25 out of 44) resulted in higher word error rate. Fully 80% of combinations (35 out of 44) resulted in an decrease in BLEU.

The results above show that the simple MAX approach simply does not improve translation quality for the majority of language pairs. In addition, Akiba et al. (2002) report (in a bilingual setting) that the hypothesis chosen by MAX ranking often differs from the hypothesis chosen by a human performing manual ranking.

For many MAX combinations, an improvement in WER was matched by an improvement in BLEU, and an increase in WER was matched with a lower BLEU score. However, 26% of language pairs showed an improvement in WER but a decline in BLEU.

4.3.2 Extending MAX to a Log-linear Framework

Given that most current statistical translation systems are based on a log-linear combination of features rather than a noisy channel model, the question immediately arrives whether MAX might work if the log-linear score of each sentence is used in the argmax calculation (equation 4.2) instead of the noisy channel product. We define MAXLL as follows:

$$\hat{e} = \arg\max_{e}\{\max_{n} exp(\sum_{i} \lambda_i h_i(e, f_n))\} \quad (4.3)$$

$$= \arg\max_{e,n}\{exp(\sum_{i} \lambda_i h_i(e, f_n))\} \quad (4.4)$$

To test this method, we combine all 45 system pairs (as in section 4.3.1) but use MAXLL in place of MAX. This method performs quite badly. The number of systems for which BLEU increases is the same as for MAX (9 out of 45 pairs); however, for pairs in which MAXLL does poorly, it performs worse than MAX. The worst performance comes from the es & nl pair, where MAXLL scores -5.7 BLEU worse than the best of either system individually. This poor performance should not be surprising, as the scores returned by each system are not comparable.

4.4 PROD Ranking

The MAX method provides a simple method to choose the best output from among two or three bilingual translation systems. However, it fails to effectively make use of larger numbers of source languages; in fact, translation quality degrades when additional source languages are incorporated. The PROD method presented in Och and Ney (2001) addresses this shortcoming in MAX. Kay (1997) observed that if multiple translation engines independently produce the same hypothesis, that is strong evidence that the hypothesis is a good one. The PROD method follows this insight.

The PROD method attempts to approximate a true multi-source decoding algorithm by incorporating probabilities associated with each bilingual translation model. In this approach, given n source languages, the best translation $\hat{e}$ is defined as

$$\hat{e} = \arg\max_{e} \{p(e) \cdot \prod_{n=1}^{N} p(f_n|e)\} \tag{4.5}$$

Paul et al. (2005) explore a related technique in the bilingual setting, where a hypotheses is selected if its average translation model times language model score is significantly higher than competing hypotheses.

4.4.1 Constraint Decoding

The PROD method requires that for each target hypothesis e, a translation probability $p(f_n|e)$ must be calculated for each source language sentence f_n. Each target hypothesis e is produced by one of the bilingual decoders described earlier in section 4.2. Each sentence f_n is given as a source sentence.

The standard phrase-based decoder produces a 1-best or n-best list of hypotheses when given a source sentence. There is no guarantee that a particular target sentence e will appear in a decoder's n-best list of hypotheses. So, in order to calculate $p(f_n|e)$ for every source language n, we modified the Moses decoder to permit *constraint decoding*. When constraint decoding is used, the phrase-based search is constrained so that only hypotheses which are consistent with the desired target output are considered.

The standard phrase-based decoding model (Koehn, 2004) creates "stacks" of translation options that cover contiguous phrases in the input sentence. Each translation option stores a target language phrase. The decoder attempts to trace a path through the translation options, creating partial hypotheses as it proceeds, in such a way that all source words are covered by a translation option, resulting in a complete hypothesis.

Each partial hypothesis represents a partial translation into the target language. Constraint decoding is defined by restricting the creation of partial hypotheses. Whenever a partial hypothesis would be constructed, the partial translation for that partial hypothesis is examined. If the partial translation is compatible with the desired target sentence, the partial hypothesis is constructed. If the partial translation is not compatible, meaning it is not a prefix to the desired target sentence, the partial hypothesis is pruned.

In this way, a desired target sentence can be provided to the decoder as a constraint; the desired target sentence will be produced as the result as long as the decoder's model and parameters are capable of reaching the desired target sentence.

4.4.2 Experiments using PROD

The decoder provides feature values, including $p(f|e)$, for each sentence that it successfully translates. By using constraint decoding, and providing the decoder with both source sentence and desired target sentence, the translation model probabilities required for PROD can be obtained.

Unfortunately, the vast majority of target sentences presented to the constraint decoder resulted in failure. In these cases, the decoder was not able to reach the desired target sentence from the provided source sentence given the translation model, the language model, and the feature parameters. Table 4.9 presents a sample of results that illustrate this problem.

	da-en	de-en	es-en	fr-en
% reachable	10.5	9.8	11.5	10.6

Table 4.9: Percentage of sentences reachable by the Swedish-English system when constrained by the output of the listed systems.

Attempts to increase the number of reachable constraint sentences by turning off all pruning during constraint decoding did not lead to a substantial increase in the number of reachable sentences. Even worse, the particular sentences reachable in the test set was not the same across the various translation systems. As as result, the decoder was unable to provide $p(f_n|e)$ for all required source languages n in the vast majority of test sentences.

It is reasonable to ask whether the PROD method could be applied for those minority of sentences which are reachable. What conditions are necessary in order to apply the PROD method when just two source languages are used? In other words, are we able to use equation 4.5 to determine $\hat{e}$? Equation 4.5 requires that we know $p(f_n|e)$. Consider the concrete example where Spanish-English and French-

English outputs are to be ranked using the PROD method. The Spanish system translates the Spanish input into English hypothesis e_{es} and provides $p(f_{es}|e_{es})$. Likewise the French system translates the French input into English hypothesis e_{fr} and provides $p(f_{fr}|e_{fr})$. If the Spanish system is able to successfully translate the Spanish source sentence f_{es} into e_{fr} using constraint decoding, $p(f_{es}|e_{fr})$ becomes available. Likewise $p(f_{fr}|e_{es})$ becomes available if the French system is able to translate f_{fr} into e_{es}. Only a very small number of test sentences fulfill these conditions for each pair of systems. For three or more systems, the problem is even worse.

4.4.3 Discussion on PROD

Och and Ney (2001) do not discuss the problem of unreachable sentences when calculating $p(f_n|e)$ for PROD. We now briefly examine why our experiments found so few sentences to be reachable by a constrained phrase based decoder, while no such problem was reported by Och and Ney (2001).

The first possible factor that presents itself is the data used in the experiments. The test corpus used by Och and Ney (2001) was extracted from the *Bulletin of the European Union* and was restricted so that all reference sentences in the test set were 10 to 14 words long. By contrast, the Europarl test05 test set includes reference sentences up to 135 words long. In the experiments in section 4.4.2, the sentences reachable during constraint decoding have an average length of 14.2 words. We note that this is longer than the *maximum* sentence length used by Och and Ney (2001). The average reference sentence length in our complete test set is 29.0 words.

If we restrict our test set to only those sentences where the reference has 10-14 words, the percentage of reachable sentences increases from approximately 10% of the entire test corpus to approximately 25% of the subset of 10-14 word sentences for a given pair of systems. In other words, even when only short sentences are considered, a large majority cannot be reached during constraint decoding.

The second possible factor is the choice of translation search algorithm. The experiments presented in this paper used the standard phrase based Moses decoder, modified to allow constraint decoding[1]. Och and Ney (2001) use an alignment template decoder which implements an early phrase-based translation model.

Phrase based translation allows a single word to be translated as a phrase only if the word and phrase were aligned during training. Consider the case where the decoder needs to translate source word a, and adjacent target words xyz still need to be generated; no phrase $a \rightarrow xyz$ exists in the phrase table, but entries for $a \rightarrow x$,

[1] Moses decoder, subversion revision 1857

$a \rightarrow y$, and $a \rightarrow z$ all do exist. A word based decoder might be able to deal with this by assigning a a fertility of three, then translating the three words individually, but a phrase-based system cannot.

We are unable to fully explain how Och and Ney (2001) were able to reach all hypotheses during decoding that are required to calculate $p(f_n|e)$ for all system combinations that they report for PROD.

Given the positive results initially reported for PROD, we believe that there is still value in replicating this technique, at minimum for use as a baseline as more advanced techniques are developed.

4.5 Summary of Existing Work

We have shown that significant gains in translation quality are possible using hypothesis ranking, but that the MAX technique is not a reliable method for hypothesis ranking. We have also shown that limitations in current decoding techniques prevent the use of PROD with phrase-based systems. Our findings show a limit to the claim by Och and Ney (2001) that this method for combining multiple source languages is independent of translation models. In particular, PROD is useful only insofar as $p(f_n|e)$ can be reasonably approximated for an arbitrary source language sentence f_n when constrained by an arbitrary target language sentence e.

Chapter 5

Planned Work

Когда сделался этот шум, собрался народ и пришел в смятение, ибо каждый слышал их говорящих его наречием.

فلما صار هذا الصوت اجتمع الجمهور وتحيّروا لان كل واحد كان يسمعهم يتكلمون بلغته .

Et ce bruit ayant eu lieu, il s'assembla une multitude, qui fut confondue de ce que chacun les entendait parler dans sa propre langue.

At this sound, they gathered in a large crowd, but they were confused because each one heard them speaking in his own language.

– Acts 2:6

5.1 Overview

Virtually all existing research in machine translation assumes a single source language and a single target language. Yet as Kay (1980) points out, if a document is translated into more than one target language, it will likely be translated into many other target languages.

Multi-parallel corpora provide the same texts in multiple languages. Large multi-parallel corpora have become available to researchers in recent years. Such corpora a rich source of information which could be exploited to reduce ambiguity and improve translation choices.

We have surveyed the state of the art in techniques to exploit multiple languages. Our survey found only three publications which explicitly examine multilingual translation. Och and Ney (2001) used translation model scores to rank hypotheses from multiple bilingual systems; our existing work (chapter 4) has shown

that MAX is not as reliable as originally reported, and that PROD cannot be applied as-is to existing phrase-based translation. Matusov et al. (2006) used standard system combination techniques to combine the English outputs from Chinese-English and Japanese-English translation systems; this preliminary showed that system combination approaches can be used in a multi-lingual setting to substantially improve translation results. Finally, Callison-Burch et al. (2008) report preliminary results that indicate promising results when applying system combination techniques with more than two source languages.

Initial research (Matusov et al., 2006; Callison-Burch et al., 2008) has indicated that standard system combination techniques for consensus decoding can be successfully applied to multiple source languages. We propose more thorough multilingual system combination experiments using consensus decoding (section 5.2) for use as a baseline for other techniques.

No research has successfully applied hypothesis ranking techniques to multiple source languages using standard phrase based or hierarchical phrase based statistical machine translation methods. We propose a thorough examination of hypothesis ranking techniques in a multilingual context using state-of-the-art statistical machine translation systems (section 5.3).

No published research has shown how standard phrase based translation or hierarchical phrase based translation can be directly adapted to use multiple source languages. We propose to apply existing lattice input techniques for machine translation to multiple source languages (section 5.4).

No published research has shown how multilingual translation could be directly modeled as part of the decoding process. We propose to develop novel techniques for multi-synchronous parsing (section 5.5) to model multiple source languages and multiple target languages.

The ultimate proposed scientific contribution of this proposal will be to answer the question: how can multilingual resources be best exploited to improve the quality of machine translation?

5.2 Consensus Decoding

Given a common target language, consensus network decoding (Mangu et al., 2000) techniques align multiple candidate translations into a weighted word lattice called a consensus network. The weights in the network are designed to encode the level of confidence that a word arc in the network represents the consensus among the candidate translations.

The network can be intersected with a separately trained n-gram language model. The result is word lattice with an identical topology to the original consensus net-

work, but with weights adjusted according to the likelihood in the language model the paths through the network.

The path through the weighted lattice with the best score is interpreted to be the consensus translation. Consensus decoding has been extensively studied recently in the context of bilingual translation (Bangalore et al., 2001; Jayaraman and Lavie, 2005; Rosti et al., 2007). The technique has been found to be highly effective as a means to improve translation quality; consensus translations have consistently achieved BLEU scores substantially higher than any of the translation systems being combined.

Preliminary results (Matusov et al., 2006; Callison-Burch et al., 2008) indicate that consensus decoding can be successfully used to combine the outputs of systems with different source languages. We propose to use consensus decoding as a solid baseline against which to compare other multi-source translation techniques.

The experiments in the following sections will use the Europarl multi-parallel corpus. Unfortunately, the previously published results cannot be used for direct comparison with our proposed experiments. Matusov et al. (2006) reports results translating from Chinese and Japanese into English. The multilingual experiments in Callison-Burch et al. (2008) are conducted using many of the same language pairs as in Europarl; but the corpus used (News Commentary) is different than ours (Europarl). Additionally, the multilingual experiments in Callison-Burch et al. (2008) are very preliminary — only high-level results are reported.

We propose the following experiments:

- Train bilingual translation systems for the 110 core EU language pairs in Europarl. The training corpus will be Europarl. Phrase based systems (using Moses) and hierarchical phrase based systems (using Joshua) will be trained for each language pair.
- Using each system, translate a common test corpus.
- Using each of the 11 Europarl languages as a target language, perform consensus decoding for each target language. Multiple experiments will be performed for each target language exploring various numbers and combinations of source languages.

5.3 Hypothesis Ranking

Straightforward approaches to multi-source translation require training multiple bilingual translation systems. Each translation system must be capable of translating from a different source language into a common target language. Given a

multi-parallel text to be translated, each source input is translated using the appropriate bilingual translation system. Consensus decoding then constructs a consensus network, applies an n-gram language model, and extracts a novel consensus translation from the lattice.

A simpler approach would be to choose one output hypotheses from the multiple machine translation systems. To choose which hypothesis represents the best translation, any one of several techniques can be applied to rank the output hypothesis.

5.3.1 Language Model Ranking

Existing research (Kaki et al., 1999; Callison-Burch and Flourney, 2001; Nomoto, 2004) has proposed using a language model to rank the output hypothesis of multiple bilingual translation systems. This approach follows the intuition that the hypothesis with the highest language model score will be the most fluent. In previous research all translation systems used a common source language and a common target language.

We propose the following novel experiments:

- Re-use the bilingual translation systems trained in section 5.2 for the 110 core EU language pairs in Europarl.
- Re-use the bilingual system output hypothesis generated in experiments in section 5.2.
- Using each of the 11 Europarl languages as a target language, perform language model hypothesis ranking for each target language. Multiple experiments will be performed for each target language exploring various numbers and combinations of source languages.

5.3.2 PROD using Hierarchical Oracle Extraction

Existing research (Och and Ney, 2001) has proposed using translation models to rank the output of multiple bilingual translation systems in a multi-source setting. Our existing work in section 4.4 shows that the PROD technique proposed by Och and Ney cannot be replicated using standard phrase based techniques, due to an extremely high number of unreachable hypotheses.

Given the positive results reported by Och and Ney, additional work is warranted to successfully reproduce the PROD technique. Zhifei Li (p.c.) has developed a technique for finding the closest reachable translation to a reference translation in the shared forest constructed by a hierarchical phrase based decoder. Recall

that the problem with applying PROD was that translations constructed by one system are often not reachable by another system. We propose to use Li's oracle extraction technique to circumvent this problem. The translation output of one system can be provided as a "reference translation" to other systems. The oracle extraction technique will find the closest translation that is reachable by the system. It is an empirical question whether these "closest reachable" translations will serve as an acceptable substitute for the true hypothesis when performing PROD hypothesis ranking.

We propose the following novel work:

- Adapt the PROD technique to allow use of the closest translation reachable by the translation system, instead of only allowing the exact translation (which is often unreachable).
- Re-use the bilingual system output hypothesis generated in experiments in section 5.2.
- Using the oracle extraction technique of Zhifei Li (p.c.) to find the closest translation to a reference reachable by the hierarchical translation system, perform experiments using the adapted PROD technique. Multiple experiments will be performed for each target language exploring various numbers and combinations of source languages.

5.3.3 Weighted Hypothesis Ranking

The MAX and PROD hypothesis ranking methods of Och and Ney (2001) give equal consideration to the contribution of each translation model. This is potentially problematic, especially for MAX, since each translation model may not be equally good.

$$\hat{e} = \arg\max_{e,n} \{p(e) \cdot p(f_n|e)\} \tag{5.1}$$

$$\hat{e} = \arg\max_{e} \{p(e) \cdot \prod_{n=1}^{N} p(f_n|e)\} \tag{5.2}$$

Recall that MAX chooses the hypothesis whose combined language model and translation model score is greater than that of any other hypothesis (equation 5.1). PROD chooses a hypothesis such that the combination of the language model and all available translation models is greater than that of any other hypothesis.

To address the problem that translation model scores are not necessarily comparable, a weight could be assigned to each system. The weighted MAX and PROD hypothesis ranking methods are shown in equations 5.3 and 5.4, respectively.

$$\hat{e} = \arg\max_{e,n} \{p(e) \cdot p(f_n|e)^{\lambda_n}\} \tag{5.3}$$

$$\hat{e} = \arg\max_{e} \{p(e) \cdot \prod_{n=1}^{N} p(f_n|e)^{\lambda_n}\} \tag{5.4}$$

Och and Ney report only informal experiments where such weighting is performed. They simply state that in these informal experiments the system weights did not diverge much from 1. Our experiments in chapter 4 found that MAX does not consistently produce positive results, in contradiction to the results originally reported by Och and Ney (2001).

We propose the following experiments:

- Implement a method to perform system weighting for use in equations 5.3 and 5.4.
- Re-run the full suite of MAX experiments from section 4.3 using system weighting to determine if system weighting yields consistently positive results for MAX hypothesis ranking.
- Re-run the full suite of PROD experiments from section 5.3.2 using system weighting to determine if system weighting improves results for PROD hypothesis ranking.

5.3.4 Inside Score Ranking

Existing hypothesis ranking strategies use translation model or language model scores to select a hypothesis. These techniques select a hypothesis based on an approximate measure of translation system confidence in that hypothesis. The intuition is that a hypothesis which the translation system is extremely confident in may be a better translation than a hypothesis which a system is not confident in. Can other, perhaps more accurate, measures of system confidence be calculated?

Recall that translation in a hierarchical phrase based system is essentially a parsing task. One result of a successful parse is a shared forest that represents possible translations. Typically, the best tree from the forest is selected as the hypothesized translation. In most forests, the string of words that correspond to a hypothesis could be generated by any one of several trees in the forest.

To calculate the confidence of a translation, we propose as a novel strategy to calculate the probability of the hypothesis given the shared forest. This probability is represented as the inside probability (Baker, 1979).

We propose the following experiments:

- Modify the Joshua hierarchical phrase based translation system to allow calculation of inside probabilities for hypotheses in a shared forest.
- Run experiments using the previously trained translation systems which use the inside probability to rank hypotheses. Compare this result with other ranking techniques.
- Run experiments which combine the inside probability with translation model probabilities to determine if the inside probability can improve the effectiveness of other ranking techniques, such as MAX and PROD.

5.4 Multi-Source Translation using Lattice Input

Multi-source translation methods attempt to exploit alternative representations of a single source sentence to derive a higher quality translation that could be constructed if only a single representation of the source were available. In this proposal, we have assumed that multiple representations of a source will necessarily be in multiple source languages. However, multi-source inputs may be alternative representations of an input sentence in a single source language.

Dyer (2007) proposes the use of a *confusion network* to model multiple morphological analyses of a morphologically rich source language, Czech. Dyer found that this monolingual multi-source input helped to improve translation quality. Earlier work (Bertoldi et al., 2007) successfully used confusion networks to model multiple hypothesized speech transcriptions for spoken machine translation. In these approaches, phrase based stack decoders are modified to accepts a confusion network as input instead of a single sentence.

Dyer et al. (2008) extends the multi-source techniques of Dyer (2007) from simple confusion networks to general lattice inputs. Dyer et al. also show how the chart parsing techniques used in hierarchical phrase based translation can be generalized to accept lattice inputs. Improved translation quality is seen when lattices are used to represent multiple word segmentations for Chinese input and multiple morphological analyses for Arabic input.

In this section, we propose novel experiments which further explore the use of lattice inputs in multi-source translation.

5.4.1 Monolingual Multi-Source

Bertoldi et al. (2007) and other earlier work has shown that lattices can successfully model multiple interpretations of a monolingual speech transcription, as input to translation systems. We propose to further explore the idea of lattices for representing monolingual source input.

In many use cases of translation, a user needs to translate from a language they understand into a foreign language. When the translation system does poorly in such cases, the user may be able to reword a problematic sentence at little cost. If a user, or an automatic pre-processing module, is able to provide one or more paraphrases of an input sentence, the paraphrases and the original sentence can be encoded in a word lattice which then is provided to the translation system.

We propose the following experiments, in collaboration with Chris Dyer and Chris Callison-Burch:

- Finish modifying the Joshua hierarchical phrase based translation system to accept general word lattice input.
- Train hierarchical phrase based models for English-Chinese and English-Arabic using GALE Chinese-English and Arabic-English corpora.
- GALE provides test sets for Chinese-English and Arabic-English with multiple English references for each sentence. Create English word lattices using the multiple English versions of each sentence.
- Run English-Chinese and English-Arabic experiments using the English word lattices as input.

5.4.2 Multilingual Multi-Source

Lattice techniques for translation could also be applied for multilingual inputs. In such cases, word lattices can be constructed which represent the aligned multilingual input sentences.

This technique would be the first multi-source translation method which attempts to take advantage of multiple source languages in the decoding process itself. This contrasts with hypothesis ranking methods, which simply choose the a single hypothesis from a list of hypotheses, and with consensus decoding methods, which construct an *output* lattice from the available hypotheses and choose a possibly novel translation by following the best path through the output lattice. Taking advantage of multiple source languages in the decoding process is a major step forward, one that has the potential to significantly improve translation quality over consensus decoding and hypothesis ranking.

We propose the following experiments:

- Re-use the translation models trained on Europarl for earlier experiments.
- Generate multilingual source lattices using multiple Europarl languages.
- Run experiments which use these source lattices as input for various target languages, using the Joshua hierarchical decoder. If time permits, also run these experiments using the Moses phrase based decoder.

5.5 Multi-Synchronous Decoding

Hierarchical phrase based decoding use chart parsing techniques with a synchronous grammar to parse a single source input and simultaneously construct a synchronous tree that represents a single target output.

A traditional grammar includes a single language component on the right-hand side of each rule. A synchronous grammar includes two synchronous components on the right-hand side of each rule, one for a source language and one for a target language.

We define a *multi-synchronous grammar* as a synchronous grammar with multiple components to the right-hand side of each rule. In this section, we propose novel techniques to use multi-synchronous grammars in statistical machine translation.

5.5.1 Hierarchical Multi-Target Translation

The language model serves to encourage translations which are fluent in the target language. The use of a language model in the decoding process is an extremely critical requirement to obtain high quality translations; unfortunately, this integration can slow down the decoding process by effectively blowing up the size of the non-terminal set. The use of cube pruning (Chiang, 2007; Huang and Chiang, 2007) can be used to greatly mitigate any slowdowns caused by language model integration, at little or no cost to translation quality.

We propose a novel technique designed to improve translation quality through *multi-synchronous parsing* and the use of multiple language models in different target languages. We propose to modify the hierarchical phrase based decoding process to accept a multi-syncronous grammar; each rule would have one source language component on the right-hand side and two or more target language components on the right-hand side.

This technique would translate from a single source language simultaneously into two or more target languages. The decoding process would integrate one language model for each target language. The intuition is that translations which are fluent across all target translations are at minimum more likely to be consistent translations, and should also be of higher quality.

The major research question to be resolved is whether multiple language models can be efficiently integrated into the decoding process. We propose to generalize the cube pruning algorithm to accomplish this integration. We also propose the following experiments:

- Extend the Joshua hierarchical phrase based decoder allow multiple target languages.
- Translate Europarl languages using the multi-target multi-synchronous method proposed above, and compare these results to earlier bilingual experiments.

5.5.2 Hierarchical Multi-Source Translation

The chart parsing algorithm in hierarchical phrase based translation iterates over each starting index, ending index, and split point index in the source input, resulting in cubic runtime $O(n^3)$ where n is the number of words in the input sentence.

We propose a novel technique to extend hierarchical phrase based translation to accept multiple source inputs. We propose to extend the chart parsing algorithm to accept a multi-synchronous grammar with multiple source components and a single target component. We expect this technique to result in higher quality translations.

However, this extension to chart parsing comes at a high computational cost. The runtime of this extended algorithm is $O(n^{3\ell})$ where ℓ is the number of source languages.

We propose the following experiments:

- Extend the Joshua hierarchical phrase based decoder allow multiple source languages.
- Attempt to translate Europarl languages using the multi-source multi-synchronous method proposed above, and determine if this approach is tractable in practice.
- If this technique is tractable, compare the results to earlier bilingual experiments.
- Explore novel techniques to incrementally parse one source language at a time.

5.5.3 Approximate Inference using Loopy Belief Propagation

Standard phrase based decoding models translation as finite state transduction. Hierarchical phrase based decoding models translation as parsing. However, we see in section 5.5.2 that extending chart parsing to multiple source languages comes at a high computational cost.

To work around the high computational complexity of exact multi-synchronous parsing with multiple source inputs, we turn to an alternative modeling strategy. Smith and Eisner (2008) explore the use of graphical models in the context of monolingual and bilingual parsing. They propose that such parsing tasks can be modeled with undirected graphical models. Their work uses loopy belief propagation as an approximate inference technique to solve the parsing problem.

We propose to explore extensions to Smith and Eisner (2008) to use graphical models and loopy belief propagation as an approximate, but tractable, inference technique to solve the multi-synchronous multi-source parsing problem.

5.5.4 Phrase Based Multi-Source Translation

As another possible alternative to multi-synchronous hierarchical decoding, we propose exploring possible mechanisms for multi-source translation extensions to standard phrase based decoding.

Klein and Manning (2001) show that diverse parsing algorithms can be generalized as a common set of operations on hypergraphs. There is reason to expect that this work could be further generalized to a larger set of decoding algorithms, including phrase based decoding.

We propose to examine whether multi-source phrase based translation can be pursued in the framework of generalized hypergraph decoding.

5.6 Conclusion

Multi-parallel texts provide a rich source of information which could be exploited to reduce ambiguity and improve translation choices. Despite this rich promise, very little research has been conducted in this area. This work proposes novel research to explore the area of statistical multilingual translation.

We propose adapting existing bilingual techniques (consensus decoding, language model reranking) to the multi-source case. We propose novel work that leverages an existing parsing technique (oracle extraction) for use in multi-source hypothesis ranking (PROD). We propose in-depth exploration of system weighting for hypothesis ranking. We propose a novel hypothesis ranking technique (inside probability).

We propose novel work that uses lattice input to translate monolingual multi-source input. We propose novel work that uses lattice input to translate multilingual multi-source input.

We propose novel work in multi-synchronous parsing. We propose examining whether multi-synchronous multi-target translation can improve translation quality, and whether multi-synchronous multi-source translation is tractable. We propose the use of undirected graphical models and approximate inference as a potential method to deal with multi-source multi-synchronous parsing in a tractable manner. Finally, we propose that generalized hypergraph decoding techniques may be applicable to multi-source phrase based translation.

Multilingual translation is an open research topic with enormous potential. We have high hopes that this research will open the door to higher quality machine translation.

Bibliography

Alfred V. Aho and Jeffery D. Ullman. Syntax directed translations and the pushdown assembler. *Journal of computer and system sciences*, 3:37–56, 1969.

Yasuhiro Akiba, Taro Watanabe, and Eiichiro Sumita. Using language and translation models to select the best among outputs from multiple mt systems. In *Proc. COLING*, 2002.

James Baker. Trainable grammars for speech recognition. In D.H. Klatt and J.J. Wolf, editors, *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pages 547–550, 1979.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proc. ACL*, 2005. URL `http://www.aclweb.org/anthology-new/W/W05/W05-0909.pdf`.

Srinivas Bangalore, German Bordel, and Giuseppe Riccardi. Computing consensus translation from multiple machine translation systems. In *Proc. ASRU*, 2001.

Adam Berger, Vincent Della Pietra, and Stephen Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22 (1):39–72, March 1996. URL `http://www.aclweb.org/anthology-new/J/J96/J96-1002.pdf`.

Nicola Bertoldi, Richard Zens, and Marcello Federico. Speech translation by confusion network decoding. In *Proc. ICASSP*, April 2007.

Peter Brown, John Cocke, Stephen Della Pietra, Vincent Della Pietra, Frederick Jelinek, John Lafferty, Robert Mercer, and Paul Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16:79–85, 1990. URL `http://www.aclweb.org/anthology-new/J/J90/J90-2002.pdf`.

Peter Brown, Vincent Della Pietra, Peter deSouza, Jenifer Lai, and Robert Mercer. Class-based n-gram models of natural language. *Computational Linguistics*,

1992. URL http://www.aclweb.org/anthology-new/J/J92/J92-4003.pdf.

Peter Brown, Vincent Della Pietra, Stephen Della Pietra, and Robert Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19:263–311, 1993. URL http://www.aclweb.org/anthology-new/J/J93/J93-2003.pdf.

Chris Callison-Burch. Co-training for statistical machine translation. Master's thesis, U. Edinburgh, 2002.

Chris Callison-Burch and Raymond Flourney. A program for automatically selecting the best output from multiple machine translation engines. In *Proc. MT Summit VIII*, pages 63–66, 2001.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. (Meta-) evaluation of machine translation. In *Proc. WMT*, pages 136–158, 2007.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Further meta-evaluation of machine translation. In *Proc. WMT*, pages 70–106, 2008. URL http://www.aclweb.org/anthology/W/W08/W08-0309.

David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proc. ACL*, pages 263–270, 2005.

David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228, 2007.

Noam Chomsky. Three models for language. *IRE Transactions on Information Theory*, 2:113–124, 1956.

J. Cocke and J. I. Schwartz. Programming languages and their compilers. Technical report, Courant Institute of Mathematical Sciences, New York University, 1970.

Trevor Cohn and Mirella Lapata. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proc. ACL*, pages 728–735, 2007.

Chris Dyer. The 'noisier channel': translation from morphologically complex languages. In *Proc. ACL Workshop on Statistical Machine Translation*, 2007. URL http://www.aclweb.org/anthology-new/W/W07/W07-0729.pdf.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. Generalizing word lattice translation. In *Proc. ACL*, pages 1012–1020, June 2008. URL `http://www.aclweb.org/anthology-new/P/P08/P08-1115.pdf`.

Helge Dyvik. Translations as semantic mirrors: From parallel corpus to wordnet. Technical report, University of Bergen, 2002.

Jay Earley. An efficient context-free parsing algorithm. *CACM*, 13(2):94–102, 1970.

Andreas Eisele. Parallel corpora and phrase-based statistical machine translation for new language pairs via multiple intermediaries. In *Proc. LREC*, 2006.

Tomaž Erjavec. MULTEXT-East version 3: Multilingual morphosyntactic specifications, lexicons and corpora. In *Proc. LREC*, pages 1535–1538, 2004.

Robert Frederking and Sergei Nirenburg. Three heads are better than one. In *Proc. ANLP*, pages 95–100, 1994.

William Gale and Kenneth W. Church. Identifying word correspondences in parallel texts. In *Proceedings of the Fourth DARPA Speech and Natural Language Workshop*, 1991.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve Deneefe, Wei Wang, and Ignacio Thayer. Scalable inference and training of context-rich syntactic translation models. In *Proc. ACL*, 2006. URL `http://www.aclweb.org/anthology-new/P/P06/P06-1121.pdf`.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. Fast decoding and optimal decoding for machine translation. In *Proc. ACL*, pages 228–235, Toulouse, France, July 2001. URL `http://www.aclweb.org/anthology-new/P/P01/P01-1030.pdf`.

Liang Huang and David Chiang. Forest rescoring: Faster decoding with integrated language models. In *Proc. ACL*, 2007.

Shyamsundar Jayaraman and Alon Lavie. Multi-engine machine translation guided by explicit word matching. In *Proc. EAMT*, 2005.

Frederick Jelinek. Fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, pages 675–685, 1969. URL `http://researchweb.watson.ibm.com/journal/rd/136/ibmrd1306D.pdf`.

Satoshi Kaki, Setsuo Yamada, and Eiichiro Sumita. Scoring multiple translations using character n-gram. In *Proc. NLPRS*, pages 298–302, 1999.

T. Kasami. An efficient recognition and syntax analysis algorithm for context free languages. Technical Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Bedford, MA, 1965.

Martin Kay. The proper place of men and machines in language translation. Xerox PARC working paper, 1980. Published 1997 in *Machine Translation* 12:3–23.

Martin Kay. Algorithm schemata and data structures in syntactic processing. In Barbara Grosz, Karen Spärck-Jones, and Bonnie Webber, editors, *Readings in natural language processing*, pages 35–70. Morgan Kaufmann Publishers Inc., San Francisco, 1986.

Martin Kay. The proper place of men and machines in language translation. *Machine Translation*, 12:3–23, 1997. First appeared as a Xerox PARC working paper in 1980.

Dan Klein and Christopher D. Manning. Parsing and hypergraphs. In *Proc. IWPT*, 2001. URL `http://www.cs.berkeley.edu/~klein/papers/klein_and_manning-parsing_and_hypergraphs-IWPT_2001.pdf`.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit X*, 2005.

Philipp Koehn. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proc. AMTA*, 2004.

Philipp Koehn, Franz Joseph Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pages 127–133, Edmonton, Canada, 2003. URL `http://www.iccs.inf.ed.ac.uk/~pkoehn/publications/phrase2003.pdf`.

Philipp Koehn, 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. Moses: Open source toolkit for statistical machine translation. In *ACL Demonstration Session*, 2007.

Shankar Kumar, Franz Och, and Wolfgang Macherey. Improving word alignment with bridge languages. In *Proc. EMNLP-CoNLL*, pages 42–50, 2007.

Zhifei Li and Sanjeev Khudanpur. A scalable decoder for parsing-based machine translation with equivalent language model state maintenance. In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical*

Translation (SSST-2), pages 10–18, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W08/W08-0402.

Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400, Oct 2000.

Gideon Mann and David Yarowsky. Multipath translation lexicon induction via bridge languages. In *Proc. NAACL*, 2001.

Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proc. EMNLP*, Philadelphia, Pennsylvania, July 2002.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. Computing consensus translation from multiple machine translation systems using enhanced hypothesis alignment. In *Proc. EACL*, pages 33–40, 2006.

Tadashi Nomoto. Multi-engine machine translation with voted language model. In *Proc. ACL*, pages 494–501, 2004.

Franz Och. Minimum error rate training in statistical machine translation. In *Proc. ACL*, pages 160–167, Sapporo, Japan, July 2003. URL http://www.fjoch.com/acl03.pdf.

Franz Och and Hermann Ney. Statistical multi-source translation. In *Proc. MT Summit VIII*, 2001.

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

Franz Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449, 2004. URL http://www.aclweb.org/anthology-new/J/J04/J04-4002.pdf.

Franz Och, Christoph Tillman, and Hermann Ney. Improved alignment models for statistical machine translation. In *Proc. SIGDAT-EMNLP*, 1999.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*, pages 311–318, 2001. URL http://acl.ldc.upenn.edu/P/P02/P02-1040.pdf.

Michael Paul, Takao Doi, Youngsook Hwang, Kenji Imamura, Hideo Okuma, and Eiichiro Sumita. Nobody is perfect: ATR's hybrid approach to spoken language translation. In *Proc. IWSLT*, pages 55–62, 2005.

Philip Resnik, Mari Broman Olsen, and Mona Diab. The Bible as a parallel corpus: Annotating the 'book of 2000 tongues'. *Computers and the Humanities*, 33(1–2): 129–153, 1999.

Antti-Veikko I. Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie J. Dorr. Combining outputs from multiple machine translation systems. In *Proc. NAACL-HLT*, pages 228–235, 2007.

Marcus Sammer and Stephan Soderland. Building a sense-distinguished multilingual lexicon from monolingual corpora and bilingual lexicons. In *Proc. MT Summit XI*, Copenhagen, Sept 2007.

Claude Shannon. Prediction and entropy of printed English. *Bell System Technical Journal*, 30:50–64, 1951. URL `http://www.princeton.edu/~wbialek/rome/refs/shannon_51.pdf`.

Michel Simard. Text-translation alignment: Three languages are better than two. In *Proc. EMNLP*, 1999.

David Smith and Jason Eisner. Dependency parsing by belief propagation. In *Proc. EMNLP*, 2008.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proc. AMTA*, pages 223–231, 2006. URL `http://www.cs.umd.edu/~snover/pub/amta06/ter_amta.pdf`.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proc. LREC*, pages 2142–2147, 2006.

UN. United Nations parallel text. LDC Catalog No.: LDC94T4A, 1994.

Masao Utiyama and Hitoshi Isahara. A comparison of pivot methods for phrase-based statistical machine translation. In *Proc. NAACL/HLT*, pages 484–491, 2007.

Warren Weaver. Translation. Memo, 1949. Published 1955 in *Machine translation of languages: fourteen essays*.

Dekai Wu and Hongsing Wong. Machine translation with a stochastic grammatical channel. In *Proc. ACL*, pages 1408–1414, 1998. URL `http://www.aclweb.org/anthology-new/P/P98/P98-2230.pdf`.

Hua Wu and Haifeng Wang. Pivot language approach for phrase-based statistical machine translation. In *Proc. ACL*, pages 856–863, 2007.

Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proc. ACL*, pages 303–310, 2001. URL `http://www.aclweb.org/anthology-new/P/P01/P01-1067.pdf`.

D.H. Younger. Recognition and parsing of context-free languages in time n cubed. *Information and Control*, 10(2):189–208, 1967.