Phrase Alignment Models for Statistical Machine Translation

John Sturdy DeNero

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2010-161
http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-161.html

December 16, 2010

Phrase Alignment Models for Statistical Machine Translation

by

John Sturdy DeNero

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Engineering — Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Dan Klein, Chair
Professor Stuart Russell
Professor Tom Griffiths
Professor David Chiang

Fall 2010

Phrase Alignment Models for Statistical Machine Translation

by

John Sturdy DeNero

Abstract

Phrase Alignment Models for Statistical Machine Translation

by

John Sturdy DeNero

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Dan Klein, Chair

The goal of a *machine translation* (MT) system is to automatically translate a document written in some human input language (e.g., Mandarin Chinese) into an equivalent document written in an output language (e.g., English). This task—so simple in its specification, and yet so rich in its complexities—has challenged computer science researchers for 60 years. While MT systems are in wide use today, the problem of producing human-quality translations remains unsolved.

Statistical approaches have substantially improved the quality of MT systems by effectively exploiting *parallel corpora*: large collections of documents that have been translated by people, and therefore naturally occur in both the input and output languages. Broadly characterized, statistical MT systems translate an input document by matching fragments of its contents to examples in a parallel corpus, and then stitching together the translations of those fragments into a coherent document in an output language.

The central challenge of this approach is to distill example translations into reusable parts: fragments of sentences that we know how to translate robustly and are likely to recur. Individual words are certainly common enough to recur, but they often cannot be translated correctly in isolation. At the other extreme, whole sen-

tences can be translated without much context, but rarely repeat, and so cannot be recycled to build new translations.

This thesis focuses on acquiring translations of *phrases*: contiguous sequences of a few words that encapsulate enough context to be translatable, but recur frequently in large corpora. We automatically identify phrase-level translations that are contained within human-translated sentences by partitioning each sentence into phrases and aligning phrases across languages. This alignment-based approach to acquiring phrasal translations gives rise to statistical models of phrase alignment.

A statistical phrase alignment model assigns a score to each possible analysis of a sentence-level translation, where an analysis describes which phrases within that sentence can be translated and how to translate them. If the model assigns a high score to a particular phrasal translation, we should be willing to reuse that translation in new sentences that contain the same phrase. Chapter 1 provides a non-technical introduction to phrase alignment models and machine translation. Chapter 2 describes a complete state-of-the-art phrase-based translation system to clarify the role of phrase alignment models. The remainder of this thesis presents a series of novel models, analyses, and experimental results that together constitute a thorough investigation of phrase alignment models for statistical machine translation.

Chapter 3 presents the formal properties of the class of phrase alignment models, including inference algorithms and tractability results. We present two specific models, along with statistical learning techniques to fit their parameters to data. Our experimental evaluation identifies two primary challenges to training and employing phrase alignment models, and we address each of these in turn.

The first broad challenge is that generative phrase models are structured to prefer very long, rare phrases. These models require external pressure to explain observed translations using small, reusable phrases rather than large, unique ones. Chapter 4 describes three Bayesian models and a corresponding Gibbs sampler to address this

challenge. These models outperform the word-level models that are widely employed in research and production MT systems.

The second broad challenge is structural: there are many consistent and coherent ways of analyzing a translated sentence using phrases. Long phrases, short phrases, and overlapping phrases can all simultaneously express correct, translatable units. However, no previous phrase alignment models have leveraged this rich structure to predict alignments. We describe a discriminative model of multi-scale, overlapping phrases that outperforms all previously proposed models.

The cumulative result of this thesis is to establish model-based phrase alignment as the most effective approach to acquiring phrasal translations. Only phrase alignment models are able to incorporate statistical signals about multi-word constructions into alignment decisions and score coherent phrasal analyses of full sentence pairs. As a result, phrase alignment models outperform classical word-level models in both generative and discriminative settings. This result is fundamental to the field: the models proposed in this thesis address a general, language-independent alignment problem that arises in all state-of-the-art statistical machine translation systems in use today.

Acknowledgements

This dissertation is a direct result of my having spent five years in a truly exceptional research group at Berkeley, working within an exciting global research community. Every person in our research group and many people from around the world have influenced my work. I have highlighted the largest contributions below, but I am grateful for them all.

I consider myself tremendously lucky to have been advised by Dan Klein. At Berkeley, Dan created a research environment that was at the same time thrilling, productive, light-hearted, motivating, cooperative, and fun. Dan's energetic commitment to research and mentorship has rubbed off on each one of us who has had the opportunity to work with him. He has taught me so many things: how to structure a research project, how to evaluate an idea, how to find the flaws in published work, how to find issues with my own ideas, how to teach, how to iterate, how to experiment, how to find a job, and even how to navigate the cocktail hour of an academic conference.

Dan is a brilliant problem solver who has a unique grasp of the relationship between natural language and computation. These qualities certainly contributed to my successes as a graduate student and to the contents of this thesis. But, I'll most fondly remember Dan's generosity. He has given me far more of his time and guidance than I could have expected or asked for. He created excellent opportunities for me, prepared me to take advantage of them, and coached me through the trickiest parts. He's also quite the matchmaker—at least between students and problems. He introduced me to the problem of phrase alignment in my first six weeks of graduate school, and I've been smitten ever since. Graduate school has been such a highlight of my life; I can't thank Dan enough for making it so.

I have also worked with an amazing set of people at Berkeley. My first paper with Dan Gillick and James Zhang taught me how much more I could learn and accomplish working with other smart people than by myself. Alex Bouchard-Cote taught me virtually all that I know about sampling and Bayesian statistics. (Alex, please forgive me for the notational shortcuts I've taken in this thesis.) Aria Haghighi not only taught me how to write down a principled statistical model (as my first GSI), but also how to hack. The set of ICML and NIPS papers I read is entirely defined by searching the author list for the name Percy Liang. Adam Pauls dared to work with me on all manner of translation projects when everyone else abandoned me for other (less frustrating) problems. David Burkett has become my new role model for understated talent. John Blitzer taught me to talk about computer science just for the fun of it. David Hall taught me that CS research is so much better when properly engineered. Mohit Bansal taught me to ask more questions. All of you, along with the rest of the Berkeley community, have kept me smiling for many years.

I would also like to thank my many external collaborators. David Chiang has been an excellent technical mentor who demonstrated to me how to conduct a truly thorough experimental investigation. Kevin Knight and Daniel Marcu continue to ensure that machine translation is an exciting research area in which to work and constantly draw new talented people into the field. Chris Callison-Burch keeps conjuring up new translation-related problems for us to solve. I enjoyed working with my collaborators at Google Research so much that I've come back for more.

My wife, Jessica Wan, has been immensely patient and supportive through the many long nights that led up to this dissertation. I think she now knows more NLP buzzwords ("Dirichlet process") and names of researchers ("Sharon Goldwater") than most second-year graduate students in the field. The rest of my family has also earned my deepest gratitude for continually trying to understand what it is that I do, even if they can't quite fathom why I do it.

Five years is a long time. My five years at Berkeley changed my perspective, my interests, and my understanding of the world in profound and lasting ways. I'm already nostalgic for the great times and great company I found at Cal. Thank you all who contributed to my journey.

To my wife Jessica, for always
complimenting and complementing me perfectly.

Contents

Chapter 1

Introduction

This thesis proposes model-based techniques to solve the central learning problem in statistical machine translation: how to identify phrasal translations in parallel corpora. This chapter provides a non-technical introduction to statistical machine translation and the phrase alignment problem. Chapter 2 provides a detailed technical description of a full phrase-based translation pipeline, which includes the existing baseline approach to identifying phrasal translations. The novel contributions of this thesis begin in Chapter 3.

1.1 Statistical Machine Translation

Soon after the first electronic computers became available, Warren Weaver (1949) proposed that computers might one day be able to take as input a document written in some natural human language, and automatically produce an equivalent document written in some target language—a task that we now refer to as *machine translation* (MT). After 60 years of research and development, the machine translation services that are now freely and widely available receive hundreds of millions of requests each week (Helft, 2010). The approach that underlies today's best performing systems is

rooted in statistical learning theory.

Broadly characterized, statistical MT systems translate a source-language input document by matching fragments of its contents to documents that have already been translated by people, and then stitching together the corresponding human translations of those fragments. All knowledge of how to translate is implicitly expressed in a large collection of human-translated documents, called a *parallel corpus*. Parallel corpora are naturally occurring phenomena: many news articles, transcriptions of government proceedings, corporate marketing materials, and personal websites are regularly published in multiple languages.

Statistical MT systems are *statistical* in that they choose among many possible ways of translating a document using statistics gathered from a parallel corpus. They employ statistical learning techniques to make those decisions. In particular, the parameters of a system are trained to guide the translation engine toward outputs that closely match human reference translations.

This general approach to translating using parallel corpora is complicated by the fact that fragment-level translations must be discovered in translated documents. Documents do not come annotated with how they decompose into smaller, translatable units. This discovery process is challenging because language is full of nuance and ambiguity, and any given language fragment, however short or long, can and will be translated in many different ways. Nonetheless, this task of discovering translatable fragments is fundamental to statistical MT systems and is the primary focus of this thesis.

1.2 The Task of Translating Sentences

As a simplifying assumption, machine translation systems generally assume that each *sentence* within a document can be translated independently of the rest. Sentence

independence affects all aspects of the task: training conditions, test conditions, and evaluation measures. We will consider each of these sub-tasks in turn in order to precisely define a translation task.

1.2.1 Learning from Example Translations

The vast majority of statistical MT systems today are trained on parallel corpora that consist of *sentence pairs*.[1] Each sentence pair is a pair of sentences, each of which is a translation of the other.[2]

The collection of sentence pairs is analyzed as a whole in order to train a model of translation that will generalize effectively to new sentences. In particular, a phrase-based translation system searches the parallel corpus for *phrase pairs*. A phrase pair is a pair of contiguous sequences of words—one phrase in each language—that are translation-equivalent in some contexts. These phrase pairs are stored along with their frequency statistics, and they serve as the building blocks of novel translations.

Statistical MT systems do vary in the precise form of the patterns that they harvest from parallel corpora; some allow discontiguous phrases or syntactically annotated fragments. However, the common character of statistical MT systems is that they discover and count translated language fragments (e.g., phrase pairs) automatically in parallel corpora using statistical learning techniques.

1.2.2 Translating New Sentences

Open-domain machine translation systems are typically tasked with translating novel sentences that do not appear in any parallel corpus available to the system. Statistical

[1] Such collections are typically denoted sentence-aligned parallel corpora.

[2] No regard is given to which sentence was the original (or whether they are both translations of some original third sentence).

systems consider many candidate translations, each formed by recycling fragments from previously observe sentences, and choose among those candidates using a linear scoring function called a model. The space of candidates is large for several reasons. For an input sentence to be translated, there are typically many ways of partitioning that sentence into phrases that have been observed previously in the parallel corpus. Phrases are often observed multiple times with different phrasal translations. Beyond choosing a phrasal segmentation of the input and a way of translating each phrase, a system must decide how to order those translations into a coherent sentence in the target language. Considering all of these variants when translating naturally occurring sentences, which can often contain 60 words or more, is a challenging search problem.

Statistical translation models integrate several sources of information into their scoring functions. The two most important sources are frequency statistics on the phrase pairs used and the *language model*, a phrase-factored model of how likely a proposed output sentence is to occur in the output language. These model features are discussed in more detail in Chapter 2.

1.2.3 Evaluating System Output

The evaluation of machine translation systems is an active research area in itself. While human judgements of output quality are regularly solicited, the field primarily relies on automatic measures of output quality to track progress. Automatic measures have several advantages: they are replicable, fast, easily shared, easily employed, and cost effective. Of course, the challenge of evaluation is to devise a measure of output quality that correlates well with true output quality and human judgments.

Throughout this thesis, we use the Bilingual Evaluation Understudy (BLEU) metric (Papineni *et al.*, 2002). Despite some known issues, BLEU is the de facto standard for automatically evaluating statistical machine translation systems. It is a precision-

based evaluation measure that quantifies the overlap in contiguous phrases between the proposed output translation and a set of human reference translations. BLEU collects precision statistics on a per-sentence basis, but these statistics are aggregated over a test corpus to provide a more statistically robust evaluation.

1.3 The Role of Alignment Models

The core statistical learning problem in machine translation is to automatically discover and count the phrase pairs present in a large parallel corpus. For a single sentence pair, we must first determine how the words, idioms, and constructions of each sentence correspond to those of its translation. This problem of determining correspondence is called *alignment*. Determining the alignment between translations allows us to identify phrase pairs.

Alignment models are statistical models that score proposed alignments between sentences. In this thesis, we will consider both generative models, which are trained to explain the observed sentences via an unobserved alignment, and discriminative models, which are trained to match human-generated reference alignments. While discriminative models have been shown to outperform generative ones, generative models are preferred in practice because they do not require annotated alignments—instead they induce alignments automatically from raw parallel corpora.

1.4 Word Alignment and Phrase Alignment

Phrase-based statistical machine translation systems generate translations that decompose into phrases, where each output phrase corresponds to exactly one input phrase, and *vis versa*. These symmetric phrasal matchings have proven to serve as a useful structured output space that can be scored efficiently and effectively to generate

high quality translations.

However, the alignment models that are used to extract phrasal translation pairs from parallel corpora typically do not employ this symmetric phrase matching structure, but instead allow words to align more freely. In particular, the classic *word alignment models* still in widespread use today allow alignment structures that do not respect phrasal contiguity and are not symmetric. While subsequent chapters describe this contrast in detail, it will suffice for present purposes to observe that there is a discrepancy between the structure of alignments used to analyze a parallel corpus and the structure of translations generated by a phrase-based machine translation system. This discrepancy has various negative consequences: heuristics must "clean up" the spurious alignments generated by classical models, alignment models are unable to learn multi-word phrasal patterns in the data directly, and the structural divergence forces the use of relatively crude estimators for phrasal models. All of these issues can be addressed by replacing word alignment models with phrase alignment models.

1.5 Contributions of this Thesis

The purpose of this thesis is to evaluate the following hypothesis: phrase alignment models will lead to higher quality translations relative to word alignment models in phrase-based machine translation systems.

In order to evaluate this hypothesis, we describe learning techniques for alignment models that directly predict symmetric phrase alignments. The output of these phrase alignment models is compared to state-of-the-art word alignment baselines. The effectiveness of these techniques is established experimentally in a series of alignment and machine translation experiments.

The specific contributions of this thesis include:

- Complexity analyses for the phrase alignment model class,
- Inference algorithms for this model class,
- Empirical results for phrase-factored generative models,
- An analysis of the degenerate behavior of maximum likelihood estimators for phrase-factored generative models,
- Bayesian priors for joint and conditional generative models.
- A Gibbs sampler for phrase-factored models,
- Empirical results for Bayesian generative models,
- Empirical results for discriminative phrase models, and
- Definitions, inference algorithms, and empirical results for *extraction set* models.

The cumulative result of this thesis is to establish model-based phrase alignment as the most effective approach to acquiring phrasal translations. Only phrase alignment models are able to incorporate statistical signals about multi-word constructions into alignment decisions and score coherent phrasal analyses of full sentence pairs. As a result, phrase alignment models outperform classical word-level models in both generative and discriminative settings. This result is fundamental to the field: the models proposed in this thesis address a general, language-independent alignment problem that arises in all state-of-the-art statistical machine translation systems in use today.

Chapter 2

Phrase-Based Statistical Machine Translation

Phrase-based statistical MT has become a dominant approach to fast, large-scale, open-domain machine translation. This chapter describes the key technical details of such systems and concludes with an overview of the baseline approach to discovering phrase pairs. We compare to this baseline throughout this thesis.

2.1 Notation for Indexing Sentence Pairs

We consider the task of translating an input sentence $\mathbf{f}$ into an output sentence $\mathbf{e}$.[1] Sentences will be treated as sequences of word tokens, each of which is drawn from the vocabulary of word types $\mathcal{E}$ and $\mathcal{F}$. Within a sentence, e_i and f_j denote the words in position i of $\mathbf{e}$ and in position j of $\mathbf{f}$, respectively, where $\mathbf{e}$ and $\mathbf{f}$ are 0-indexed.

Phrase-based translation operates on contiguous subsequences of sentences, which

[1]The variables $\mathbf{f}$ and $\mathbf{e}$ are standard for historical reasons, with $\mathbf{f}$ denoting a French sentence and $\mathbf{e}$ denoting an English sentence. All examples and experiments in this thesis translate some non-English language into English, in order to remain consistent with this notation and to ensure that examples are intelligible to English-speaking readers.

we call *phrases*. We use fencepost indexing to denote phrases: the span $e_{[g:h)}$ with $g < h$ refers to the phrase of $\mathbf{e}$ beginning at position g and ending at position $h - 1$. Likewise, $[k : \ell)$ denotes a span of $\mathbf{f}$.

2.2 Training Pipeline for Phrase-Based MT

Phrase-based translation systems are typically batch-trained offline using some fixed set of data. Once a model of translation is learned, it is applied to new sentences, but the model is not adapted or trained online. Training the model involves a pipeline of steps that combine statistical learning techniques, heuristics, and large-scale data processing.

2.2.1 Phrasal Model Representation

The phrase-based model we wish to learn via this pipeline scores the space of well-formed, phrase-segmented, and phrase-aligned sentence pairs, often called *derivations*. A derivation $d = (\mathbf{e}, \mathbf{f}, \mathcal{P})$ consists of the input word sequence $\mathbf{f}$, the output sequence $\mathbf{e}$, and a set $\mathcal{P}$ of aligned spans, e.g. $[g : h) \Leftrightarrow [k : \ell)$.

In well-formed derivations, $\mathcal{P}$ defines a phrasal partition of each sentence, along with a bijective (one-to-one and onto) mapping of the phrases in $\mathbf{f}$ to the phrases in $\mathbf{e}$. Formally, let the expression $\sqcup_{s \in \mathcal{S}} s = \mathcal{T}$ denote that the set of spans S is a partition of the set $\mathcal{T}$: all s are *pairwise disjoint*, but the union over all $s \in \mathcal{S}$ equals T. In a well-formed derivation $(\mathbf{e}, \mathbf{f}, \mathcal{P})$, the following two properties must hold:

$$\sqcup_{[g:h) \Leftrightarrow [k:\ell) \in \mathcal{P}} \; [g : h) = [0 : |\mathbf{e}|)$$

$$\sqcup_{[g:h) \Leftrightarrow [k:\ell) \in \mathcal{P}} \; [k : \ell) = [0 : |\mathbf{f}|) \; .$$

The model itself is a linear model that conditions on the input $\mathbf{f}$, and factors over features ϕ_T on the aligned spans of $\mathcal{P}$, as well as features ϕ_L on k-length spans of the output $\mathbf{e}$, where k is the order of a *language model*.

$$s(\mathbf{e}, \mathbf{f}, \mathcal{P}) = \theta \cdot \left[\sum_{[g:h) \Leftrightarrow [k:\ell) \in \mathcal{P}} \phi_T(e_{[g:h)}, f_{[k:\ell)}) + \sum_{i=0}^{|\mathbf{e}-k|} \phi_L(e_{[i:i+k)}) \right] . \tag{2.1}$$

Conceptually, the features ϕ_T are typically functions of the *phrase pair type*, and they collectively ensure that each phrase is adequately translated. The features ϕ_L promote fluency and proper length of the output. Of course, this linear form is quite general, and a host of additional effective features have been proposed that blur the distinction between adequacy-focused "translation model" features and fluency-focused "language model" features (Carpuat and Wu, 2007; Gimpel and Smith, 2008; Blackwood *et al.*, 2008). Despite these advances, most systems use a low dimensional feature space with between 6 and 30 features. Each feature typically requires substantial data processing to compute, in the form of aggregating statistics over a large corpus of text. The canonical features that drive translation performance are described in detail below, along with a brief survey of techniques to learn θ.

2.2.2 Training Data

Training begins with a sentence-aligned parallel corpus that is divided into a large *training set* (often millions of sentences) in which phrase pairs are discovered, and a small *tuning set* (typically on the order of 2,000 sentences) used to learn θ. Parallel corpora are collected as collections of documents. These documents are divided into sentences using a sentence splitter, e.g. Gillick (2009), and those sentences are paired using a sentence aligner.

In addition to the parallel corpus, a large monolingual corpus in the output lan-

guage is used to train a language model. We will not summarize the rich literature on language modeling, but we do note that language models play a central and vital role in modern statistical translation systems. Language models are likelihood-trained Markov models of word sequences that are estimated from corpus n-gram counts and smoothed, for example using Kneser-Ney smoothing (Kneser and Ney, 1995) or a computationally convenient alternative, stupid back-off (Brants *et al.*, 2007).

2.2.3 Phrase Pair Scoring

From the sentence pairs in the training set, reusable phrase pairs are identified, counted, and scored. This stage of the pipeline has two complementary goals:

Phrase Pair Selection Among all pairs of input and output phrases that co-occur in the same sentence pair, what subset should we consider to be reusable phrase pairs $(\bar{e}, \bar{f})$.

Relative Frequencies Given an input-language phrase $\bar{f}$, how often do we observe $\bar{e}$ as its translation? This relative frequency statistic is denoted $\mathrm{P}_{\mathrm{rel}}(\bar{e}|\bar{f})$. We also compute the complementary statistic $\mathrm{P}_{\mathrm{rel}}(\bar{f}|\bar{e})$ over possible translations for each $\bar{e}$.

In addition to relative frequency statistics, systems often collect additional statistics based on the lexical items contained within phrases (Och and Ney, 2004). In aggregate, these features constitute ϕ_T in Equation 2.1

The specific models and learning techniques that select phrase pairs and estimate their relative frequencies are the primary focus of this dissertation. Section 2.3 describes the baseline approach that is in common use today by state-of-the-art machine translation systems.

2.2.4 Tuning Translation Models

The full model used to select translations of novel inputs is the linear model defined in Equation 2.1, which is parameterized by a weight vector θ. The most common learning objective in machine translation is direct loss minimization on a tuning set T of sentence pairs $(\mathbf{e}, \mathbf{f})$:

$$\theta^* = \arg\min_{\theta} L(\left\{ \arg\max_{(\mathbf{e},\mathbf{f},\mathcal{P}) \in \mathcal{D}(\mathbf{f})} s(\mathbf{e}, \mathbf{f}, \mathcal{P}) : \mathbf{f} \in T \right\}, \{\mathbf{e}^* : (\mathbf{e}^*, \mathbf{f}) \in T\}) \ . \tag{2.2}$$

Above, $\mathcal{D}(\mathbf{f})$ is the set of all derivations $(\mathbf{e}, \mathbf{f}, \mathcal{P})$ that match $\mathbf{f}$. This non-continuous, non-differentiable, and often volatile objective function is typically optimized using a coordinate descent procedure called minimum error rate training (Och, 2003).

2.2.5 Selecting Translations

Most statistical MT systems select an output $\mathbf{e}$ for novel input $\mathbf{f}$ by choosing the highest scoring derivation $(\mathbf{e}, \mathbf{f}, \mathcal{P})$ that matches $\mathbf{f}$.

$$\arg\max_{(\mathbf{e},\mathbf{f},\mathcal{P}) \in \mathcal{D}(\mathbf{f})} s(\mathbf{e}, \mathbf{f}, \mathcal{P}) \ .$$

This quantity also appears in Equation 2.2. Finding the highest scoring derivation requires solving a search problem in the space of derivations.

Efficient search over the space of derivations can be performed via beam search and dynamic programming (Koehn *et al.*, 2003). We note that many other decision objectives have been proposed that combine multiple derivations (Blunsom and Osborne, 2008), multiple output candidates (Kumar and Byrne, 2004; Li *et al.*, 2009; DeNero *et al.*, 2009), and even multiple systems (Rosti *et al.*, 2007; DeNero *et al.*, 2010). These extensions are important to improving the performance of statistical

MT systems, but they are beyond the scope of this thesis.

2.3 Baseline Phrase Pair Extraction and Scoring

This section presents the standard approach to selecting and scoring phrase pairs in a state-of-the-art phrase-based machine translation system, elaborating on the summary described in Section 2.2.3. This stage of a pipeline is itself a multi-stage learning procedure, which includes word alignment, phrase pair extraction, and relative frequency estimation.

2.3.1 Word Alignment

The pioneering work in statistical MT posited that sentence-to-sentence translation could be modeled as a word-level stochastic process, wherein each word of $\mathbf{f}$ is generated independently by selecting some word e of $\mathbf{e}$ and then drawing a word f from a learned *bilexical* distribution $\mathrm{P}(F = f|E = e)$, which is a conditional multinomial over word types in $\mathcal{F}$. Variants on these original models are still used today as the primary means of inferring an alignment between the words of $\mathbf{f}$ and $\mathbf{e}$ in a sentence pair.

The most widely used word alignment models assume that each word in $\mathbf{f}$ aligns to (and is therefore generated from) exactly one word in $\mathbf{e}$, but place no restriction on how many different $f \in \mathbf{f}$ can align to some $e \in \mathbf{e}$. Hence, the structure of an analysis of a sentence pair is a many-to-one alignment from elements of $\mathbf{f}$ to elements of $\mathbf{e}$.

To capture these structural assumptions, alignment models posit a vector of integer-valued alignments $\mathbf{a}$, where $a_j = i$ indicates that word j of $\mathbf{f}$ aligns to word i of $\mathbf{e}$. During training, $\mathbf{a}$ is a latent variable in a model that has the following conditional

parametric form, where the vector-valued random variables $\mathbf{F}$ and $\mathbf{E}$ take vectors of word type values from $\mathcal{F}$ and $\mathcal{E}$, and $\mathbf{A}$ is a random variable over vectors of integers.

$$\mathrm{P}(\mathbf{F} = \mathbf{f}, \mathbf{A} = \mathbf{a} | \mathbf{E} = \mathbf{e}) = \prod_{j=0}^{|\mathbf{f}|-1} \mathrm{P}_d(A_j = i | A_0 \dots A_{j-1}) \cdot \mathrm{P}_b(F_j = f_j | E_i = e_i) \ .$$

Above, P_d is a Markov distribution over alignment positions, referred to as a *distortion model*, and P_b is the bilexical conditional multinomial over word types. Specific alignment models vary in the parametric form and Markov assumptions of P_d. They may also contain an additional term to parameterize the number of alignments to a particular position i in $\mathbf{e}$, referred to as fertility model. However, the prediction of word alignment models are primarily driven by P_b. The parameters of P_d and P_b are set to maximize the likelihood of the observed $\mathbf{f}$ for each $\mathbf{e}$, summing out over the hidden $\mathbf{a}$; this optimization can be achieved with the expectation-maximization algorithm (Brown *et al.*, 1993).

Generative alignment models of this form can be queried for their predictions about the set $\mathcal{A}$ of word-to-word alignment links that describe the lexical correspondence between the two translations in a sentence pair. $\mathcal{A}$ is typically taken to be either the Viterbi alignment,

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} \mathrm{P}(\mathbf{F} = \mathbf{f}, \mathbf{A} = \mathbf{a} | \mathbf{E} = \mathbf{e})$$

$$\mathcal{A} = \left\{ (i, j) : \mathbf{a}^*_j = i \right\} \ ,$$

or a collection of alignment links (i, j) whose posterior exceeds a fixed threshold t,

$$\mathrm{P}(\mathbf{A} = \mathbf{a}|\mathbf{F} = \mathbf{f}, \mathbf{E} = \mathbf{e}) \propto \mathrm{P}(\mathbf{F} = \mathbf{f}, \mathbf{A} = \mathbf{a}|\mathbf{E} = \mathbf{e})$$

$$\mathrm{P}(A_j = i|\mathbf{F} = \mathbf{f}, \mathbf{E} = \mathbf{e}) = \sum_{\mathbf{a}:a_j=i} \mathrm{P}(\mathbf{A} = \mathbf{a}|\mathbf{F} = \mathbf{f}, \mathbf{E} = \mathbf{e})$$

$$\mathcal{A} = \{(i, j) : t \leq \mathrm{P}(A_j = i|\mathbf{F} = \mathbf{f}, \mathbf{E} = \mathbf{e})\} \ .$$

Note that in the latter case, $\mathcal{A}$ may contain links from a position j to multiple i if $t \leq 0.5$, violating the one-to-many structure assumed by the model.

Typically, two word alignment models are trained on the same parallel corpus. One generates $\mathbf{f}$ conditioned on $\mathbf{e}$, and the other generates $\mathbf{e}$ conditioned on $\mathbf{f}$. Hence, we can generate two predicted alignments $\mathcal{A}_{\mathbf{f}}$ and $\mathcal{A}_{\mathbf{e}}$, respectively. A variety of heuristics exist that merge these two alignment vectors into a single set of links (Och *et al.*, 1999; DeNero and Klein, 2007). These heuristics attempt to filter out erroneous alignment links and produce an alignment structure that is more amenable to the phrase extraction procedure that follows.

2.3.2 Phrase Pair Extraction

From word-aligned sentence pairs, we can enumerate and count all of the phrase pairs that are *consistent* with the word alignment. These phrase pairs serve as the building blocks of new translations, and their counts derive the most important statistical features for scoring translations.

Phrase pair extraction defines a mapping from $\mathcal{A} = \{(i, j)\}$ to an *extraction set* of bispans $R_n(\mathcal{A}) = \{[g : h) \Leftrightarrow [k : \ell)\}$, where each bispan links target span $e_{[g:h)}$ to

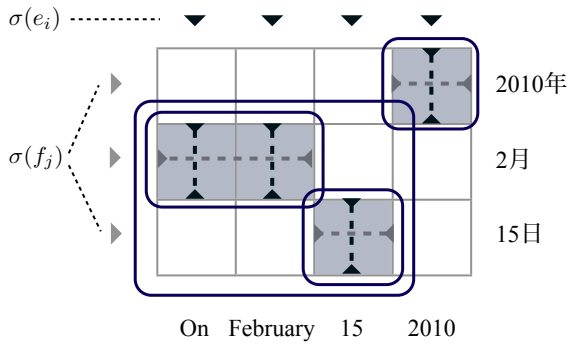


Figure 2.1: A word alignment $\mathcal{A}$ (shaded grid cells) defines projections $\sigma(e_i)$ and $\sigma(f_j)$, shown as dotted lines for each word in each sentence. The extraction set $R_3(\mathcal{A})$ includes all bispans licensed by these projections, shown as rounded rectangles.

source span $f_{[k:\ell)}$. The maximum phrase length parameter n dictates that

$$\forall [g:h) \Leftrightarrow [k:\ell) \in R_n(\mathcal{A}) : \max(h-g, \ell-k) \leq n \ .$$

We can describe this mapping via word-to-phrase projections, as illustrated in Figure 2.1. Let word e_i project to the phrasal span $\sigma(e_i)$, where

$$\sigma(e_i) = \left[\min_{j \in J_i} j \ , \ \max_{j \in J_i} j + 1 \right) \tag{2.3}$$

$$J_i = \{j : (i,j) \in \mathcal{A}\} \ .$$

and likewise each word f_j projects to a span of $\mathbf{e}$. Then, $[g:h) \Leftrightarrow [k:\ell) \in R_n(\mathcal{A})$ if and only if

$$\sigma(e_i) \subseteq [k:\ell) \qquad \forall i \in [g:h) \tag{2.4}$$

$$\sigma(f_j) \subseteq [g:h) \qquad \forall j \in [k:\ell)$$

That is, every word in one of the phrasal spans must project within the other. This mapping is deterministic, and so we can interpret a word-level alignment $\mathcal{A}$ as also specifying the phrasal rules that should be extracted from a sentence pair.

Among the bispans extracted from a word alignment are *minimal* bispans, which do not contain extracted bispans within them, and *composed* bispans that contain two or more bispans within their bounds. While the phrase pairs corresponding to minimal bispans are sufficient to describe the lexical correspondence between the two sentences of a sentence pair, composed phrase pairs capture additional context and therefore can improve output quality at translation time.

Words that do not appear in any of the links of $\mathcal{A}$ are called *null-aligned* words, and are often treated as special cases. A variety of approaches to handling null-aligned words have been suggested (Och and Ney, 2004; Ayan and Dorr, 2006). According to the definition above, for null-aligned i, $\sigma(e_i) = \emptyset$, and so null-aligned words are included in phrase pairs. A common restriction is to dictate that $R_n(\mathcal{A})$ include a bispan $[g : h) \Leftrightarrow [k : \ell)$ only if none of the projections of the boundary words of the bispan — $\sigma(e_g)$, $\sigma(e_{h-1})$, $\sigma(f_k)$, or $\sigma(f_{\ell-1})$ — are empty.

2.3.3 Relative Frequency Features

All phrase pair types $(\bar{e}, \bar{f})$ are then scored by a pair of relative frequency features, which express how often a given translation has been observed for a given phrase, relative to other observe translations of that phrase:

$$\mathrm{P}_{\mathrm{rel}}(\bar{e}|\bar{f}) = \frac{\mathrm{count}(\bar{e}, \bar{f})}{\sum_{\bar{e}'} \mathrm{count}(\bar{e}', \bar{f})} \quad ; \quad \mathrm{P}_{\mathrm{rel}}(\bar{f}|\bar{e}) = \frac{\mathrm{count}(\bar{e}, \bar{f})}{\sum_{\bar{f}'} \mathrm{count}(\bar{e}, \bar{f}')} \tag{2.5}$$

Above, $\mathrm{count}(\cdot)$ denotes how many times a phrase pair corresponded to some bispan $[g : h) \Leftrightarrow [k : \ell) \in R_n(\mathcal{A})$ for a word-aligned sentence pair $(\mathbf{e}, \mathbf{f}, \mathcal{A})$ in the parallel training corpus.

These conditional distributions effectively distinguish between frequent and infrequent phrasal patterns in the parallel corpus. However, we note that the pipelined approach by which these probabilities have been estimated does not correspond to a probabilistic generative model of the data. In particular, these phrase pair counts do not capture the uncertainty of the underlying word alignment models. Moreover, they are never re-estimated, despite the underlying uncertainty about how phrases correspond to phrases in a sentence pair. Hence, this pipeline for learning phrase pairs does not use the frequency information computed via Equation 2.5 to propose better alignments of the training corpus. Nonetheless, this baseline approach to phrase pair extraction and scoring has proven effective and robust in a large number of different systems and language pairs.

Chapter 3

Phrase Alignment Models

As described in Chapter 2, phrase-based systems capture statistical knowledge of translation equivalence by collecting phrase pairs and their frequency statistics from parallel corpora. These statistics are typically gathered by fixing a word alignment and applying a deterministic phrase extraction procedure to word-aligned sentence pairs. Alternatively, phrase pairs can be identified and counted using a statistical *phrase alignment model*.

Phrase alignment models are statistical models that score proposed alignments according to the phrase pairs they contain, rather than by their word-to-word links. That is, the score of a set of alignment links $\mathcal{A}$ factors over the phrases implied by $\mathcal{A}$, rather than word links within it.

Phrase alignment models have several advantages over the standard word-alignment-based pipeline presented in Section 2.3:

1. Phrase alignment models induce distributions over phrase pairs that have interpretable probabilistic semantics. That is, they express uncertainty over the phrase pairs that should be counted in a sentence pair.

2. Training these models allows us to propagate frequency information about

multi-word phrasal patterns among different sentences in a corpus, so that repeated instances of idioms reinforce a correct analysis. Word alignment models assume a word-to-word correspondence and independent generation of each word. As a result, they often fail to identify and properly analyze multi-word patterns in the data.

3. The independence assumptions of phrase models match those of phrase-based translation models. Thus, phrase alignment models unify data representation across the training and application stages of a machine translation pipeline.

This chapter investigates the computational and statistical properties of the phrase alignment model class. In particular, we focus on *phrase-factored* alignment models, which have scoring functions that factor over disjoint phrases. In addition to general complexity results, we present two specific probabilistic models within this model class.

3.1 The Phrase-Factored Model Class

Phrase-factored alignment models share the same derivation structure as phrase-based translation systems, described in Section 2.2.1. They assign a score or probability to all possible tuples $(\mathbf{e}, \mathbf{f}, \mathcal{P})$, where $\mathbf{e}$ and $\mathbf{f}$ are word sequences, and $\mathcal{P}$ is a set of bispans that together describe phrasal partitions $\mathbf{e}$ and $\mathbf{f}$, along with a bijective mapping from the phrases in $\mathbf{e}$ to the phrases in $\mathbf{f}$:[1]

$$\sqcup_{[g:h)\Leftrightarrow[k:\ell)\in\mathcal{P}} \ [g:h) = [0:|\mathbf{e}|)$$

$$\sqcup_{[g:h)\Leftrightarrow[k:\ell)\in\mathcal{P}} \ [k:\ell) = [0:|\mathbf{f}|) \ .$$

[1] $\sqcup_{s\in\mathcal{S}} \ s = \mathcal{T}$ denote that the set of spans S is a partition of the sequence $\mathcal{T}$: all s are *pairwise disjoint*, and $\bigcup_s s = T$.

The model score is a sum over potentials ϕ on bispans, where ϕ may take sentence-specific values that condition on the word sequences $\mathbf{e}$ and $\mathbf{f}$:

$$s(\mathbf{e}, \mathbf{f}, \mathcal{P}) = \sum_{[g:h) \Leftrightarrow [k:\ell)} \phi([g:h) \Leftrightarrow [k:\ell)) \ . \tag{3.1}$$

Equation 3.1 defines the factorization of scores for the phrase-factored alignment model class. The precise definition of $\phi(\cdot)$ varies with the particulars of the model.

Because the model score factors over phrase pairs in Equation 3.1, we can define the score of all possible derivations $(\mathbf{e}, \mathbf{f}, \mathcal{P})$ for a sentence pair simply by assigning a potential to each possible bispan within it. Let a *weighted sentence pair* $(\mathbf{e}, \mathbf{f}, \phi)$ include a real-valued potential function $\phi : \{[g:h) \Leftrightarrow [k:\ell)\} \rightarrow \mathbb{R}$, which scores bispans. For the purpose of analyzing inference complexity of this model class, we impose no additional restrictions on ϕ.

Phrase-factored alignment models can have a probabilistic interpretation. Any model can be used to induce a distribution over derivations,

$$\Pr(\mathbf{e}, \mathbf{f}, \mathcal{P}) \propto \prod_{[g:h) \Leftrightarrow [k:\ell)} \exp(\phi([g:h) \Leftrightarrow [k:\ell))) \ ,$$

which can be normalized to produce a distribution over well-formed alignments,

$$\Pr(\mathcal{P}|\mathbf{e}, \mathbf{f}) = \frac{\Pr(\mathbf{e}, \mathbf{f}, \mathcal{P})}{\sum_{\mathcal{P}'} \Pr(\mathbf{e}, \mathbf{f}, \mathcal{P}')} \ ,$$

which can in-turn be marginalized to produce probabilities of individual bispans,

$$\Pr([g:h) \Leftrightarrow [k:\ell)|\mathbf{e}, \mathbf{f}) = \sum_{\mathcal{P}:[g:h) \Leftrightarrow [k:\ell) \in \mathcal{P}} \Pr(\mathcal{P}|\mathbf{e}, \mathbf{f}) \ .$$

Furthermore, generative probabilistic models generally consider each bispan to be

drawn from some distribution, and therefore,

$$0 \leq \exp(\phi([g:h) \Leftrightarrow [k:\ell))) \leq 1 \ .$$

3.2 Inference in Phrase-Factored Models

Model *inference* refers to the set of computations that systematically aggregate information across the entire space of derivations for a sentence pair. In the case of phrase-factored alignment models, inference procedures allow us to predict the highest scoring $\mathcal{P}$ for a sentence pair $(\mathbf{e}, \mathbf{f})$, as well as to predict the posterior probability of a particular $[g:h) \Leftrightarrow [k:\ell)$ appearing in $\mathcal{P}$. Computing these phrase posteriors serves as the basis for the probabilistic learning techniques that we explore in Section 3.6.

Exact inference in this model class is challenging. While the number of bispans is quartic in the length of the sentence pair, the number of phrase alignments is exponential. In this section, we show that search for the highest scoring derivation is NP-hard, while computing bispan posteriors is #P-hard. We first define the exact inference problems of interest and then analyze their complexity classes. These complexity results were originally published by DeNero and Klein (2008).

For some restricted combinatorial spaces of alignments—those that arise in ITG-based phrase models (Cherry and Lin, 2007) or local distortion models (Zens *et al.*, 2004)—inference can be accomplished using polynomial time dynamic programs. We analyze these cases in Section 3.3.

3.2.1 Inference Problem Definitions

We consider four related inference problems for weighted sentence pairs. Let $\mathbb{P}$ be the set of all phrase alignments $\mathcal{P}$ for a sentence pair $(\mathbf{e}, \mathbf{f})$.

Optimization, $\mathcal{O}$: Given $(\mathbf{e}, \mathbf{f}, \phi)$, find $\max_{\mathcal{P} \in \mathbb{P}} s(\mathbf{e}, \mathbf{f}, \mathcal{P})$.

Decision, $\mathcal{D}$: Given $(\mathbf{e}, \mathbf{f}, \phi)$, decide if $\exists \mathcal{P} \in \mathbb{P} : s(\mathbf{e}, \mathbf{f}, \mathcal{P}) \geq 1$.

$\mathcal{O}$ arises in predicting the highest scroing $\mathcal{P}$ from a model. This inference problem arises in the Viterbi approximation to EM that assumes probability mass is concentrated at the mode of the posterior distribution over alignments. It also arises in prediction and discriminative training. $\mathcal{D}$ is the corresponding decision problem for $\mathcal{O}$, useful in analysis.

In addition to maximizing over $\mathbb{P}$, we can also define summing inference problems. Let $\mathbb{P}(g, h, k, \ell) = \{\mathcal{P} : [g : h) \Leftrightarrow [k : \ell) \in \mathcal{P}\}$.

Expectation, $\mathcal{E}$: Given $(\mathbf{e}, \mathbf{f}, \phi)$ and g, h, k, ℓ, compute $\sum_{\mathcal{P} \in \mathbb{P}(g,h,k,\ell)} s(\mathbf{e}, \mathbf{f}, \mathcal{P})$.

Sum, $\mathcal{S}$: Given $(\mathbf{e}, \mathbf{f}, \phi)$, compute $\sum_{\mathcal{P} \in \mathbb{P}} s(\mathbf{e}, \mathbf{f}, \mathcal{P})$.

$\mathcal{E}$ arises in computing sufficient statistics for re-estimating phrase translation probabilities (E-step) when training generative models. A polynomial time algorithm for $\mathcal{E}$ implies a polynomial time algorithm for $\mathcal{S}$, because we can compute the total sum by adding together the sums for disjoint subsets. One such decomposition of $\mathbb{P}$ is

$$\mathbb{P} = \bigcup_{j=1}^{|\mathbf{e}|} \bigcup_{k=0}^{|\mathbf{f}|-1} \bigcup_{l=k+1}^{|\mathbf{f}|} \mathbb{P}(0, h, k, \ell) \ .$$

3.2.2 Complexity of $\mathcal{O}$ and $\mathcal{D}$

For the space $\mathbb{P}$ of well-formed phrase alignments, problems $\mathcal{E}$ and $\mathcal{O}$ have long been suspected of being NP-hard, first asserted but not proven in Marcu and Wong (2002). We give a novel proof that $\mathcal{O}$ is NP-hard, showing that $\mathcal{D}$ is NP-complete by reduction from 3-SAT, the boolean satisfiability problem for sets of clauses with up to three literals. This result holds despite the fact that the related problem of finding

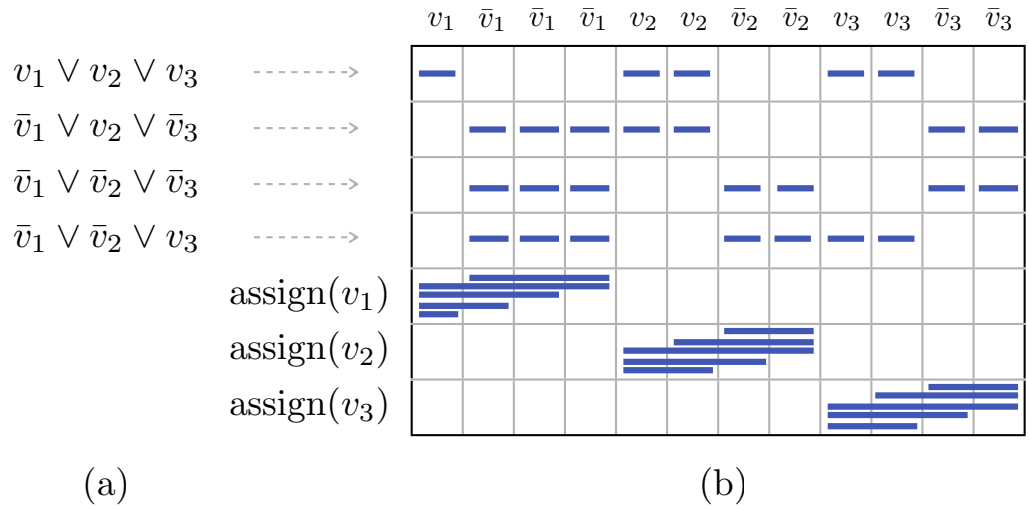


Figure 3.1: (a) The clauses of an example 3-SAT instance with $\mathbf{v} = (v_1, v_2, v_3)$. (b) The weighted sentence pair $\text{wsp}(\mathbf{v}, \mathbf{C})$ constructed from the 3-SAT instance. All links that have $\phi = 1$ are marked with a blue horizontal stripe. Stripes in the last three rows demarcate the alignment options for each assign(v_n), which consume all words for some literal.

an optimal matching in a weighted bipartite graph (the ASSIGNMENT problem) is polynomial-time solvable using the Hungarian algorithm.

A reduction proof of NP-completeness gives a construction by which a known NP-complete problem can be solved via a newly proposed problem. From a 3-SAT instance, we construct a weighted sentence pair for which alignments with positive score correspond exactly to the SAT solutions. Since 3-SAT is NP-complete and our construction requires only polynomial time, we conclude that $\mathcal{D}$ is NP-complete. D is certainly in NP: given an alignment $\mathcal{P}$, it is easy to determine whether or not $\phi(\mathcal{P}) \geq 1$.

3-SAT: Given vectors of boolean variables $\mathbf{v} = (v_1, \ldots, v_n)$ and propositional clauses[2] $\mathbf{C} = (C_1, \ldots, C_m)$, decide whether there exists an assignment to $\mathbf{v}$ that simul-

[2]A clause is a disjunction of literals. A literal is a bare variable v_k or its negation $\bar{v}_k$. For instance, $v_2 \vee \bar{v}_7 \vee \bar{v}_9$ is a clause.

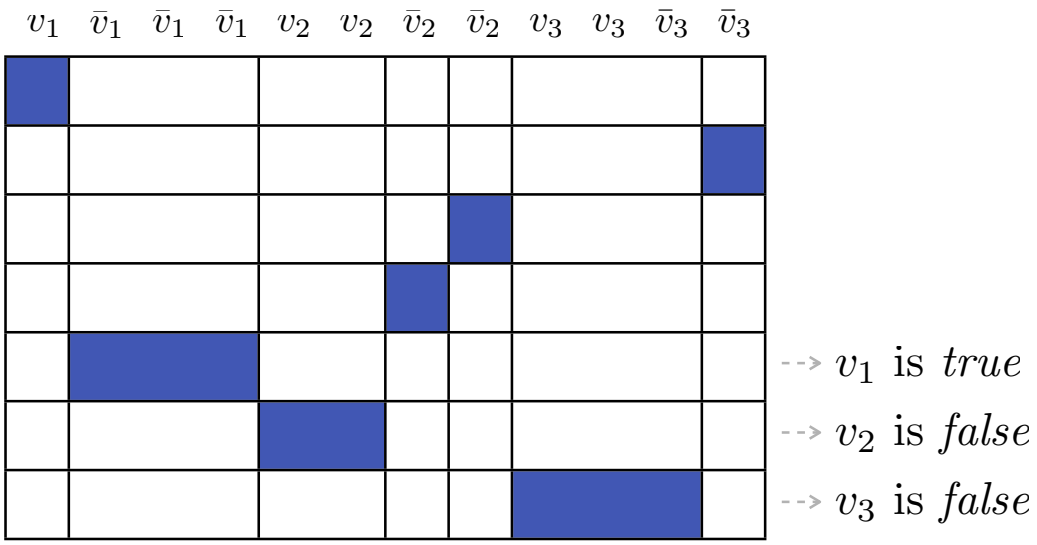


Figure 3.2: A particular bijective alignment that has score 1 under our SAT construction can be interpreted as predicting a satisfying assignment for the original SAT instance.

taneously satisfies each clause in $\mathbf{C}$.

For a 3-SAT instance $(\mathbf{v}, \mathbf{C})$, we construct $\mathbf{f}$ to contain one word for each clause, and $\mathbf{e}$ to contain several copies of the literals that appear in those clauses. ϕ scores only alignments from clauses to literals that satisfy the clauses. The crux of the construction lies in ensuring that no variable is assigned both *true* and *false*. The details of constructing such a weighted sentence pair $\text{wsp}(\mathbf{v}, \mathbf{C}) = (\mathbf{e}, \mathbf{f}, \phi)$, described below, are also depicted in Figure 3.1. An example solution and interpretation of this construction appears in Figure 3.2.

1. $\mathbf{f}$ contains a word for each C, followed by an assignment word for each variable, $\text{assign}(v)$.

2. $\mathbf{e}$ contains $c(t)$ consecutive words for each literal t, where $c(t)$ is the number of times that t appears in any clause.

Then, we set $\phi(\cdot, \cdot) = 0$ everywhere except:

3. For all clauses C and each satisfying literal t, and each one-word phrase e in $\mathbf{e}$ containing t, $\phi(e, f_C) = 1$. f_C is the one-word phrase containing C in $\mathbf{f}$.

4. The assign(v) words in $\mathbf{f}$ align to longer phrases of literals and serve to consistently assign each variable by using up inconsistent literals. They also align to unused literals to yield a bijection. Let $\mathbf{e}_{(t,n)}$ be the phrase in $\mathbf{e}$ containing all literals t and n negations of t. By construction, $\mathbf{e}_{(t,n)}$ is unique. Also, let $f_{\text{assign}(v)}$ be the one-word phrase for assign(v). Then, $\phi(e_{[k:t)}, f_{\text{assign}(v)}) = 1$ for $t \in \{v, \bar{v}\}$ and all applicable n.

Claim 1. *If wsp*$(\mathbf{v}, \mathbf{C})$ *has an alignment* $\mathcal{P}$ *with* $s(\mathcal{P}) \geq 1$*, then* $(\mathbf{v}, \mathbf{C})$ *is satisfiable.*

Proof. The score implies that $\mathbf{f}$ aligns using all one-word phrases and $\forall a_i \in \mathcal{P}, \phi(a_i) = 1$. By condition 4, each $f_{\text{assign}(v)}$ aligns to all $\bar{v}$ or all v in $\mathbf{e}$. Then, assign each v to *true* if $f_{\text{assign}(v)}$ aligns to all $\bar{v}$, and *false* otherwise. By condition 3, each C must align to a satisfying literal, while condition 4 assures that all available literals are consistent with this assignment to $\mathbf{v}$, which therefore satisfies $\mathbf{C}$. □

Claim 2. *If* $(\mathbf{v}, \mathbf{C})$ *is satisfiable, then wsp*$(\mathbf{v}, \mathbf{C})$ *has an alignment* $\mathcal{P}$ *with* $s(\mathcal{P}) = 1$*.*

Proof. We construct such an alignment $\mathcal{P}$ from the satisfying assignment $\mathbf{v}$. For each C, we choose a satisfying literal t consistent with the assignment. Align f_C to the first available t token in $\mathbf{e}$ if the corresponding v is *true*, or the last if v is *false*. Align each $f_{\text{assign}(v)}$ to all remaining literals for v. □

Claims 1 and 2 together show that $\mathcal{D}$ is NP-complete. Since solving $\mathcal{O}$ would trivially provide the correct decision for $\mathcal{D}$, $\mathcal{O}$ is NP-hard.

3.2.3 Complexity of $\mathcal{E}$ and $\mathcal{S}$

With another construction, we can show that $\mathcal{S}$ is #P-hard, meaning that it is at least as hard as any #P-complete problem. #P is a class of counting problems related

to NP, and #P-hard problems are NP-hard as well.

Counting Perfect Matchings, CPM Given a bipartite graph G with $2n$ vertices, count the number of matchings of size n.

For a bipartite graph G with edge set $E = \{(v_j, v_l)\}$, we construct $\mathbf{e}$ and $\mathbf{f}$ with n words each, and set $\phi(e_{[j-1\ :j)}, f_{[l-1\ :l)}) = 1$ and 0 otherwise. The number of perfect matchings in G is the sum $\mathcal{S}$ for this weighted sentence pair. CPM is #P-complete (Valiant, 1979), so $\mathcal{S}$ and $\mathcal{E}$ are #P-hard.

3.3 Polynomial-Time Subclasses

While inference in unrestricted phrase alignment models is NP-hard, polynomial-time inference procedures do exist for some restricted classes of models. In particular, restricting the reordering of phrases yields polynomial-time dynamic programming solutions.

3.3.1 Monotonic Alignments

A monotonic phrase alignment has the property that the kth phrase of $\mathbf{e}$ aligns to the kth phrase of $\mathbf{f}$, for all k. Under this restriction, The highest scoring $\mathcal{P}$ can be computed using a polynomial-time left-to-right dynamic program (Problem $\mathcal{O}$). Let $m(i, j)$ denote the most probable derivation for the sentence pair prefix $(\mathbf{e}_{[0:i)}, \mathbf{f}_{[0:j)})$, with $m(0, 0) = 0$. Then,

$$m(i,j) = \max_{i':i'<i} \max_{j':j'<j} \{m(i',j') + \phi([i':i) \Leftrightarrow [j':j))\}$$

Let $n = \max(|\mathbf{e}|, |\mathbf{f}|)$ be the number of words in a sentence pair. Then, the dynamic program that corresponds to this recurrence has $\Theta(n^2)$ states, each of which

is computed via a maximum over $\Theta(n^2)$ terms, yielding a $\Theta(n^4)$ algorithm. Limiting the length of phrases to k reduces the order of growth to $\Theta(n^2 \cdot k^2)$. Replacing max with $\sum$ and appropriately redefining m gives an identically structured algorithm for summing over monotonic alignments.

Similar dynamic programs exist for *distortion-limited* phrase alignment models that allow only local reordering patterns. For example, a constraint that the phrase of $\mathbf{e}$ in position k aligns to some phrase of $\mathbf{f}$ with position in the range $[k-r, k+r]$ is polynomial in n, but exponential in r.

3.3.2 Inversion Transduction Grammars

Monotonic and distortion-limited alignments have *linear* ordering restrictions. We can also define a subclass of phrase alignment models that have polynomial-time inference procedures through *hierarchical* restriction on reordering. The most prevalent example of a hierarchical reordering restriction corresponds to an bracketing inversion transduction grammar (ITG), which is a binary synchronous grammar (Wu, 1997).

An ITG derivation T is a binary tree in which each node is labeled with a bispan of $(\mathbf{e}, \mathbf{f})$. The root of T is labeled with the bispan containing the whole sentence pair. T is ITG if for every parent node x labeled by bispan $[g:h) \Leftrightarrow [k:\ell)$ with children y and z, there is some i with $g < i < h$ and j with $k < j < \ell$ such that either

- y is labeled by $[g:i) \Leftrightarrow [k:j)$ and z is labeled by $[i:h) \Leftrightarrow [j:\ell)$, or
- y is labeled by $[i:h) \Leftrightarrow [k:j)$ and z is labeled by $[g:i) \Leftrightarrow [j:\ell)$.

We say that T is an ITG derivation of the phrasal alignment $\mathcal{P}$ if $\mathcal{P}$ is the set of bispans at the leaves of T. We say that a phrase alignment $\mathcal{P}$ is ITG if there exists some T that is a derivation of $\mathcal{P}$. Every ITG derivation T defines exactly one bijective phrase alignment $\mathcal{P}$, and the score for T is the score for its corresponding $\mathcal{P}$.

The class of ITG derivations also admits a polynomial-time Viterbi inference procedure. Let $m(g, h, k, \ell)$ denote the highest scoring ITG derivation of the bispan $[g : h) \Leftrightarrow [k : \ell)$. Then,

$$m(g,h,k,\ell) = \max \left\{ \begin{array}{l} \phi([g:h) \Leftrightarrow [k:\ell)) \\ \max_{i:g<i<h} \max_{j:k<j<\ell} \{m(g,i,k,j) + m(i,h,j,\ell)\} \\ \max_{i:g<i<h} \max_{j:k<j<\ell} \{m(i,h,k,j) + m(g,i,j,\ell)\} \end{array} \right\} .$$

This recurrence implies $\Theta(n^4)$ dynamic programming states, each of which requires a max over $\Theta(n^2)$ terms to compute, yielding a $\Theta(n^6)$ dynamic program. Unlike the case of linear reordering constraints, limiting the length of phrases does not affect the order of growth of this algorithm.

This order of growth applies only to binary derivations. As with distortion-limited linear distortions, inference procedures exist for r-ary synchronous grammars that are polynomial in n but exponential in r.

3.4 Inference Procedures for Phrase Alignment

Although $\mathcal{O}$ is NP-hard for the general case of phrase-factored alignment models, we present three algorithms to solve the problem. While in the worst case, these algorithms must either be approximate or require a running time that is exponential in sentence length, they have all been employed effectively in experimental settings.

3.4.1 Previous Work: Greedy Hill Climbing

Marcu and Wong (2002) developed an approximation to $\mathcal{O}$. Given a weighted sentence pair, high scoring phrases are linked together greedily to reach an initial alignment.

Then, local operators are applied to hill-climb $\mathbb{P}$ in search of the maximum $\mathcal{P}$. This procedure can also approximate $\mathcal{E}$ by collecting weighted bispan counts as the space is traversed. Birch *et al.* (2006) suggest using word alignments both to find better initialization points and to constrain the space of phrase alignments explored during hill climbing.

3.4.2 An Exponential-Time Dynamic Program

As with the polynomial-time subclasses of the phrase-factored alignment model class, we can solve $\mathcal{O}$ using dynamic programming. Let $\mathbf{j}$ be a subset of the word positions in $\mathbf{f}$. Furthermore, let $m(i, \mathbf{j})$ denote the maximal derivation covering the prefix $\mathbf{e}_{[0:i)}$ of $\mathbf{e}$ and the (possibly discontiguous) subsequence of $\mathbf{f}$ selected by $\mathbf{j}$. Then,

$$m(i, \mathbf{j}) = \max_{i' < i} \max_{[k:\ell):\mathbf{j}=\mathbf{j}' \cup [k:\ell)} m(i', \mathbf{j}') + \phi([i' : i) \Leftrightarrow [k : \ell)) \ .$$

While each value of m requires only $\Theta(n^3)$ to compute, corresponding to possible values of i' and $[k : \ell)$, the state space is exponential in the length of $\mathbf{f}$ because $\mathbf{j}$ is an arbitrary subset of $[0 : |\mathbf{f}|)$. In practice, the set of dynamic programming states can typically be controlled by disallowing any bispans that violate the word alignment predictions of a simpler model (DeNero *et al.*, 2006). However, our implementation of this dynamic program did have to be terminated before completion on certain sentences because of long running times.

3.4.3 An Integer Linear Programming Construction

We can also cast $\mathcal{O}$ as an integer linear programming (ILP) problem, for which many optimization techniques are known. This section gives an ILP construction for an arbitrary weighted sentence pair $(\mathbf{e}, \mathbf{f}, \phi)$ and evaluates ILP inference experimentally.

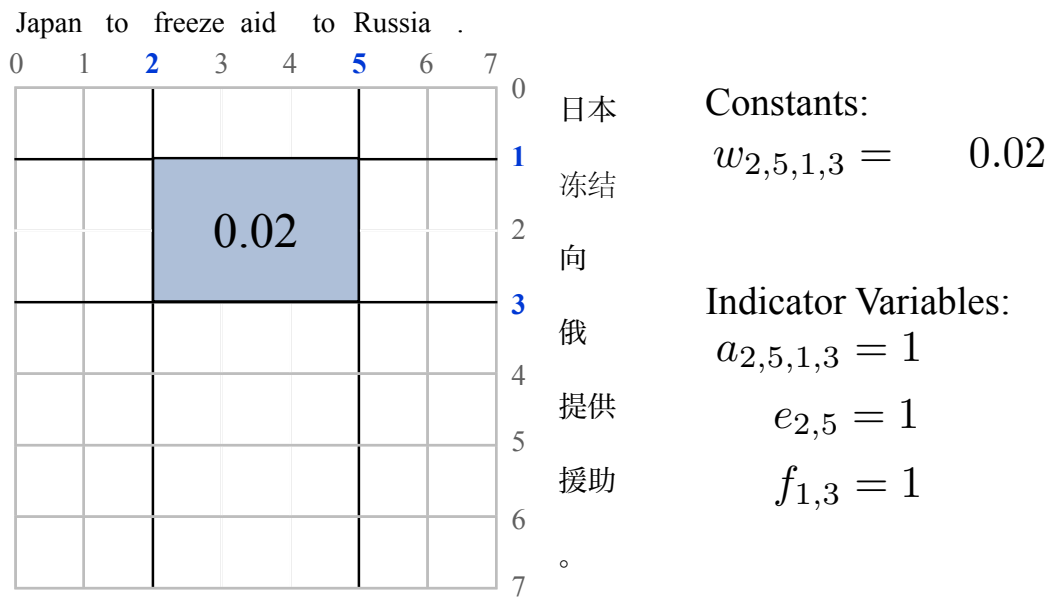


Figure 3.3: This example depicts the indicator variables and constants in the integer linear program construction of $\mathcal{O}$ that must be set in order to properly describe the inclusion of the single shaded bispan above.

We aim to construct an ILP with a solution that will yield the highest scoring phrase alignment $\mathcal{P}$ for a weighted sentence pair $(\mathbf{e}, \mathbf{f}, \phi)$. First, we introduce binary indicator variables $a_{g,h,k,\ell}$ denoting whether $[g:h) \Leftrightarrow [k:\ell) \in \mathcal{P}$. Next, we introduce binary indicators $e_{g,h}$ and $f_{k,\ell}$ that denote whether some $[g:h) \Leftrightarrow \cdot$ or $\cdot \Leftrightarrow [k:\ell)$ appear in $\mathcal{P}$, respectively. Finally, we represent the weight function ϕ as a weight vector in the program: $w_{g,h,k,\ell} = \phi[g:h) \Leftrightarrow [k:\ell)$. Figure 3.3 shows an example bispan weight, along with the relevant constants and indicator variables that encode the decision to include a bispan in $\mathcal{P}$.

Now, we can express a linear program that, when restricted to integer-valued solutions and optimized, will yield the optimal $\mathcal{P}$ of our weighted sentence pair.

$$\max \quad \sum_{g,h,k,\ell} w_{g,h,k,\ell} \cdot a_{g,h,k,\ell} \tag{3.2}$$

$$\text{s.t.} \quad \sum_{g,h:g<i\leq h} e_{g,h} = 1 \qquad \forall i : 1 \leq i \leq |e| \tag{3.3}$$

$$\sum_{k,\ell:k<j\leq \ell} f_{k,\ell} = 1 \qquad \forall j : 1 \leq j \leq |f| \tag{3.4}$$

$$e_{g,h} = \sum_{k,\ell} a_{g,h,k,\ell} \qquad \forall g, h \tag{3.5}$$

$$f_{k,\ell} = \sum_{g,h} a_{g,h,k,\ell} \qquad \forall k, \ell \tag{3.6}$$

with the following constraints on index variables:

$$g < h \ ; \ 0 \leq g < |\mathbf{e}| \ ; \ 0 < h \leq |\mathbf{e}| \ .$$

$$k < \ell \ ; \ 0 \leq k < |\mathbf{f}| \ ; \ 0 < l \leq |\mathbf{f}| \ .$$

The objective function (equation 3.2) is $s(\mathbf{e}, \mathbf{f}, \mathcal{P})$ for $\mathcal{P}$ implied by $\{a_{i,j,k,\ell} = 1\}$. Constraint equation 3.3 ensures that the English phrases form a partition of $\mathbf{e}$ – each word in $\mathbf{e}$ appears in exactly one phrase — as does equation 3.4 for $\mathbf{f}$. Constraint equation 3.5 ensures that each phrase in $\mathbf{e}$ appears in exactly one phrase-to-phrase alignment link, and that phrases not in $\mathbf{e}$ do not appear in the alignment (and likewise constraint 3.6 for $\mathbf{f}$).

Using an off-the-shelf ILP solver,[3] we were able to quickly and reliably find the globally optimal alignment. Model weights ϕ used in our timing experiments were

[3]We used Mosek: `www.mosek.com`.

Sentences per hour on a four-core server	20,000
Frequency of optimal solutions found	93.4%
Frequency of ϵ-optimal solutions found	99.2%

Table 3.1: An integer linear programming solver regularly finds a phrase alignment within $\epsilon = 10^{-5}$ of the optimal score for instances of weighted sentence pairs drawn from a typical machine translation corpus.

derived from relative frequency counts from a large corpus, as described in Section 2.3. Table 3.1 shows that ILP inference is accurate and efficient.

3.5 Generative Phrase-Factored Models

Now that we have developed a collection of inference algorithms for the phrase-factored alignment model class, we now define two generative models that we will apply to the task of phrase alignment.

Generative probabilistic modeling is an approach to statistical learning that follows from assuming that some stochastic process generated a collection of observed data instances. Learning involves selecting model parameters that effectively explain the observed data, thereby recovering the particular details of the assumed stochastic process. The two generative models described in this section correspond to stochastic processes that generate sentence pairs, phrase by phrase.

Probabilistic models assign probability mass to outcomes in their domain. In our case, the domain of interest is the discrete space of all sentence pairs.

3.5.1 Models with Latent Variables

To capture the patterns of the data effectively, generative models often include latent variables or latent structure: information that is *assumed* to exist but is not observed in the data. Clustering is a canonical example of latent variable modeling, in which

the data are assumed to have been generated conditioned upon some class or cluster.

In the case of alignment modeling, the correspondence structure between two sentences is latent. For word alignment, this correspondence is encoded as a vector $\mathbf{a}$ that encodes word-to-word alignment links, as described in Section 2.3.1. For generative phrase alignment models, the latent structure is $\mathcal{P}$, the set of bispans that form a partition of each sentence and a bijective alignment between the phrases in those partitions.

While latent variables are unobserved, they can be inferred from the model. In particular, alignment often involves querying the generative model for the posterior distribution over the latent alignment variable conditioned upon the observed input and output sentences.

3.5.2 Joint Generative Model

We first describe the symmetric joint model of Marcu and Wong (2002), which we extend in Chapter 4. A three-step generative process constructs a phrase-aligned sentence pair $(\mathbf{e}, \mathbf{f}, \mathcal{P})$.

1. Choose a number of component phrase pairs $n = |\mathcal{P}|$.

2. Draw n phrase pairs p_i independently from a multinomial distribution θ over phrase pair types, yielding a vector $\mathbf{p}$ of phrase pairs.

3. Choose a permutation π of $\mathbf{p}$ that defines the ordering in the input language. The ordering for the output language is defined by $\mathbf{p}$.[4]

In this process, all phrases in both sentences are aligned one-to-one via the generative process. A pair $(\mathbf{p}, \pi)$ describes a phrase-aligned sentence pair $(\mathbf{e}, \mathbf{f}, \mathcal{P})$, where

[4]We choose the input to reorder without loss of generality.

the set of bispans $\mathcal{P}$ contains all positional information from $\mathbf{p}$ and π, and the sentence pair $(\mathbf{e}, \mathbf{f})$ contains all lexical information. This change from our canonical representation is necessary because drawing phrase pairs from θ determines phrasal segmentation, phrase length, and lexical content simultaneously.

In this model, we parameterize the choice of n using a geometric distribution, denoted P_{G}, with stop parameter $p_{\$}$:

$$\mathrm{P}(n) = \mathrm{P}_{\mathrm{G}}(n; p_{\$}) = p_{\$} \cdot (1 - p_{\$})^{n-1} \quad .$$

The type of each aligned phrase pair is drawn from a multinomial distribution θ, which must be learned.

$$p_i \sim \theta$$

We fix a simple distortion model, using an exponential decay based on the start position of two phrases:

$$D([g:h) \Leftrightarrow [k:\ell) || \mathbf{e}|, |\mathbf{f}|) = b^{\left|g - k \cdot \frac{|\mathbf{e}|}{|\mathbf{f}|}\right|} \tag{3.7}$$

Above, $|\mathbf{e}|$ and $|\mathbf{f}|$ are the lengths of the sentences. Using this parametric form, we can define the probability of a permutation of the foreign phrases as proportional to the product of position-based distortion penalties for each phrase:

$$\mathrm{P}(\pi|\mathbf{p}) \propto \prod_{[g:h) \Leftrightarrow [k:\ell) \in \mathcal{P}} D([g:h) \Leftrightarrow [k:\ell) || \mathbf{e}|, |\mathbf{f}|) \quad .$$

This distortion model encourages the start positions g and k of each bispan to have similar positions in their respective sentences, after adjusting for differing total sentence lengths. This model component also factors over phrases, abiding by the

factorization restriction of the phrase-factored model class. This positional distortion model was deemed to work well relative to alternatives by Marcu and Wong (2002).

We can now state the joint probability for a phrase-aligned sentence consisting of n phrase pairs:

$$\mathrm{P}_\theta(\mathbf{e}, \mathbf{f}, \mathcal{P}) \propto \mathrm{P}_\mathrm{G}(|\mathcal{P}|; p_\$) \prod_{[g:h)\Leftrightarrow[k:\ell)\in\mathcal{P}} \theta(e_{[g:h)}, f_{[k:\ell)}) \cdot D([g:h) \Leftrightarrow [k:\ell) || \mathbf{e}|, |\mathbf{f}|) \ .$$

While this model has several free parameters in addition to θ, we fix them to reasonable values to focus learning on the phrase pair distribution θ.[5] The model can be properly normalized by summing over all possible derivations.

Sentence pairs do not always contain equal information on both sides, and so we revise the generative story to include unaligned phrases in both sentences. When generating each component of a sentence pair, we first decide whether to generate an aligned phrase pair or, with probability $p_\emptyset$, an unaligned phrase.[6]

To unify notation, we denote unaligned phrases as phrase pairs with one side equal to a *null symbol*, $\emptyset$. An unaligned output span is referred to by a bispan $[g:h) \Leftrightarrow \emptyset$ and a phrase pair type $(e_{[g:h)}, \emptyset)$. Similar notation can describe unaligned input spans.

3.5.3 Conditional Generative Model

The following conditional phrase-factored alignment model scores the same phrase alignment derivations, but in a different way. Rather than jointly generating both sentences by drawing from a multinomial over phrase pairs, this model generates only the output sentence $\mathbf{e}$ and the bispan set $\mathcal{P}$, conditioned on the input sentence $\mathbf{f}$.

The generative process that corresponds to this conditional model has two steps,

[5] Parameters were chosen by hand during development on a small training corpus. $p_\$ = 0.1$ and $b = 0.85$ in experiments.

[6] We strongly discouraged unaligned phrases in order to align as much of the corpus as possible: $p_\emptyset = 10^{-10}$ in experiments.

assuming an observed input sentence $\mathbf{f}$.

1. Uniformly choose a set of spans $\{[k:\ell)\}$ in $\mathbf{f}$ that form a partition of the output sentence into phrases: $\sqcup[k:\ell) = [0:|\mathbf{f}|)$.

2. For each span $[k:\ell)$, choose a corresponding position $[g:h)$ in the English sentence and establish the alignment $[g:h) \Leftrightarrow [k:\ell)$, then generate exactly one output phrase type $e_{[g:h)}$ conditioned on input phrase type $f_{[k:\ell)}$.

The likelihood that a particular phrase-aligned sentence pair $(\mathbf{e}, \mathbf{f}, \mathcal{P})$ will be generated by this model, given an input sentence $\mathbf{f}$, has the following parametric form:

$$\mathrm{P}_\theta(\mathbf{e}, \mathcal{P}|\mathbf{f}) \propto \mathrm{P}(\{[k:\ell)\}\,|\mathbf{f}) \prod_{[g:h)\Leftrightarrow[k:\ell)\in\mathcal{P}} \theta(e_{[g:h)}|f_{[k:\ell)}) \cdot D([g:h) \Leftrightarrow [k:\ell)||\mathbf{e}|, |\mathbf{f}|)\ . \tag{3.8}$$

Again, we parameterize the choice of $[g:h)$ given $[k:\ell)$ using the distortion model defined in Equation 3.7.

This conditional model, while similar to the joint model in Marcu and Wong (2002), has parameters θ that share the conditional form of relative frequency features in phrase-based translation (Section 2.3.3).

3.6 Learning Generative Model Parameters

We now turn to the problem of fitting the parameters θ of these generative phrase models. In this section, we focus on the conditional model above. This experimental study was originally published by DeNero *et al.* (2006).

3.6.1 Maximum Likelihood Estimation

A generative model assigns a *likelihood* to every sentence pair within its domain. We have already defined the likelihood of a phrase-aligned sentence pair $\mathrm{P}_\theta(\mathbf{e}, \mathbf{f}, \mathcal{P})$ via

Equation 3.8. This likelihood is a useful criterion for model selection. For example, the parameters of word alignment models are most often chosen to maximize the likelihood of the training set under the model, as described in Section 2.3.1. The likelihood of a training set under our conditional model is a function of the conditional phrase distribution θ, and takes the form:

$$\mathcal{L}(\theta) = \prod_{(\mathbf{e},\mathbf{f})} \sum_{\mathcal{P}} \mathrm{P}_\theta(\mathbf{e}, \mathbf{f}, \mathcal{P}) \ .$$

We can optimize $\mathcal{L}$ using the expectation maximization (EM) algorithm. $\mathcal{L}$ is non-convex, due to the coupling of the likelihood terms by the latent alignment. However, EM will find a local maximum in the likelihood function.

For this model, the E-step of EM requires us to compute the expected number of times that each phrase type $\bar{e}$ is aligned to each phrase type $\bar{f}$. In each iteration of EM, we re-estimate each phrase translation probability by summing expected phrase counts $c(\bar{e}, \bar{f})$ from the data given the current model parameters:

$$\theta_{new}(\bar{e}|\bar{f}) = \frac{c(\bar{e}, \bar{f})}{\sum_{\bar{e}'} c(\bar{e}', \bar{f})}$$

$$c(\bar{e}, \bar{f}) = \sum_{(\mathbf{e},\mathbf{f}) \in T} \sum_{\mathcal{P}} \left[\mathrm{count}(\bar{e}, \bar{f}, \mathbf{e}, \mathbf{f}, \mathcal{P}) \cdot \mathrm{P}_\theta(\mathbf{e}, \mathbf{f}, \mathcal{P}) \right] \ .$$

Above, the expression $\mathrm{count}(\bar{e}, \bar{f}, \mathbf{e}, \mathbf{f}, \mathcal{P})$ is the count of the number of times phrase type $(\bar{e}, \bar{f})$ appears in the derivation $(\mathbf{e}, \mathbf{f}, \mathcal{P})$.

As discussed previously, the sum over all possible $\mathcal{P}$ is intractable, requiring time exponential in the length of the sentences. Moreover, the number of possible phrase pairs grows too large to fit in memory. To address both of these problems, we use the exponential-time dynamic program described in Section 3.4.2, but constrain $\mathcal{P}$ to be compatible with a word alignment produced by a simpler model. That is, we allow

only $\mathcal{P}$ consisting of bispans $[g:h) \Leftrightarrow [k:\ell)$ that can be *extracted* according to the definition in Section 2.3.2

The word alignments we use to constrain $\mathcal{P}$ are generated using the same techniques common in phrase-based pipelines, as described in Section 2.3.1. In particular, we use IBM Model 4 word alignments generated by the GIZA++ software package (Brown *et al.*, 1993; Och and Ney, 2003), and combine two directional alignments using the *grow-diag* combination heuristic (Och *et al.*, 1999).

This word-based constraint has two important effects. First, we force $\mathrm{P}(\bar{e}|\bar{f}) = 0$ for all phrase pairs not compatible with the word-level alignment for some sentence pair. This restriction reduces the total legal phrase pair types from approximately 250 million to 17 million for 100,000 training sentences. Second, the time to compute the E-step is reduced dramatically. In practice, we can compute each sentence pair's contribution in under a second.

On the other hand, constraining with word alignments is not an ideal remedy for intractability. Due to errors in the word-level alignments and non-literal translations, this constraint ruled out approximately 54% of the training set. That is, given the word-level alignment, no well-formed phrase alignment was possible under a maximum phrase length restriction of 3. Furthermore, the phrase alignment model is unable to recover from errors made by the word aligner, although it can select how to analyze phrasal patterns.

Subsequent to the original publication of this approach (DeNero *et al.*, 2006), several similar approaches have also computed phrase alignment posteriors using word alignments to constrain inference (Birch *et al.*, 2006; Cherry and Lin, 2007; Zhang *et al.*, 2008).

3.6.2 Experimental Results

Maximum likelihood estimation via EM produces θ, a conditional multinomial distribution over phrase pair types. These parameters can take the place of the phrasal relative frequency features derived from word alignments, as described in Section 2.2. These relative frequency features serve as our baseline for comparison, and we denote them θ_{RF}.

We evaluate θ and θ_{RF} as features in an end-to-end translation system from English to French. All training and test sentence pairs were drawn from the French-English section of the Europarl sentence-aligned corpus (Koehn, 2002). We tested translation performance on the first 1,000 unique sentences of length 5 to 15 in the corpus and trained on sentences of length 1 to 60 starting after the first 10,000. We omitted the tuning step of the standard pipeline described in Section 2.2 because we only include three features: a language model, a distortion model, and a single conditional phrase score (either θ or θ_{RF}), each combined with weight 1.

The language model was generated from the Europarl corpus using the SRI Language Modeling Toolkit (Stolcke, 2002). The publicly available Pharaoh package performed search in the space of derivations (Koehn *et al.*, 2003). A maximum phrase length of 3 was used for all experiments. The final translation output is evaluated using BLEU, a precision-based metric that compares phrases in the output to human-generated reference translations (Papineni *et al.*, 2002).

Figure 3.4 compares the BLEU scores using each estimate. The expectation maximization algorithm for training θ was initialized with the baseline parameters θ_{RL}, so the θ_{RL} curve can be equivalently labeled as iteration 0. The model-based estimate θ underperforms its heuristic initialization. This pattern of performance was also observed by Koehn *et al.* (2003), which compated θ_{RL} to features derived from the joint phrase-factored alignment model described in Section 3.5.2 and originally

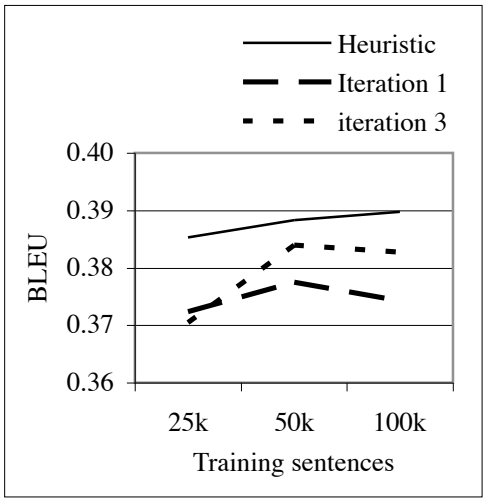


Figure 3.4: Statistical re-estimation using a generative phrase model degrades BLEU score relative to its heuristic initialization using relative frequency counts based on word alignments.

by Marcu and Wong (2002).

Thus, the first iteration of EM increases the observed likelihood of the training sentences while simultaneously degrading translation performance on the test set. As training proceeds, performance on the test set levels off after three iterations of EM. The system never achieves the performance of its initialization parameters.

3.6.3 Analysis of Experimental Results

Learning θ degrades translation quality in large part because EM learns overly determinized segmentations and translation parameters, overfitting the training data and failing to generalize. The primary increase in expressiveness between generative word-level and phrase-level models is due to the additional latent segmentation variable assumed by phrase alignment models. Although we impose a uniform distribution over segmentations, they nonetheless play a crucial role during training. We will characterize this effect through aggregate statistics and translation examples

shortly, but begin by demonstrating the model's capacity to overfit the training data.

We first return to the motivation behind introducing and learning variable-resolution phrases in machine translation. For any language pair, there are contiguous strings of words whose collocational translation is non-compositional; that is, they translate together differently than they would in isolation. For instance, *chat* in French generally translates to *cat* in English, but *appelez un chat un chat* is an idiom which translates to *call a spade a spade*. Introducing phrases allows us to translate *chat un chat* atomically to *spade a spade* and vice versa.

The structure of our model, which has fixed probability mass to distribute across competing phrasal analyses of the training corpus, encourages θ to learn that *chat* should never be translated to *spade* in isolation. On the other hand, the baseline approach which lacks this structural learning pressure is only sensitive to the fact that *chat* and *spade* co-occur regularly. Hence, translating *I have a spade* can lead to a lexical error in the baseline, and we observed this error in our system. This error should be corrected by learning θ via a phrase alignment model.

However, imposing competition among segmentations introduces a new problem: true translation ambiguity can also be spuriously explained by the segmentation. In our generative model, counter-intuitively deterministic translation parameters can yield higher likelihoods than intuitive parameters. Consider the french fragment *carte sur la table*, which could translate to *map on the table* or *notice on the chart*. Using these two sentence pairs as training, one would hope to capture the ambiguity in the parameter table as:

French	English	$\theta(e\|f)$
carte	*map*	0.5
carte	*notice*	0.5
carte sur	*map on*	0.5
carte sur	*notice on*	0.5
sur	*on*	1.0
...	...	...
table	*table*	0.5
table	*chart*	0.5

Assuming we only allow non-degenerate segmentations and disallow non-monotonic distortions, this parameter table yields a marginal likelihood $\Pr(\mathbf{f}|\mathbf{e}) = 0.25$ for both sentence pairs – the intuitive result given two independent lexical ambiguities. However, the following table yields a likelihood of 0.28 for both sentences:[7]

French	English	$\theta(e\|f)$
carte	*map*	1.0
carte sur	*notice on*	1.0
carte sur la	*notice on the*	1.0
sur	*on*	1.0
sur la table	*on the table*	1.0
la	*the*	1.0
la table	*the table*	1.0
table	*chart*	1.0

According to the parameters above, a translation can be chosen deterministically

[7]For example, summing over the first translation expands to $\frac{1}{7}(\theta(map \mid carte)\theta(on\ the\ table \mid sur\ la\ table) + \theta(map \mid carte)\theta(on \mid sur)\theta(the\ table \mid la\ table))$.

given the segmentation! All notion of lexical ambiguity has been lost.

Hence, a higher likelihood can be achieved by allocating some source-side phrases to certain translations while reserving overlapping phrases for others, thereby failing to model the real ambiguity that exists across the language pair. Also, notice that the phrase *sur la* can take on an arbitrary distribution over any english phrases without affecting the likelihood of either sentence pair. Not only does this counterintuitive parameterization give a high data likelihood, but it is also a fixed point of the EM algorithm.

The phenomenon demonstrated above poses a problem for generative phrase models in general. The ambiguous process of translation can be modeled either by the latent segmentation variable or the phrase translation probabilities. In some cases, optimizing the likelihood of the training corpus selects for the former when we would prefer the latter. We next investigate how this problem manifests in θ and its effect on translation quality.

3.6.4 Analysis of Learned parameters

The parameters of θ differ from the heuristically extracted parameters θ_{RL} in that the conditional distribution over English translations for some French words is sharply peaked for θ compared to the flat distributions of θ_{RL}.

To quantify the notion of peaked distributions over phrase translations, we compute the entropy of the distribution for each French phrase:

$$H(\theta) = \sum_{\bar{e}} \theta(\bar{e}|\bar{f}) \log_2 \theta(\bar{e}|\bar{f})$$

The average entropy for the most common 10,000 phrases in the learned table was 1.45, comparable to 1.54 for the heuristic table. The difference between the tables becomes much more striking when we consider a histogram of entropies for phrases in

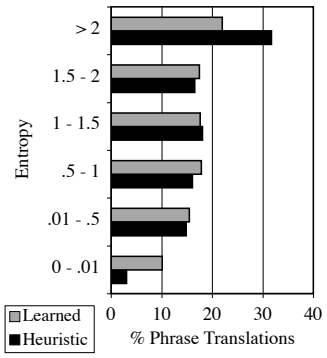


Figure 3.5: Many more French phrases have very low entropy under the learned parameterization.

Figure 3.5. In particular, the learned table has many more phrases with entropy near zero. The most pronounced entropy differences often appear for common phrases. The ten most common phrases in the French corpus, along with the entropies of their translation distributions, are shown in Figure 3.6.

As more probability mass is reserved for fewer translations, many of the alternative translations under θ_{RL} are assigned prohibitively small probabilities. In translating 1,000 test sentences, for example, no phrase translation with $\theta(\bar{e}|\bar{f})$ less than 10^{-5} was used in a final output translation. Given this empirical threshold, nearly 60% of entries in θ are unusable, compared with 1% in θ_{RL}. In practice, θ is much more sparse than θ_{RL}

3.6.5 Modeling Effects on End-to-End Translation

While this determinism of θ may be desirable in some circumstances, we found that the ambiguity captured by θ_{RL} is often preferable at translation time. In particular, the pattern of translation-ambiguous phrases receiving spuriously peaked distributions (as described in section 3.6.3) introduces new translation errors relative to the baseline.

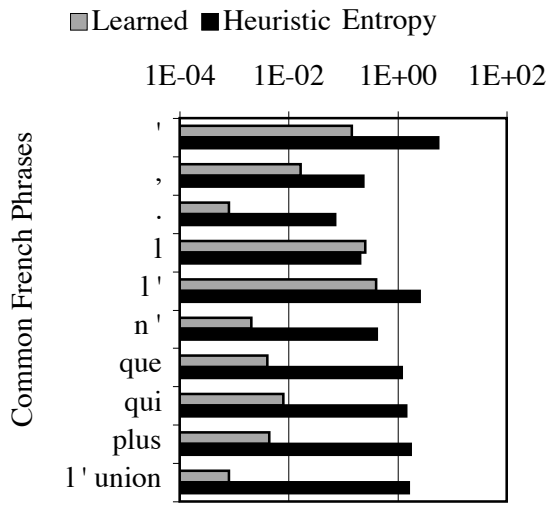


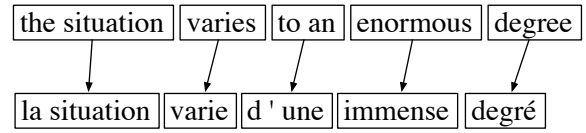
Figure 3.6: Entropy of 10 common French phrases. Several learned distributions have very low entropy.

Training θ with a generative phrase alignment model does improve some translations. The issue that motivated training a generative model is addressed correctly: for a word that translates differently alone than in the context of an idiom, the translation probabilities more accurately reflect this. For instance, the translation distribution for *chat* has been corrected through the learning process. The heuristic process gives the incorrect translation *spade* with 61% probability, while the statistical learning approach gives *cat* with 95% probability.

While such targeted examples of improvement are encouraging, the trend of spurious determinism overwhelms this benefit by introducing errors in four ways, each of which will be explored in turn.

1. Useful phrase pairs are assigned very low probability.

2. A proper translation for a phrase can be blocked by another translation with spuriously high probability.

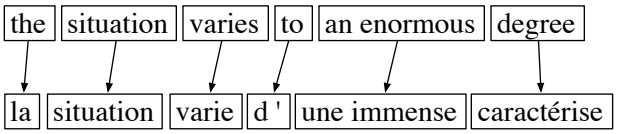


caractérise

English	$\phi(e\|f)$
characterises	0.49
characterised	0.21
permeate	0.05
features	0.05
typifies	0.05

degré

English	$\phi(e\|f)$
degree	0.49
level	0.38
extent	0.02
amount	0.02
how	0.01



caractérise

English	$\phi(e\|f)$
degree	0.998
characterises	0.001
characterised	0.001

degré

English	$\phi(e\|f)$
degree	0.64
level	0.26
extent	0.10

Figure 3.7: Spurious determinism in the learned phrase parameters degrades translation quality.

3. Error-prone ambiguous phrases become active during translation.

4. The language model cannot distinguish between different translation options as effectively due to deterministic translation model distributions.

The first effect follows from our observation in section 3.6.4 that many phrase pairs are unusable due to vanishingly small probabilities. Some of the entries that are unusable in θ would be beneficial to translation, as evidenced by the fact that removing these phrases from θ_{RL} reduces BLEU score by over 5%.

The second effect is more subtle. Consider the sentence in Figure 3.7. While there exists some agreement error and awkwardness in both translations, the translation generated under the heuristically extracted parameters θ_{RL} is comprehensible to native speakers. On the other hand, the learned translation incorrectly translates *degree*, degrading translation quality. Notice also that the translation probabilities from heuristic extraction are non-deterministic. On the other hand, the translation system makes a significant lexical error on this simple sentence when parameterized by θ: the use of *caractérise* in this context is incorrect. This error arises from a

sharply peaked distribution over English phrases for *caractérise*.

This example illustrates a recurring problem: errors do not necessarily arise because a correct translation is not available. Notice that a preferable translation of degree as *degré* is available under both parameterizations. *Degré* is not used, however, because of the peaked distribution of a competing translation candidate. In this way, very high probability translations can effectively block the use of more appropriate translations.

What is furthermore surprising and noteworthy in this example is that the learned, near-deterministic translation for *caractérise* is not a common translation for the word. Not only does the statistical learning process yield low-entropy translation distributions, but occasionally the translation with undesirably high conditional probability does not have a strong surface correlation with the source phrase. This example is not unique; during different initializations of the EM algorithm, we noticed such patterns even for common French phrases such as *de* and *ne*.

The third source of errors is closely related: common phrases that translate in many ways depending on the context can introduce errors if they have a spuriously peaked distribution. For instance, consider the lone apostrophe, which is treated as a single token in our data set (Figure 3.8). The shape of the heuristic translation distribution for the phrase is intuitively appealing, showing a relatively flat distribution among many possible translations. Such a distribution has very high entropy. On the other hand, the learned table translates the apostrophe to *the* with probability very near 1.

Selecting ambiguous phrases that require context to translate correctly will invariably introduce errors. The flatness of the distribution of θ_{RL} ensures that the single apostrophe will rarely be used because no single outcome has high enough probability to promote its use. On the other hand, using the peaked entry $\theta(the|')$ incurs virtually no cost to the score of a translation.

Heuristic		Learned	
English	$\theta_{RL}(e\|f)$	English	$\theta(e\|f)$
our	0.10	*the*	0.99
that	0.09	*,*	$4.1 \cdot 10^{-3}$
is	0.06	*is*	$6.5 \cdot 10^{-4}$
we	0.05	*to*	$6.3 \cdot 10^{-4}$
next	0.05	*in*	$5.3 \cdot 10^{-4}$

Figure 3.8: Translation probabilities for an apostrophe, the most common French phrase in our tokenized parallel corpus. The learned table contains a highly peaked distribution.

The final source of error stems from interactions between the language and translation models. The selection among translation choices via a language model is hindered by the determinism of the translation model. This effect appears to be less significant than the previous three. We note, however, that adjusting the language and translation model weights during decoding does not close the performance gap between θ_{RL} and θ.

3.6.6 Interpolating Model-Based and Heuristic Estimators

In light of the low-entropy of θ, we could hope to improve translations by retaining entropy. There are several strategies we have considered to achieve this.

The simplest strategy to increase entropy is to interpolate the heuristic and learned phrase tables. Training on 100,000 sentences, this approach yields a BLEU score of 0.386, outperforming θ and nearly equaling the performance of θ_{RL}. Further improvements come from varying the weight of interpolation, which yielded an improvement over θ_{RL} of up to 1.0 BLEU. However, this simple approach does little to remedy the most problematic cases. For example, a near-zero entropy distribution averaged with a very flat distribution still leaves almost half the probability mass on one outcome and the rest spread thinly among the rest.

Additionally, we modified the training loop to prevent convergence to low entropy

distributions. To start, we interpolated the output of each iteration of EM with its input, thereby maintaining some entropy from the initialization parameters. BLEU score increased to a maximum of 39.4 using this technique, outperforming the heuristic by a slim margin of 0.5 BLEU.

3.6.7 Summary of Findings for Likelihood-Trained Models

Re-estimating phrase translation probabilities using a generative model holds the promise of improving upon heuristic techniques. However, the combinatorial properties of a phrase-based generative model have problematic effects. Parameter estimates that explain lexical ambiguity using segmentation variables can in some cases yield higher data likelihoods by abusing the latent segmentation variable in order to condition on rare events. These rare events in turn promote low-entropy conditional phrase distributions, which generate errors at translation time.

While the experiments presented here have focused on the conditional phrase alignment model, the joint model also falls into degenerate learning patterns when trained to maximize likelihood. Rather than selecting for rare conditioning events, the joint model selects for large phrases in order to explain the training data using as few draws from its joint phrase pair multinomial as possible. As a result, the joint model often fails to learn to translate individual words and short phrases, instead focusing only on longer phrases that may not recur at translation time.

The following chapters address these learning challenges. Chapter 4 introduces prior distributions on θ for both joint and conditional models that coerce learning toward more useful results. Chapter 5 applies discriminative learning, which avoids these pitfalls by training explicitly toward human-generated alignment annotations.

Chapter 4

Bayesian Phrase Alignment Models

This chapter describes priors for the conditional and joint generative phrase-factored alignment models defined in Section 3.1. The purpose of these priors is to bias learning toward phrase alignment analyses that are useful at translation time. As such, the priors do not encode interpretable prior knowledge of the correct alignment, as their name might suggest, but instead express correctional pressure to mitigate the degenerate learning behavior of maximum likelihood estimation described in Chapter 3.

In particular, we wish to correct two degenerate learning patterns. In both models, the maximum likelihood objective pressures the model to explain each sentence pair using a small number of large phrases, because each phrase introduces an additional multiplicative term into the likelihood expression. In the conditional model, training for maximum likelihood also causes the model to explain lexical ambiguity in the data using segmentation ambiguity, and thereby to select rare input phrases.

The Bayesian priors we describe correct for these degenerate patterns in two ways:

1. They express a strong preference for short phrases rather than long ones. Short phrases are more common and therefore more reusable at translation time.

2. A preference is expressed for reusing phrase types across the analyses of multiple sentence pairs in the training corpus. Because common phrases are more readily available to be reused, this preference also promotes analyses that include common phrase pairs.

The computational machinery that enables us to express these priors are the *Dirichlet Process*—a simple prior over multinomials with unbounded dimension—and *collapsed Gibbs sampling*—an approximate inference technique with desirable convergence properties. We first describe these techniques and apply them to phrase alignment, and then we evaluate our improved models experimentally.

4.1 Bayesian Priors for Generative Models

Bayesian modeling treats model parameters as additional random variables that have associated distributions. This additional distribution over parameters, referred to as a prior, grants us the flexibility to adjust our learning objective while maintaining the same structure and parameterization of the underlying model. That is, we retain our model definition but abandon the maximum likelihood training criterion.

4.1.1 From Parameter Estimation to Expected Counts

The purpose of phrase alignment modeling is to replace the phrasal relative frequency features used in a standard phrase-based translation pipeline, which are based on counts of phrases extracted from word-level alignments:

$$\phi_{RL}(\bar{e}, \bar{f}) = \frac{\text{count}(\bar{e}, \bar{f})}{\sum_{\bar{e}'} \text{count}(\bar{e}', \bar{f})} \tag{4.1}$$

In Section 3.6, we focused on selecting a single value for the parameter θ of a phrase alignment model that explained the training data well. We then used the

value of θ instead of relative frequency statistics in a translation pipeline. In the case of our conditional phrase alignment model estimated using a maximum likelihood criterion, the entries of θ are equivalent to the expected relative frequency of phrases in the training corpus under the model:

$$\theta(\bar{e}|\bar{f}) = \frac{\mathbb{E}_\theta\left[\text{count}(\bar{e},\bar{f})\right]}{\sum_{\bar{e}'}\mathbb{E}_\theta\left[\text{count}(\bar{e}',\bar{f})\right]} \tag{4.2}$$

Comparing Equations 4.1 and 4.2, we can see the contrast between the standard word-alignment-based method and the phrase-alignment-model-based method of computing relative frequency features. The former uses fixed word alignments to collect counts, while the latter collects expected counts that reflect alignment uncertainty.

The Bayesian framework also lets us collect expected counts of aligned phrase pairs. The core differences between maximum likelihood training and the Bayesian approach we present below are (a) we consider θ to be a random variable with an explicit model prior $\text{P}(\theta)$, and (b) the phrase pair count expectations under the Bayesian model is not computed with respect to a particular value of θ, but instead as

$$\mathbb{E}\left[\text{count}(\bar{e},\bar{f})\right] = \int_\theta \text{P}(\theta)\cdot\mathbb{E}_\theta\left[\text{count}(\bar{e},\bar{f})\right]\cdot d\theta \tag{4.3}$$

where $\mathbb{E}_\theta\left[\text{count}(\bar{e},\bar{f})\right]$ denotes the expected count of a phrase pair under a particular value of the parameter θ. Integrating out the parameter in this way allows us to take into account our uncertainty over θ in our expectation, rather than having to select a particular point estimate of θ. In the end, only the expected counts will serve as features in a translation model. Hence, the experiments and analysis portions of this chapter will not focus on properties of θ, but instead on properties of the expected phrase pair counts computed by integrating over θ.

Computing expected counts explicitly, rather than focusing on a parameter estimate, also allows flexibility in the definition of $\text{count}(\cdot)$. The baseline extraction ap-

proach from Section 2.3.2 includes both minimal and composed phrase pairs. Model parameters θ reference only minimal bispans. In this chapter, we compute an expected count wherein both minimal and composed phrase pairs are counted, which better corresponds to the baseline.

4.1.2 Inference in Bayesian Models

In a Bayesian model, where parameters are treated as random variables, we aim to compute expectations that reflect our uncertainty over the values of those parameters, as in Equation 4.3. In other terms, rather than learning a model from some parameterized space, our goal is to find expectations under a fixed Bayesian model with random parameters. Thus, we collapse the distinction between learning and inference that played a prominent role in Chapter 3.

However, adopting a DP prior over the parameters of our phrase pair multinomials fundamentally changes our inference problem. Because the multinomial parameters are unknown latent variables, the expectations of phrase alignments in different sentence pairs are coupled together. We can no longer perform inference on each sentence pair independently. Instead, we must treat the training corpus as a whole.

Sampling allows us to decouple inference for sentence pairs from each other by fixing parts of the latent structure. While approximate, sampling methods can provide unbiased estimators of posterior expectations under a model, which converge to the true values under a model in the limit. The following section describes such a sampler for phrase alignment models that allows us to incorporate our Bayesian priors.

4.2 A Gibbs Sampler for Phrase Alignments

Our models involve observed sentence pairs, which in aggregate we can call x, latent phrase alignments, which we can call z, and parameters θ. For exposition purposes, we describe a Gibbs sampling algorithm for computing expected counts of phrases under $\mathrm{P}(z|x,\theta)$ for fixed θ, as in Equation 4.2. In this case, each bispan $[g:h) \Leftrightarrow [k:\ell)$ in each sentence pair has some fixed potential $\exp(\phi([g:h) \Leftrightarrow [k:\ell)))$ that it contributes to the probability of any phrase alignments $\mathcal{P}$ that contains $[g:h) \Leftrightarrow [k:\ell)$, where $\phi([g:h) \Leftrightarrow [k:\ell))$ is some function of the parameters θ. In Section 4.3, we extend this method to compute expectations under $\mathrm{P}(z|x)$, with θ marginalized out entirely, as in Equation 4.3.

In a Gibbs sampler, we start with a *complete* alignment, state z_0, which sets all latent variables to some initial configuration. We then produce a sequence of sample states z_i, each of which differs from the last by some small local change. The samples z_i are guaranteed (in the limit) to consistently approximate the conditional distribution $\mathrm{P}(z|x,\theta)$ (or $\mathrm{P}(z|x)$ later). Therefore, the average counts of phrase pairs in the samples converge to expected counts under the model.

Gibbs sampling is not new to the natural language processing community (Teh, 2006; Johnson *et al.*, 2007a). However, it is usually used as a search procedure akin to simulated annealing, rather than for approximating expectations (Goldwater *et al.*, 2006; Finkel *et al.*, 2007). Our application is also atypical for an NLP application in that we use an approximate sampler not only to include Bayesian prior information (section 4.3), but also because computing phrase alignment expectations exactly is a #P-hard problem, as we showed in Section 3.2.

4.2.1 Related Work

Expected phrase pair counts under $\mathrm{P}(z|x,\theta)$ have been approximated before in order to run EM. Marcu and Wong (2002) employed the local hill climbing approach described in Section 3.4.1, and DeNero *et al.* (2006) employed the exponential-time dynamic program described in Section 3.4.2, pruned by word alignments. Subsequent work relied heavily on word alignments to constrain inference, even under reordering models that admit polynomial-time E-steps (Cherry and Lin, 2007; Zhang *et al.*, 2008).

None of these approximations are consistent, and they offer no method of measuring their biases. Gibbs sampling is not only consistent in the limit, but also conveniently integrates our Bayesian priors. Of course, sampling has liabilities as well: we do not know in advance how long we need to run the sampler to approximate the desired expectations "closely enough."

Snyder and Barzilay (2008) describe a Gibbs sampler for a bilingual morphology model very similar in structure to ours. However, the basic sampling step they propose – resampling all segmentations and alignments for a sequence at once – requires a #P-hard computation. While this asymptotic complexity was apparently not prohibitive in the case of morphological alignment, where the sequences are short, it is prohibitive in phrase alignment, where the sentences are often very long.

4.2.2 Sampling with the Swap Operator

Our Gibbs sampler repeatedly applies each of five operators to each position in each training sentence pair. Each operator freezes all of the current state z_i except a small local region, determines all the ways that region can be reconfigured, and then chooses a (possibly) slightly different z_{i+1} from among those outcomes according to the conditional probability of each, given the frozen remainder of the state. This

frozen region of the state is called a *Markov blanket* (denoted m), and plays a critical role in proving the correctness of the sampler.

The first operator we consider is SWAP, which changes alignments but not segmentations. It freezes the set of aligned spans in each sentence, then picks two output spans e_1 and e_2.[1] All bispans are frozen except the two that contain e_1 and e_2: $e_1 \Leftrightarrow f_1$ and $e_2 \Leftrightarrow f_2$. SWAP chooses between retaining bispans $e_1 \Leftrightarrow f_1$ and $e_2 \Leftrightarrow f_2$ (outcome o_0), or swapping their to create $e_1 \Leftrightarrow f_2$ and $e_2 \Leftrightarrow f_1$ (outcome o_1).

SWAP chooses stochastically in proportion to each outcome's posterior probability: $\mathrm{P}(o_0|m, x, \theta)$ and $\mathrm{P}(o_1|m, x, \theta)$. Each bispan in each outcome contributes to these posteriors its local potential,

$$\psi(e, f) = \exp(\phi(e, f)) \ .$$

The outcome of SWAP is thus drawn from the following binomial distribution:

$$\mathrm{P}(o_0|m, x, \theta) = \frac{\psi(e_1 \Leftrightarrow f_1)\psi(e_2 \Leftrightarrow f_2)}{\psi(e_1 \Leftrightarrow f_1)\psi(e_2 \Leftrightarrow f_2) + \psi(e_1 \Leftrightarrow f_2)\psi(e_2 \Leftrightarrow f_1)}$$

$$\mathrm{P}(o_1|m, x, \theta) = \frac{\psi(e_1 \Leftrightarrow f_2)\psi(e_2 \Leftrightarrow f_1)}{\psi(e_1 \Leftrightarrow f_1)\psi(e_2 \Leftrightarrow f_2) + \psi(e_1 \Leftrightarrow f_2)\psi(e_2 \Leftrightarrow f_1)} \ .$$

To see that these are in fact the posteriors of the two outcomes under a phrase-factored model, note that the remaining terms in the full data likelihood are equivalent in either case because they are all contained within the Markov blanket m, which is constant across outcomes.

Operators in a Gibbs sampler require certain conditions to guarantee that averaging over samples created by applying them many times will yield an unbiased

[1] We also apply the operator to two input phrases, but we focus on the output case for exposition.

estimator of the model posterior. First, they must choose among all possible configurations of the unfrozen local state. Second, immediately re-applying the operator from any outcome must yield the same set of outcome options as before.[2] If these conditions are not met, the sampler may no longer be guaranteed to yield consistent approximations of the posterior distribution. A full formal justification of this step and others appears in Section 4.2.6.

A subtle issue arises with SWAP as defined: should it also consider an outcome o_2 of $e_1 \Leftrightarrow \emptyset$ and $e_2 \Leftrightarrow \emptyset$ that removes alignments? No part of the frozen state is changed by removing these alignments, so the first Gibbs condition dictates that we must include o_2. However, after choosing o_2, when we reapply the operator to positions e_1 and e_2, we freeze all alignments except $e_1 \Leftrightarrow \emptyset$ and $e_2 \Leftrightarrow \emptyset$, which prevents us from returning to o_0. Thus, we fail to satisfy the second condition.

Fortunately, the problem is not with SWAP, but with our justification of it: we can salvage SWAP by augmenting its Markov blanket. Given that we have selected $e_1 \Leftrightarrow f_1$ and $e_2 \Leftrightarrow f_2$, we not only freeze all other alignments and phrase boundaries, but also the number of aligned phrase pairs. With this count held invariant, o_2 is not among the possible outcomes of SWAP given m. Moreover, regardless of the outcome chosen, SWAP can immediately be reapplied at the same location with the same set of outcomes.

As we have defined SWAP, it can manipulate an unaligned phrase if it chooses it initially as e_1 and chooses some aligned e_2. Then, the outcomes are

o_0**:** $e_1 \Leftrightarrow \emptyset$; $e_2 \Leftrightarrow f_2$

o_1**:** $e_1 \Leftrightarrow f_2$; $e_2 \Leftrightarrow \emptyset$

On the other hand, if SWAP selects two unaligned phrases, then no change is possible.

[2]These are two sufficient conditions to guarantee that the Metropolis-Hastings acceptance ratio of the sampling step is 1.

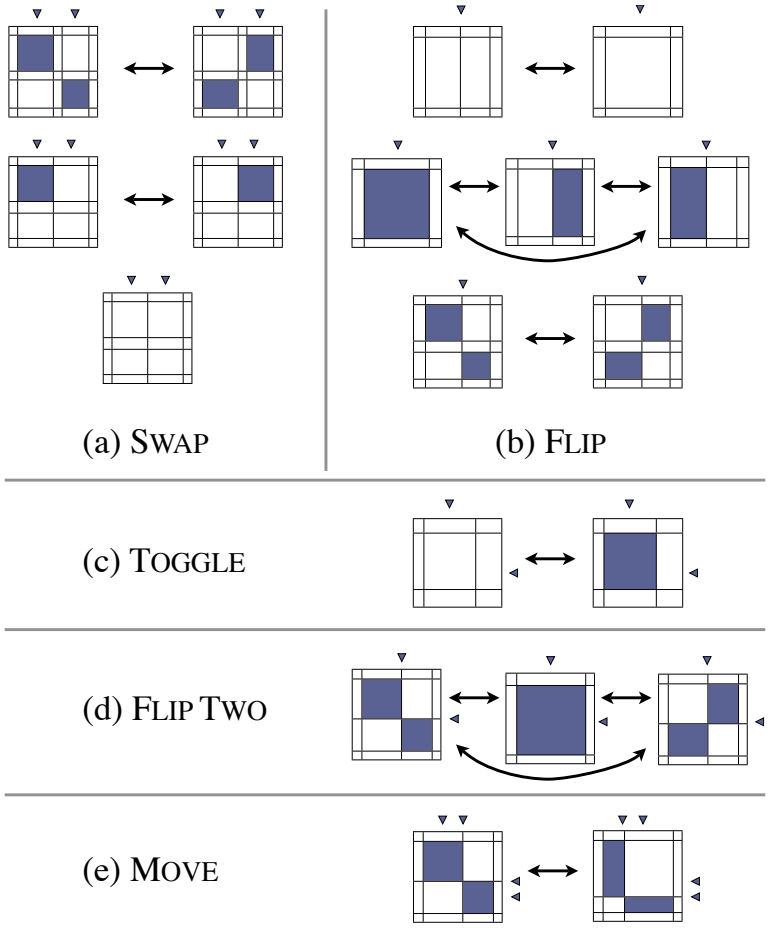


Figure 4.1: Each local operator manipulates a small portion of a single alignment. Relevant phrases are exaggerated for clarity. The outcome sets (depicted by arrows) of each possible configuration are fully connected. Certain configurations cannot be altered by certain operators, such as the final configuration in SWAP. Unalterable configurations for TOGGLE have been omitted for space.

All the possible starting configurations and outcome sets for SWAP appear in Figure 4.1(a).

4.2.3 The Flip operator

SWAP can arbitrarily shuffle alignments, but we need a second operator to change the actual phrase boundaries. The FLIP operator changes the status of a single

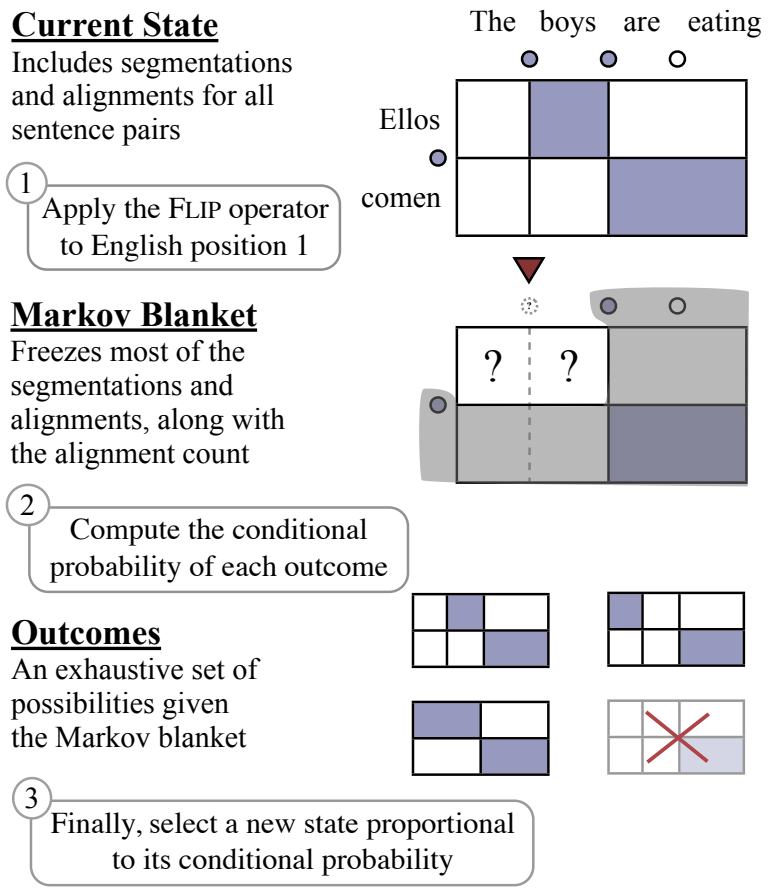


Figure 4.2: The three steps involved in applying the FLIP operator. The Markov blanket freezes all segmentations except English position 1 and all alignments except those for *Ellos* and *The boys*. The blanket also freezes the number of alignments, which disallows the lower right outcome.

segmentation position[3] to be either a phrase boundary or not. In this sense FLIP is a bilingual analog of the segmentation boundary flipping operator of Goldwater *et al.* (2006).

Figure 4.2 diagrams the operator and its Markov blanket. First, FLIP chooses any between-word position in either sentence. The outcome sets for FLIP vary based on the current segmentation and adjacent alignments, and are depicted in Figure 4.1(b).

[3]A segmentation position is a position between two words that is also potentially a boundary between two phrases in an aligned sentence pair.

For FLIP to satisfy the Gibbs conditions, we must augment its Markov blanket to freeze not only all other segmentation points and alignments, but also the number of aligned phrase pairs. Otherwise, we end up allowing outcomes from which we cannot return to the original state by reapplying FLIP. Consequently, when a position is already segmented and both adjacent phrases are currently aligned, FLIP cannot unsegment the point because it can't create two aligned phrase pairs with the one larger phrase that results.

4.2.4 The Toggle operator

Both SWAP and FLIP freeze the number of alignments in a sentence. The TOGGLE operator, on the other hand, can add or remove individual alignment links. In TOGGLE, we first choose an e_1 and f_1. If $e_1 \Leftrightarrow f_1 \in \mathcal{P}$ or both e_1 and f_1 are $\emptyset$, we freeze all segmentations and the rest of the alignments, and choose between including $e_1 \Leftrightarrow f_1$ in the alignment or leaving both e_1 and f_1 unaligned. If only one of e_1 and f_1 are aligned, or they are not aligned to each other, then TOGGLE does nothing.

4.2.5 A Complete Sampler

Together, FLIP, SWAP and TOGGLE constitute a complete Gibbs sampler that consistently samples from the posterior of a probabilistic phrase-factored model. Not only are these operators valid Gibbs steps, but they also can form a path of positive probability from any source state to any target state in the space of phrase alignments (formally, the induced Markov chain is *irreducible*). Such a path can at worst be constructed by unaligning all phrases in the source state with TOGGLE, composing applications of FLIP to match the target phrase boundaries, then applying TOGGLE to match the target alignments.

We include two more local operators to speed up the rate at which the sampler

explores the hypothesis space. FLIP TWO simultaneously flips an English and a foreign segmentation point (to make a large phrase out of two smaller ones or vice versa), while MOVE shifts an aligned phrase boundary to the left or right. Figure 4.1 depicts the outcome sets for these operators.

The purpose of including FLIP TWO and MOVE is to ensure that two high-probability regions of the alignment space are connected with a high-probability path. As a result of adding these operators, the number of discovered phrase pairs increased by 13% for an equal number of sampling iterations.

4.2.6 Justification of Gibbs Steps

We now analyze the sampler that results from the alternate application of the operators we have defined. To show that samples generated in this way converge to a stationary distribution that is the posterior of the model, we must verify that the operators are indeed Gibbs steps (Berg, 2004).[4]

Gibbs sampling is often seen in the context of a graphical model: we pick a node in the model and resample it given its Markov blanket. Analogously, for each operator, we need to find the largest event that characterizes which part of the state is invariant across all outcomes of the operator. A step is Gibbs if this invariant exists, and the operator chooses appropriately among all positive-probability outcomes.

Consider FLIP at English position i, illustrated in Figure 4.2. Let I_i be the interval corresponding to the foreign segment to which the English phrase containing word i aligns (possibly null). Here, we sample from the distribution conditioning on the current value of $E_{-i}, F, I_-, I_i \cup I_{i-1}$.[5] $I_i \cup I_{i-1}$ is the portion of the foreign sentence covered by I_i and I_{i-1}. Note that we condition on the current value of $I_i \cup I_{i-1}$ in

[4]We must also verify that from any source to any target state, there is a path of positive probability. This condition was shown in Section 4.2.5.

[5]Here I_- is the alignments of phrases outside of e_s and $E_{-i} = E_1, \ldots, E_{i-1}, E_{i+1}, \ldots, E_{|\mathbf{e}|}$.

order to ensure that each outcome aligns only and exactly to the current projection of the phrases bordering i.

FLIP TWO i, j can be treated similarity: the conditioning event is $E_{-i}, F_{-j}, I_{-}, I_i \cup I_{i-1}$. SWAP, on the other hand gives more flexibility on the alignment, but none on the segmentation: the event is given by the value of $E, F, I_{-\{i,i'\}}$.

For the MOVE k, k' operator, we will need a different representation of the state space. Let $P_1 < P_2 < \cdots < P_N$ be the positions in the source sentence where the partitions fall (similarily, we have $Q_1 < \cdots < Q_M$ for the target sentence). Let $G_1, \ldots, G_N$ be $\mathbb{N} \cup \{null\}$ valued random variables, where G_i points to the index of the segment in the target side it is aligned to (if any). We can then write the conditioning event as the current value of $P_{-k}, Q_{-k'}, G_{-\{k,k'\}}, \{G_k, G_{k'}\}$. Again, the random set $G_{-\{k,k'\}}, \{G_k, G_{k'}\}$ ensures that the modification preserves the alignment up to a swap and only allows resizing the split point.

Conditioning on these invariants, one can check that each operator covers every outcome possible. Therefore, these are Gibbs steps. We verified this theoretical result programmatically by computing Metropolis-Hastings acceptance ratios for each operator, which were always 1.

4.2.7 Expected Phrase Pair Counts

With our sampling procedure in place, we can now estimate the expected number of times a given phrase pair occurs in our data, for fixed θ, using a Monte-Carlo average,

$$\frac{1}{N}\sum_{i=1}^{N} \text{count}(\bar{e}, \bar{f}, x, z_i) \xrightarrow{a.s.} \mathbb{E}_\theta \left[\text{count}(\bar{e}, \bar{f}, x)\right] \quad .$$

Where $\text{count}(\bar{e}, \bar{f}, x, z_i)$ is the number of times that the phrase pair type $\bar{e} \Leftrightarrow \bar{f}$ appears in the observed dataset x under phrase alignment z_i.

The left hand side above is simple to compute; we count aligned phrase pairs in each sample we generate. We count both the phrase pairs directly aligned to each other and larger *composed* phrase pairs that include multiple contiguous sub-phrases that are aligned. This phrase pair counting is identical to the baseline phrase extraction procedure defined in Section 2.3.2, if each aligned bispan $[g : h) \Leftrightarrow [k : \ell)$ is interpreted as a dense set of word-to-word alignment links:

$$\{(i, j) : i \in [g : h) \ \wedge \ j \in [k : \ell)\} \ .$$

In practice, rather than counting phrase pairs after every sampling step, we only count phrase pairs after applying every operator to every position in every sentence (one iteration). By the strong law of large numbers for Markov Chains, this average converges to the true expectation in the limit. For experiments, we ran the sampler for 100 iterations.

4.3 Bayesian Priors for Phrase Models

The Gibbs sampler we have presented allows us to estimate expected phrase counts under a fixed parameter θ. With slight modification described below in Section 4.3.5, it also enables us to estimate expected counts under a prior distribution.

$$\int_{\theta} \mathrm{P}(\theta) \cdot \mathbb{E}_{\theta} \left[\mathrm{count}(\bar{e}, \bar{f}, x) \right] \cdot d\theta$$

In this section, we define Bayesian models that treat θ as a random variable. We select priors that are designed to prevent the degenerate learning behavior identified in Section 3.6.7, which we revisit below.

4.3.1 Model Degeneracy

Consider the joint model from Section 3.5.2. The structure of the joint model penalizes explanations that use many small phrase pairs: each phrase pair token incurs the additional expense of generation and distortion. In fact, the maximum likelihood estimate of the model puts mass on phrase pair types $\bar{e} \Leftrightarrow \bar{f}$ that span entire training sentences, explaining the training corpus with one phrase pair per sentence. Constraining the model to a maximum phrase length does little to address this degenerate behavior: maximum likelihood estimation will select as long phrases as it can, ignoring the short, reliable phrases that generalize well to new examples.

Previous phrase alignment work has primarily mitigated this tendency by constraining the inference procedure used for learning and prediction. For example, allowed phrase pairs have been constrained by word alignments and linguistic features (Birch *et al.*, 2006) or by disallowing phrase pairs that can be decomposed into contiguous sub-phrases (Cherry and Lin, 2007; Zhang *et al.*, 2008). However, the problem lies with the model, and therefore should be corrected in the model, rather than the inference procedure.

Model-based solutions appear in the literature as well, though typically combined with word alignment constraints on inference. A sparse Dirichlet prior coupled with variational EM was explored by Zhang *et al.* (2008), but it did not avoid the degenerate solution. Moore and Quirk (2007a) proposed a new conditional model structure that does not cause large and small phrases to compete for probability mass. May and Knight (2007) added additional model terms to balance the cost of long and short derivations in a syntactic alignment model.

However, no previous work has defined a model that avoids degenerate solutions while allowing all possible phrase alignments during learning and inference.

4.3.2 The Dirichlet Process

The priors we employ are instances of the Dirichlet Process (DP), a close cousin to the more well-known Dirichlet distribution (Ferguson, 1973). The standard Dirichlet is the conjugate prior to a finite multinomial, such as a distribution over a fixed set of phrase pairs. The DP is an extension of the Dirichlet to multinomials over infinite outcome spaces. In principle, there can be an infinite number of phrase pair types, and so the DP is a natural choice for our domain.

The Dirichlet distribution and the DP distribution have similar parameterizations. A K-dimensional Dirichlet can be parameterized with a *concentration parameter* $\alpha > 0$ and a *base distribution* $M_0 = (\mu_1, \ldots, \mu_{K-1})$, with $\mu_i \in (0, 1)$.[6] This parameterization has an intuitive interpretation: under these parameters, the average of independent samples from the Dirichlet will converge to M_0. That is, the average of the ith element of the samples will converge to μ_i. Hence, the base distribution M_0 characterizes the sample mean. The concentration parameter α only affects the variance of the draws.

The DP is an infinite-dimensional extension to the Dirichlet distribution. We can parameterize the Dirichlet process with a concentration parameter α (that affects only the variance) and a base distribution M_0 that determines the mean of the samples. Just as in the finite Dirichlet case, M_0 is simply a probability distribution, but now with countably infinite support: all possible phrase pairs in our case. In practice, we can use an unnormalized M_0 (a base measure) by appropriately rescaling α.

Goldwater *et al.* (2009) give an intuitive interpretation of the DP: a generative process equipped with a cache. Suppose we have generated n phrase pairs so far. The next one can be generated either by emitting a previously used value from the cache

[6] This parametrization is equivalent to the standard pseudo-counts parametrization of K positive real numbers. The bijection is given by $\alpha = \sum_{i=1}^{K} \tilde{\alpha}_i$ and $\mu_i = \tilde{\alpha}_i / \alpha$, where $(\tilde{\alpha}_1, \ldots, \tilde{\alpha}_K)$ are the pseudo-counts.

or by constructing a new random one using the base distribution. Old values are reused with probability proportional to the number of times they were used before, inducing the rich-get-richer property of the Dirichlet process.

4.3.3 A Dirichlet Process Prior for the Joint Model

In the joint phrase alignment model defined in Section 3.5.2, phrase pair types are drawn from a multinomial θ over the countably infinite set of possible phrase pair types. θ ranges over both standard and null-aligned phrase pairs. In order to define our Bayesian model, we must now define θ in terms of two multinomials: θ_{J} is a distribution over standard phrase pair types, and θ_{N} is a multinomial over null-aligned phrase pairs. Then, any phrase pair can be generated via the following distribution:

$$\theta(\bar{e} \Leftrightarrow \bar{f}) = \begin{cases} p_{\emptyset} \cdot \theta_{\mathrm{N}}(\bar{e} \Leftrightarrow \bar{f}) & \text{if } \bar{e} = \emptyset \ \vee \ \bar{f} = \emptyset \\ (1 - p_{\emptyset}) \cdot \theta_{\mathrm{J}}(\bar{e} \Leftrightarrow \bar{f}) & \textit{otherwise} \end{cases}$$

We control the degenerate behavior by placing DP prior over θ_{J}, the multinomial distribution over standard aligned phrase pairs. To do so, we select a base measure that strongly prefers shorter phrases, encouraging the model to use large phrases only when it has sufficient evidence for them.

$$\theta_{\mathrm{J}} \sim DP(M_0, \alpha_j) \tag{4.4}$$

$$M_0(\bar{e} \Leftrightarrow \bar{f}) = \left[\mathrm{P}_f(\bar{f}) P_{\mathrm{WA}}(\bar{e}|\bar{f}) \cdot \mathrm{P}_e(\bar{e}) P_{\mathrm{WA}}(\bar{f}|\bar{e})\right]^{\frac{1}{2}} \tag{4.5}$$

$$\mathrm{P}_f(\bar{f}) = \mathrm{P}_{\mathrm{G}}(|\bar{f}|; p_s) \cdot \left(\frac{1}{n_f}\right)^{|\bar{f}|} \tag{4.6}$$

$$\mathrm{P}_e(\bar{e}) = \mathrm{P}_{\mathrm{G}}(|\bar{e}|; p_s) \cdot \left(\frac{1}{n_e}\right)^{|\bar{e}|} \quad . \tag{4.7}$$

P_{WA} is the IBM model 1 likelihood of one phrase conditioned on the other, which factors over the individual words contained in each phrase (Brown *et al.*, 1993). P_f and P_e are uniform over types for each phrase length: the constants n_f and n_e denote the vocabulary size of the foreign and English languages, respectively, and P_{G} is a geometric distribution.

In Equation 4.4, θ_{J} is drawn from a DP centered on the geometric mean of two joint distributions over phrase pairs, each of which is composed of a monolingual unigram model and a lexical translation component. This prior has two advantages. First, we pressure the model to use smaller phrases by increasing p_s ($p_s = 0.8$ in experiments). Second, we encourage sensible phrase pairs by incorporating IBM Model 1 distributions in Equation 4.5. This use of word alignment distributions is notably different from word alignment constraints: we are supplying prior knowledge that phrases will generally follow word alignments, though with enough corpus evidence they need not (and often do not) do so in the posterior samples. The model proved largely insensitive to changes in the sparsity parameter α_j, which we set to 100 for experiments.

4.3.4 Unaligned phrases and the DP Prior

Allowing null-aligned phrases invites further degenerate megaphrase behavior: a sentence pair can be generated cheaply as two unaligned phrases that each span an entire sentence. We attempted to place a similar DP prior over θ_{N}, but surprisingly, this modeling choice invoked yet another degenerate behavior. The DP prior imposes a rich-get-richer property over the phrase pair distribution, strongly encouraging the model to reuse existing pairs rather than generate new ones. As a result, common words consistently aligned to *null*, even while suitable translations were present, simply because each null alignment reinforced the next. For instance, *the* was always unaligned.

Instead of drawing θ_{N} from a DP prior, we fix θ_{N} to a simple unigram model that is uniform over word types. This way, we discourage unaligned phrases while focusing learning on θ_{J}. For simplicity, we reuse $\text{P}_f(f)$ and $\text{P}_e(e)$ from the prior over θ_{J}.

$$\theta_{\text{N}}(\bar{e} \Leftrightarrow \bar{f}) \;=\; \begin{cases} \frac{1}{2} \cdot \text{P}_e(\bar{e}) & \text{if } \bar{f} = \emptyset \\ \frac{1}{2} \cdot \text{P}_f(\bar{f}) & \text{if } \bar{e} = \emptyset \end{cases} .$$

The $\frac{1}{2}$ represents a choice of whether the aligned phrase is in the foreign or English sentence.

This definition concludes our presentation of the joint model with a DP prior. This model was originally proposed in DeNero *et al.* (2008).

4.3.5 Collapsed Sampling with a DP Prior

Our entire model now has the general form $\text{P}(x, z, \theta_{\text{J}})$ for observed corpus x, phrase alignments z, and multinomial parameter θ_{J}; all other model parameters have been

fixed. Instead of searching for a suitable θ_{J},[7] we sample from the posterior distribution $\mathrm{P}(z|x)$ with θ_{J} marginalized out.

To this end, we convert our Gibbs sampler into a collapsed Gibbs sampler[8] using the Chinese Restaurant Process (CRP) representation of the DP (Aldous, 1985). With the CRP, we avoid the problem of explicitly representing samples θ_{J}. CRP-based samplers have served the community well in related language tasks, such as word segmentation and coreference resolution (Goldwater *et al.*, 2006; Haghighi and Klein, 2007).

The CRP representation posits a collection of *tables*. Each table t is labeled with a phrase pair type and has a number $n(t)$ of phrase pair tokens that are "seated" at that table. Let A^- be the set of aligned phrase pair tokens observed so far, each of which is assigned to some table:

$$\sum_t n(t) = |\mathrm{A}^-|$$

Then, the process dictates that the next phrase pair drawn from our model (with θ_{J} integrated out) joins an existing table with probability $\frac{n(t)}{|\mathrm{A}^-|+\alpha}$, or starts a new table with probability $\frac{\alpha}{|\mathrm{A}^-|+\alpha}$.

Thus, for a standard phrase pair $\bar{e} \Leftrightarrow \bar{f}$ (that is, neither $\bar{e}$ nor $\bar{f}$ are $\emptyset$), the conditional probability of drawing that pair takes the form:

$$\tau_{\mathrm{DP}}(\bar{e} \Leftrightarrow \bar{f}|\mathrm{A}^-) = \frac{\mathrm{N}(\bar{e} \Leftrightarrow \bar{f}, \mathrm{A}^-) + \alpha_j \cdot M_0(\bar{e} \Leftrightarrow \bar{f})}{|\mathrm{A}^-| + \alpha_j} \ , \tag{4.8}$$

where $\mathrm{N}(\bar{e} \Leftrightarrow \bar{f}, \mathrm{A}^-)$ is the number of times that $\bar{e} \Leftrightarrow \bar{f}$ appears in A^-. Note that $\mathrm{N}(\cdot)$ differs from the count$(\cdot)$ function in Section 4.2.7: the former counts only minimal phrase pairs drawn from our model, while the latter also counts composed phrase

[7] For instance, using approximate MAP EM.

[8] A collapsed sampler is simply one in which the model parameters have been marginalized out.

pairs constructed from contiguous minimal pairs.

Notice that Equation 4.8 does not directly refer to table counts, but instead depends only on the number of times that $\bar{e} \Leftrightarrow \bar{f}$ appears in z. Hence, we do not need to track the assignment of phrase pair labels to tables, but instead only the total count of each phrase pair.

Finally, we note that our phrase pair emission model is exchangeable—it assigns the same probability to any sequence of phrase pair draws, regardless of their order. Therefore, we can treat each draw conditioned on the entire rest of the corpus (the Markov blanket) as the last draw in the corpus-long sequence, and apply Equation 4.8 to any sampling operation in our corpus.[9]

To complete the definition of our collapsed sampler, we must specify the potential function $\psi([g:h) \Leftrightarrow [k:\ell))$ that expresses the full contribution to the model posterior of aligning $e_{[g:h)}$ to $f_{[k:\ell)}$ in a sentence pair, given the current alignment to the rest of the corpus. With the entire corpus coupled together via the prior, ψ depends not only on the current sentence pair $(\mathbf{e}, \mathbf{f})$, but also the multiset A^- of phrase pair tokens in the rest of the aligned corpus in the current sample. Upon defining ψ, our sampler remains exactly as it was described in Section 4.2.

$$\psi_{\mathrm{DP}}([g:h) \Leftrightarrow [k:\ell) | \mathbf{e}, \mathbf{f}, \mathrm{A}^-) = \tag{4.9}$$

$$\begin{cases} (1-p_\$) \cdot (1-p_\emptyset) \cdot \tau(e_{[g:h)} \Leftrightarrow f_{[k:\ell)} | \mathrm{A}^-) \cdot D([g:h) \Leftrightarrow [k:\ell)) & e_{[g:h)} \neq \emptyset \wedge f_{[k:\ell)} \neq \emptyset \\ (1-p_\$) \cdot p_\emptyset \cdot \theta_{\mathrm{N}}(\bar{e} \Leftrightarrow \bar{f}) & \text{otherwise} \quad . \end{cases}$$

Above, $D([g:h) \Leftrightarrow [k:\ell))$ is the absolute position distortion component of the joint

[9]Note that the expression for τ changes slightly under conditions where two phrase pairs being changed simultaneously coincidentally share the same lexical content. Details of these fringe conditions have been omitted, but were included in our implementation.

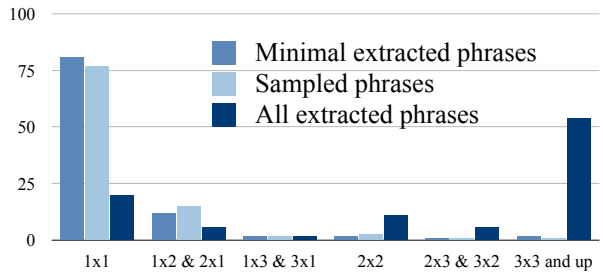


Figure 4.3: The distribution of phrase pair sizes (denoted *English length* x *foreign length*) favors small phrases under the model.

model defined in Section 3.5.2, which conditions on *sentencepair*. Respectively, $p_\$$ and $p_\emptyset$ are the phrase stop and null emission probabilities of the model. It is a signature property of the collapsed sampler that A^- must be updated after every sampling operation, thereby coupling together inference all sentence pairs in the corpus. In an uncollapsed sampler, $\tau(e_{[g:h)} \Leftrightarrow f_{[k:\ell)}|\mathrm{A}^-)$ would be replaced by $\theta_\mathrm{J}(e_{[g:h)} \Leftrightarrow f_{[k:\ell)})$ for a fixed parameter θ_J, and inference in two different sentence pairs would be independent.

4.3.6 Degeneracy Analysis

Figure 4.3 shows a histogram of phrase pair sizes in the distribution of expected counts under the model. As reference, we show the size distribution of both *minimal* and *all* phrase pairs extracted from word alignments using the standard heuristic. Our model tends to select minimal phrases, only using larger phrases when well motivated.[10]

This result alone is important: a model-based solution with no inference constraint has yielded a non-degenerate distribution over phrase lengths. Note that our sampler does find the degenerate solution quickly under a uniform prior, confirming that the model, and not the inference procedure, is selecting these small phrases.

[10]The largest phrase pair found was 13 English words by 7 Spanish words.

4.3.7 The Hierarchical Dirichlet Process

The base distribution of a DP prior M_0 is a multinomial over the same outcome space as the process itself. In the Bayesian modeling paradigm, that base distribution can also be treated as a random variable, drawn from its own prior, $\mathrm{P}(M_0)$.

When a Dirichlet process prior is used as the prior for the base distribution of another Dirichlet process, we refer to the resulting full distribution as a hierarchical Dirichlet process (HDP) (Teh *et al.*, 2005). The HDP has been applied to word segmentation (Goldwater *et al.*, 2006), language modeling (Teh, 2006), and syntactic analysis (Johnson *et al.*, 2007b; Liang *et al.*, 2007; Finkel *et al.*, 2007).

4.3.8 A Hierarchical Prior for the Joint Model

While our DP prior over θ_{J} encourages the reuse of phrase pairs, it does not have the capacity to encourage the reuse of monolingual phrases across different phrase pairs. However, we can define a hierarchical prior over θ_{J} that promotes such sharing. Our HDP prior draws monolingual distributions θ_{E} and θ_{F} from a DP and θ_{J} from their cross-product:

$$\theta_{\mathrm{J}} \sim DP(M'_0, \alpha_j) \tag{4.10}$$

$$M'_0(\bar{e} \Leftrightarrow \bar{f}) = \left[\theta_{\mathrm{F}}(\bar{f}) P_{\mathrm{WA}}(\bar{e}|\bar{f}) \cdot \theta_{\mathrm{E}}(\bar{e}) P_{\mathrm{WA}}(\bar{f}|\bar{e})\right]^{\frac{1}{2}}$$

$$\theta_{\mathrm{F}} \sim DP(\mathrm{P}_f, \alpha_h) \tag{4.11}$$

$$\theta_{\mathrm{E}} \sim DP(\mathrm{P}_e, \alpha_h) \quad . \tag{4.12}$$

This prior promotes phrase pair reuse via Equation 4.10, but also encourages novel phrase pairs to be composed of phrases that have been used before. However, it

shares the same base distribution over monolingual phrases defined by Equations 4.6 and 4.7 of the non-hierarchical DP model.

The sampling potential *psi* differs from Equation4.9 in the definition of τ. While the assignment of phrase pairs to tables in the CRP representation of the non-hierarchical DP could be summarized by A^-, this hierarchical prior requires us to track table assignments explicitly.

Let E^- be the set of tables in the upper-tier DP defined by Equation 4.12, which each have a label $l(t) = \bar{e}$ and a count $c(t)$ that tallies the number of instances of $\bar{e}$ that are assigned to t. Likewise, let F^- be the table assignments for the DP in Equation 4.11. Then,

$$\tau_{\mathrm{HDP}}(\bar{e} \Leftrightarrow \bar{f} | \mathrm{A}^-, E^-, F^-) = \frac{\mathrm{N}(\bar{e} \Leftrightarrow \bar{f}, \mathrm{A}^-) + \alpha_j \cdot \tau_{M_0}(\bar{e} \Leftrightarrow \bar{f} | E^-, F^-)}{|\mathrm{A}^-| + \alpha_j} \tag{4.13}$$

$$\tau_{M_0}(\bar{e} \Leftrightarrow \bar{f} | E^-, F^-) = \left[\tau_f(\bar{f}|F^-) P_{\mathrm{WA}}(\bar{e}|\bar{f}) \cdot \tau_e(\bar{e}|E^-) P_{\mathrm{WA}}(\bar{f}|\bar{e})\right]^{\frac{1}{2}}$$

$$\tau_f(\bar{f}|F^-) = \frac{\sum_{t \in all f : l(t) = \bar{f}} c(t) + \alpha_h \cdot \mathrm{P}_f(\bar{f})}{\sum_{t \in F^-} c(t) + \alpha_h} \tag{4.14}$$

$$\tau_e(\bar{e}|E^-) = \frac{\sum_{t \in E^- : l(t) = \bar{e}} c(t) + \alpha_h \cdot \mathrm{P}_f(\bar{f})}{\sum_{t \in E^-} c(t) + \alpha_h} \tag{4.15}$$

Care must be taken to update E^- and F^- appropriately after an operator has been applied. Table counts are updated only if the phrase pair is drawn from the latter term of the numerator of Equation 4.13 (true for all novel phrase pairs). Then, a particular table is selected for $\bar{e}$ and $\bar{f}$ respectively according to the numerators of Equations 4.14 and 4.15, and the counts of those tables must be incremented.

The HDP prior gives a similar distribution over phrase sizes as the one depicted in Figure 4.3. This model was also originally defined in DeNero *et al.* (2008).

4.3.9 A Hierarchical Prior for the Conditional Model

The conditional phrase alignment model defined in Section 3.5.3 also exhibits degenerate behavior. In particular, the model prefers to select rare phrases to condition upon, so that it can assume a spuriously low entropy conditional distribution for each conditioning event, as described in Section 3.6.3.

We propose two changes to the model. First, we let the model explain both sentences, rather than just one. The sentence $\mathbf{e}$ is generated monolingually phrase-by-phrase from a multinomial with parameter θ_{E}. Then, the sentence $\mathbf{f}$ is generated by drawing one phrase $\bar{f}$ conditioned on each $\bar{e}$ used to generate $\mathbf{e}$, using the multinomial $\theta_{f|e}$. Those phrases are then reordered to form $\mathbf{f}$.

Second, we impose priors on θ_{E} and $\theta_{f|e}$.

$$\theta_{\mathrm{E}} \sim DP(\mathrm{P}_e, \alpha_h)$$

$$\theta_{\mathrm{F}} \sim DP(\mathrm{P}_f, \alpha_h)$$

$$\theta_{f|e} \sim DP(\theta_{\mathrm{F}}, \alpha_j) \;\; \forall \text{ English phrase types } \bar{e},$$

where P_f and P_e are the flat distributions defined in Equations 4.6 and 4.7. This conditional HDP prior prefers shorter phrases in both the input and output sentences, while allowing for phrases of any length. The prior explicitly encourages the model to reuse output segments previously seen, rather than generating many phrases only once, and likewise encourages input phrases to be shared across different conditioning environments. This model was originally proposed in DeNero and Bouchard-Côté (2008). All three Bayesian models are compared in Figure 4.4

Again, our collapsed sampler can be adapted to draw samples under this distribution. The bispan potential $\psi([g:h) \Leftrightarrow [k:\ell))$ has the same form as the conditional model potential defined in Section 3.5.3. However, in the place of the fixed parameter

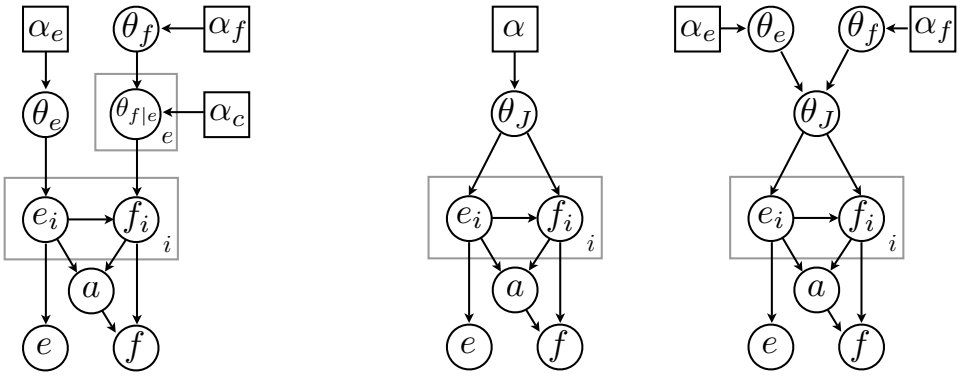


Figure 4.4: Three graphical models for the generation of a single sentence pair. The left model is the conditional hierarchical model presented in this paper. The plate with index i is over the phrase pairs of the sentence, the one with index e is over the infinite set of potential output phrase types. The center and right models are the two joint models over phrase pairs, with fixed and hierarchical base measures respectively.

$\theta(\bar{f}|\bar{e})$, we have a τ expression that conditions on the table assignments F^- and E^-.

$$\tau_{\mathrm{C}}(\bar{e} \Leftrightarrow \bar{f}|\mathrm{A}^-, E^-, F^-) = \tau_e(\bar{e}|E^-) \cdot \frac{\mathrm{N}(\bar{e} \Leftrightarrow \bar{f}, \mathrm{A}^-) + \alpha_c \cdot \tau_f(\bar{f}|F^-)}{|\mathrm{A}^-| + \alpha_j}$$

Above, τ_e and τ_f are defined identically to their joint model counterparts in Equations 4.14 and 4.15.

4.4 Bayesian Modeling Experiments

Having defined our models and collapsed sampling procedures, we now turn to the task of using the models' predictions in a machine translation system. The state-of-the-art baseline approach to scoring phrases is based on the relative frequency of aligned phrase pairs, as described in Chapter 2. That is, we define count$(\bar{e}, \bar{f})$ as the number of times that phrase type **e** aligned to phrase type **f** in our training corpus, and annotate phrase pairs with relative frequency features:

$$\mathrm{P}_{\mathrm{rel}}(\bar{f}|\bar{e}) = \frac{\mathrm{count}(\bar{e}, \bar{f})}{\sum_{\bar{e}'} \mathrm{count}(\bar{e}', \bar{f})}$$

$$\mathrm{P}_{\mathrm{rel}}(\bar{e}|\bar{f}) = \frac{\mathrm{count}(\bar{e}, \bar{f})}{\sum_{\bar{f}'} \mathrm{count}(\bar{e}, \bar{f}')}$$

We change this definition minimally by redefining $\mathrm{count}(\mathbf{e}, \mathbf{f})$ to be the expected number of times that $\mathbf{e}$ aligns to $\mathbf{f}$ under our model; approximated by our sampler.

4.4.1 Word-Alignment Baseline

We trained Moses on all Spanish-English Europarl sentences up to length 20 (177k sentences) using GIZA++ Model 4 word alignments and the *grow-diag-final-and* combination heuristic (Koehn *et al.*, 2007; Och and Ney, 2003; Koehn, 2002), which performed better than any alternative combination heuristic. The baseline estimates come from extracting phrases up to length 7 from the word alignment. We used a bidirectional lexicalized distortion model that conditioned on both Spanish and English phrases, along with their orientations. Our 5-gram language model was trained on 38.3 million words of Europarl using Kneser-Ney smoothing using the SRILM toolkit (Stolcke, 2002).

We tuned and tested on development corpora for the 2006 translation workshop. The parameters for each system were tuned separately using minimum error rate training (Och, 2003). Results are scored with lowercased, tokenized NIST BLEU (Papineni *et al.*, 2002), and exact match METEOR (Agarwal and Lavie, 2007).

The baseline system gives a BLEU score of 29.8, as shown in Table 4.1.

4.4.2 Joint Bayesian Model Performance

We initialized the sampler with a configuration derived from the word alignments generated by the baseline. We greedily constructed a phrase alignment from the word

Estimate	Phrase Pair Count	NIST BLEU	Exact Match meteor
Word Alignments	4.4M	29.8	52.4
Joint Model with DP Prior	3.7M	30.1	52.7
Joint Model with HDP Prior	3.1M	30.1	52.6

Table 4.1: BLEU results for learned distributions improve over a heuristic baseline.

alignment by identifying minimal phrase pairs consistent with the word alignment in each region of the sentence. We then ran the sampler for 100 iterations through the training data. Each iteration required 12 minutes under the DP prior, and 30 minutes under the HDP prior. Total running time for the HDP model neared two days on an eight-core machine with 16 Gb of RAM.

Both the DP and HDP estimators outperformed the baseline, even though the total number of phrase pairs decreased. The hierarchical prior does not improve performance over the non-hierarchical DP, but it does further reduce the number of phrase pairs. This model sparsity is desirable for efficiency reasons.

4.4.3 Conditional Bayesian Model Performance

The conditional model shares the same sampler and estimation procedure as the joint models. We compared them in a separate experiment; Table 4.2 shows that the conditional model slightly underperforms the joint models. The priors for these models share many components, and so it is not surprising that the models perform similarly. Note that due to various changes in our MT system over time, these results are not directly comparable with Table 4.1.

Model	Prior	BLEU Score
Joint Model	Dirichlet process prior	30.9
Joint Model	Hierarchical Dirichlet process	**31.0**
Conditional Model	Hierarchical Dirichlet process	30.8

Table 4.2: The joint phrase alignment model slightly outperforms the conditional model under Bayesian priors.

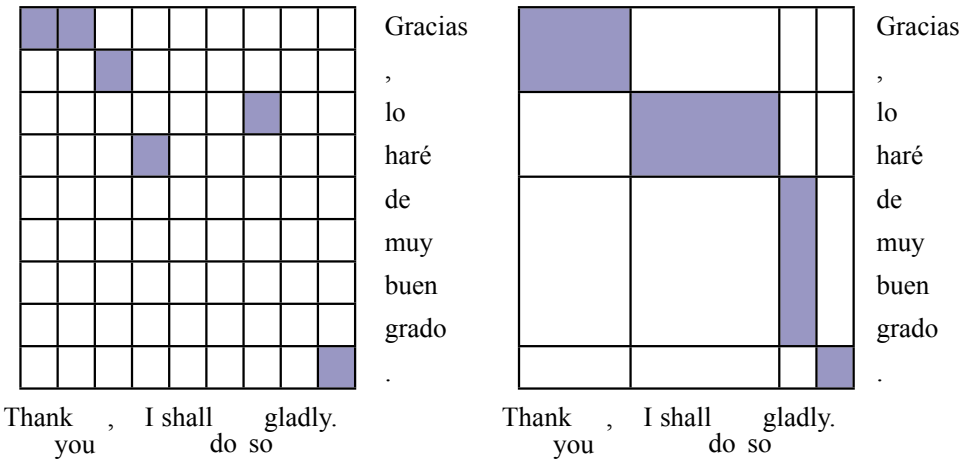
4.4.4 Summary of Experimental Findings

Bayesian models based on the Dirichlet Process, combined with a Gibbs sampler, allow phrase alignment models to outperform their word alignment counterparts while inducing sparser models, which simultaneously improves translation-time efficiency. Previously published positive results for phrase alignment models had always constrained the output of the phrase model by the predictions of a word-level model (Birch *et al.*, 2006; Cherry and Lin, 2007). These sampled models use word alignments for initialization—just as state-of-the-art word alignment models use simpler word-level models for initialization—but allow the phrase model to correct alignment errors. An example of a corrected error appears in Figure 4.5. Subsequent work on Bayesian alignment has upheld this result (Blunsom *et al.*, 2009).

4.4.5 Segmentation and Composition

Phrase alignment models are *segmented models*: they must choose some segmentation of their structured output that can either have many small components or few large ones. Simple likelihood objectives for segmented models have a strong bias toward using few segments if each segment is generated independently under the model. Therefore, it is no surprise in our case that phrase alignment models required a strong prior to promote the use of small, reusable components.

On the other hand, the success of phrase-based *translation* models is widely at-



(a) example word alignment **(b) example phrase alignment**

Figure 4.5: (a) An example word alignment from the baseline technique does not correctly analyze the multi-word idiomatic phrase *de muy buen grado*, which translates as *glady*. (b) Despite being initialized by this erroneous word alignment, the phrase alignment model recovers a correct analysis of the sentence pair.

tributed to their ability to detect and reuse long sequences of training sentence words in order to translate novel inputs. In the case of translation, reusing longer phrases is not only empirically effective but intuitively reasonable: any number of local agreement constraints, word choice biases, and reordering phenomena can be captured in together in a single long phrase.

The tension between the phrase alignment model's requirement to use small phrases and the translation model's requirement to use longer phrases was resolved via composed phrases (Section 4.2.7). That is, we gathered statistics about larger phrase pairs that were generated as multiple contiguous phrase pairs under the alignment model. While this solution is effective, it begs an additional question: how can we use statistics about longer phrase pairs to inform the phrase alignment. The independence assumptions of these generative models prevent us from sharing information across shorter and longer phrases, but the discriminative models in the following chap-

ters will enable us to consider composed and minimal phrases simultaneously during alignment.

Chapter 5

Discriminative Phrase Alignment

The previous two chapters investigated generative phrase alignment models. The objective function that guides learning in generative models is the probability of the observed training corpus, expressed either as a likelihood or a posterior given a prior. This objective does not reference the correctness of alignments at all — we only hypothesize that correct alignments will yield a high-probability explanation of the observed data. By contrast, the discriminative models investigated in this chapter allow us to define a learning objective based on a set of reference or gold alignments created by a human annotator. Rather than expecting correct alignments to emerge as latent aspects of a generative process, we will expect to learn correct alignments directly by correcting the alignment errors made by a model.

5.1 Discriminative Learning

Discriminative learning methods provide state-of-the-art performance in many structured prediction tasks in natural language processing, such as part-of-speech tagging (Lafferty *et al.*, 2001) and syntactic parsing (Huang, 2009). Moreover, the translation models that the MT community uses to generate final translation outputs are typi-

cally trained discriminatively. Discriminative linear models have replaced the more simplistic noisy channel model that dominated early statistical machine translation work, due to improved performance (Och, 2003).

Discriminative models directly predict the output y (in our case, an alignment) conditioned upon the observed input x (in our case, a sentence pair). In structured prediction tasks, discriminative models are typically trained to optimize a task-specific loss function $L(y_g, y)$, which measures the degree of error for a hypothesis y relative to a gold reference output y_g.

Linear discriminative models score hypotheses y as

$$\text{score}(y) = \theta \cdot \phi(y)$$

where $\phi(y)$ is a vector of real-valued features which expose the aspects of the structured output y that are relevant to the prediction task, and θ is a vector of model weights. The model can be used to predict an output y_m by maximizing this score:

$$y_m = \arg\max_y \theta \cdot \phi(y)$$

Discriminative learning procedures are often characterized as mistake-driven, as their objectives often include the highest scoring prediction y_m and its loss, which together characterizes the mistakes of the model.

Statistical learning theory provides a wide selection of discriminative learning techniques. In this thesis, we focus on a loss-augmented generalization of the classic perceptron algorithm, referred to as the margin-infused relaxed algorithm (Crammer and Singer, 2003).

5.1.1 Margin-Infused Relaxed Algorithm

The margin-infused relaxed algorithm (MIRA) is an online, margin-based learning method for linear models. MIRA has been applied to a variety of natural language processing tasks with consistent success, such as dependency parsing (McDonald *et al.*, 2005), labeled chunking (Shimizu and Haas, 2006), and machine translation (Watanabe, 2007; Chiang *et al.*, 2008).

Online learning algorithms consider one example at a time; in our case, one sentence pair $x = (\mathbf{e}, \mathbf{f})$ along with its reference alignment y_g. The model weights θ are updated with each example, and the resulting weights are then used for the following example. Online algorithms differ from batch learning algorithms, like the expectation maximization learning of Chapter 3, which apply the same model weights to many examples and then update model weights to optimize an objective function over the entire dataset.

Like the perceptron, MIRA only considers the reference y_g and the highest scoring prediction under the current model, y_m. MIRA updates θ away from the feature vector for the highest scoring prediction $\phi(y_m)$ and toward the feature vector for the reference output $\phi(y_g)$.

$$\theta \leftarrow \theta + \tau \cdot (\phi(y_g) - \phi(y_m)) \tag{5.1}$$

where τ is the minimal step size that ensures the resulting θ prefers y_g over y_m by a margin as large as the loss of y_m:

$$\theta \cdot \phi(y_g) \geq \theta \cdot \phi(y_m) + L(y_g, y_m)$$

The step size in the direction $\phi(y_g) - \phi(y_m)$ that achieves this margin condition

can be expressed in closed form:

$$s(y_g, y_m) = \frac{L(y_g, y_m) - \theta \cdot (\phi(y_g) - \phi(y_m))}{||\phi(y_g) - \phi(y_m)||_2^2}$$

Where the vector 2-norm above is defined as

$$||v||_2^2 = \sum_{i=1}^{|v|} v_i^2 \ .$$

To prevent any one example from influencing θ too much (for instance, due to a very high loss for some y_m), learning is regularized by capping the step size τ with some constant C:

$$\tau = \min(C, s(y_g, y_m)) \ .$$

The online update defined by Equation 5.1 is performed for each example k times. The training examples are presented in a random order for each of k passes through the training set. As an additional form of regularization, we average together the k weight vectors θ_i obtained after each pass through the training dataset. Experiments presented in this thesis set k to 30 and C to 0.01. These values were selected based on preliminary alignment experiments.

To apply MIRA to the problem of phrase alignment, we must supply the following components:

- A vector of features $\phi(y)$ that characterize a phrase alignment y.
- An inference procedure to produce $y_m = \arg\max_y \theta \cdot \phi(y)$.
- A reference feature extractor to give $\phi(y_g)$ from y_g.
- A loss function $L(y_g, y_m)$.

5.2 Previous Work on Discriminative Alignment

Discriminative methods have been shown to yield state-of-the-art performance for the task of word alignment (Moore *et al.*, 2006; Lacoste-Julien *et al.*, 2006). This section briefly reviews previous work on discriminative word alignment, focusing particular attention on its influence on the phrase alignment models we propose in the remainder of this chapter.

Several variants of discriminative aligners were proposed simultaneous, which varied in learning techniques, inference procedures, model factorization, and output constraints (Liu *et al.*, 2005; Ayan *et al.*, 2005; Taskar *et al.*, 2005; Moore, 2005; Ittycheriah and Roukos, 2005). Despite their variety, these aligners all shared the same input and output conditions. The discriminative models themselves map from an input sentence pair $(\mathbf{e}, \mathbf{f})$ to an output set of word alignment links $\mathcal{A}$.

Features are defined primarily on individual links $(i, j) \in \mathcal{A}$, but may also be defined on tuples of links $((i, j)_1, \ldots, (i, j)_k)$. Those features may refer to any part of the sentence pair. Features also leverage large parallel corpora that do not include reference alignments. These unsupervised corpora are used to collect surface statistics such as co-occurrence counts, as well as to train unsupervised word alignment models. The predictions of those unsupervised models can then be used as features in discriminative aligners. Unlike large-scale discriminative modeling efforts, which often rely on millions of low-occurrence but highly specific features to make predictions, the most successful discriminative aligners heavily leverage real-valued features that encode frequency patterns collected from these large unsupervised parallel corpora.

5.2.1 Perceptron-Trained Word Alignment

Among the most successful efforts in this space is a two-stage aligner trained with averaged perceptron (Moore, 2005; Moore *et al.*, 2006). The final version of this aligner

achieved state-of-the-art performance on a French-English parallel corpus excerpted from the Canadian Hansards corpus and hand aligned for use in evaluating alignment techniques (Och and Ney, 2003). This line of work contributed several findings that we leverage in our phrase aligner:

- While at-most-one-to-one or one-to-many constraints on alignments have proven computationally convenient for unsupervised models (Brown *et al.*, 1993; Melamed, 2000), they must be relaxed to maximize performance. Aligners benefit from being able to align multiple words from either sentence to a word or words in the other sentence.

- Achieving state-of-the-art performance from a supervised aligner requires using the predictions of an unsupervised aligner as input. Strong performance can be achieved through cleverly engineered features based on surface statistics and multi-stage alignment, but unsupervised models generate the best known features.

- The choice of learning technique has a small impact on performance relative to the feature representation, amount of data, and structural output restrictions used in the model.

The discriminative phrase aligners we develop in this chapter therefore rely on the predictions of unsupervised word alignment models, employ only a single learning technique (MIRA), and allow many-to-many word alignments in the form of phrase-to-phrase alignments.

5.2.2 Discriminative Inversion Transduction Grammars

Inference in the space of word alignments that include many-to-many alignments leads to challenging inference problems. For instance, Section 3.2.2 proved that finding the

highest scoring phrase alignment under an unrestricted phrase-factored model is NP-Hard.

A natural way to circumvent this hardness result is to restrict the space of phrase alignments to a phrasal inversion transduction grammar (ITG), as defined in Section 3.3.2. This alignment space admits polynomial time inference algorithms.

A discriminative alignment model based on ITG has also been proposed previously (Cherry and Lin, 2006). However, this model restricted the output to at-most-one-to-one word alignments. The next section extends this approach to phrase alignments.

5.3 Disjoint Phrase Alignment Models

We now define a discriminative model that conditions upon a sentence pair $(\mathbf{e}, \mathbf{f})$ and scores a phrasal ITG alignment $\mathcal{P} \in \text{ITG}(\mathbf{e}, \mathbf{f})$. $\mathcal{P}$ is a set of pairwise-disjoint bispans that partition each sentence, as in the phrase-factored models of Section 3.1. In this discriminative model, each bispan corresponds to a feature vector $\phi([g:h) \Leftrightarrow [k:\ell))$. The feature vector for a phrase alignment $\phi(\mathcal{P})$ is the sum of the feature vectors of its component bispans, which can be distilled into a single score via an inner product with weight vector θ.

$$\text{score}(\mathcal{P}) = \theta \cdot \sum_{[g:h) \Leftrightarrow [k:\ell) \in \mathcal{P}} \phi([g:h) \Leftrightarrow [k:\ell)) \ .$$

This disjoint phrase model is a featurized extension to the phrase-factored model class defined by Equation 3.1, linear both in the dimensions of ϕ and the elements of $\mathcal{P}$.

By construction, every phrase alignment $\mathcal{P} \in \text{ITG}(\mathbf{e}, \mathbf{f})$ corresponds to at least one binary synchronous tree T with a set of terminal productions $\text{trm}(T) = \mathcal{P}$. These synchronous trees play an important role in model inference for ITG. Extending our model to the space of trees, we define the score of a tree T in terms of its terminal

productions.

$$\text{score}(T) = \theta \cdot \sum_{[g:h) \Leftrightarrow [k:\ell) \in \text{trm}(T)} \phi([g:h) \Leftrightarrow [k:\ell)) \ .$$

While there may be many trees that correspond to the same $\mathcal{P}$, all trees with a common terminal set will have the same model score.

$$\text{trm}(T) = \text{trm}(T') \rightarrow \text{score}(T) = \text{score}(T') \ .$$

Because trees are scored entirely by their terminals, we may refer to some $\mathcal{P}$ and its corresponding T interchangeably in many contexts.

Finding the highest-scoring phrase alignment

$$\mathcal{P}_m = \arg\max_{\mathcal{P} \in \text{ITG}(\mathbf{e},\mathbf{f})} \theta \cdot \phi(\mathcal{P})$$

requires parsing under a max-sum semi-ring using a synchronous grammar that assigns zero weight to non-terminal productions, and a weight of $\theta \cdot \phi([g:h) \Leftrightarrow [k:\ell))$ to terminal productions. This inference procedure requires $O(n^6)$ time for a sentence pair $(\mathbf{e}, \mathbf{f})$ with maximum length $n = \max(|\mathbf{e}|, |\mathbf{f}|)$, as described in Section 3.3.2.

5.3.1 Word-Level Projection and Loss

We can interpret a phrase alignment as a word alignment by computing the set of word alignment links $\mathcal{A}$ that are contained within the bispans of $\mathcal{P}$:

$$\mathcal{A}(\mathcal{P}) = \bigcup_{[g:h) \Leftrightarrow [k:\ell) \in \mathcal{P}} \{(i,j) : i \in [g:h) \wedge j \in [k:\ell)\} \ . \tag{5.2}$$

Using this projection, we can define a word-level loss function for some $\mathcal{P}$ relative to any reference word alignment $\mathcal{A}_g$ that simply sums the number of missed sure

alignment links (recall errors) with the number of erroneous proposed links (precision errors). This loss function has been used regularly in state-of-the-art discriminative word aligners (Taskar *et al.*, 2005; Moore, 2005).

5.3.2 Features on Phrase Alignment Links

As with any discriminative alignment model, the effectiveness of this model depends largely on the set of features used. These features regularly refer to an unsupervised probabilistic aligner trained on a large parallel corpus that supplies feature information. We use the hidden Markov model aligner implemented in the Berkeley Aligner software package (Liang *et al.*, 2006; DeNero and Klein, 2007). We used the following features on a bispan $[g:h) \Leftrightarrow [k:\ell)$ in order to score our model.

Unsupervised Link Posteriors Let $P(i,j|\mathbf{e},\mathbf{f})$ be the posterior link probabilities of aligning i to j in an unsupervised model. Then, we average that value for all links included within the bispan $[g:h) \Leftrightarrow [k:\ell)$.

Bias The bias feature is 1 for all bispans.

Neighbor Posteriors The maximum posterior link probability in an unsupervised model of any (i,j) that is directly adjacent to $[g:h) \Leftrightarrow [k:\ell)$.

Extraction from Unsupervised Links An indicator of whether $[g:h) \Leftrightarrow [k:\ell)$ would be extractable from the full alignment predicted by an unsupervised model.

Identity Whether $e_{[g:h)}$ is identical to $f_{[k:\ell)}$ (often true for proper names, numbers, etc.).

Dictionary Whether $e_{[g:h)}$ appears as a translation of $f_{[k:\ell)}$ in a bilingual dictionary.

Our Chinese-English experiments, we used the Chinese-English dictionary distributed by the Linguistic Data Consortium.

Shape Indicator features for the size of the bispan in words (and in characters, for Chinese).

Numerical Whether $e_{[g:h)}$ and $f_{[k:\ell)}$ contain only numerical characters.

Punctuation Whether $e_{[g:h)}$ and $f_{[k:\ell)}$ contain only punctuation characters.

Lexical Indicator features on the lexical content of $e_{[g:h)}$ and $f_{[k:\ell)}$, but where all words except the k most common have been mapped to a single rare word token. In experiments we set $k = 50$, thereby storing lexical features for only the most common 50 words in each language.[1]

Fertility On 1-by-n or n-by-1 bispans, for $n > 1$, we include indicator features of the fertility of common words. This notion of fertility also appears in some generative word alignment models (Brown *et al.*, 1993).

Preliminary experiments indicated that all of these features improved the performance of the model.

5.3.3 Agenda-Based Inference and Pruning

While the space of phrasal ITG alignments admits a polynomial-time inference algorithm, the naive $O(n^6)$ dynamic program spends a great deal of time considering alternatives that can easily be ruled out by pruning the search space with a simpler model. An unsupervised model is required to supply features for our discriminative aligner, and so it is a natural source of pruning information as well.

[1] A larger number of lexical features have been shown to help in discriminative aligners (Moore *et al.*, 2006). However, incorporating them can require tweaks to the learning algorithm and staged training — complications that are orthogonal to the goals of this thesis.

We adopt a pruning scheme broadly similar to the one proposed in Cherry and Lin (2007). We refuse to consider any bispan $[g : h] \Leftrightarrow [k : \ell)$ that violates more than 3 alignment links produced by the unsupervised aligner.

Pruning has only a minimal effect on the maximum possible performance of our model. The oracle alignment error rate for the block ITG model class is 1.4%; the oracle alignment error rate for this pruned subset of ITG is 2.0%. Hence, even with a hard constraint based on word alignments, the right model can achieve a nearly perfect analysis of our hand-aligned test set. Maximum violation count constraints like this one are much less restrictive than the constraints described in Section 3.4.2, which disallowed violations entirely. We note that while each bispan can violate at most three links, the entire alignment can violate many more in total.

With this pruning approach, a large number of bispans are excluded. In addition to those bispans excluded directly by this pruning criterion, many large bispans are effectively eliminated simply because they cannot be constructed using the smaller bispans that remain after pruning. Nonetheless, a straightforward dynamic programming approach to parsing would systematically test all ways of building each non-excluded bispan, which can cause long running times despite the fact that the vast majority of the search space as been excluded via the pruning criterion.

To take advantage of the sparsity that results from pruning, we use an agenda-based parser that orders search states from small to large, where we define the size of a bispan as the total number of words contained within it:

$$\text{size}([g : h) \Leftrightarrow [k : \ell)) = h - g + \ell - k \ .$$

For each size, we maintain a separate agenda. Only when the agenda for size k is exhausted does the parser proceed to process the agenda for size $k + 1$. This parsing approach iterates topologically through spans from small to large, as would

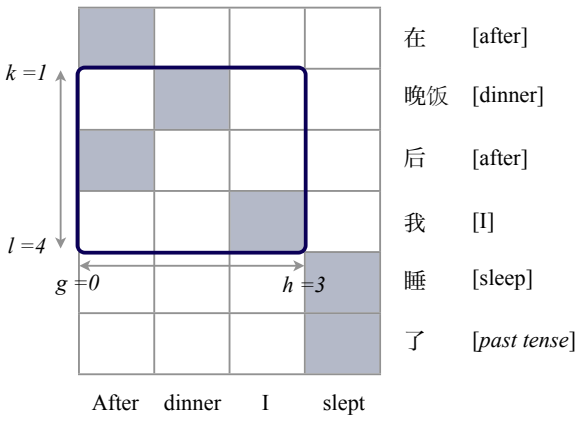


Figure 5.1: A* search for pseudo-gold ITG alignments uses an admissible heuristic for bispans that counts the number of gold links outside of $f_{[k:\ell)}$ but within $e_{[g:h)}$. Above, the heuristic is 1, which is also the minimal number of alignment errors that an ITG alignment will incur using this bispan.

a standard dynamic program, but differs in that it organizes computation around a list of successfully constructed bispans, rather than considering all possible bispans. Related work has demonstrated that sparsity in an ITG can also be exploited via a two-step dynamic program (Dyer, 2009).

5.3.4 Pseudo-Gold ITG Alignments

So far, we have supplied the following ingredients required by MIRA to train our discriminative model: a feature representation, a loss function, and an inference method for finding $\mathcal{A}_m$. We also need to supply a feature representation of $\mathcal{A}_g$, the reference alignment.

However, some hand-annotated alignments $\mathcal{A}_t$ are outside of the block ITG model class. Hence, we update toward the extraction set for a pseudo-gold alignment $\mathcal{A}_g \in$

$\text{ITG}(\mathbf{e}, \mathbf{f})$ with minimal distance from the true reference alignment $\mathcal{A}_t$.

$$\mathcal{A}_g = \arg\min_{\mathcal{A} \in \text{ITG}(\mathbf{e},\mathbf{f})} |\mathcal{A} \cup \mathcal{A}_t - \mathcal{A} \cap \mathcal{A}_t| \tag{5.3}$$

Equation 5.3 asks for the block ITG alignment $\mathcal{A}_g$ that is closest to a reference alignment $\mathcal{A}_t$, which may not lie in $\text{ITG}(\mathbf{e}, \mathbf{f})$. We search for $\mathcal{A}_g$ using A* bitext parsing (Klein and Manning, 2003). Each search state, which corresponds to some bispan $[g : h) \Leftrightarrow [k : \ell)$, is scored by the number of errors within the bispan plus the number of $(i, j) \in A_t$ such that $j \in [k : \ell)$ but $i \notin [g : h)$ (recall errors). As an admissible heuristic for the future cost of a bispan $[g : h) \Leftrightarrow [k : \ell)$, we count the number of $(i, j) \in \mathcal{A}_t$ such that $i \in [g : h)$ but $j \notin [k : \ell)$, as depicted in Figure 5.1. These links will become recall errors eventually. A* search with this heuristic makes no errors, and the time required to compute pseudo-gold alignments is negligible.

5.3.5 Using Disjoint Phrase Alignments for Translation

We have finished defining our discriminative model over disjoint ITG phrase alignments. Once we train such a model, we can query it for a phrase alignment $\mathcal{P}_m$ for each sentence pair in our parallel corpus. Using the word-level projection $\mathcal{A}(\mathcal{P})$ defined by Equation 5.2, we can integrate this aligner into a phrase-based pipeline in precisely the same manner as a word alignment model would be integrated. That is, we extract all phrase pairs up to size n that are licensed by the disjoint phrase alignment, $\text{R}_n(\mathcal{A}(\mathcal{P}_m))$, defined in Section 2.3.2.

Of course, $\mathcal{P} \subseteq \text{R}_n(\mathcal{A}(\mathcal{P}))$ for any well-formed phrase alignment $\mathcal{P}$, as phrase extraction will additionally include all of the possibly overlapping composed phrase pairs that can be formed by grouping together contiguous elements of $\mathcal{P}$, while $\mathcal{P}$ contains only a set of disjoint bispans that partition the sentence pair.

While both our alignment model and translation model have a bijective phrasal

structure, we still have a mismatch between our alignment model — which factors over disjoint bispans — and the phrase pairs that are tallied from the output alignment — which includes both large and small overlapping phrase pairs. Section 5.4 addresses this discrepancy.

5.4 Extraction Set Models

The baseline phrase pair extraction and scoring method described in Section 2.3 consists of a two-stage pipeline; a parallel corpus is aligned at the word level, and then phrase pairs are extracted from word-aligned sentence pairs. None of the models we have considered so far take into account the composed phrase pairs that will be extracted from a predicted alignment. Word alignment models consider only individual words. The generative phrase-factored and discriminative disjoint models considered in this thesis consider only minimal phrase pairs. As a result, these models cannot adjust their output based on composed phrase pairs, nor utilize features on composed phrase pairs to make alignment decisions.

In this section, we develop a model-based alternative to phrasal rule extraction, which merges the standard two-stage pipeline into a single step. We call the resulting approach an *extraction set* model, where an extraction set is the collection of phrase pairs $\mathrm{R}_n(\mathcal{A})$ that is extracted from an alignment $\mathcal{A}$.

Predicting extraction sets provides additional discriminative power relative to word aligners or minimal phrase aligners. Moreover, the structure of our model directly reflects the purpose of alignment models in a phrase-based translation pipeline, which is to discover sets of aligned phrase pairs that can be reused at translation time. Extraction sets also play a central role in hierarchical and syntactic translation systems. This relationship is investigated more fully in DeNero and Klein (2010), where extraction set models were originally proposed.

5.4.1 Extraction Set Definition

The input to our model is an unaligned sentence pair $(\mathbf{e}, \mathbf{f})$, and the output is an extraction set of phrasal translation rules $R_n(\mathcal{A})$, coupled with an underlying word alignment $\mathcal{A}$.

We restrict the model to propose alignments that correspond to a disjoint phrase alignment $\mathcal{A}(\mathcal{P})$, where $\mathcal{P} \in \text{ITG}(\mathbf{e}, \mathbf{f})$ and the conversion to a link set $\mathcal{A}$ is defined by Equation 5.2. In the context of an extraction set model, it is the word-to-word links rather than the phrasal bispans of $\mathcal{P}$ that dictate the set of extracted phrase pairs. Therefore, we view phrase alignments as word alignments and write $\mathcal{A} \in \text{ITG}(\mathbf{e}, \mathbf{f})$ to denote the set of word-level alignments permitted by the model.

We briefly review the rule extraction function $R_n(\cdot)$, which is described in Section 2.3.2. Consider an alignment $\mathcal{A}$. Let word e_i project to the phrasal span $\sigma(e_i)$, where

$$\sigma(e_i) = \left[\min_{j \in J_i} j \ , \ \max_{j \in J_i} j + 1 \right) \tag{5.4}$$

$$J_i = \{j : (i, j) \in \mathcal{A}\}$$

and likewise each word f_j projects to a span of $\mathbf{e}$. Then, $R_n(\mathcal{A})$ includes a bispan $[g : h) \Leftrightarrow [k : \ell)$ iff

$$\sigma(e_i) \subseteq [k : \ell) \qquad \forall i \in [g : h)$$

$$\sigma(f_j) \subseteq [g : h) \qquad \forall j \in [k : \ell)$$

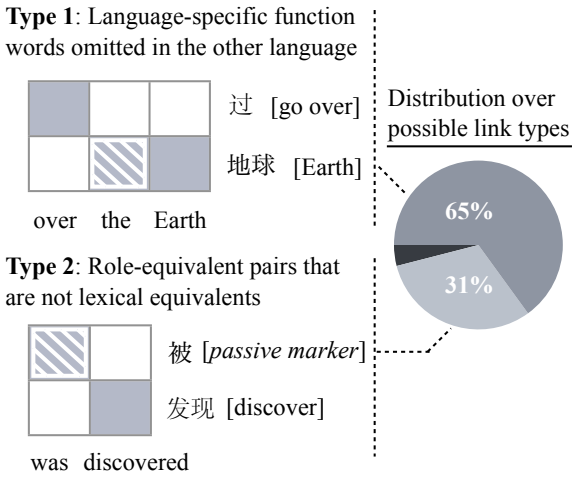


Figure 5.2: Examples of two types of possible alignment links (striped). These types account for 96% of the possible alignment links in our data set.

5.4.2 Possible and Null Alignment Links

We have not yet accounted for two special cases in annotated corpora: *possible* alignments and *null* alignments. To analyze these annotations, we consider a particular data set: a hand-aligned portion of the NIST MT02 Chinese-to-English test set, which has been used in previous alignment experiments (Ayan *et al.*, 2005; DeNero and Klein, 2007; Haghighi *et al.*, 2009).

Possible links account for 22% of all alignment links in these data, and we found that most of these links fall into two categories. First, possible links are used to align function words that have no equivalent in the other language, but colocate with aligned content words, such as English determiners. Second, they are used to mark pairs of words or short phrases that are not lexical equivalents, but which play equivalent roles in each sentence. Figure 5.2 shows examples of these two use cases, along with their corpus frequencies.[2]

[2]We collected corpus frequencies of possible alignment link types ourselves on a sample of the hand-aligned data set.

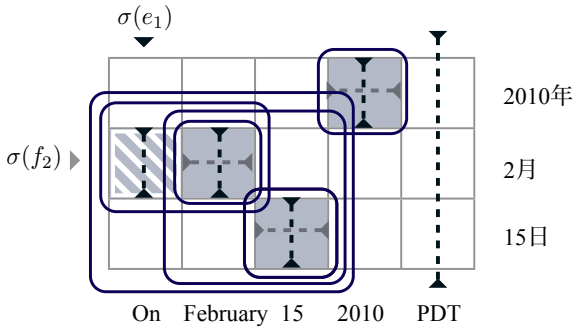


Figure 5.3: Possible links constrain the word-to-phrase projection of otherwise unaligned words, which in turn license overlapping phrases. In this example, $\sigma(f_2) = [1:2)$ does not include the possible link at $(1, 0)$ because of the sure link at $(1, 1)$, but $\sigma(e_1) = [1:2)$ does use the possible link because it would otherwise be unaligned. The word "PDT" is null aligned, and so its projection $\sigma(e_4) = [-1:4)$ extends beyond the bounds of the sentence, excluding "PDT" from all phrase pairs.

On the other hand, null alignments are used sparingly in our annotated data. More than 90% of words participate in some alignment link. The unaligned words typically express content in one sentence that is absent in its translation.

Figure 5.3 illustrates how we interpret possible and null links in our projection. The notion of possible links is typically not included in extraction procedures because most aligners predict only sure links. However, we see a natural interpretation for possible links in rule extraction: they license phrasal rules that both include and exclude them. We exclude null alignments from extracted phrases because they often indicate a mismatch in content.

We achieve these effects by redefining the projection operator σ. Let $\mathcal{A}^{(s)}$ be the subset of $\mathcal{A}$ that are *sure* links, then let the index set J_i used for projection σ in

Equation 5.4 be

$$J_i = \begin{cases} \{j : (i,j) \in \mathcal{A}^{(s)}\} & \text{if } \exists j : (i,j) \in \mathcal{A}^{(s)} \\ \{-1, |\mathbf{f}|\} & \text{if } \nexists j : (i,j) \in \mathcal{A} \\ \{j : (i,j) \in \mathcal{A}\} & \text{otherwise} \end{cases}$$

Here, J_i is a set of integers, and $\sigma(e_i)$ for null aligned e_i will be $[-1 : |\mathbf{f}| + 1)$ by Equation 5.4.

Of course, the characteristics of our aligned corpus may not hold for other annotated corpora or other language pairs. However, we hope that the overall effectiveness of our modeling approach will influence future annotation efforts to build corpora that are consistent with this interpretation.

5.4.3 A Linear Model of Extraction Sets

We now define a linear model that scores extraction sets. We restrict our model to score only *coherent* extraction sets $R_n(\mathcal{A})$, those that are licensed by an underlying word alignment $\mathcal{A}$ with sure alignments $\mathcal{A}^{(s)} \subseteq \mathcal{A}$. Conditioned on a sentence pair $(\mathbf{e}, \mathbf{f})$ and maximum phrase length n, we score extraction sets via a feature vector $\phi(\mathcal{A}^{(s)}, R_n(\mathcal{A}))$ that includes features on sure links $(i,j) \in \mathcal{A}^{(s)}$ and features on the bispans in $R_n(\mathcal{A})$ that link $[g : h)$ in $\mathbf{e}$ to $[k : \ell)$ in $\mathbf{f}$:

$$\phi(\mathcal{A}^{(s)}, R_n(\mathcal{A})) = \sum_{(i,j) \in \mathcal{A}^{(s)}} \phi_a(i,j) \quad + \sum_{[g:h) \Leftrightarrow [k:\ell) \in R_n(\mathcal{A})} \phi_b([g:h) \Leftrightarrow [k:\ell))$$

Because the projection operator $R_n(\cdot)$ is a deterministic function, we can abbreviate $\phi(\mathcal{A}^{(s)}, R_n(\mathcal{A}))$ as $\phi(\mathcal{A})$ without loss of information, although we emphasize that $\mathcal{A}$ is a set of sure and possible alignments, and $\phi(\mathcal{A})$ does not decompose as a sum of vectors on individual word-level alignment links. Our model is parameterized by a

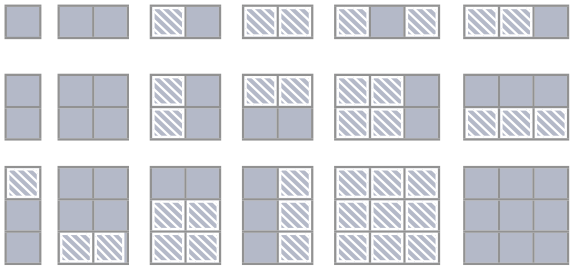


Figure 5.4: Above, we show a representative subset of the block alignment patterns that serve as terminal productions of the ITG that restricts the output space of our model. These terminal productions cover up to $n = 3$ words in each sentence and include a mixture of sure (filled) and possible (striped) word-level alignment links.

weight vector θ, which scores an extraction set $R_n(\mathcal{A})$ as $\theta \cdot \phi(\mathcal{A})$.

In summary, our model scores all $R_n(\mathcal{A})$ for $\mathcal{A} \in \text{ITG}(\mathbf{e}, \mathbf{f})$ where $\mathcal{A}$ can include block terminals of size up to n. In our experiments, $n = 3$. Unlike previous work, we allow possible alignment links to appear in the block terminals, as depicted in Figure 5.4. We do not include features on possible links (although adding such features would be straightforward). Instead, the desirability of possible links is assessed indirectly via the phrasal rules they introduce.

5.4.4 Extraction Set Loss Function

In order to focus learning on predicting the right bispans, we use an extraction-level loss $L(\mathcal{A}_g, \mathcal{A}_m)$: an F-measure of the overlap between bispans in $R_n(\mathcal{A}_m)$ and $R_n(\mathcal{A}_g)$. This measure has been proposed previously to evaluate alignment systems (Ayan and Dorr, 2006). Based on preliminary translation results during development, we chose

bispan F_5 as our loss:

$$\mathrm{Pr}(\mathcal{A}_g, \mathcal{A}_m) = |R_n(\mathcal{A}_g) \cap R_n(\mathcal{A}_m)|/|R_n(\mathcal{A}_m)|$$

$$\mathrm{Rc}(\mathcal{A}_g, \mathcal{A}_m) = |R_n(\mathcal{A}_g) \cap R_n(\mathcal{A}_m)|/|R_n(\mathcal{A}_g)|$$

$$\mathrm{F}_5(\mathcal{A}_g, \mathcal{A}_m) = \frac{(1+5^2) \cdot \mathrm{Pr}(\mathcal{A}_g, \mathcal{A}_m) \cdot \mathrm{Rc}(\mathcal{A}_g, \mathcal{A}_m)}{5^2 \cdot \mathrm{Pr}(\mathcal{A}_g, \mathcal{A}_m) + \mathrm{Rc}(\mathcal{A}_g, \mathcal{A}_m)}$$

$$L(\mathcal{A}_g, \mathcal{A}_m) = 1 - \mathrm{F}_5(\mathcal{A}_g, \mathcal{A}_m)$$

F_5 favors recall over precision. Previous alignment work has shown improvements from adjusting the F-measure parameter (Fraser and Marcu, 2006). In particular, Lacoste-Julien *et al.* (2006) also chose a recall-biased objective.

Optimizing for a bispan F-measure penalizes alignment mistakes in proportion to their rule extraction consequences. That is, adding a word link that prevents the extraction of many correct phrasal rules, or which licenses many incorrect rules, is strongly discouraged by this loss.

5.4.5 Additional Features on Extraction Sets

The output space of our extraction set model includes an underlying minimal phrase alignment, and so we can use the same set of features in our extraction set model that we used in our minimal phrase alignment model.

In addition, extraction set models allow us to incorporate the same phrasal relative frequency statistics that drive phrase-based translation performance, described in Section 2.3.3. To implement these frequency features, we score phrase pairs according to the baseline procedure described in Section 2.3 using the same unsupervised model that we use for features and pruning. Then, we score phrase pairs in an extraction set using the resulting relative frequency features.

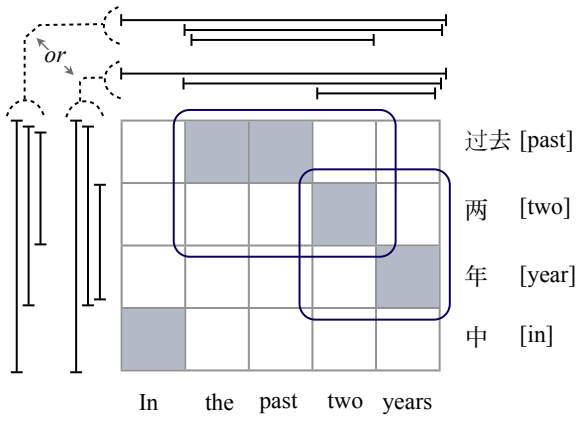


Figure 5.5: Both possible ITG decompositions of this example alignment will split one of the two highlighted bispans across constituents.

We also include monolingual phrase features that expose useful information to the model. For instance, in Chinese-English parallel corpora, English bigrams beginning with "the" are often extractable phrases: they align contiguously to the Chinese sentence. English trigrams with a hyphen as the second word are typically extractable, meaning that the first and third words align to consecutive Chinese words. When any conjugation of the word "to be" is followed by a verb, indicating passive voice or progressive tense, the two words tend to align together.

In total, our final model includes 4,249 individual features, dominated by various instantiations of lexical templates.

5.4.6 Extraction Set Inference

To train and apply our model, we must find the highest scoring extraction set under our model, $R_n(\mathcal{A}_m)$, which we also require at test time. Although we have restricted $\mathcal{A}_m \in \text{ITG}(\mathbf{e}, \mathbf{f})$, our extraction set model does not factor over ITG productions, and so the dynamic program for a vanilla block ITG will not suffice to find $\arg\max_{\mathcal{A}} \theta \cdot \phi(\mathcal{A})$. To see this, consider the extraction set in Figure 5.5. An ITG decomposition of the

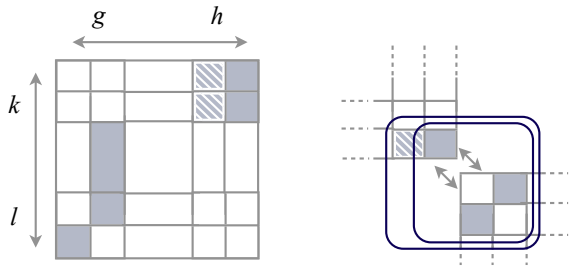


Figure 5.6: Augmenting the ITG grammar states with the alignment configuration in an $n-1$ deep perimeter of the bispan allows us to score all overlapping phrasal rules introduced by adjoining two bispans. The state must encode whether a sure link appears in each edge column or row, but the specific location of edge links is not required.

underlying alignment imposes a hierarchical bracketing on each sentence, and some bispan in the extraction set for this alignment will cross any such bracketing. Hence, the score of some licensed bispan will be non-local to the ITG decomposition.

If we treat the maximum phrase length n as a fixed constant, then we can define a polynomial-time dynamic program to search the space of extraction sets. An ITG derivation for some alignment $\mathcal{A}$ decomposes into two sub-derivations for $\mathcal{A}_L$ and $\mathcal{A}_R$.[3] The model score of $\mathcal{A}$, which scores extraction set $R_n(\mathcal{A})$, decomposes over $\mathcal{A}_L$ and $\mathcal{A}_R$, along with any phrasal bispans licensed by adjoining $\mathcal{A}_L$ and $\mathcal{A}_R$.

$$\theta \cdot \phi(\mathcal{A}) = \theta \cdot \phi(\mathcal{A}_L) + \theta \cdot \phi(\mathcal{A}_R) + I(\mathcal{A}_L, \mathcal{A}_R)$$

where $I(\mathcal{A}_L, \mathcal{A}_R)$ is $\theta \cdot \sum \phi(g, h, k, l)$ summed over licensed bispans $e_{[g:h)} \Leftrightarrow f_{[k:\ell)}$ that overlap the boundary between $\mathcal{A}_L$ and $\mathcal{A}_R$.[4]

In order to compute $I(\mathcal{A}_L, \mathcal{A}_R)$, we need certain information about the alignment configurations of $\mathcal{A}_L$ and $\mathcal{A}_R$ where they adjoin at a corner. The state must represent

[3] We abuse notation in conflating an alignment $\mathcal{A}$ with its derivation. All derivations of the same alignment receive the same score, and we only compute the max, not the sum.

[4] We focus on the case of adjoining two aligned bispans. Our algorithm easily extends to include null alignments, but we focus on the non-null setting for simplicity.

(a) the specific alignment links in the $n-1$ deep corner of each $\mathcal{A}$, and (b) whether any sure alignments appear in the rows or columns extending from those corners.[5] With this information, we can infer the bispans licensed by adjoining $\mathcal{A}_L$ and $\mathcal{A}_R$, as in Figure 5.6.

Applying our score recurrence yields a polynomial-time dynamic program. This dynamic program is an instance of ITG bitext parsing, where the grammar uses symbols to encode the alignment contexts described above. This context-as-symbol augmentation of the grammar is similar in character to augmenting symbols with lexical items to score language models during hierarchical decoding (Chiang, 2007).

5.4.7 Coarse-to-Fine Inference and Pruning

As with the disjoint phrase alignment model of Section 5.3, we can benefit from pruning using the predictions of the underlying unsupervised word alignment model. Again, we generate phrase alignments using an agenda-based ITG parser to take advantage of the sparsity of legal alignments under our pruning criterion.

We also employ coarse-to-fine search to speed up inference (Charniak and Caraballo, 1998). In the coarse pass, we search over the space of ITG alignments, but score only features on alignment links and bispans that are local to terminal productions. This simplification eliminates the need to augment grammar symbols, and so we can exhaustively explore the (pruned) space. We then compute outside scores for bispans under a max-sum semiring (Goodman, 1996). In the fine pass with the full extraction set model, we impose a maximum beam size of 10,000 hypotheses for each agenda. We order states on agendas by the sum of their inside score under the full model and the outside score computed in the coarse pass, pruning all states not within the fixed agenda beam size.

[5] The number of configuration states does not depend on the size of $\mathcal{A}$ because corners have fixed size, and because the position of links within rows or columns is not needed.

Search states that are popped off agendas are indexed by their corner locations for fast look-up when constructing new states. For each corner and size combination, built states are maintained in sorted order according to their inside score. This ordering allows us to stop combining states early when the results are falling off the agenda beams. Similar search and beaming strategies appear in many decoders for machine translation (Huang and Chiang, 2007; Koehn and Haddow, 2009; Moore and Quirk, 2007b).

5.4.8 Relationship to Previous Work

Our extraction set model is similar in design to the discriminative word alignment models described in Section 5.2. However, our model is the first to consider the output space of extraction sets. Two related lines of work also support our hypothesis that considering the overlapping structure of extraction sets should improve alignment quality for machine translation.

Kääriäinen (2009) trains a *translation* model discriminatively using features on overlapping phrase pairs. That work differs from ours in that it uses fixed word alignments and focuses on translation model estimation, while we focus on alignment and translate using standard relative frequency estimators.

Deng and Zhou (2009) present an alignment combination technique that uses phrasal features. Our approach differs in two ways. First, their approach is tightly coupled to the input alignments, while we perform a full search over the space of ITG alignments. Also, their approach uses greedy search, while our search is optimal aside from pruning and beaming. Despite these differences, their strong results reinforce our claim that phrase-level information is useful for alignment.

5.5 Experimental Results

We evaluate our extraction set model by the bispans it predicts, the word alignments it generates, and the translations generated by an end-to-end phrase-based system. Table 5.1 compares the four systems described below, including an unsupervised baseline. All supervised aligners were optimized for bispan F_5.

Unsupervised Baseline: Joint HMM. We trained and combined two HMM alignment models (Ney and Vogel, 1996) using the Berkeley Aligner.[6] We initialized the HMM model parameters with jointly trained Model 1 parameters (Liang *et al.*, 2006), combined word-to-word posteriors by averaging (soft union), and decoded with the competitive thresholding heuristic of DeNero and Klein (2007), yielding a state-of-the-art unsupervised baseline.

Disjoint phrase alignment model. We discriminatively trained a phrasal ITG aligner with only sure links, using block terminal productions up to 3 words by 3 words in size. This supervised baseline is a reimplementation of the MIRA-trained model of Haghighi *et al.* (2009). We use the same features and parser implementation for this model as we do for our extraction set model to ensure a clean comparison. To remain within the alignment class, MIRA updates this model toward a pseudo-gold alignment with only sure links. This model does not score any overlapping bispans.

Extraction Set Coarse Pass. We add possible links to the output of the block ITG model by adding the mixed terminal block productions described in Section 5.4.3. This model scores overlapping phrasal rules contained within terminal blocks that result from including or excluding possible links. However, this model does not score bispans that cross bracketing of ITG derivations.

Full Extraction Set Model. Our full model includes possible links and features

[6] http://code.google.com/p/berkeleyaligner

on extraction sets for phrasal bispans with a maximum size of 3. Model inference is performed using the coarse-to-fine scheme described in Section 5.4.7.

5.5.1 Data

In this experiment, we focus exclusively on Chinese-to-English translation. We performed our discriminative training and alignment evaluations using a hand-aligned portion of the NIST MT02 test set, which consists of 150 training and 191 test sentences (Ayan and Dorr, 2006). We trained the baseline HMM on 11.3 million words of FBIS newswire data, a comparable dataset to those used in previous alignment evaluations on our test set (DeNero and Klein, 2007; Haghighi *et al.*, 2009).

Our end-to-end translation experiments were tuned and evaluated on sentences up to length 40 from the NIST MT04 and MT05 test sets. For these experiments, we trained on a 22.1 million word parallel corpus consisting of sentences up to length 40 of newswire data from the GALE program, subsampled from a larger data set to promote overlap with the tune and test sets. This corpus also includes a bilingual dictionary. To improve performance, we retrained our aligner on a retokenized version of the hand-annotated data to match the tokenization of our corpus.[7] We trained a language model with Kneser-Ney smoothing on 262 million words of newswire using SRILM (Stolcke, 2002).

5.5.2 Word and Phrase Alignment

The first panel of Table 5.1 gives a word-level evaluation of all four aligners. We use the alignment error rate (AER) measure: precision is the fraction of sure links in the system output that are sure or possible in the reference, and recall is the fraction of sure links in the reference that the system outputs as sure. For this evaluation,

[7] All alignment results are reported under the annotated data set's original tokenization.

	Word			Bispan				Moses
	Pr	Rc	AER	Pr	Rc	F_1	F_5	BLEU
Unsupervised HMM	84.0	76.9	19.6	69.5	59.5	64.1	59.9	33.2
Minimal Phrase ITG	83.4	83.8	16.4	**75.8**	62.3	68.4	62.8	33.6
Extraction Set Coarse	82.2	**84.2**	16.9	70.0	72.9	71.4	72.8	34.2
Extraction Set Full	**84.7**	84.0	**15.6**	69.0	**74.2**	**71.6**	**74.0**	**34.4**

Table 5.1: Experimental results demonstrate that the full extraction set model outperforms supervised and unsupervised baselines in evaluations of word alignment quality, extraction set quality, and translation. In the *BLEU* evaluation, all systems used a bilingual dictionary included in the training corpus. The *BLEU* evaluation of supervised systems also included rule counts from the Joint HMM to compensate for parse failures.

possible links produced by our extraction set models are ignored. The full extraction set model performs the best by a small margin, although it was not tuned for word alignment.

The second panel evaluates bispan precision and recall, corresponding to the extraction set loss function described in Section 5.4.4.[8] To compete fairly, all models were evaluated on the full extraction sets induced by the word alignments they predicted. Again, the extraction set model outperformed the baselines, particularly on the F_5 measure for which the discriminative aligners were trained.

Our coarse pass extraction set model performed nearly as well as the full model. We believe these models perform similarly for two reasons. First, most of the information needed to predict an extraction set can be inferred from word links and phrasal rules contained within ITG terminal productions. Second, the coarse-to-fine inference may be constraining the full phrasal model to predict similar output to the coarse model. This similarity persists in translation experiments.

[8]While pseudo-gold approximations to the annotation were used for training, the evaluation is always performed relative to the original human annotation.

5.5.3 Translation Experiments

We evaluate the alignments predicted by our model using Moses, a state-of-the-art phrase-based MT system with lexicalized phrasal reordering (Koehn *et al.*, 2007). Moses is an open-source implementation of the statistical MT approach described in Chapter 2.

Moses does not accept possible links as input, although possible links do exist in the output of our extraction set aligners. To interface Moses with our extraction set models, we produced three sets of sure-only alignments from our model predictions: one that omitted possible links, one that converted all possible links to sure links, and one that includes each possible link with 0.5 probability. These three sets were aggregated and rules were extracted from all three.

The training set we used for MT experiments is quite heterogenous and noisy compared to our alignment test sets, and the supervised aligners did not handle certain sentence pairs in our parallel corpus well. In some cases, pruning based on consistency with the HMM caused parse failures, which in turn caused training sentences to be skipped. To account for these issues, we added counts of phrasal rules extracted from the baseline HMM to the counts produced by supervised aligners.

Our extraction set model predicts the set of phrases extracted by the Moses pipeline, and so the estimation techniques for the alignment model and translation model both share a common underlying representation: extraction sets. Empirically, we observe a BLEU score improvement of 1.2 over the unsupervised baseline and 0.8 over the block ITG supervised baseline. This substantial improvement in translation quality indicates that discriminative extraction set models represent an effective approach to alignment.

5.5.4 Analysis

The alignments predicted by the best performing extraction set model differ systematically from the predictions of the unsupervised baseline model. The most prominent change is structural: our extraction set model never allows a non-contiguous set of words to align to a single word. Such an event is rare, but not forbidden, in the baseline aligner. Of course, heuristic techniques like competitive thresholding can also enforce phrasal contiguity, but are not sensitive to phrasal statistics (DeNero and Klein, 2007).

Additionally, extraction set models correctly align many function words that are incorrectly analyzed by word alignment models. Function words are closed-class lexical items in a language that typically play a structural role, such as determiners, prepositions, and auxiliary verbs. These words are so common, often occurring multiple times in a sentence, that bilexical statistics do not reliably resolve their alignments. Consider the example in Figure 5.7. In the unsupervised baseline aligner, all instances of the word "the" are unaligned, along with one comma and the prepositions "for" and "to". The extraction set aligner corrects all of these omissions.

However, the extraction set model does not eliminate all errors in the baseline model. In particular, proper names (e.g., "FIFA") and multi-word expressions (e.g., "mid-size") are often mis-analyzed because our phrasal features fail to identify their translations. Features that identify proper names and compound expressions, perhaps by incorporating additional sources of information, would likely address these errors. Of course, a certain small fraction of errors are due to the structural restrictions of the ITG model class; only relaxing the model or allowing post-processing would resolve these errors.

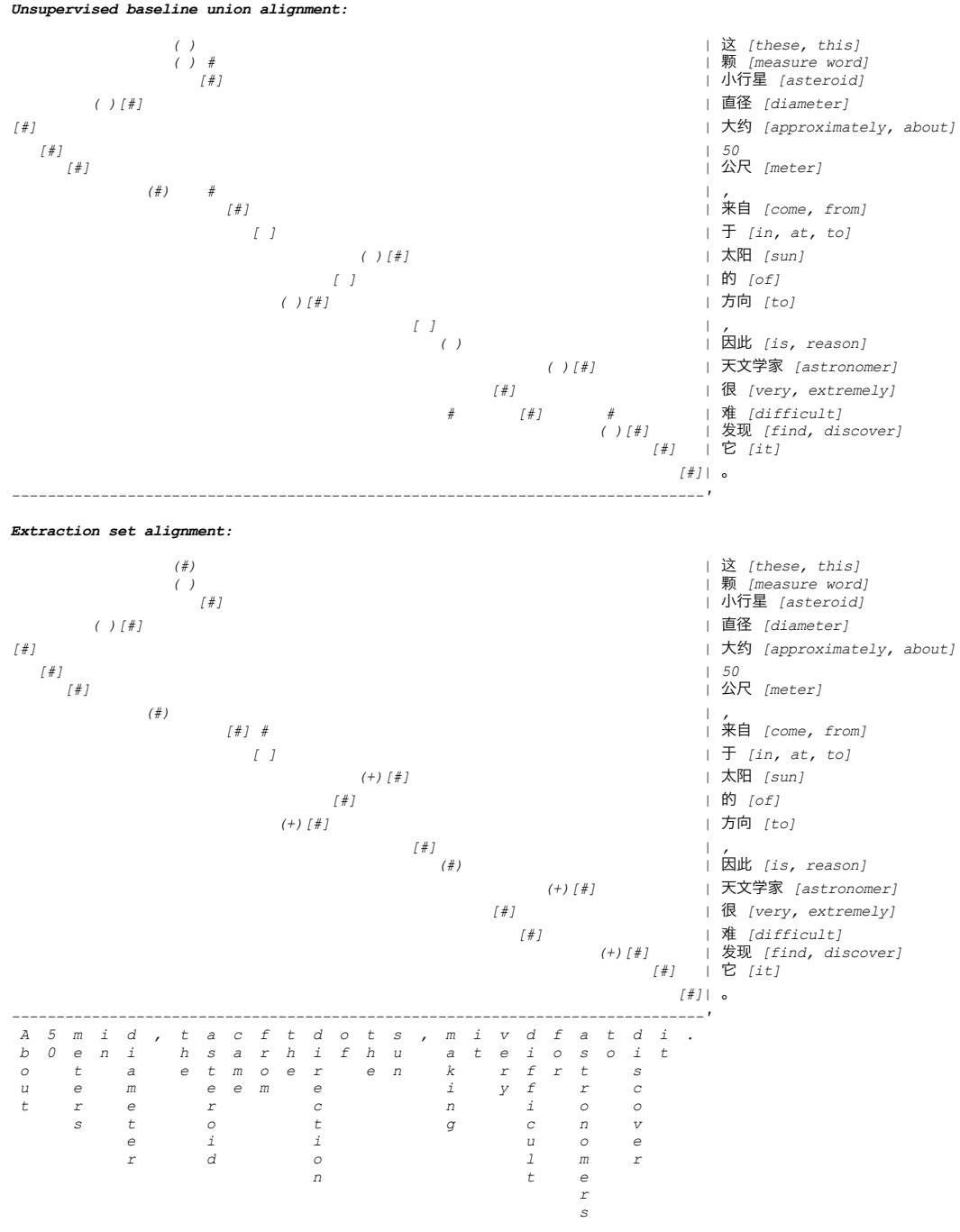


Figure 5.7: An example alignment from the unsupervised baseline (top) and the extraction set model (bottom). Hand annotated correct alignments are marked as [ ] for sure alignments and ( ) for possible alignments. Sure predictions are marked with #, and possible predictions are marked with +.

Chapter 6

Conclusion

This thesis has investigated many aspects of phrase alignment models for machine translation, including inference algorithms, generative models, Bayesian priors, and discriminative models. We found that phrase alignment models can offer quality improvements over state-of-the-art word-alignment alternatives. This conclusion comes as no surprise—the purpose of alignment models in a phrase-based translation pipeline is to identify phrasal translation rules, and considering phrasal patterns while predicting alignments has the capacity to improve alignment quality.

However, realizing these benefits required identifying and addressing a range of learning and inference challenges that arise in phrase alignment models.

1. The fundamental inference procedures of computing maximal alignments and bispan expectations both proved to be computationally intractable. However, pruned dynamic programs and sampling procedures allow for these inference problems to be solved quickly in practice.

2. Maximum likelihood estimators for both joint and conditional models of phrase alignment lead to degenerate parameter estimates that do not provide useful alignment output. However, Bayesian priors based on the Dirichlet process suc-

cessfully suppress problematic parameter values, providing translation quality improvements and sparser models relative to a state-of-the-art word alignment baseline.

3. Phrase factored models diverge structurally from the extraction sets used for translation model feature estimation, because they do not score composed phrase pairs. Extraction set models relax the independence assumption of phrase-factored models. Unifying the model structure used for alignment and translation also improved translation quality.

After addressing these challenges and others, our original hypothesis proved true: modeling the phrasal correspondence between languages does improve alignment quality. Moreover, these alignment improvements provide translation quality improvements in state-of-the-art, end-to-end, phrase-based machine translation systems.

These improvements came from both Bayesian generative models (Chapter 4) and discriminative extraction set models (Chapter 5). These two approaches differ primarily in their data conditions: generative phrase alignment models can be trained directly from parallel corpora, while discriminative models require a small hand-aligned dataset. Discriminative models generally outperform generative models in alignment because they incorporate this additional data source. However, generative models are preferred in cases where hand-aligned data is not available.

The methods also differ on other dimensions:

1. The Bayesian phrase alignment models defined in this thesis do not contain a notion of composed phrase pairs because they factor over a phrasal segmentation of each sentence. On the other hand, our extraction set model simultaneously incorporate statistics about minimal and composed phrase pairs to make alignment decisions.

2. Bayesian models are more computationally expensive to employ, because sampling requires iterating many times over a large parallel corpus. Discriminative training focuses only on a small hand-aligned dataset, and training time is dominated by computing features based on word alignment models.

3. However, generative models adapt naturally to parallel corpora that contain data from disparate domains because they are estimated on a full corpus. Discriminative models can easily overfit the small hand-aligned dataset on which they are trained. In fact, we achieved our best results in Chapter 5 by including both the discriminatively trained alignments from our extraction set model and generatively trained alignments from the baseline word aligner in a phrase-based MT system.

In summary, data conditions and computational resources will determine what method is appropriate in a given statistical MT system. These differences also highlight new challenges. In future work, we hope to incorporate statistical information about composed phrase pairs into generative models, scale Bayesian model inference to larger corpora, and adapt discriminative models to out-of-domain data. These improvements would certainly promote more wide-spread adoption of phrase alignment methods in statistical machine translation.

Bibliography

(Agarwal and Lavie, 2007) Abhaya Agarwal and Alon Lavie. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the ACL Workshop on Statistical Machine Translation*, 2007.

(Aldous, 1985) David Aldous. Exchangeability and related topics. In *École d'été de probabilitiés de Saint-Flour*, Berlin, 1985. Springer.

(Ayan and Dorr, 2006) Necip Fazil Ayan and Bonnie J. Dorr. Going beyond AER: An extensive analysis of word alignments and their impact on MT. In *Proceedings of the Association for Computational Linguistics*, 2006.

(Ayan *et al.*, 2005) Necip Fazil Ayan, Bonnie J. Dorr, and Christof Monz. Neuralign: combining word alignments using neural networks. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005.

(Berg, 2004) Bernd A. Berg. *Markov Chain Monte Carlo Simulations and Their Statistical Analysis*. World Scientific, Singapore, 2004.

(Birch *et al.*, 2006) Alexandra Birch, Chris Callison-Burch, and Miles Osborne. Constraining the phrase-based, joint probability statistical translation model. In *Proceedings of the Conference for the Association for Machine Translation in the Americas*, 2006.

(Blackwood *et al.*, 2008) Graeme Blackwood, Adri de Gispert, and William Byrne. Phrasal segmentation models for statistical machine translation. In *Proceedings of the International Conference on Computational Linguistics*, 2008.

(Blunsom and Osborne, 2008) Phil Blunsom and Miles Osborne. Probabilistic inference for machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008.

(Blunsom *et al.*, 2009) Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. A Gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of the Association for Computational Linguistics*, 2009.

(Brants *et al.*, 2007) Thorsten Brants, Ashok C. Popat, Peng Xu, Franz Josef Och, and Jeffrey Dean. Large language models in machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2007.

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

(Carpuat and Wu, 2007) Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2007.

(Charniak and Caraballo, 1998) Eugene Charniak and Sharon Caraballo. New figures of merit for best-first probabilistic chart parsing. In *Computational Linguistics*, 1998.

(Cherry and Lin, 2006) Colin Cherry and Dekang Lin. Soft syntactic constraints for word alignment through discriminative training. In *Proceedings of the Association for Computational Linguistics*, 2006.

(Cherry and Lin, 2007) Colin Cherry and Dekang Lin. Inversion transduction grammar for joint phrasal translation modeling. In *Proceedings of the NAACL-HLT Workshop on Syntax and Structure in Statistical Translation*, 2007.

(Chiang *et al.*, 2008) David Chiang, Yuval Marton, and Philip Resnik. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008.

(Chiang, 2007) David Chiang. Hierarchical phrase-based translation. *Computational Linguistics*, 2007.

(Crammer and Singer, 2003) Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003.

(DeNero and Bouchard-Côté, 2008) John DeNero and Alexandre Bouchard-Côté. A hierarchical Dirichlet process prior for a conditional model of phrase alignment. In *NIPS Workshop on Unsupervised Models in Natural Language Processing*, 2008.

(DeNero and Klein, 2007) John DeNero and Dan Klein. Tailoring word alignments to syntactic machine translation. In *Proceedings of the Association for Computational Linguistics*, 2007.

(DeNero and Klein, 2008) John DeNero and Dan Klein. The complexity of phrase alignment problems. In *Proceedings of the Association for Computational Linguistics*, 2008.

(DeNero and Klein, 2010) John DeNero and Dan Klein. Discriminative modeling of extraction sets for machine translation. In *Proceedings of the Association for Computational Linguistics*, 2010.

(DeNero *et al.*, 2006) John DeNero, Dan Gillick, James Zhang, and Dan Klein. Why generative phrase models underperform surface heuristics. In *Proceedings of the NAACL Workshop on Statistical Machine Translation*, 2006.

(DeNero *et al.*, 2008) John DeNero, Alexandre Bouchard-Côté, and Dan Klein. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008.

(DeNero *et al.*, 2009) John DeNero, David Chiang, and Kevin Knight. Fast consensus decoding over translation forests. In *Proceedings of the Association for Computational Linguistics and IJCNLP*, 2009.

(DeNero *et al.*, 2010) John DeNero, Shankar Kumar, Ciprian Chelba, and Franz Josef Och. Model combination for machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2010.

(Deng and Zhou, 2009) Yonggang Deng and Bowen Zhou. Optimizing word alignment combination for phrase table training. In *Proceedings of the Association for Computational Linguistics*, 2009.

(Dyer, 2009) Chris Dyer. Two monolignual parses are better than one (synchronous parse). In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2009.

(Ferguson, 1973) Thomas S Ferguson. A Bayesian analysis of some nonparametric problems. In *Annals of Statistics*, 1973.

(Finkel *et al.*, 2007) Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. The infinite tree. In *Proceedings of the Association for Computational Linguistics*, 2007.

(Fraser and Marcu, 2006) Alexander Fraser and Daniel Marcu. Semi-supervised training for statistical word alignment. In *Proceedings of the Association for Computational Linguistics*, 2006.

(Gillick, 2009) Dan Gillick. Sentence boundary detection and the problem with the U.S. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2009.

(Gimpel and Smith, 2008) Kevin Gimpel and Noah Smith. Rich source-side context for statistical machine translation. In *Proceedings of the ACL Workshop on statistical machine translation*, 2008.

(Goldwater *et al.*, 2006) Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the Association for Computational Linguistics*, 2006.

(Goldwater *et al.*, 2009) Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 2009.

(Goodman, 1996) Joshua Goodman. Parsing algorithms and metrics. In *Proceedings of the Association for Computational Linguistics*, 1996.

(Haghighi and Klein, 2007) Aria Haghighi and Dan Klein. Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of the Association for Computational Linguistics*, 2007.

(Haghighi *et al.*, 2009) Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. Better word alignments with supervised ITG models. In *Proceedings of the Association for Computational Linguistics*, 2009.

(Helft, 2010) Miguel Helft. Googles computing power refines translation too. *New York Times*, 2010.

(Huang and Chiang, 2007) Liang Huang and David Chiang. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the Association for Computational Linguistics*, 2007.

(Huang, 2009) Liang Huang. Forest reranking: Discriminative parsing with non-local feature. In *Proceedings of the Association for Computational Linguistics*, 2009.

(Ittycheriah and Roukos, 2005) Abraham Ittycheriah and Salim Roukos. A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2005.

(Johnson *et al.*, 2007a) Mark Johnson, Thomas Griffiths, and Sharon Goldwater. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proceedings of the Association for Computational Linguistics*, 2007.

(Johnson *et al.*, 2007b) Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. Adaptor grammars: a framework for specifying compositional nonparametric Bayesian models. In *Proceedings of Neural Information Processing Systems*, 2007.

(Kääriäinen, 2009) Matti Kääriäinen. Sinuhe—statistical machine translation using a globally trained conditional exponential family translation model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2009.

(Klein and Manning, 2003) Dan Klein and Chris Manning. A* parsing: Fast exact Viterbi parse selection. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2003.

(Kneser and Ney, 1995) Reinhard Kneser and Hermann Ney. Improved backing-off of M-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1995.

(Koehn and Haddow, 2009) Philipp Koehn and Barry Haddow. Edinburghs submission to all tracks of the WMT2009 shared task with reordering and speed improvements to Moses. In *Proceedings of the ACL Workshop on Statistical Machine Translation*, 2009.

(Koehn *et al.*, 2003) Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2003.

(Koehn *et al.*, 2007) 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 *Proceedings of the Association for Computational Linguistics: Demonstration track*, 2007.

(Koehn, 2002) Philipp Koehn. Europarl: A multilingual corpus for evaluation of machine translation. 2002.

(Kumar and Byrne, 2004) Shankar Kumar and William Byrne. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2004.

(Lacoste-Julien *et al.*, 2006) Simon Lacoste-Julien, Ben Taskar, Dan Klein, and Michael I. Jordan. Word alignment via quadratic assignment. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2006.

(Lafferty *et al.*, 2001) John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.

(Li *et al.*, 2009) Zhifei Li, Jason Eisner, and Sanjeev Khudanpur. Variational decoding for statistical machine translation. In *Proceedings of the Association for Computational Linguistics and IJCNLP*, 2009.

(Liang *et al.*, 2006) Percy Liang, Ben Taskar, and Dan Klein. Alignment by agreement. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2006.

(Liang *et al.*, 2007) Percy Liang, Slav Petrov, Michael Jordan, and Dan Klein. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2007.

(Liu *et al.*, 2005) Yang Liu, Qun Liu, and Shouxun Lin. Log-linear models for word alignment. In *Proceedings of the Association for Computational Linguistics*, 2005.

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

(May and Knight, 2007) Jonathan May and Kevin Knight. Syntactic re-alignment models for machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2007.

(McDonald *et al.*, 2005) Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of the Association for Computational Linguistics*, 2005.

(Melamed, 2000) I. Dan Melamed. Models of translational equivalence among words. *Computational Linguistics*, 2000.

(Moore and Quirk, 2007a) Robert Moore and Chris Quirk. An iteratively-trained segmentation-free phrase translation model for statistical machine translation. In *Proceedings of the ACL Workshop on Statistical Machine Translation*, 2007.

(Moore and Quirk, 2007b) Robert C. Moore and Chris Quirk. Faster beam-search decoding for phrasal statistical machine translation. In *Proceedings of MT Summit XI*, 2007.

(Moore *et al.*, 2006) Robert C. Moore, Wen tau Yih, and Andreas Bode. Improved discriminative bilingual word alignment. In *Proceedings of the Association for Computational Linguistics*, 2006.

(Moore, 2005) Robert C. Moore. A discriminative framework for bilingual word alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2005.

(Ney and Vogel, 1996) Hermann Ney and Stephan Vogel. HMM-based word alignment in statistical translation. In *Proceedings of the Conference on Computational linguistics*, 1996.

(Och and Ney, 2003) Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 2003.

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

(Och *et al.*, 1999) Franz Josef Och, Christopher Tillman, and Hermann Ney. Improved alignment models for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1999.

(Och, 2003) Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the Association for Computational Linguistics*, 2003.

(Papineni *et al.*, 2002) Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*, 2002.

(Rosti *et al.*, 2007) 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 *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2007.

(Shimizu and Haas, 2006) Nobuyuki Shimizu and Andrew Haas. Exact decoding for jointly labeling and chunking sequences. In *Proceedings of the Association for Computational Linguistics*, 2006.

(Snyder and Barzilay, 2008) Benjamin Snyder and Regina Barzilay. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of the Association for Computational Linguistics*, 2008.

(Stolcke, 2002) Andreas Stolcke. SRILM an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, 2002.

(Taskar *et al.*, 2005) Ben Taskar, Simon Lacoste-Julien, and Dan Klein. A discriminative matching approach to word alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2005.

(Teh *et al.*, 2005) Yee Whye Teh, David Blei, and Michael Jordan. Hierarchical Dirichlet processes. In *Proceedings of Neural Information Processing Systems*, 2005.

(Teh, 2006) Yee Whye Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the Association for Computational Linguistics*, 2006.

(Valiant, 1979) Leslie G. Valiant. The complexity of computing the permanent. In *Theoretical Computer Science*, 1979.

(Watanabe, 2007) Taro Watanabe. Online large-margin training for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2007.

(Weaver, 1949) Warren Weaver. Translation. *Machine translation of languages: fourteen essays,*, 1949.

(Wu, 1997) Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 1997.

(Zens *et al.*, 2004) Richard Zens, Hermann Ney, Taro Watanabeand, and E. Sumita. Reordering constraints for phrase based statistical machine translation. In *Proceedings of the International Conference on Computational Linguistics*, 2004.

(Zhang *et al.*, 2008) Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of the Association for Computational Linguistics*, 2008.