UNSUPERVISED LEARNING OF WORD ALIGNMENTS FOR STATISTICAL MACHINE TRANSLATION

by

Coşkun Mermer

B.S., Electrical and Electronics Engineering, Bilkent University, 1998

M.S., Electrical Engineering, University of Washington, 2001

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Electrical and Electronics Engineering

Boğaziçi University

2019

ABSTRACT

UNSUPERVISED LEARNING OF WORD ALIGNMENTS FOR STATISTICAL MACHINE TRANSLATION

Word alignment is a crucial first step in learning statistical translation models. In this dissertation, we propose a Bayesian approach to unsupervised learning of word alignments by introducing a sparse prior on the parameters of IBM word alignment models. In the original approach, word translation probabilities are estimated using the expectation-maximization (EM) algorithm. In the proposed approach, they are random variables with a prior and are integrated out during inference, where collapsed Gibbs sampling is used. The inferred word alignments are evaluated in a statistical machine translation (SMT) setting, experimenting with several language pairs and sizes of corpora and comparing against the EM and variational Bayes (VB) methods. We show that Bayesian inference outperforms both EM and VB in the majority of test cases, effectively addresses the high-fertility rare word problem in EM and unaligned rare word problem in VB, achieves higher agreement and vocabulary coverage rates than both, and leads to smaller phrase tables. We also propose a method for unsupervised learning of the optimal segmentation for SMT. We augment the original Morfessor monolingual segmentation model with a word alignment model so that the new model optimizes the posterior probability of the parallel training corpus according to a generative segmentation-translation model. In order to speed up computation, we propose an incremental method for approximate translation likelihood calculation and a parallelizable search algorithm, which improves the performance of even the monolingual segmentation. We use the proposed method to segment the Turkish side in a Turkish-to-English SMT system and find that the bilingual model results in more intuitive segmentations but does not yield a further significant increase in BLEU scores.

ÖZET

İSTATİSTİKSEL MAKİNE ÇEVİRİSİ İÇİN KELİME HİZALAMALARININ GÖZETİMSİZ ÖĞRENİMİ

Kelime hizalama, istatistiksel çeviri modeli öğreniminde kritik öneme sahip bir ilk aşamadır. Bu tez çalışmasında IBM kelime hizalama modellerinin parametreleri üzerine seyrek bir önsel getirerek kelime hizalamalarının gözetimsiz öğrenimi için Bayesçi bir yaklaşım önerilmektedir. Orijinal yöntemde kelime çeviri olasılıkları beklenti-eniyileme (EM) yöntemiyle kestirilmektedir. Önerilen yöntemde ise bu olasılıklar bir önsel dağılıma sahip rastsal değişkenlerdir ve daraltılmış Gibbs örneklemesi kullanılarak çıkarım esnasında tümlevi alınmaktadır. Çıkarımı yapılan hizalamalar bir istatistiksel makine çevirisi (SMT) ortamında birçok dil çifti ve derlem büyüklükleri üzerinde EM ve değişimsel Bayes (VB) ile kıyaslanarak değerlendirilmektedir. Önerilen Bayesçi yöntemin sınama senaryolarının çoğunluğunda diğer iki yöntemden üstünlüğü, EM yöntemindeki yüksek doğurganlıklı nadir kelime ve VB yöntemindeki hizalanmamış nadir kelime problemlerine etkin çözüm getirdiği, iki yöntemden de daha yüksek uzlaşım ve dağarcık kapsama oranı elde ettiği, ve daha küçük öbek tablolarını mümkün kıldığı gösterilmektedir. Tezde aynı zamanda SMT için en uygun bölütlemenin gözetimsiz öğrenimi için de bir yöntem önerilmektedir. Orijinal Morfessor tek dilli bölütleme modeli bir kelime hizalama modeliyle geliştirilmektedir, böylece yeni model paralel eğitim derleminin üretken bir bölütleme-hizalama modeline göre sonsal olasılığını en iyiler. Hesaplamayı hızlandırmak amacıyla, yaklaşık çeviri olabilirliğini hesaplamak için artımsal bir yöntem ve aynı zamanda tek dilli bölütlemenin de başarımını iyileştiren paralelleştirilebilen bir arama yordamı önerilmektedir. Önerilen yöntem bir Türkçeden İngilizceye SMT sisteminde Türkçe tarafı bölütlemek için kullanılmış ve iki dilli modelin daha sezgisel bölütlemelere yol açmasına rağmen BLEU skorlarında daha öte bir belirgin artış sağlamadığı gözlenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\mathbf{A}$	Alignments for the parallel corpus $(\mathbf{E}, \mathbf{F})$
$\mathbf{a}$	Alignments for the s-th sentence pair; $\mathbf{a} = a_1 \cdots a_J$
a_j	Alignment index for the j-th target word
$\mathbf{E}$	Source language corpus, consisting of S sentences
$\mathbf{e}$	s-th source sentence; $\mathbf{e} = e_1 \cdots e_I$
e_i	i-th source word
e_0	"Null" source word accounting for any unaligned words in $\mathbf{f}$
$\mathbf{F}$	Target language corpus, consisting of S sentences
$\mathbf{f}$	s-th target sentence; $\mathbf{f} = f_1 \cdots f_J$
f_j	j-th target word
I	length of source sentence
J	length of target sentence
$N_{e,f}$	Number of times e is aligned to f in parallel corpus $(\mathbf{E}, \mathbf{F})$
$n_{e,f,s}$	Number of times e is aligned to f in sentence pair s
S	Number of sentences in parallel corpus $(\mathbf{E}, \mathbf{F})$
s	Index of a sentence pair
$\mathbf{t}_e$	Word translation probability distribution; $\mathbf{t}_e = t_{e,1} \cdots t_{e,V_F}$
$t_{e,f}$	Word translation probability; $t_{e,f} = P(f\|e)$
$\mathbf{T}$	A $V_E \times V_F$ table of translation probabilities
V_E	Size of the source corpus vocabulary (including the null word)
V_F	Size of the target corpus vocabulary
$\boldsymbol{\Theta}$	Hyperparameters of the model; $\boldsymbol{\Theta} = \boldsymbol{\Theta}_1 \cdots \boldsymbol{\Theta}_{V_E}$
$\boldsymbol{\Theta}_e$	Hyperparameter vector for the Dirichlet distribution from which $\mathbf{t}_e$ is drawn from; $\boldsymbol{\Theta}_e = \theta_{e,1} \cdots \theta_{e,V_F}$
$\theta_{e,f}$	Individual values of the hyperparameter vector $\boldsymbol{\Theta}_e$

LIST OF ACRONYMS/ABBREVIATIONS

AER	Alignment Error Rate
BLEU	Bilingual Evaluation Understudy
BTEC	Basic Travel Expressions Corpus
CRP	Chinese Restaurant Process
EM	Expectation-Maximization
GS	Gibbs Sampling
HMM	Hidden Markov Model
IWSLT	International Workshop on Spoken Language Translation
LDC	Linguistic Data Consortium
LM	Language Model
MAP	Maximum A Posteriori
MCEM	Monte Carlo EM
MERT	Minimum Error Rate Training
NMT	Neural Machine Translation
OOV	Out-of-Vocabulary
PDF	Probability Density Function
SCFG	Synchronous Context-Free Grammar
SMT	Statistical Machine Translation
TER	Translation Edit Rate
VB	Variational Bayes
WMT	Workshop on Statistical Machine Translation

1. INTRODUCTION

In training statistical machine translation (SMT) systems, the parameters/feature values of the translation models are estimated from parallel corpora. Whether the employed models are the widely-used phrase-based models [1] or the more recent tree-based models [2, 3], a crucial first step in training is word alignment [4]. These models make use of the estimated word alignments for constraining the set of candidates in phrase or grammar rule extraction. As such, the coverage and the accuracy of the learned phrase/rule translation models are strongly correlated with those of the word alignment. Therefore, good word alignment algorithms are important since they affect the remaining steps of SMT system training.

Given a sentence-aligned parallel corpus, the goal of the word alignment is to identify the mapping between the source and target words in parallel sentences. Since word alignment information is usually not available during corpus generation and human annotation is costly, the task of word alignment is considered as an unsupervised learning problem.

State-of-the-art word alignment models, such as IBM Models [5], hidden Markov model (HMM) [6], and the jointly-trained symmetric HMM [7], contain a large number of parameters (such as word translation, transition, and fertility probabilities) that need be estimated in addition to the desired alignment variables. The common method of inference in such models is expectation-maximization (EM) [8] or an approximation to EM when exact EM is intractable. The EM algorithm finds the value of parameters that maximizes the likelihood of the observed variables. However, with many parameters to be estimated without any prior, EM tends to explain the training data by overfitting the parameters. Moreover, EM is generally prone to getting stuck in a local maximum of the likelihood. Finally, EM is based on the assumption that there is one fixed value of parameters that explains the data, i.e., EM-inferred word alignments do not take into account other probable values of the parameters.

Another problem in word alignment arises when aligning a morphologically diverging language pair (e.g., Turkish-English) such that there is usually a granularity mismatch when using word alignment models. Word-based models treat a space-separated token (a "word") as the smallest unit in the model. On the other hand, Turkish is an agglutinative language where words can carry several morphemes in the form of suffixes. As a result, a Turkish word can correspond to a multi-word (sometimes non-contiguous) phrase when paired with a morphologically simpler language such as English. Figure 1.1 illustrates this morphological divergence on a example English-Turkish translation pair.

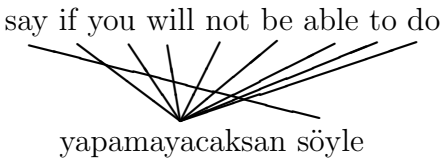

Figure 1.1. An English-Turkish translation pair and its ground-truth word alignment.

Encoding so many morphemes in a word leads to fast-growing vocabulary, data sparsity issues in estimating model parameters, and high degree of out-of-vocabulary (OOV) problems during run-time. For example, even though there are a total of about 150 distinct lexical suffixes in Turkish, the number of possible word derivations are practically unlimited, posing a huge problem for word-based models.

Furthermore, 1-to-N alignment models (such as the popular IBM Models 1-5 and the HMM alignment model) are usually run in both directions before symmetrization. This creates a problem when the target language in the alignment generation model is the morphologically-rich language. For example in Figure 1.1, English-to-Turkish generative model requires the Turkish word "yapamayacaksan" to align to only one English word, which does not accurately capture the true translation process. As a result, naively applying word-based alignment/translation models to parallel corpora involving Turkish is not optimal.

A logical solution to this problem is morphological analysis, i.e., tokenizing the individual morphemes, since the translation process is assumed to preserve meaning and the smallest meaning-bearing unit in language is the morpheme. Therefore, one expects a better correspondence between a translation pair on the morpheme level leading to more accurate alignments, except for the language-specific idiosyncrasies (for example, the semantics contributed by the English word/morpheme "the" does not have an overt morpheme counterpart in Turkish, a similar example is the lack of an accusative marker in English).

For agglutinative languages (and, to an extent, even for inflecting languages such as Arabic, English etc.), morphological analysis can be approximated by segmentation, i.e., splitting surface word forms into multiple "morphs". For the Turkish word "yapamayacaksan" in Figure 1.1, segmentation into its surface morphemes results in a more fine-grained and scalable morpheme-based alignment as shown in Figure 1.2.

yap	+a	+ma	+yacak	+sa	+n
do	be able to	not	will	if	you

Figure 1.2. Morphemes and their correspondences to the words in the English translation of the Turkish word *yapamayacaksan* ('if you will not be able to do')

Our motivation in this study is to improve on the existing statistical machine translation (SMT) models for Turkish. We focus on improving the learning of alignments and modeling of morphology, hopefully leading to better translation models (e.g., phrase tables). We also would like to preserve the two traits of SMT that has lead to its widespread success, namely learning without requiring human involvement and language independence. Hence we narrow our interest to the unsupervised learning methods.

The work presented in this thesis can be divided into two parts. In the first part, we present a method for Bayesian inference of word alignments that outperforms

the existing maximum-likelihood solutions (Chapter 3). In the second part, we propose a method for unsupervised determination of the optimal segmentation for SMT (Chapter 4).

1.1. Contributions of the Thesis

This study contributes to the steps (i)–(ii) of translation model training described in Figure 1.3. Improvements in modeling and inference of these fundamental steps (tokenization and alignment) are expected to result in better translation models, and eventually better decoding (i.e., translation) performance. Since our ultimate goal is improving the translation quality, we measure the utility of our proposed algorithms on the end-to-end translation performance of the SMT system.

(i) Tokenization (i.e., determination of what constitutes a "word")
(ii) Word alignment (in both directions)
(iii) Symmetrization of word alignments
(iv) Phrase-pair/SCFG rule extraction
(v) Computation of phrase-/rule-based features (e.g., probabilities estimated from the counts in the training corpus)

Figure 1.3. Steps in learning a translation model.

The main contributions of this thesis are:

- Bayesian treatment of IBM word aligment Models 1 and 2 (Section 3.3). We treat the model parameters as multinomial-distributed random variables with sparse Dirichlet priors and integrate over all parameter values during inference.
- Derivation of a Gibbs sampler for the proposed Bayesian alignment models (Section 3.3.3 and Appendix A) as well as equivalence of the Chinese Restaurant Process view of word alignment (Section 3.3.4).
- Extensive end-to-end evaluation of the alternative unsupervised word alignment methods (EM, Gibbs sampling and variational Bayes) and their combina-

tions on several language pairs and data sizes (Section 3.4), including sampling analysis of the Gibbs sampler (Section 3.6).

- Detailed intrinsic evaluation of the word alignments obtained by these methods (Section 3.5).
- A bilingual extension of the Morfessor segmentation algorithm that performs sub-word segmentation by taking into account both sides of the SMT training corpus (Section 4.3.2).
- An incremental method for approximate translation likelihood calculation in order to speed up the proposed bilingual segmentation method (Section 4.3.3).
- A parallel search algorithm for speeding up using multiple CPUs that is also both applicable and beneficial to the original (monolingual) version of Morfessor (Section 4.3.4).

Parts of the work in this thesis has been published before as follows: The Bayesian treatment of word alignment (Chapter 3) was first presented in [9] and then elaborated with extensive evaluation in [10]. The proposed unsupervised segmentation method (Chapter 4) was presented in [11] and [12].

1.2. Organization of the Thesis

We start with a brief overview of statistical machine translation, the main setting and motivation in our work, in Chapter 2. The state-of-the-art log-linear approach and its components are presented. Then the usual steps of training an SMT system are outlined.

In Chapter 3 we present a method for Bayesian inference of word alignments that outperforms the existing maximum-likelihood solutions. We re-formulate the original IBM Model 1 as a multinomial generative model and introduce a conjugate prior on the model parameters. This allows us to integrate out the parameters and infer the word alignments using Gibbs sampling. We describe the Gibbs sampling procedure and a modified sampling algorithm that enables multi-threaded implementation. An extension of the approach to IBM Model 2 is also presented. The proposed method is

evaluated on various language pairs with varying sizes of corpora. The Gibbs samplers and the inferred alignments are analyzed in detail.

In Chapter 4 we propose a method for unsupervised determination of the optimal segmentation for SMT. To improve on the commonly-used Morfessor algorithm, which utilizes only monolingual information, a bilingual model is proposed that utilizes both sides of the training corpus. In order to speed up computation, an incremental method of computing approximate likelihoods and a parallel search method are proposed. The proposed method is evaluated in a Turkish-English SMT setting.

Finally, conclusion and future research directions are discussed in Chapter 5. Special attention is reserved for relating the work in this thesis to the most recent MT paradigm, neural machine translation (NMT). Potential impact of the presented work on NMT is outlined.

2. STATISTICAL MACHINE TRANSLATION

In most of today's SMT systems, the probability of a translation hypothesis e given the source sentence f is formulated as a log-linear model:

$$P(e|f) = \frac{1}{Z(f)} \exp(\sum_{k=1}^{K} \lambda_k h_k(e, f)) \tag{2.1}$$

Here, h_k represent the feature functions, K the number of features in the model and λ_k the feature weights. Z is essentially a normalization factor so that $\sum_e P(e|f) = 1$.

The decision rule for the best hypothesis e^* is a direct maximization over the posterior $P(e|f)$:

$$e^* = \arg\max_e P(e|f) \tag{2.2}$$

$$= \arg\max_e \sum_{k=1}^{K} \lambda_k h_k(e, f) \tag{2.3}$$

Note that the value of Z never needs to be computed since it is a common denominator in the probabilities of all hypotheses. We also utilize the monotonicity of the exponential function and work directly on the linear combination of features.

Every system today uses an assortment of features h_k with K usually around 10–15. The most commonly-used features are:

- Word translation probabilities in both translation directions
- SCFG rule or phrase (depending on the system) translation probabilities in both translation directions
- Language model probability of the hypothesis ($h_{LM}(e, f) = P(e)$)
- Word/phrase/rule count in the hypothesis
- Reordering (distortion) model cost.

Note that the values for most of these features depend on the particular "derivation" that produces e from f, e.g., which set of SCFG rules were applied or which phrase-pairs were used etc. It is the decoder's task to search for the hypothesis (or derivation) with the highest total score according to (2.3).

Many of the feature functions listed above are probabilities, which need to be estimated (from data) in a training step before decoding. SMT system training usually consists of the following steps:

(i) Learn the model-based feature functions.
   - Learn a translation model, as shown in Figure 1.3.
   - Learn a language model (usually an N-gram model).
   - Learn other models, if any (e.g., some systems use probabilistic reordering models, additional part-of-speech (POS) language models etc.).

(ii) Learn the feature weights, usually using the minimum error rate training (MERT) algorithm [13].

3. BAYESIAN WORD ALIGNMENT

In this chapter we present a Bayesian approach to word alignment inference in IBM Models 1 and 2. In the classical maximum-likelihood approach, word translation probabilities (i.e., model parameters) are estimated using the expectation-maximization (EM) algorithm. In the proposed approach, word translation probabilities are random variables with a prior and are integrated out during inference. We use Gibbs sampling to infer the word alignment posteriors. The inferred word alignments are compared against EM and variational Bayes (VB) inference in terms of their end-to-end translation performance on several language pairs and types of corpora up to 15 million sentence pairs. Experimental results show that Bayesian inference outperforms both EM and VB. Further analysis reveals that the proposed method effectively addresses the high-fertility rare word problem in EM and unaligned rare word problem in VB, achieves higher agreement and vocabulary coverage rates than both, and leads to smaller phrase tables.

3.1. Introduction

Word alignment is a crucial early step in the training pipeline of most statistical machine translation (SMT) systems [4]. Whether the employed models are phrase-based or tree-based, they use the estimated word alignments for constraining the set of candidates in phrase or grammar rule extraction [1–3]. As such, the coverage and the accuracy of the learned phrase/rule translation models are strongly correlated with those of the word alignment. Given a sentence-aligned parallel corpus, the goal of the word alignment is to identify the mapping between the source and target words in parallel sentences. Since word alignment information is usually not available during corpus generation and human annotation is costly, the task of word alignment is considered as an unsupervised learning problem.

State-of-the-art word alignment models, such as IBM Models [5], hidden Markov model (HMM) [6], and the jointly-trained symmetric HMM [7], contain a large num-

ber of parameters (such as word translation, transition, and fertility probabilities) that need be estimated in addition to the desired alignment variables. The common method of inference in such models is expectation-maximization (EM) [8] or an approximation to EM when exact EM is intractable. The EM algorithm finds the value of parameters that maximizes the likelihood of the observed variables. However, with many parameters to be estimated without any prior, EM tends to explain the training data by overfitting the parameters. A well-documented example of overfitting in EM-estimated word alignments is the case of rare words, where some rare words act as "garbage collectors" aligning to excessively many words on the other side of the sentence pair [14–16]. Moreover, EM is generally prone to getting stuck in a local maximum of the likelihood. Finally, EM is based on the assumption that there is one fixed value of parameters that explains the data, i.e., EM gives a point estimate.

We propose a Bayesian approach in which we utilize a prior distribution on the parameters. The alignment probabilities are inferred by integrating over all possible parameter values. We treat the word translation probabilities as multinomial-distributed random variables with a sparse Dirichlet prior. Inference is performed via Gibbs sampling, which samples the posterior alignment distribution. We compare the EM and Bayesian alignments on the case of IBM Models 1 and 2. The inferred alignments are evaluated in terms of end-to-end translation performance on various language pairs and corpora.

The remainder of this chapter is organized as follows: The related literature is reviewed in Section 3.2. The proposed model and the inference algorithm are presented in Section 3.3. The experiments are described and their results are presented in Section 3.4. A detailed analysis of the resulting alignments, sampling settings, and BLEU variance are provided in Sections 3.5–3.7, followed by the conclusions in Section 3.8.

3.2. Related Work

Problems with the standard EM estimation of IBM Model 1 were pointed out by Moore [16]. A number of heuristic changes to the estimation procedure, such as

smoothing the parameter estimates, were shown to reduce the alignment error rate, but the effects on translation performance were not reported. Zhao and Xing [17] address the data sparsity issue using symmetric Dirichlet priors in parameter estimation and they use variational EM to find the maximum *a posteriori* (MAP) solution. Vaswani *et al.* [18] encourage sparsity in the translation model by placing an ℓ_0 prior on the parameters and then optimize for the MAP objective.

Zhao and Gildea [19] use sampling in their proposed fertility extensions to IBM Model 1 and HMM, but they do not place any prior on the parameters. Their inference method is stochastic EM (also known as Monte Carlo EM), a maximum-likelihood technique in which sampling is used to approximate the expected counts in the E-step. Even though they report substantial reductions in the alignment error rate, the translation performance measured in BLEU does not improve.

Bayesian modeling and inference have recently been applied to several unsupervised learning problems in natural language processing such as part-of-speech tagging [20,21], word segmentation [22,23], grammar extraction [24] and finite-state transducer training [25] as well as other tasks in SMT such as synchronous grammar induction [26] and learning phrase alignments directly [27].

Word alignment learning problem was addressed jointly with segmentation learning by Xu *et al.* [28], Nguyen *et al.* [29], and Chung and Gildea [30]. As in this paper, they treat word translation probabilities as random variables (with an associated prior distribution). Both [28] and [29] place *nonparametric* priors (also known as cache models) on the parameters. Similar to our work, this enables integration over the prior distribution. In [28], a Dirichlet Process prior is placed on IBM Model 1 word translation probabilities. In [29], a Pitman-Yor Process prior is placed on word translation probabilities in a proposed bag-of-words translation model that is similar to IBM Model 1. Both studies utilize Gibbs sampling for inference. However, alignment distributions are not sampled from the true posteriors but instead are updated either by running GIZA++ [28] or using a "local-best" maximization search [29]. On the other hand, a sparse Dirichlet prior on the multinomial parameters is used in [30] to

prevent overfitting.

Bayesian word alignment with Dirichlet priors was also investigated in a recent study using variational Bayes (VB) [31]. VB is a Bayesian inference method which is sometimes preferred over Gibbs sampling due to its relatively lower computational cost and scalability. However, VB inference approximates the model by assuming independence between the hidden variables and the parameters. To evaluate the effect of this approximation, we also present and analyze the experimental results obtained using VB (Sections 3.4.3 and 3.5).

3.3. Bayesian Inference of Word Alignments

We first recap the IBM Model 1 presented in [5] and establish the notation used in this paper. Given a parallel corpus $(\mathbf{E}, \mathbf{F})$ of S sentence pairs, let $\mathbf{e}$ $(\mathbf{f})$ denote the s-th sentence in $\mathbf{E}$ $(\mathbf{F})$, and let e_i (f_j) denote the i-th $(j$-th) word among a total of I (J) words in $\mathbf{e}$ $(\mathbf{f})$[1] . We also hypothesize an imaginary "null" word e_0 to account for any unaligned words in $\mathbf{f}$. Also let V_E and V_F denote the size of the respective vocabularies.

We associate with each f_j a hidden *alignment* variable a_j whose value ranges over $[0, I]$. The set of alignments for a sentence (corpus) is denoted by $\mathbf{a}$ $(\mathbf{A})$. The model parameters consist of a $V_E \times V_F$ table $\mathbf{T}$ of word translation probabilities such that $t_{e,f} = P(f|e)$. Since f is conditioned on e, we refer to e $(\mathbf{e})$ as the "source" word (sentence) and f $(\mathbf{f})$ as the "target" word (sentence)[2] .

[1]Keeping in mind that $\mathbf{e}, \mathbf{f}, I, J$ (and $\mathbf{a}$ introduced later) are defined with respect to the s-th sentence, we drop the subscript s for notational simplicity.

[2]Historically, the source and target designations were based on the translation task, when the word alignment direction was dictated by the "noisy channel model" to be the inverse of the translation direction. Today almost all SMT systems using IBM models train alignments in both directions, decoupling the alignment direction from that of translation and nullifying the justification of the early nomenclature.

Table 3.1. List of variables in the original IBM Model 1. To simplify notation, sentence-specific subscripts are omitted from the variables **a**, **e**, **f**, I and J; their dependence on the sentence index s is implicit.

Observed variables:	
$\mathbf{e} = e_1 \cdots e_I$	s-th source (English) sentence, consisting of I words
$\mathbf{f} = f_1 \cdots f_J$	s-th target (French) sentence, consisting of J words
V_E	Size of the source corpus vocabulary (including the null word)
V_F	Size of the target corpus vocabulary
Hidden variables:	
$\mathbf{a} = a_1 \cdots a_J$	Alignments for the s-th sentence pair; for each target word f_j, a_j takes integer values in $[0, I]$ with the value 0 representing alignment to the null source word
Model parameters:	
$\mathbf{T}$	A $V_E \times V_F$ table of translation probabilities where $t_{e,f} = P(f\|e)$

The conditional distribution of the Model 1 variables given parameters $\mathbf{T}$ is expressed by the following generative model:

$$a_j|\mathbf{e} \sim \text{Uniform}(a_j; I+1)$$

$$P(\mathbf{F}, \mathbf{A}|\mathbf{E}; \mathbf{T}) = \prod_s P(\mathbf{a}|\mathbf{e}) P(\mathbf{f}|\mathbf{a}, \mathbf{e}; \mathbf{T}) \tag{3.1}$$

$$= \prod_s \frac{1}{(I+1)^J} \prod_{j=1}^{J} t_{e_{a_j}, f_j}. \tag{3.2}$$

The dependency structure of this generative model is illustrated in Figure 3.1.

The two unknowns $\mathbf{A}$ and $\mathbf{T}$ are estimated using the EM algorithm, which finds the value of $\mathbf{T}$ that maximizes the likelihood of the observed variables $\mathbf{E}$ and $\mathbf{F}$ according to the model. Once the value of $\mathbf{T}$ is known, the probability of any alignment becomes straightforward to compute.

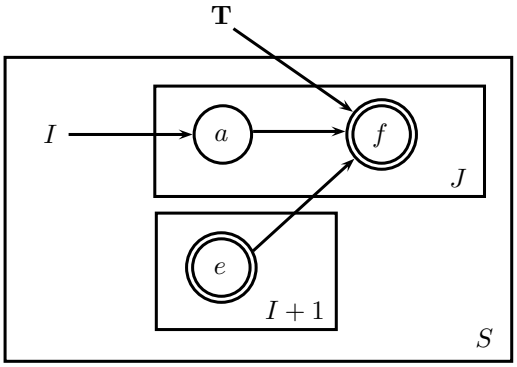

Figure 3.1. Plate representation of the generative model IBM Model 1.

In the following derivation of our proposed model, we treat the unknown $\mathbf{T}$ as a random variable. Following the Bayesian approach, we assume a prior distribution on $\mathbf{T}$ and infer the distribution of $\mathbf{A}$ by integrating over all values of $\mathbf{T}$.

3.3.1. Canonical Representation of Model 1

We first convert the token-based expression in (3.2) into a type-based one as (with $\mathbf{T}$ now a random variable):

$$P(\mathbf{F}, \mathbf{A}|\mathbf{E}, \mathbf{T}) = \prod_s \frac{1}{(I+1)^J} \prod_{e=1}^{V_E} \prod_{f=1}^{V_F} (t_{e,f})^{n_{e,f,s}} \tag{3.3}$$

$$= \prod_{e=1}^{V_E} \prod_{f=1}^{V_F} (t_{e,f})^{N_{e,f}} \cdot \prod_s \frac{1}{(I+1)^J}, \tag{3.4}$$

where in (3.3) the count variable $n_{e,f,s}$ denotes the number of times the source word type e is aligned to the target word type f in the sentence pair s, and in (3.4) $N_{e,f} = \sum_s n_{e,f,s}$.

This formulation exposes two properties of IBM Model 1 that facilitates the derivation of a Bayesian inference algorithm. First, the parametrization on $\mathbf{T}$ is in the canonical form of an *exponential family* distribution (as the inner-product of parameters $\log t_{e,f}$ and sufficient statistics $N_{e,f}$), which implies the existence of a *conjugate prior*

that simplifies calculation of the posterior.

Second, the distribution in (3.4) depends on the variables **E**, **F** and **A** only through a set of count variables $N_{e,f}$. In other words, the order of words within a sentence has no effect on the likelihood, which is called *exchangeability* or a "bag of words" model. This results in simplification of the terms when deriving the Gibbs sampler.

3.3.2. Prior on Word Translation Probabilities

For each source word type e, by definition $\mathbf{t}_e = t_{e,1} \cdots t_{e,V_F}$ form the parameters of a multinomial distribution that governs the distribution of the target words aligned to e. Hence, the conditional distribution of the j-th target word in a sentence pair is defined by:

$$f_j | \mathbf{a}, \mathbf{e}, \mathbf{T} \sim \text{Multinomial}(f_j; \mathbf{t}_{e_{a_j}}).$$

Since the conjugate prior of multinomial is the Dirichlet distribution, we choose:

$$\mathbf{t}_e | \boldsymbol{\Theta} \sim \text{Dirichlet}(\mathbf{t}_e; \boldsymbol{\Theta}_e),$$

where $\boldsymbol{\Theta}_e = \theta_{e,1} \cdots \theta_{e,V_F}$. Overall, $\boldsymbol{\Theta} = \boldsymbol{\Theta}_1 \cdots \boldsymbol{\Theta}_{V_E}$ are the hyperparameters of the model. The mathematical expression for the prior $P(\mathbf{T}; \boldsymbol{\Theta})$ is provided in (A.3) in the Appendix. The dependency structure of the proposed generative model is illustrated in Figure 3.2.

We can encode our prior expectations for $\mathbf{t}_e$ into the model by suitably setting the values of $\boldsymbol{\Theta}_e$. For example, we generally expect the translation probability distribution of a given source word type e to be concentrated on one or a few target word types. Setting $\theta_{e,f} \ll 1, \forall f$ allocates more prior weight to such sparse distributions. Figure 3.3 shows the probability density function (PDF) of an example symmetric sparse Dirichlet distribution for the case where $V_F = 3$. Figure 3.4 illustrates random samples drawn

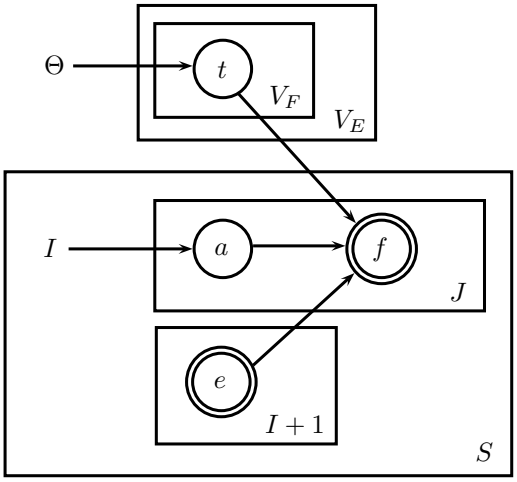

Figure 3.2. Plate representation of the proposed generative model.

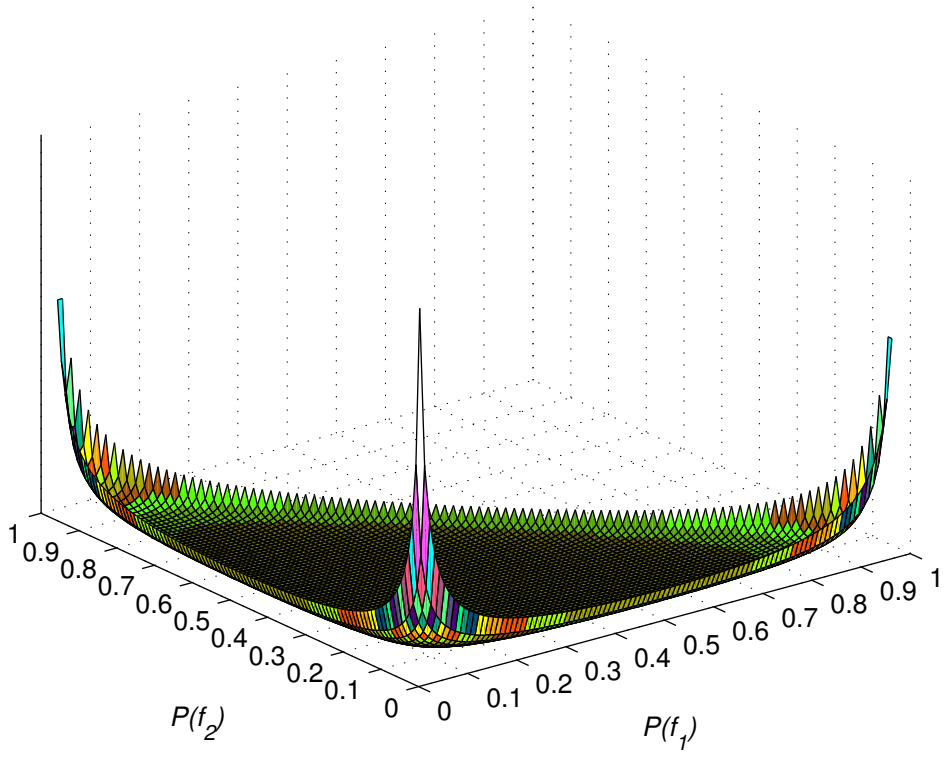

Figure 3.3. Probability density function of a sparse Dirichlet prior for a 3-word vocabulary.

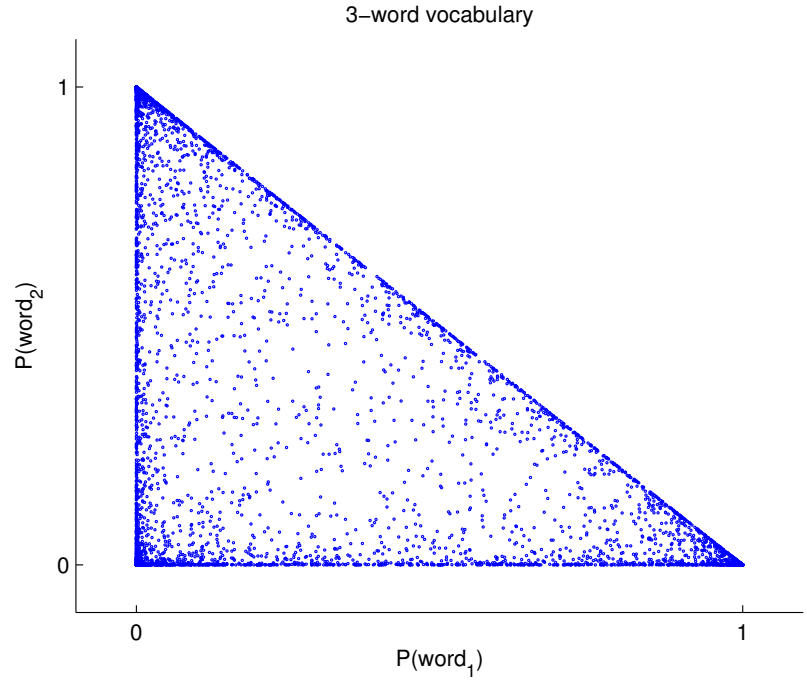

Figure 3.4. Samples from a sparse Dirichlet prior for a 3-word vocabulary.

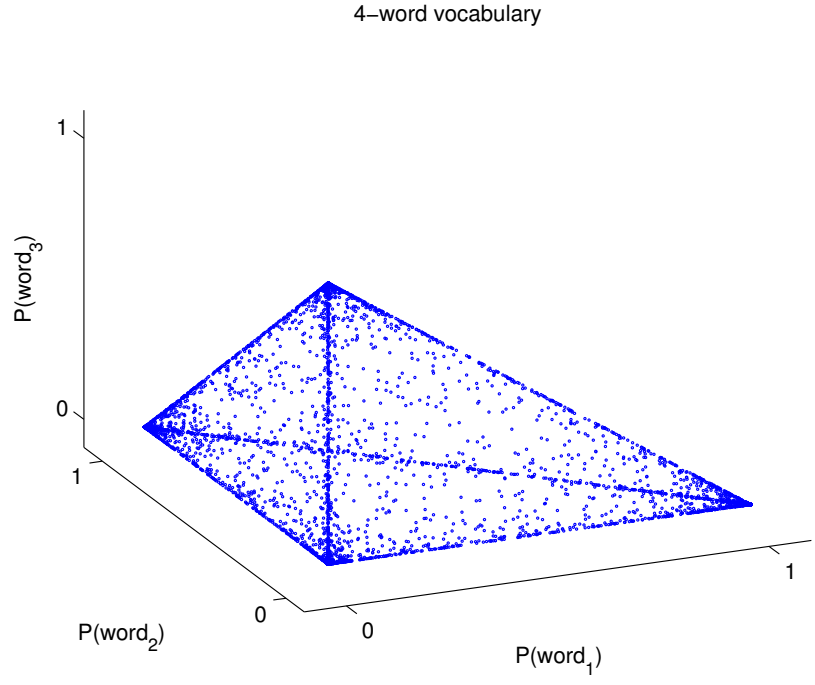

Figure 3.5. Samples from a sparse Dirichlet prior for a 4-word vocabulary.

from the PDF used in Figure 3.3. Similarly, Figure 3.5 shows random samples drawn from a sparse Dirichlet prior for the case where $V_F = 4$.

3.3.3. Inference by Gibbs Sampling

To infer the posterior distribution of the alignments $P(\mathbf{A}|\mathbf{E}, \mathbf{F}; \mathbf{\Theta})$, we use Gibbs sampling [32], a stochastic inference technique that produces random samples that converge in distribution to the desired posterior. In general, for a set of random variables $\mathbf{z} = \{z_j\}$, a Gibbs sampler iteratively updates the variables z_j one at a time by sampling its value from the distribution $P(z_j|\mathbf{z}^{\neg j})$, where the superscript $\neg j$ denotes the exclusion of the variable being sampled.

Before applying Gibbs sampling to our model in (3.4), since we are only after $\mathbf{A}$, we integrate out the unknown $\mathbf{T}$ using:

$$P(\mathbf{F}, \mathbf{A}|\mathbf{E}; \mathbf{\Theta}) = \int_{\mathbf{T}} P(\mathbf{T}; \mathbf{\Theta}) P(\mathbf{F}, \mathbf{A}|\mathbf{E}, \mathbf{T}). \tag{3.5}$$

The remaining set of variables is $\mathbf{z} = \{\mathbf{E}, \mathbf{F}, \mathbf{A}\}$, of which only $\mathbf{A}$ is unknown.

Starting from (3.5), the Gibbs sampling formula is found as (the derivation steps are outlined in the Appendix A):

$$P(a_j = i|\mathbf{E}, \mathbf{F}, \mathbf{A}^{\neg j}; \mathbf{\Theta}) \propto \frac{N^{\neg j}_{e_i, f_j} + \theta_{e_i, f_j}}{\sum_{f=1}^{V_F} N^{\neg j}_{e_i, f} + \sum_{f=1}^{V_F} \theta_{e_i, f}}. \tag{3.6}$$

Here, $N^{\neg j}_{e_i, f_j}$ denotes the number of times the source word type e_i is aligned to the target word type f_j in $\mathbf{A}$, not counting the current alignment link between f_j and e_{a_j}. We can also observe the effect of the prior, where the hyperparameters act as *pseudo-counts* added to N_{e_i, f_j}. Table 3.2 describes the complete inference algorithm. In Step 1,

Table 3.2. Alignment inference algorithm for Bayesian IBM Model 1 using Gibbs sampling.

```
Input: E, F;  Output: K samples of A
1  Initialize A
2  for k = 1 to K do
3      for each sentence pair s in (E, F) do
4          for j = 1 to J do
5              for i = 0 to I do
6                  Calculate P(a_j = i | ···) according to (3.6)
7              Sample a new value for a_j
```

$\mathbf{A}$ can be initialized arbitrarily. However, informed initializations, e.g., EM-estimated alignments, can be used for faster convergence. Once the Gibbs sampler is deemed to have converged after B burn-in iterations, we collect M samples of $\mathbf{A}$ to estimate the underlying distribution $P(\mathbf{A}|\mathbf{E}, \mathbf{F})$. To reduce correlation between these M samples, a lag of L iterations is introduced in-between. Thus the algorithm is run for a total of $K = B + M \times L$ iterations.

The phrase/rule extraction step requires as its input the most probable alignment $\mathbf{A}^* = \arg\max_{\mathbf{A}} P(\mathbf{A}|\mathbf{E}, \mathbf{F})$, which is also called the *Viterbi* alignment. Since $\mathbf{A}$ is a vector with a large number of elements, we make the assumption that the most frequent value for the vector $\mathbf{A}$ can be approximated by the vector consisting of the most frequent values for each element a_j. Hence, we select for each a_j its most frequent value in the M collected samples as the Viterbi alignment.

3.3.4. Interpretation as a Chinese Restaurant Process

We can also view the IBM Model 1 as a Chinese Restaurant Process (CRP). In this analogy, each e has its own separate restaurant. The j-th target word goes to the

restaurant of e_{a_j} and sits at a table according to (K: number of existing tables)

$$P(table_j = k) = \begin{cases} \frac{N_k}{N+\alpha} & \text{if } 1 \leq k \leq K \text{ (sits at an existing table),} \\ \frac{\alpha}{N+\alpha} & \text{if } k = K+1 \text{ (opens a new table).} \end{cases} \tag{3.7}$$

The seating at the tables defines a partition, and the CRP assigns a probability to any such partition. To map these partitions to the space of target-language vocabulary, we introduce labels (dishes) to each table (*labeled* CRP model).

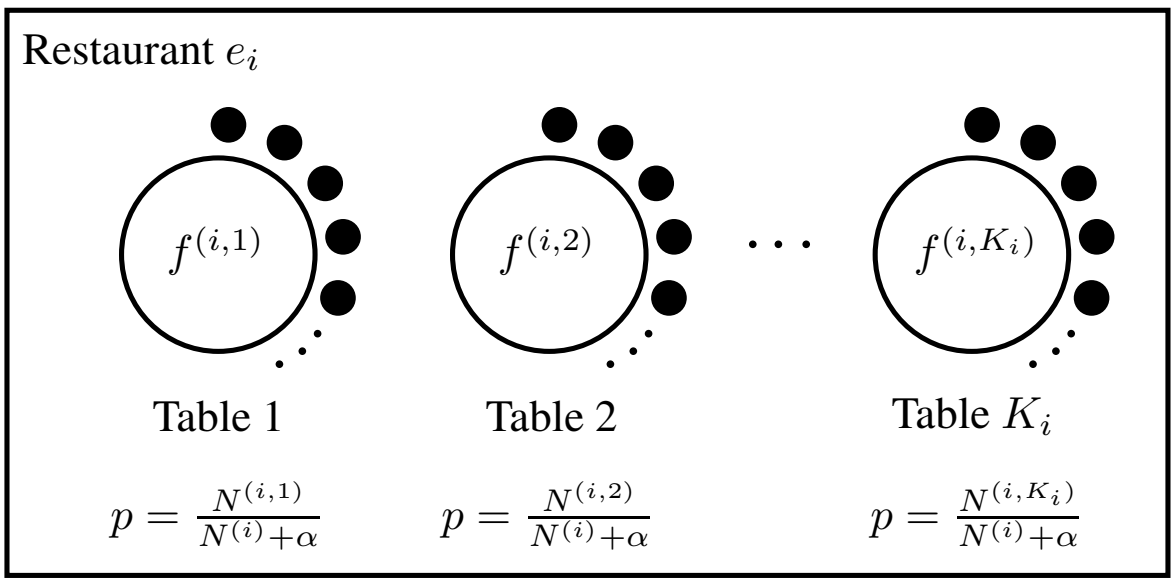

Figure 3.6. Illustration of the Chinese Restaurant Process (CRP) model.

At each table, only one dish is served. Different dishes correspond to different translations of e. When a customer opens a new table, he/she gets to choose the dish for that table from the menu (G_0), and subsequent customers of that table are served that same dish.

When the j-th customer arrives at the restaurant of e_{a_j}, the probability that he is served a particular dish f is given by

$$\begin{aligned} P(dish(j) = f) &= P(\text{sits at an existing table with } f \text{ served on it}) \\ &\quad + P(\text{opens a new table } \textbf{and} \text{ chooses } f \text{ as the dish}) \qquad (3.8) \\ &= \left(\sum_{\substack{k \in \text{ existing tables with} \\ f \text{ served on them}}} \frac{N_k}{N+\alpha} \right) + \frac{\alpha}{N+\alpha} \cdot P(f|G_0) \qquad (3.9) \\ &= \frac{N_f + \alpha \cdot P(f|G_0)}{N+\alpha} \qquad (3.10) \end{aligned}$$

The CRP model given in (3.10) allows unlimited V_F. Being able to account for an unlimited outcome space is especially useful for *hidden* random variables, where the vocabulary cannot be determined beforehand. The CRP model "lets the data choose" the appropriate vocabulary size to account for the observed data. This property is appealing in problems such as mixtures (where the number of mixtures is unknown) and word segmentation (where the underlying word types are unknown).

In the case of finite[3] $|V_F|$, (3.10) and (3.6) are equivalent, since by definition $N_f \equiv N^{\neg j}_{e_{a_j},f}$, $N = \sum_f N_f$, $\sum_f P(f|G_0) = 1$, and hyperparameters $\boldsymbol{\Theta}$ and (α, G_0) are related according to

$$\theta_{e_{a_j},f} = \alpha \cdot P(f|G_0^{(e_{a_j})}) \qquad (3.11)$$

after which (3.10) becomes

$$P(dish(j) = f) = P(f_j|\mathbf{E}, \mathbf{F}^{\neg j}, \mathbf{A}) = \frac{N^{\neg j}_{e_{a_j},f} + \theta_{e_{a_j},f}}{\sum_{f'=1}^{|V_F|} N^{\neg j}_{e_{a_j},f'} + \sum_{f'=1}^{|V_F|} \theta_{e_{a_j},f'}}. \qquad (3.12)$$

[3]Note that the number of tables is still infinite.

It follows from the graphical model topology (Figure 3.2) that the predictive distribution for the alignments $P(a_j|\mathbf{E},\mathbf{F},\mathbf{A}^{\neg j})$ is proportional to $P(f_j|\mathbf{E},\mathbf{F}^{\neg j},\mathbf{A})$ as follows:

$$
\begin{aligned}
P(a_j|\mathbf{E},\mathbf{F},\mathbf{A}^{\neg j}) &= \frac{P(\mathbf{E},\mathbf{F},\mathbf{A})}{P(\mathbf{E},\mathbf{F},\mathbf{A}^{\neg j})} && (3.13)\\
&= \frac{P(\mathbf{E},\mathbf{F}^{\neg j},\mathbf{A}^{\neg j})\cdot P(a_j,f_j|\mathbf{E},\mathbf{F}^{\neg j},\mathbf{A}^{\neg j})}{P(\mathbf{E},\mathbf{F},\mathbf{A}^{\neg j})} && (3.14)\\
&= \frac{P(\mathbf{E},\mathbf{F}^{\neg j},\mathbf{A}^{\neg j})\cdot P(a_j|\mathbf{E},\mathbf{F}^{\neg j},\mathbf{A}^{\neg j})\cdot P(f_j|\mathbf{E},\mathbf{F}^{\neg j},\mathbf{A})}{P(\mathbf{E},\mathbf{F},\mathbf{A}^{\neg j})} && (3.15)\\
&= \frac{P(\mathbf{E},\mathbf{F}^{\neg j},\mathbf{A}^{\neg j})\cdot \frac{1}{|\mathbf{e}|+1}\cdot P(f_j|\mathbf{E},\mathbf{F}^{\neg j},\mathbf{A})}{P(\mathbf{E},\mathbf{F},\mathbf{A}^{\neg j})} && (3.16)\\
&\propto P(f_j|\mathbf{E},\mathbf{F}^{\neg j},\mathbf{A}), && (3.17)
\end{aligned}
$$

the expression for which was already found in (3.12).

3.3.5. Extension to IBM Model 2

IBM Model 1 assumes that all alignments are equally probable, i.e., $P(a_j = i) = (I+1)^{-1}$. In IBM Model 2 [5], the alignment probability distribution $P(a_j)$ for a given target word at position j depends on the quadruple (i,j,I,J). This dependency is parametrized by a distortion parameter d for each quadruple such that

$$P(a_j = i|j,I,J) = d_{i,j,I,J}. \tag{3.18}$$

Note that Model 1 is a special case of Model 2 in which the parameters $d_{i,j,I,J}$ are fixed at $(I+1)^{-1}$.

Different variants of Model 2 have been proposed to reduce the number of parameters, e.g., by dropping dependence on J ($d_{i,j,I}$ [15]) or using relative distortion (d_r where $r = i - \lfloor j\frac{I}{J}\rfloor$ [6], also called "diagonal-oriented Model 2" [33]). In the following, we used the latter parametrization; the derivation for inference in the other variants would be similar.

Bayesian inference in Model 2 can be derived in an analogous manner to Model 1. Treating the set of distortion parameters, denoted by $\mathbf{d} = d_{-\max_s I} \cdots d_{\max_s I}$, as a new random variable, equations (3.2) and (3.4) can be adapted to Model 2 as:

$$P(\mathbf{F}, \mathbf{A}|\mathbf{E}, \mathbf{T}, \mathbf{d}) = \prod_s \prod_{j=1}^{J} (t_{e_{a_j}, f_j} \cdot d_{a_j - \lfloor j\frac{I}{J} \rfloor}) \tag{3.19}$$

$$= \prod_{e=1}^{V_E} \prod_{f=1}^{V_F} (t_{e,f})^{N_{e,f}} \cdot \prod_{r=-\max_s I}^{\max_s I} (d_r)^{C_r}, \tag{3.20}$$

where in (3.20) the count variable C_r stores the number of times a particular relative distortion r occurs in $\mathbf{A}$.

Since $\mathbf{d}$ form the parameters of a multinomial distribution on a_j (see (3.18)), we choose a Dirichlet prior on $\mathbf{d}$:

$$a_j|\mathbf{d} \sim \text{Multinomial}(a_j; \mathbf{d})$$
$$\mathbf{d}|\mathbf{\Phi} \sim \text{Dirichlet}(\mathbf{d}; \mathbf{\Phi}),$$

where $\mathbf{\Phi} = \phi_{-\max_s I} \cdots \phi_{\max_s I}$ are the distortion hyperparameters. Integrating out the parameters $\mathbf{T}$ and $\mathbf{d}$ results in the following Gibbs sampling formula for Bayesian IBM Model 2:

$$P(a_j = i|\mathbf{E}, \mathbf{F}, \mathbf{A}^{\neg j}; \mathbf{\Theta}, \mathbf{\Phi})$$
$$\propto \frac{N^{\neg j}_{e_i, f_j} + \theta_{e_i, f_j}}{\sum_{f=1}^{V_F} N^{\neg j}_{e_i, f} + \sum_{f=1}^{V_F} \theta_{e_i, f}} \cdot (C_r^{\neg j} + \phi_r), \tag{3.21}$$

where $r = i - \lfloor j\frac{I}{J} \rfloor$. A complete derivation is presented in the Appendix. To infer the alignments under Model 2, the only change needed in Table 3.2 is the use of (3.21) instead of (3.6) in step 6.

3.3.6. Parallel Algorithm for Multithreaded Implementation

Normally, the Gibbs sampling algorithm for Equation (3.6) can be implemented in a single processor as shown in Table 3.2. We devised a multi-threaded implementation as an approximation of Gibbs sampling as shown in Table 3.3, where the counts $N_{e,f}$ and C_r are not updated until the end of an iteration. Similar approximations have been done in scaling Gibbs sampling to large datasets using multiple parallel processors, e.g. in [34].

Table 3.3. Multithreaded Gibbs Sampling Implementation.

`main`()
Input: **E**, **F**; Output: K samples of **A**
Initialize **A**
for $k = 1$ to K **do**
for each chunk **in** $(\mathbf{E}, \mathbf{F})$ **do**
Execute `OneThread`(chunk)
for each change **in** ChangeList **do**
Update Counts

`OneThread`(chunk)
for each sentence pair s **in** chunk **do**
for $j = 1$ to J **do**
for $i = 0$ to I **do**
Calculate $P(a_j = i \mid \mathbf{A}^{\neg j}, \mathbf{E}, \mathbf{F})$
Make a random draw for a_j
Add to ChangeList

All large-data experiments reported in Sections 3.4.5 and 3.4.6 have been performed using this multi-threaded implementation.

3.4. Experimental Results

3.4.1. Experimental Setup

We evaluated the performance of the Bayesian word alignment via bi-directional translation experiments. We performed the initial experiments and analyses on small data, then tested the best performing baseline and proposed methods on large data. Furthermore, we performed some of the side investigations and compute-intensive experiments such as those concerning the alignment combination schemes, morphological segmentation, convergence and the effect of sampling settings only on the smallest of the datasets (Turkish↔English).

For Turkish↔English (T↔E) experiments, we used the travel domain BTEC dataset [35] from the annual IWSLT evaluations [36] for training, the CSTAR 2003 test set for tuning, and the IWSLT 2004 test set for testing. For Arabic↔English (A↔E), we used LDC2004T18 (news from years 2001-2004) for training, subsets of the AFP portion of LDC2004T17 (news from year 1998) for tuning and testing, and the AFP and Xinhua subsets of the respective Gigaword corpora (LDC2007T07 and LDC2007T40) for additional LM training. We filtered out sentence pairs where either side contains more than 70 words for Arabic↔English. All language models are 4-gram in the travel domain experiments and 5-gram in the news domain experiments with modified Kneser-Ney smoothing [37] and interpolation. Table 3.4 shows the statistics of the data sets used in the small-data experiments.

For each language pair, we obtained maximum-likelihood word alignments using the EM implementation of GIZA++ [15] and Bayesian alignments using the publicly available Gibbs sampling (GS) implementation [38]. As sampling settings (Section 3.3.3), we used $M = 100$; $L = 10$; and $B = 400$ for T↔E and 8000 for A↔E. We chose identical symmetric Dirichlet priors for all source words e with $\theta_{e,f} = \theta = 0.0001$ to obtain a sparse Dirichlet prior.

After alignments were obtained in both translation directions, standard phrase-based SMT systems were trained in both directions using Moses [39], SRILM [40], and ZMERT [41] tools. The translations were evaluated using the single-reference BLEU [42] metric. Alignments in both directions were symmetrized using the default heuristic in Moses ("grow-diag-final-and"). To account for the random variability in minimum error-rate training (MERT) [43], we report the mean and standard deviation of 10 MERT runs for each evaluation.

We also investigated alignment combination, both within and across alignment methods, to obtain the best possible performance. For this purpose, we obtained three alignment samples from each inference method while trying to capture as much diversity as possible. For EM, we obtained alignments after 5, 20, and 80 iterations (denoted by EM-5, EM-20, and EM-80, respectively). For GS, we ran three separate chains, two initialized with the EM alignments (denoted by GS-5 and GS-80, respectively), and to provide even more diversity, a third initialized based on co-occurrence (denoted by GS-N): Each target word was initially aligned to the source candidate that it co-occurred with the most number of times in the entire parallel corpus.

Table 3.4. Corpus statistics for each language pair in the small-data experiments. T: Turkish, E: English, A: Arabic.

	T / E	A / E
Training set:		
Sentences	20k	56k
Tokens	140k / 183k	1.5M / 1.8M
Tokens/sentence	7.0 / 9.1	27 / 33
Types	18k / 7.3k	80k / 35k
Singletons	10k / 3.2k	35k / 14k
Additional LM tokens	-	215M / 298M
Tuning set sentences	506	873
Test set sentences	500	879

3.4.2. Performance Comparison of EM and GS

Figure 3.7 compares the BLEU scores of SMT systems trained with individual EM- and GS-inferred alignments. In all cases, using GS alignments that are initialized with the alignments from EM leads to higher BLEU scores on average than using the EM alignments directly. In Section 3.5, we investigate the intrinsic differences between the EM- and GS-inferred alignments that lead to the improved translation performance.

Alignment combination across methods (heterogeneous combination) has been previously shown [44, 45] to improve the translation performance over individual alignments. Moreover, alignment combination within a method (homogeneous combination) can also cope with random variation (in GS) or overfitting (in EM).

We implemented alignment combination by concatenating the individual sets of alignments, meanwhile replicating the training corpus, and training the SMT system otherwise the same way. We experimented with various alignment combination schemes and found that combining the EM alignments from 5, 20, and 80 iterations is in general better than the individual alignments, with a similar conclusion for combining the three GS alignments described in Section 3.4.1. Further combination of these two combinations for a total of six alignments sometimes improved the performance even more. So we present the results in this section using these three combination schemes (denoted by EM(Co), GS(Co), and EM(Co)+GS(Co), respectively, in Figure 3.8).

We observe from Figure 3.8 that GS(Co) outperforms EM(Co) on average, both by itself and in combination with EM(Co), in most cases by a significant margin. However, which scheme (GS(Co) or EM(Co)+GS(Co)) is the best seems to depend on the language pair and/or dataset.

3.4.3. Comparison with Variational Bayes

Using the publicly available software [46], we experimented with variational Bayes (VB) inference using similar alignment combination schemes: combination of three

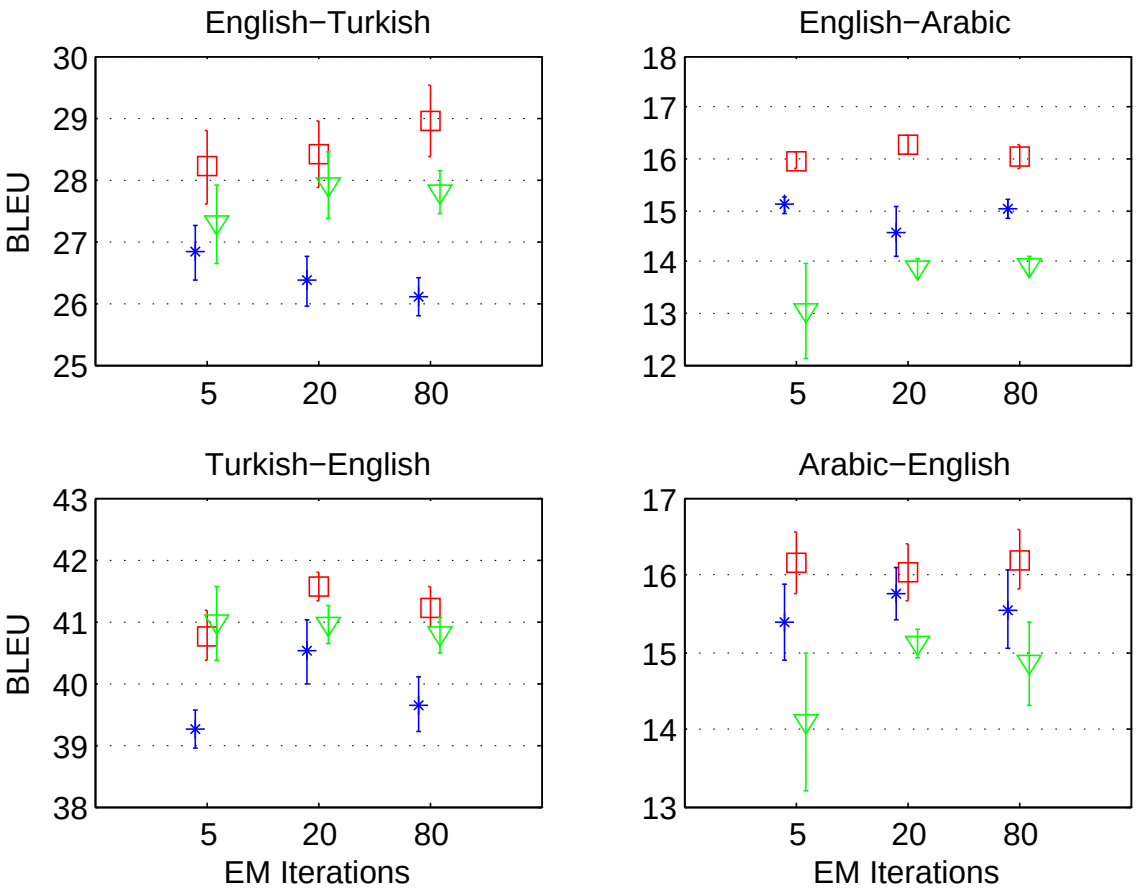

Figure 3.7. Translation performance of word alignments obtained by expectation-maximization (EM), Gibbs sampling initialized with EM (GS) and variational Bayes (VB): $*$ EM, $\square$ GS, $\triangledown$ VB.

VB-inferred alignments after 5, 20, and 80 Model 1 iterations; and further combination of it with the three EM-inferred alignments above (denoted by VB(Co) and EM(Co)+VB(Co), respectively).

The translation peformance of the individual VB alignments in Figure 3.7 shows that, compared to EM, VB achieves higher BLEU scores in T↔E but lower scores in A↔E. On the other hand, GS outperforms VB in all cases but one in Figure 3.7. As for the performance after alignment combination, Figure 3.8 shows that, for all translation directions GS(Co) leads to higher average BLEU scores compared to VB(Co), both with and without further combination with EM(Co). The performance of VB(Co) relative to EM(Co) is similar to the case for individual alignments (better in T↔E, worse in A↔E). However, EM(Co)+VB(Co) outperforms or performs as good as EM in all cases, demonstrating that Bayesian word alignment can be beneficial even with a fast, yet approximate inference method.

To explain the particularly low performance of VB in Arabic↔English, we inspected the alignments inferred by EM, GS, and VB. We found that while VB with sparse Dirichlet prior avoids excessive alignment fertilities, it leaves many rare source words unaligned. For example, the percentage of unaligned source singletons for EM-5, GS-5, and VB-5 in the English→Arabic (Arabic→English) alignments are 27%, 16%, and 69% (44%, 34%, and 71%), respectively. We believe the higher rate of unaligned singletons can lead to poorer training set coverage and lower translation performance (Section 3.5).

3.4.4. Experiments with Morphologically Segmented Corpus

Morphological preprocessing is a common practice in modern SMT systems dealing with morphologically unmatched language pairs. Thus, as a side investigation, we also experimented with morphological segmentation in the T↔E corpus to see its effect on the performance of our proposed method (morphological segmentation is also applied in the large-data A↔E experiments presented in Section 3.4.5). We used the morphological analyzer by Oflazer [47] to segment the Turkish words into lexical mor-

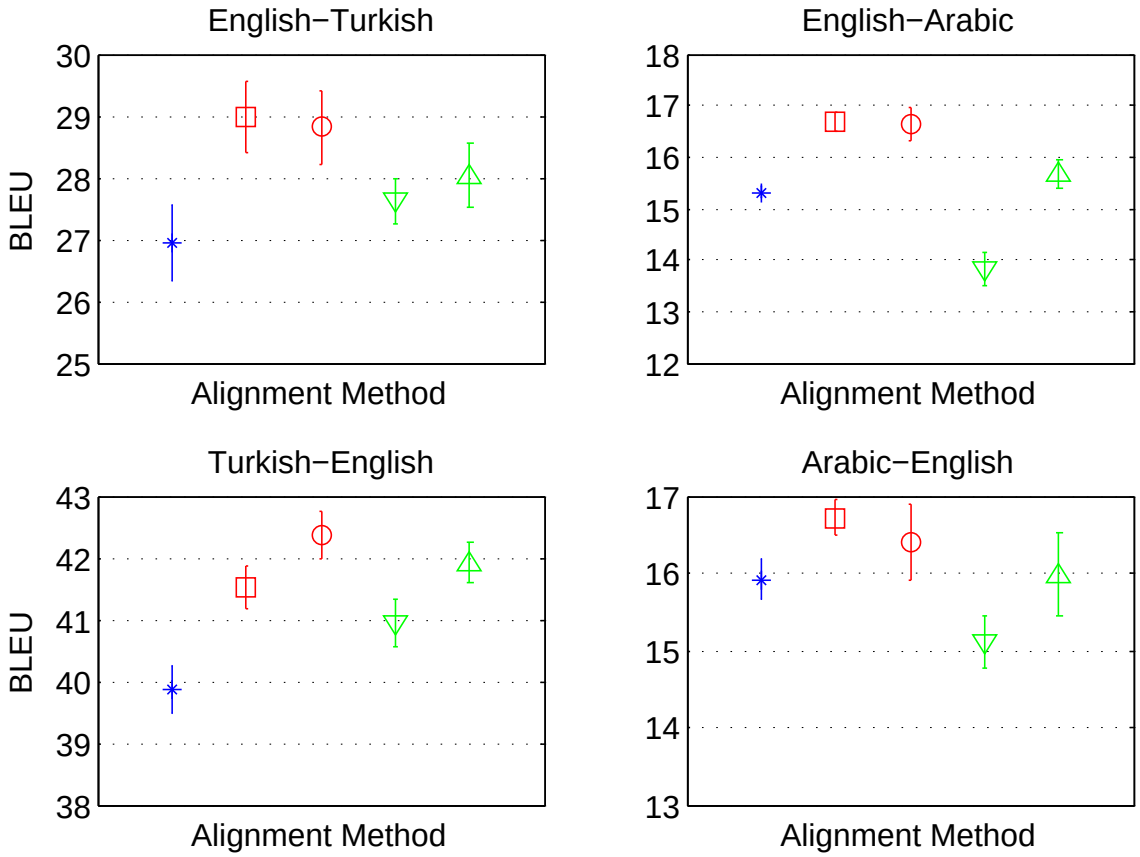

Figure 3.8. Translation performance of EM, GS and VB after applying alignment combination within and across methods: $*$ EM(Co), $\square$ GS(Co), $\circ$ EM(Co)+GS(Co), $\triangledown$ VB(Co), and $\triangle$ EM(Co)+VB(Co).

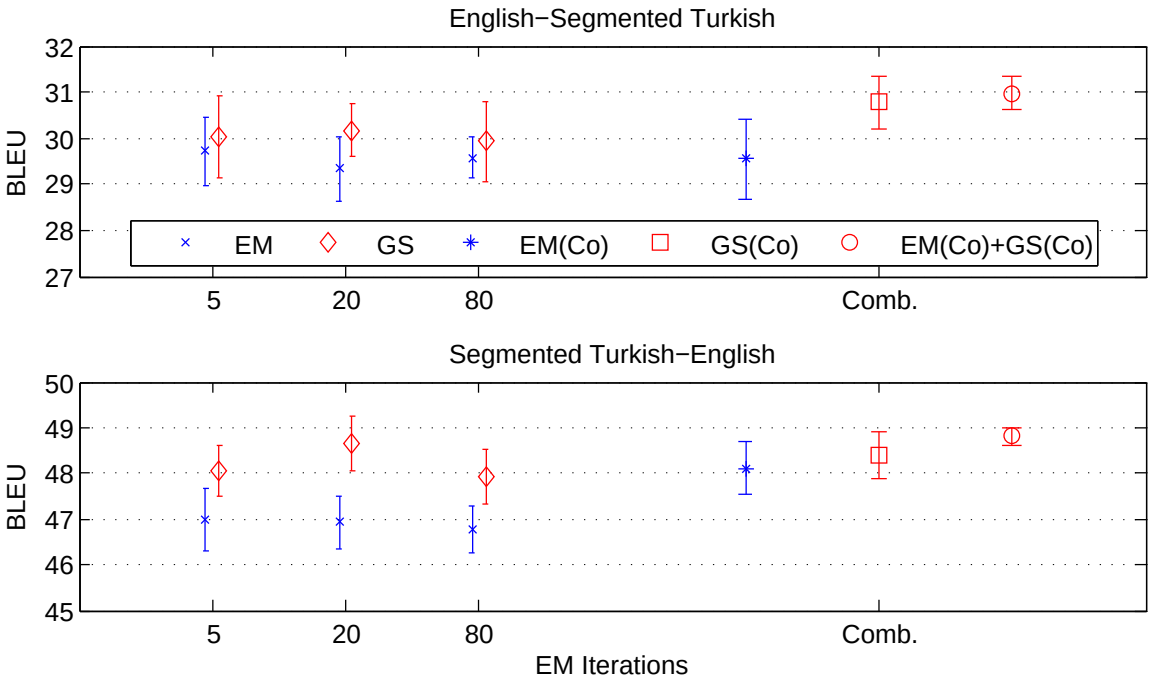

Figure 3.9. Results for the morphologically-segmented Turkish-English corpus. All BLEU scores are computed at the word level.

phemes. As a result, the vocabulary size decreased to 5.6k (from 18k, *cf.* Table 3.4), with 2.4k of them singletons. The out-of-vocabulary rate in the Turkish tuning and test sets decreased from 5.2% and 6.1% to 0.9% and 0.8%, respectively. The BLEU scores were still computed at the word level in the case of English→Turkish translation by joining the morphemes in the output.

The results in Figure 3.9 show that the advantage of GS over EM still holds in the morphologically-segmented condition in both translation directions, both individually and with combination. In addition, comparing the BLEU scores with those in Figs. 3.7 and 3.8 confirms the previous studies that applying morphological segmentation improves the translation performance significantly, especially in the morphologically poorer direction (i.e., Turkish→English).

Table 3.5. Corpus statistics for each language pair in the large-data experiments. A: Arabic, E: English, C: Czech, G: German.

	A / E	C / E	G / E
Training:			
Sentences	7.6M	15.4M	2.0M
Tokens	202M / 203M	203M / 230M	50M / 53M
Types	355k / 342k	1.53M / 1.00M	420k / 139k
LM tokens	- / 241M	265M / 1.05G	477M / 1.05G
Tuning sentences	1000	3003	3003
Test sentences	2000	3003	3003

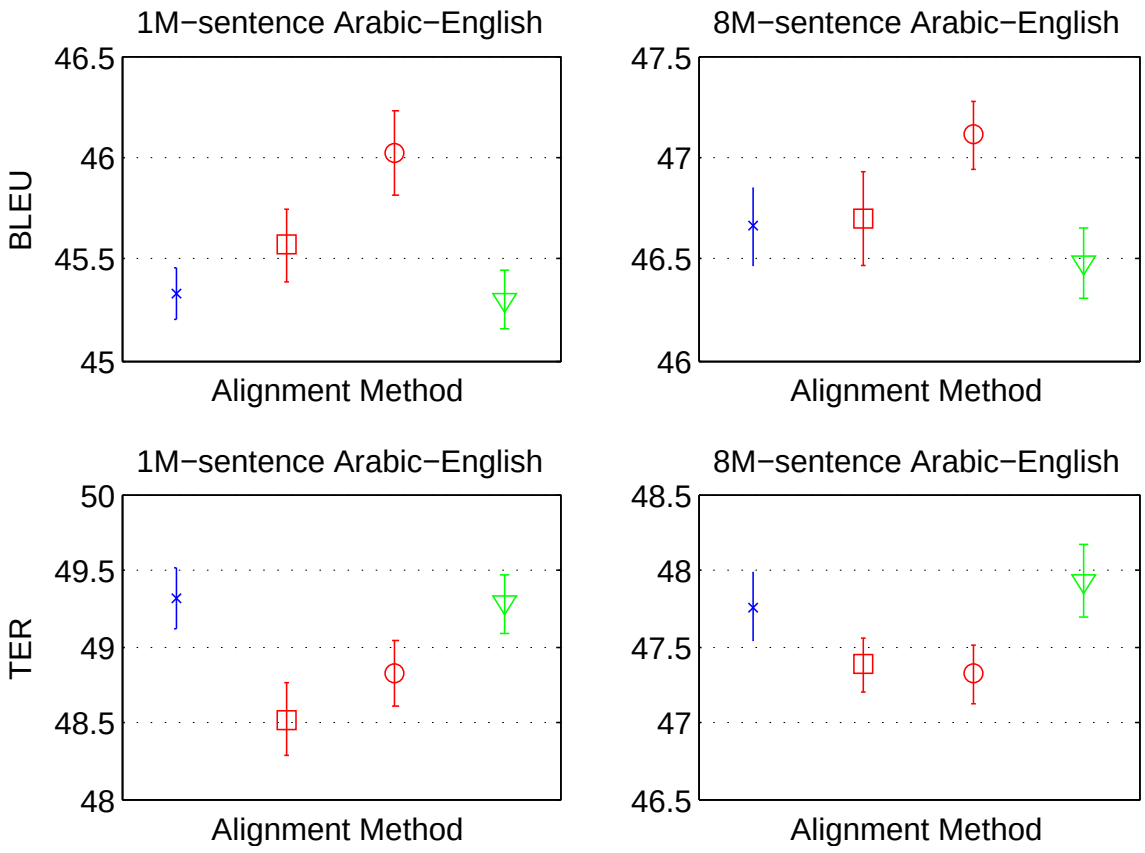

Figure 3.10. Arabic→English BLEU and TER scores of various alignment methods: ∗ EM(Co), □ GS(Co), ○ EM(Co)+GS(Co), and ▽ VB(Co).

3.4.5. Experiments on Larger Datasets

The scalability of the alignment inference methods was also tested on publicly available large datasets (Table 3.5). We used the 8-million sentence Multi-UN corpus [48] for Arabic→English translation experiments. As is common in most state-of-the-art systems for this language pair, we performed morphological segmentation on the Arabic side for the best performance (we used the MADA+TOKAN tool [49]). Note that after morphological segmentation, Arabic no longer exhibits the vocabulary characteristics of a morphologically-rich language (Table 3.5). We set aside the last 100k sentences of the corpus and randomly extracted the tuning and test sets from this subset. The English side of the parallel corpus was used for language model training.

We used the WMT 2012 [50] datasets for Czech ↔ English (C↔E) and German ↔ English (G↔E) translation experiments. The C↔E training data consisted of the Europarl, news commentary, and the 15-million sentence CzEng 1.0 [51] corpora while the G↔E training data consisted of only the Europarl and news commmentary corpora. WMT 2011 and 2012 news testsets were used for tuning and testing, respectively. The WMT 2012 monolingual news corpora covering years 2007–2011 were used for language model training.

In all large-data experiments, sentences longer than 70 words were excluded from translation model training. Gibbs sampling settings of (B, M, L) = (1000, 100, 1) were used. All language models were 4-gram. To obtain the best possible baseline, we also utilized techniques that we had previously observed to improve performance on similar corpora, such as lattice sampling [52] and search in random directions [53] during MERT and minimum Bayes risk decoding [54]. All other experimental settings (e.g., 10 MERT runs etc.) were identical to the small-data experiments (Section 3.4.1).

To conform with the majority of previous research and evaluations in these language pairs, we trained SMT systems in both directions for the WMT 2012 language pairs and in the Arabic→English direction for the Multi-UN task. For the two largest datasets (C↔E and A→E), we also experimented with 1-million sentence versions for

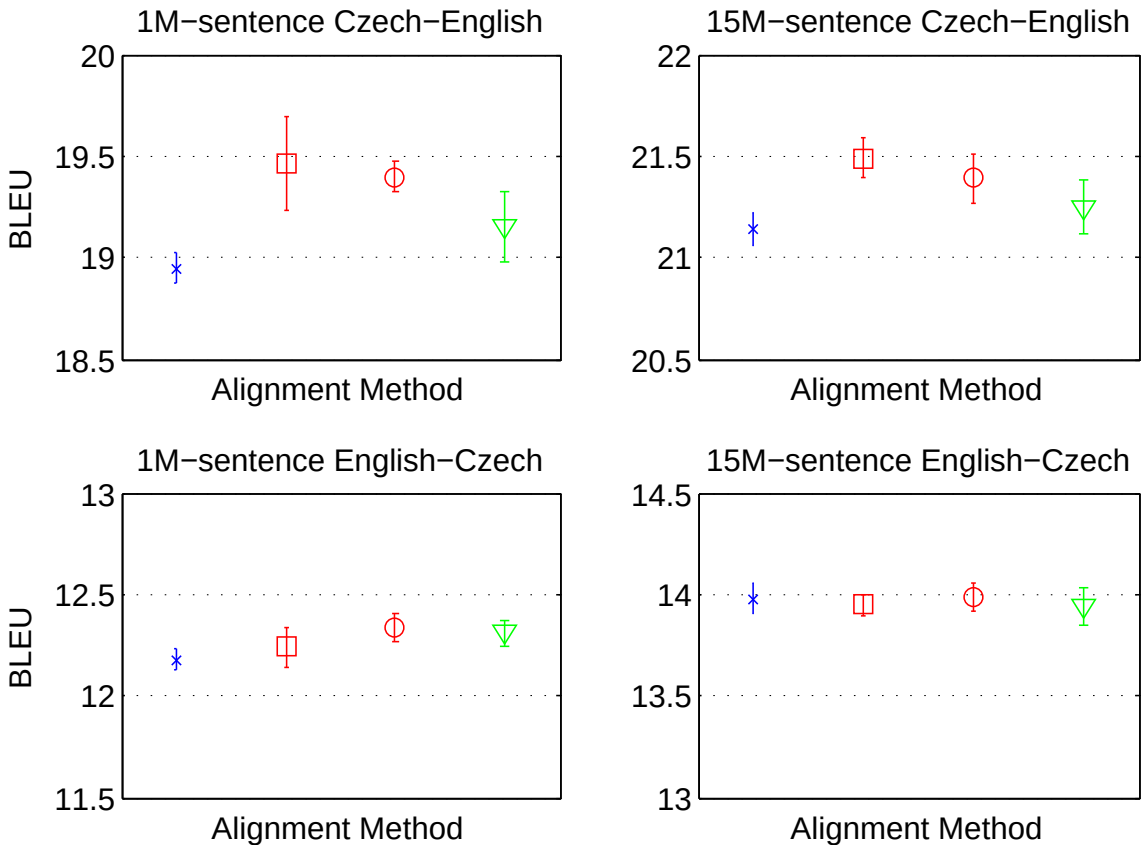

Figure 3.11. Czech↔English BLEU scores of various word alignment methods: ∗ EM(Co), □ GS(Co), ○ EM(Co)+GS(Co), and ▽ VB(Co).

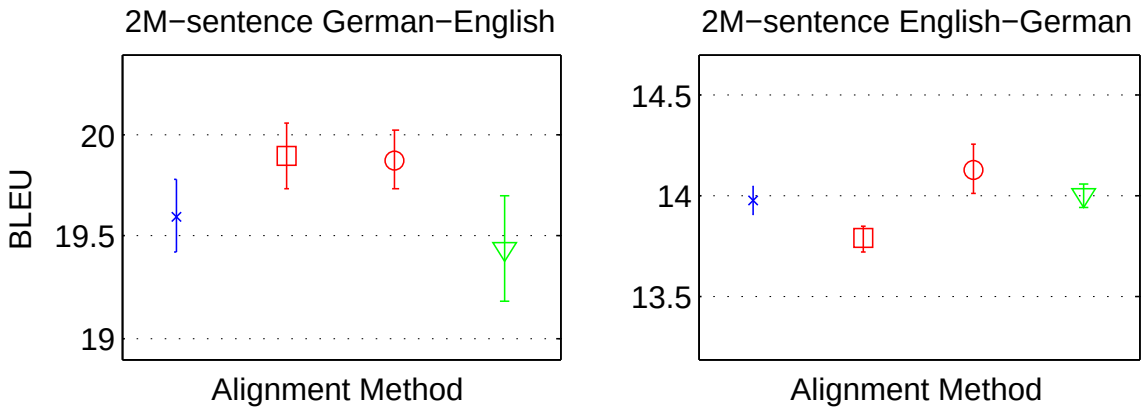

Figure 3.12. German↔English BLEU scores of various word alignment methods: ∗ EM(Co), □ GS(Co), ○ EM(Co)+GS(Co), and ▽ VB(Co).

faster development experiments and to provide an intermediate data size setting.

The results are presented in Figs. 3.10-3.12. For translation *to* English, Gibbs sampling improves over EM for all five corpora, the largest improvement achieved by GS(Co)+EM(Co) in A→E (0.5 to 0.7 BLEU mean difference) and by GS(Co) in C→E and G→E (0.3 to 0.5 BLUE mean difference). However, for translation *from* English (E→C and E→G), we do not observe a consistent improvement over EM.

For the 1-million sentence A→E task, we also report the translation error rates (TERs) [55] (bottom row of Figure 3.10). Except for the comparison between GS(Co) and EM(Co)+GS(Co) in the 1M-sentence setting, in all possible pair-wise comparisons between the alignment methods in both corpus settings, the method with the higher mean BLEU score also has the lower mean TER score[4] .

In addition, we compared the performance of some of the many possible alignment combination schemes (Figure 3.13). Not surprisingly, combination with EM(Co) helps both GS(Co) and VB(Co), and the relative ranking of the latter two does not change after combination with EM(Co). Furthermore, combination of GS(Co)+VB(Co) improves the performance slightly over EM(Co)+GS(Co).

3.4.6. Bayesian Model 2 Results

We tested the IBM Model 2 Gibbs sampling algorithm on the 1M-sentence subset of the Arabic-English Multi-UN corpus. Unlike the case of translation parameters $\mathbf{T}$, there is no clear language- and domain-independent knowledge of how the distortion parameters $\mathbf{d}$ (the distribution of a_j) should look like. Therefore, we assumed that all distortion distributions are *a priori* equally probable, which corresponds to setting the distortion hyperparameters $\phi_r = 1$ for all r. We also collapsed the counts for distortions larger in magnitude than 5, resulting in 11 total distortion count variables $N_{r\leq -5}, N_{-4}, \cdots, N_4, N_{r\geq 5}$, as done in [7].

[4]BLEU was used as the error metric for optimization in MERT.

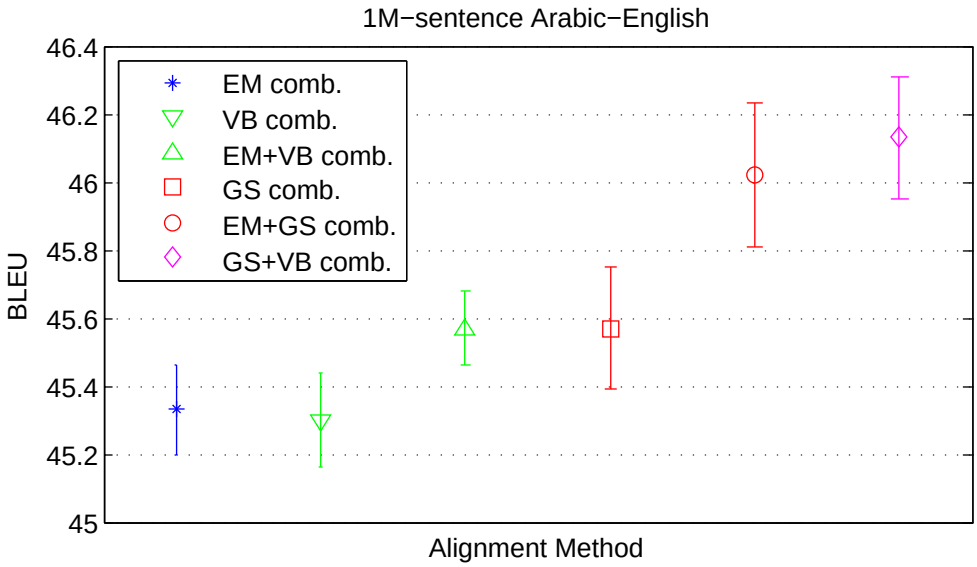

Figure 3.13. Arabic→English BLEU scores of various alignment combination schemes in the 1M-sentence translation task.

Table 3.6. BLEU scores of IBM Model 2 alignment inference methods on the 1M-sentence Arabic→English translation.

Method	Model 2 EM	Model 2 GS
BLEU	46.97 $\pm$*0.15*	47.17 $\pm$*0.14*

We compared the translation performance of the EM- and GS-inferred Model 2 alignments. Both methods are initialized with the same EM-5 alignments (i.e., 5 iterations of Model 1 EM). Model 2 EM is run for 5 iterations. Model 2 GS is estimated with $B = 1000$, $M = 100$ and $L = 1$. The results are shown in Table 3.6. Bayesian inference improves the mean BLUE score by 0.2 BLEU. Further improvement could be possible by alignment combination within and across methods, as done in Section 3.4.2.

3.5. Alignment Analysis

In order to explain the BLEU score improvements achieved by the Bayesian alignment approach and to characterize the differences between the alignments obtained by various methods, we analyzed the alignments in Figure 3.7 using several intrinsic and extrinsic evaluation metrics. As representative alignments from each method, we selected EM-5, VB-5, and GS-5.

3.5.1. Fertility Distributions

Fertility of a source word is defined as the number of target words aligned to it. In general, we expect the fertility values close to the word token ratio between the languages to be the most frequent and high fertility values to be rare. Figure 3.14 shows the fertility distributions in alignments obtained from different methods. We can observe the "garbage collecting" effect in the long tails of the EM-estimated alignments. For example, in English-Arabic Model 1 alignment using EM, 1.2% of the English source tokens are aligned with *nine or more* Arabic target words, corresponding to 22.3k total occurrences or about 0.4 occurrence per sentence. In all alignment tasks, both Bayesian methods result in fewer high-fertility alignments compared to EM. Among Bayesian inference techniques, GS is more effective than VB in avoiding high fertilities.

3.5.2. Alignment Dictionary Size

Reducing the number of unique alignment pairs has been proposed as an objective for word alignment [56,57]: it was observed during manual alignment experiments that

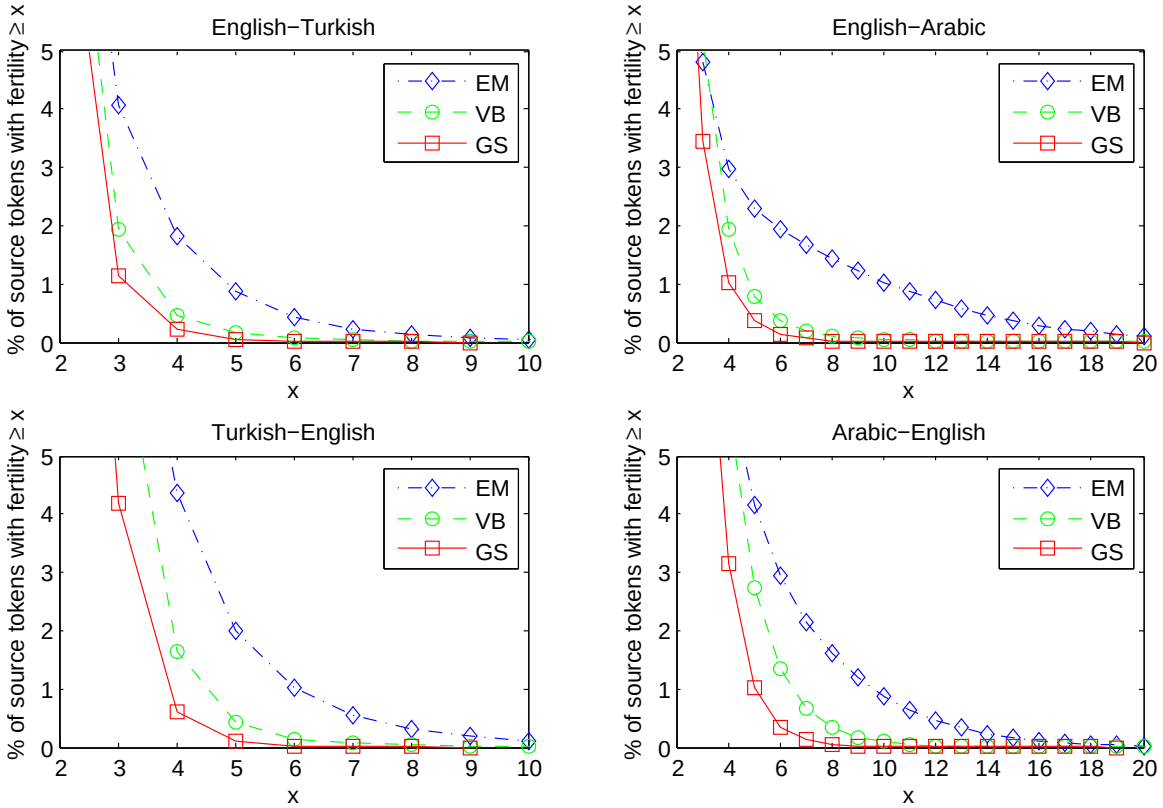

Figure 3.14. Distribution of alignment fertilities for source language tokens.

humans try to find the alignment with the most compact "alignment dictionary" (a vocabulary of unique source-target word pairs) as possible. Figure 3.15 shows that both GS and VB explain the training data using a significantly smaller alignment-pair vocabulary compared to EM.

3.5.3. Singleton Fertilities

The average alignment fertility of source singletons was proposed as an intrinsic evaluation metric in [45]. We expect lower values to correlate with better alignments. However, a value of zero could be achieved by leaving all singletons unaligned, which is clearly not desirable. Therefore, we refine the definition of this metric to calculate the average over *aligned* singletons only. The minimum value thus attainable is one. Figure 3.16 shows that both Bayesian methods significantly reduce singleton fertilities.

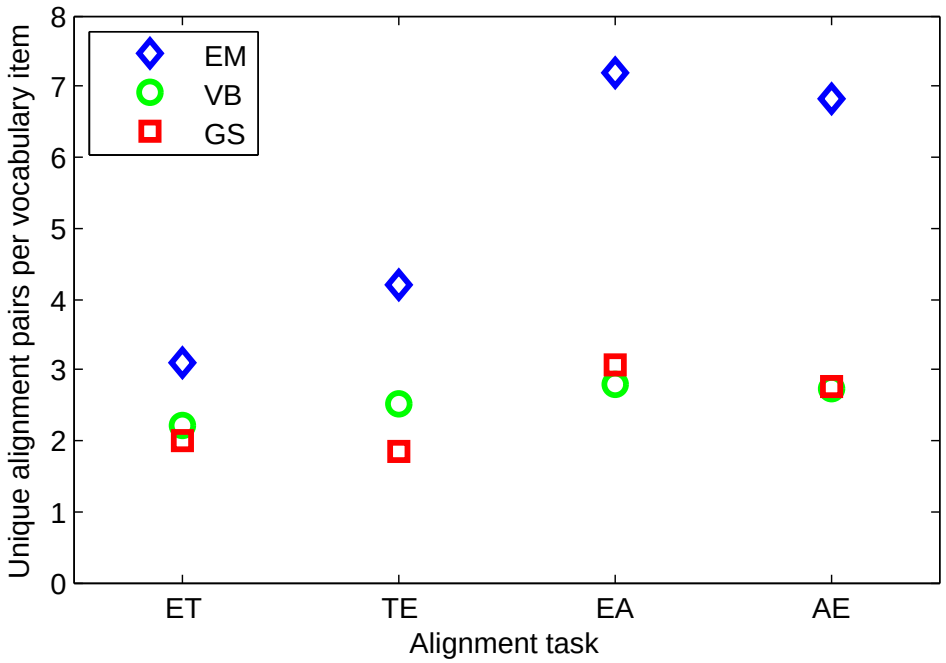

Figure 3.15. Alignment dictionary size normalized by the average of source and target vocabulary sizes.

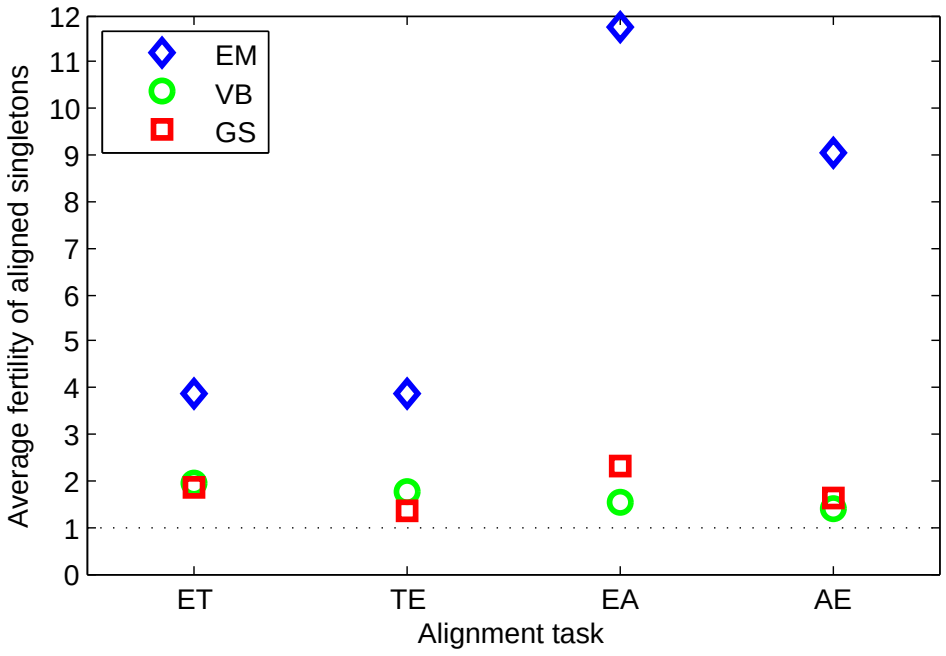

Figure 3.16. Average alignment fertility of aligned singletons.

The average fertility of aligned singletons by itself is not sufficient to accurately assess an alignment since unaligned singletons are not represented. Hence, we also report the percentage of unaligned singletons in Figure 3.17. GS has the lowest unaligned singleton rate among Model 1 inference methods. An interesting observation is that, while EM-estimated alignments suffer from rare words being assigned high fertilities (Figure 3.16), VB suffers from a high percentage of the rare words (e.g., about 70% of singletons in A↔E) being left unaligned, resulting in lower translation performance (Section 3.4.3). Our analysis agrees with the findings of Guzman *et al.* [58] that unaligned words in an alignment results in lower-quality phrase tables.

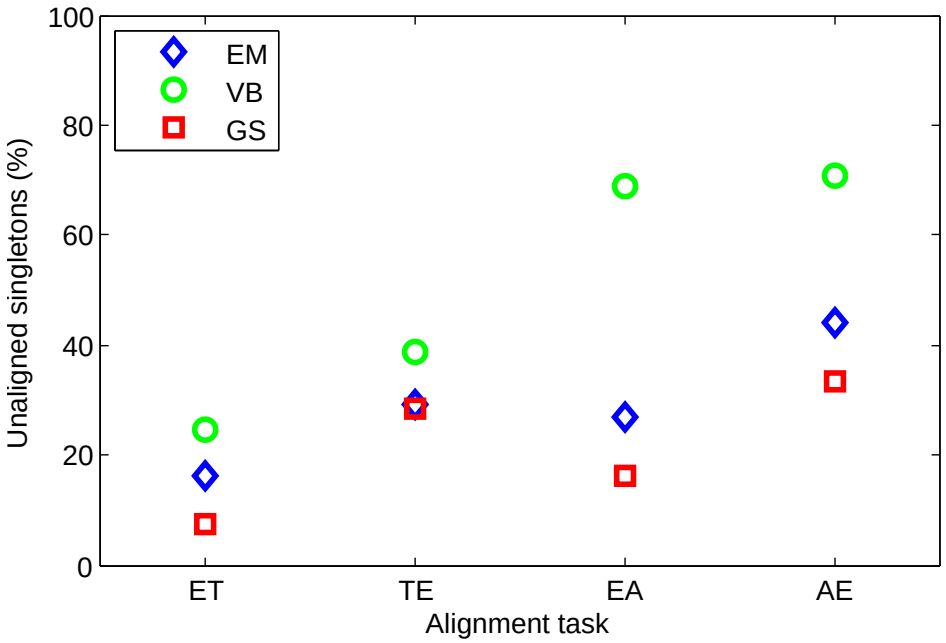

Figure 3.17. Percentage of unaligned singletons.

3.5.4. Alignment Points in Agreement

Since the IBM alignment models are one(source)-to-many(target), switching the source and target languages usually result in a different set of alignment links (or points in an alignment matrix). The intersection of the two sets consists of high-precision alignment points where both alignment models agree [7]. Since the number

of alignment points in each direction is constant (equal to the number of target words), increasing precision at the expense of recall by predicting fewer alignment points is not applicable in these models. Therefore higher agreement rate implies not only higher precision but higher recall as well. Figure 3.18 shows that GS has the highest alignment agreement rate among the alignment methods for both language pairs.

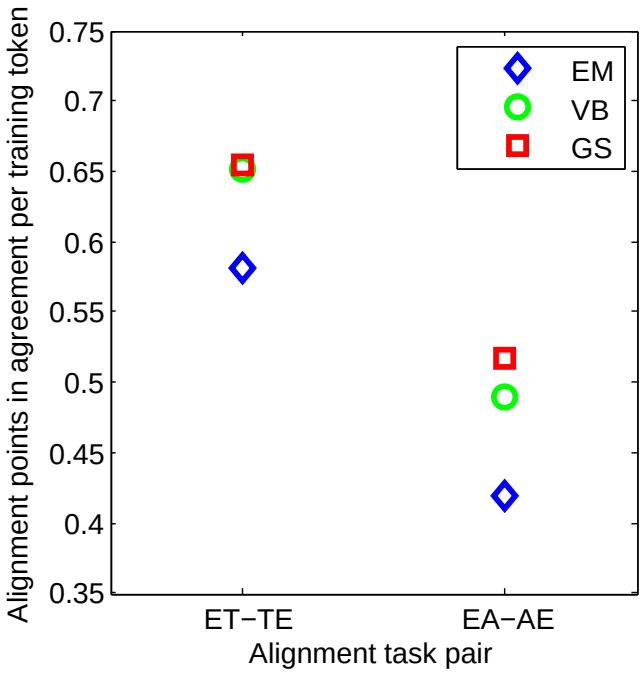

Figure 3.18. Number of symmetric alignments normalized by the average of source and target tokens.

3.5.5. Training Set Vocabulary Coverage by Phrase Table

We can also evaluate the inferred alignments extrinsically, e.g., by evaluating the SMT systems trained using those alignments. A desirable feature in a SMT system is to have as high vocabulary coverage as possible. This metric is highly sensitive to the performance of an alignment algorithm on infrequent words since they represent the majority of the vocabulary of a corpus (see Table 3.4). Figure 3.19 shows that alignment by GS leads to the best vocabulary coverage in all four alignment tasks. Note that word types that appear in the phrase table only as part of larger phrase(s) are excluded

from this metric, since such words are practically out-of-vocabulary (OOV) except only in those specific contexts.

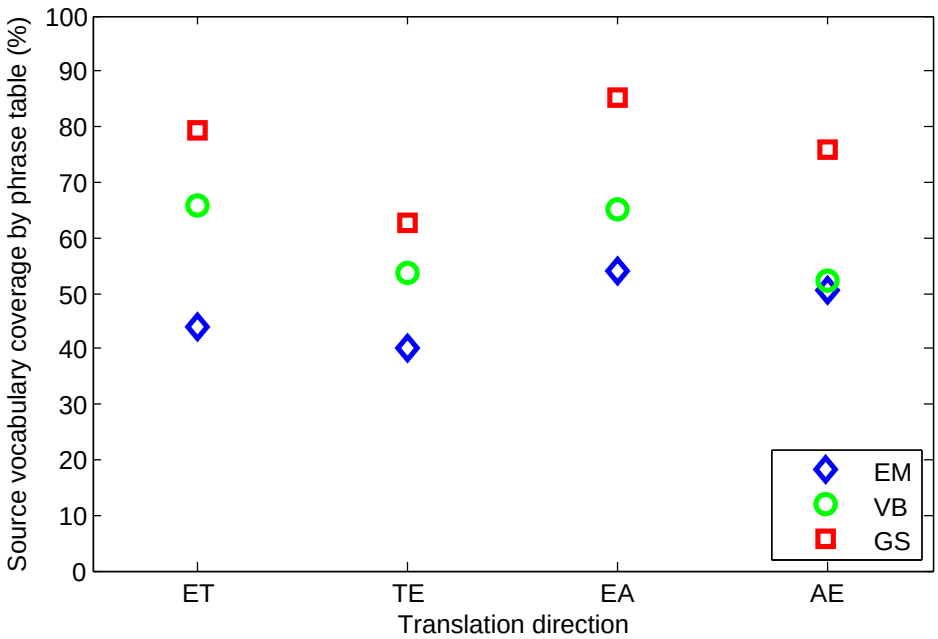

Figure 3.19. Percentage of training set vocabulary covered by single-word phrases in the phrase table.

Poor training set vocabulary coverage results in some non-OOV words being treated by the system as OOV, either dropping them from the output or leaving them untranslated. Such *pseudo-OOV* words further degrade the translation performance in addition to the OOV words. Figure 3.20 shows that GS alignments lead to the lowest rate of pseudo-OOV words.

3.5.6. Phrase Table Size

In most machine translation applications, having a small model size is valuable, e.g., to reduce the memory requirement or the start-up/access time. Alignment methods can affect the induced phrase table sizes. Figure 3.21 compares the number of phrase pairs in the SMT systems trained by different alignment methods. In the A↔E task, where model size is of more concern compared to the smaller T↔E task, GS results in significantly smaller phrase tables. This result is particularly remarkable since

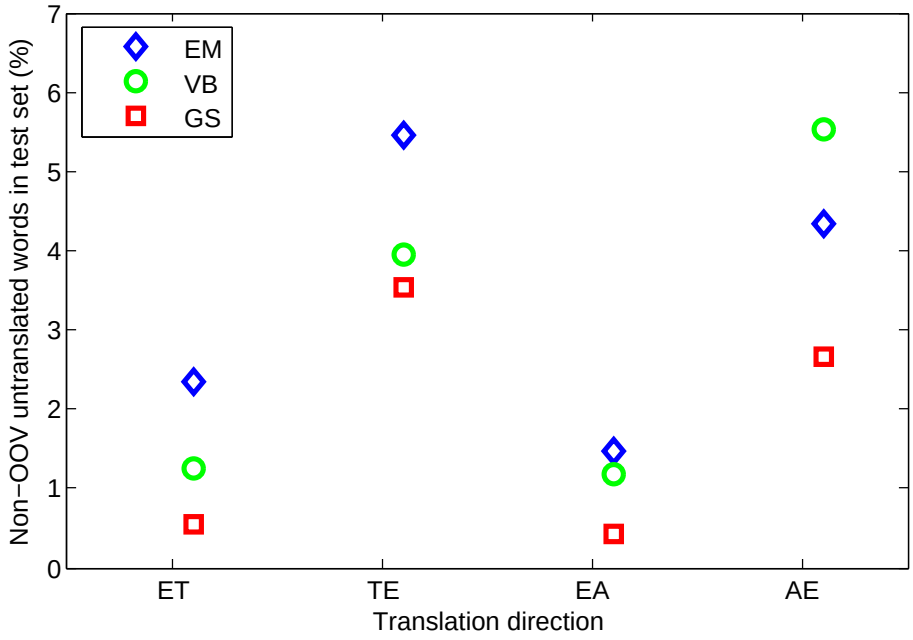

Figure 3.20. Decode-time rate of input words that are in the training vocabulary but without a translation in the phrase table.

it means that a system using GS-inferred alignments achieves more vocabulary coverage (Section 3.5.5) and higher BLEU scores (Section 3.4.2) with a smaller model size. Thanks to a larger intersection during alignment symmetrization (Figure 3.18), GS-based phrase tables contain a higher number of single-word phrase pairs (Figure 3.19). Moreover, fewer unaligned words after symmetrization lead to fewer poor-quality long phrase pairs [58].

3.5.7. Alignment Error Rate

Table 3.7 shows the alignment error rates (AERs) [15] obtained in the C↔E alignment tasks using a publicly available 515-sentence manually-aligned reference set [59]. The Bayesian methods achieve better AERs than EM in both alignment directions (denoted by "EC" and "CE"). Contrary to the ranking of the methods according to BLEU (Figure 3.11), VB achieves the best AER, which also holds true after symmetrization (denoted by "Sym."). Furthermore, the symmetrized GS-5 alignment has the worst AER in the 1M-sentence experiment. These discrepancies support earlier findings by

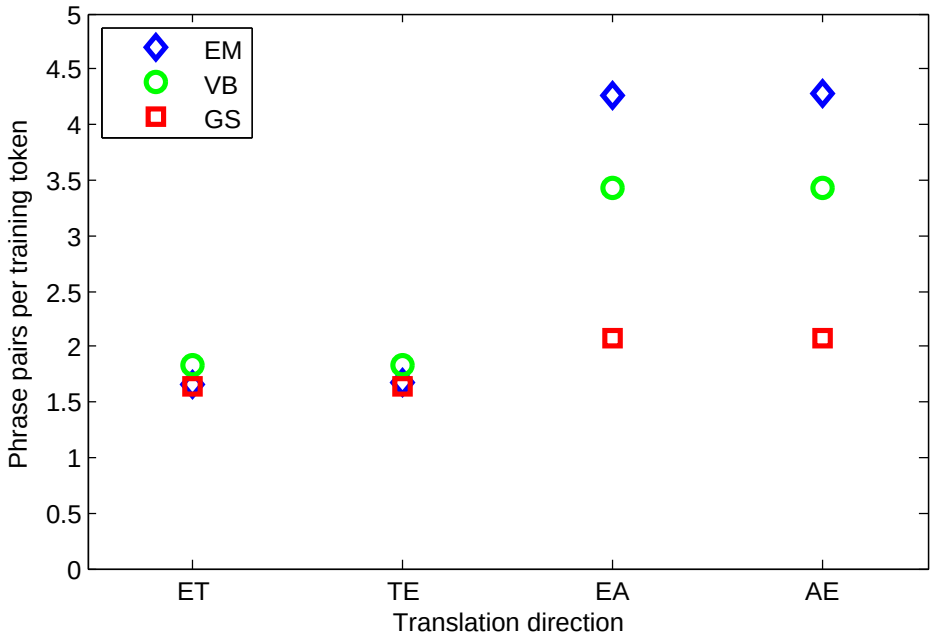

Figure 3.21. Phrase table size normalized by the average of source and target tokens.

Table 3.7. Alignment error rate (%) of the uni-directional and symmetrized Czech-English alignments.

Training set	1M sentences			15M sentences		
Direction	EC	CE	Sym.	EC	CE	Sym.
EM-5	45.1	41.4	30.9	40.6	38.4	27.7
GS-5	41.9	40.0	31.6	36.4	34.9	26.7
VB-5	37.8	36.5	28.9	31.9	32.1	24.1

several others that AER is generally not a good predictor of BLEU performance [60].

As a final remark, in Table 3.7 EM-5 enjoys a larger amount of reduction in AER via symmetrization compared to GS-5, which suggests the possibility that the default alignment symmetrization heuristic in Moses ("grow-diag-final-and") has been fine-tuned for the default EM-based alignments, and thus other symmetrization/phrase extraction methods might work better for the GS- and VB-based alignments. For example, Bayesian alignment inference could be complemented with a probabilistic model of phrase extraction, e.g. [27], which is left as a future work.

3.6. Sampling Analysis

3.6.1. Effect of Sampling Settings

We investigated the effect of changing the sampling settings B, M, and L (Section 3.3.3) on T↔E GS-N alignments. To account for the variability due to the randomness of the sampling process, we present in Figs. 3.22 and 3.23 the mean and the standard deviation of BLEU scores over eight separate chains with different random seeds. At each B value shown, eight separate SMT systems were trained. These eight runs each comprise a separate MERT run, thus error bars in Figs. 3.22 and 3.23 also include the variation due to MERT.

Figure 3.22 shows the effect of changing B with $M = 100$ and $L = 1$. In this experiment, the sampler converges after roughly a few thousand iterations. Comparing the BLEU scores in Figure 3.22 to those of the three EM-initialized samplers in Figure 3.7, where $B = 400$, for the same language pair suggests that running more iterations of Gibbs sampling can compensate for poor initializations, or equivalently, initializing with EM alignments can provide a head start in the convergence of the Gibbs chain.

Figure 3.23 compares the effect of different read-out schemes. The (M, L) settings of both $(1000, 1)$ and $(100, 10)$ collect samples over the same 1000-sample interval. We

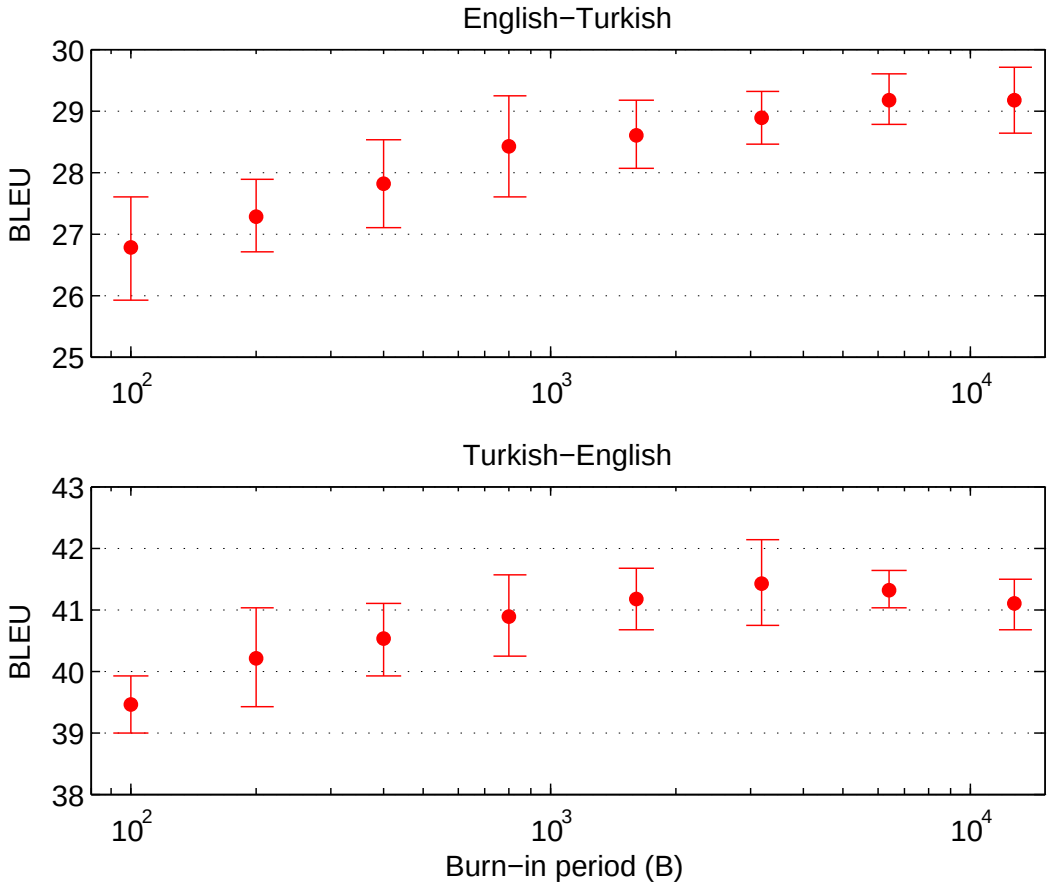

Figure 3.22. BLEU scores obtained by changing B while M=100 and L=1 (Section 3.3.3). Averages and standard deviations are over 8 separate Gibbs chains.

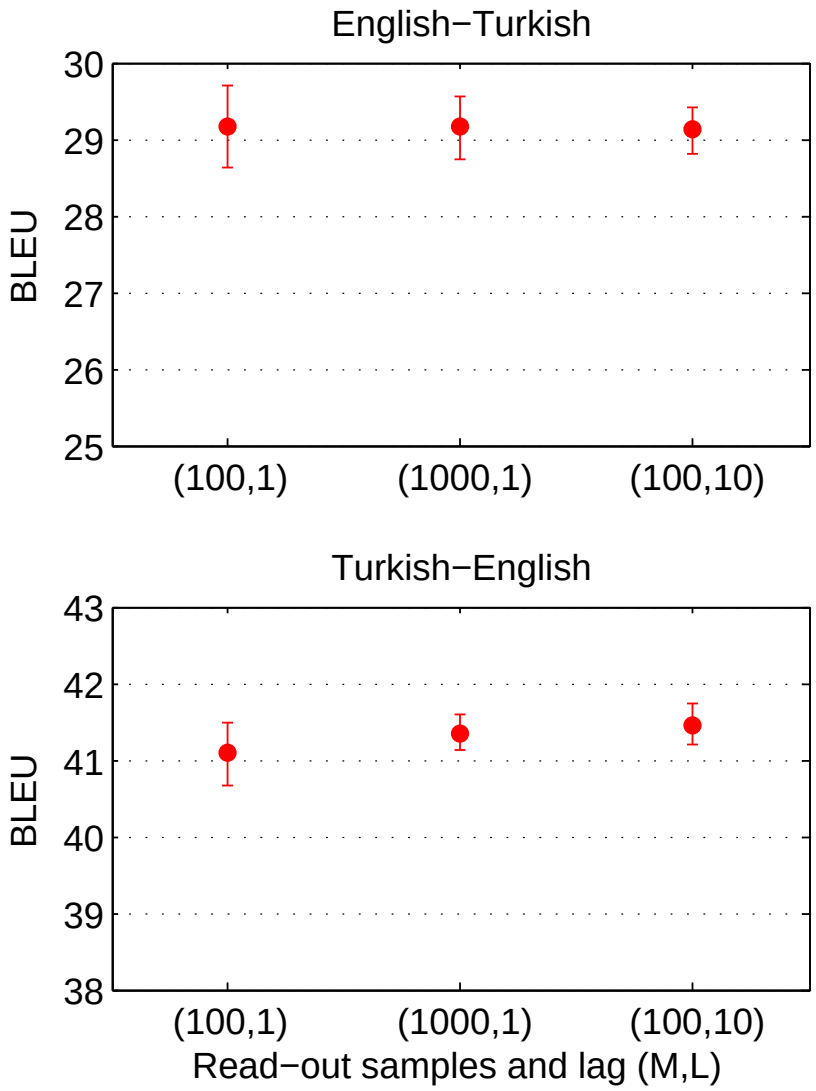

Figure 3.23. BLEU scores obtained by changing M and L while B=12800 (Section 3.3.3). Averages and standard deviations are over 8 separate Gibbs chains.

can deduce from their comparison in Figure 3.23 that including or discarding the intermediate samples does not make a significant difference. On the other hand, comparing the settings $(100, 1)$ and $(1000, 1)$ confirms our intuition that increasing the number of samples (M) leads to more reliable (smaller variance) estimates of the Viterbi alignments.

3.6.2. Convergence and Variance Between Iterations

Figure 3.24 compares the change in BLEU scores as iterations progress during both EM and GS. A separate SMT system is trained at each shown data point on the plots. Each dot in the graphs correspond to a separate SMT system trained and optimized from the alignment estimated at that iteration. In the figure, there are two main sources of BLEU score variation between the iterations: updated alignments at each iteration and randomness due to MERT.

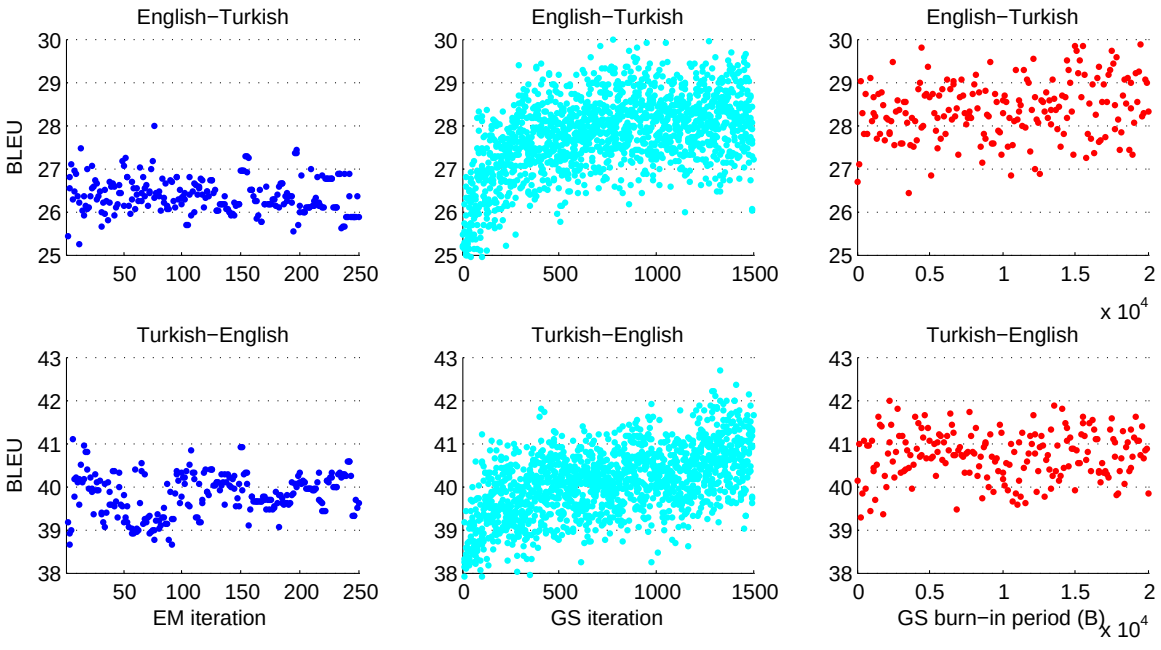

Figure 3.24. BLEU scores of alignments estimated at different iterations. Left: EM, middle: samples from the Gibbs chain, right: GS viterbi estimates with $M = 100, L = 1$. Note the difference in x-axis scales.

Comparing the BLEU scores of sample and Viterbi alignments obtained by GS, we observe smaller variance and higher average BLEU scores using Viterbi alignments. Compared to EM, GS achieves higher average BLEU scores, albeit with a larger amount of variation between iterations due to the random nature of sampling. To reduce the variation, a larger value of M (Section 3.6.1) and/or a combination of alignments at different iterations can be used.

3.6.3. Computational Complexity

The computational complexity of the Gibbs sampling algorithm in Table 3.2 is linear in the number of sentences and roughly quadratic in the average number of words per sentence. Running Gibbs sampling (Model 1) on the largest of our datasets, the 15.4M-sentence Czech-English corpus, takes on average 33 seconds per iteration (steps 3–7 in Table 3.2) using 24 threads on a 3.47GHz Intel Xeon X5690. In the case of Model 2, the average time per Gibbs sampling iteration increases to 48 seconds. For comparison, a Model 1 EM iteration on the same hardware and number of threads using MGIZA [61] takes 326 seconds on average (excluding pre-processing and initializations). In the case of Model 2, for which multi-threading is not implemented in MGIZA, an EM iteration took 1960 seconds on average. These results are summarized in Table 3.8.

Table 3.8. Execution time on 15.4 M sentence Czech-English dataset.

	Avg. EM iteration	Avg. GS iteration
Model 1	326 s @ 24 cpus	33 s @ 24 cpus
Model 2	1960 s @ 1 cpu	48 s @ 24 cpus

3.7. Lowering Variance in BLEU Scores

3.7.1. Motivation

All experiments in Section 3.6, particularly those in Figure 3.24, show large BLEU variations that reduce the signifance of the attained average improvements. One source

of variations is the random nature of Gibbs sampling. There is also noise stemming from factors unrelated to alignment such as parameter optimization and test set variance. In the following sub-sections, we attempt to reduce these two types of variations.

3.7.2. Alignment Combination

An advantage of sampling approaches that we can leverage in our application is the availability of several outputs. After convergence, since we expect different samples to be probabilistically equivalent, some of the changes from one sample to the next are expected to be constructive while some destructive. In other words, alignments obtained at different iterations are expected to have a complementary nature that can be utilized in a combination approach. These suggest that alignment combination among samples from a *single* Gibbs chain could be beneficial. We also investigate alignment combination among samples from *multiple* Gibbs chains that are started from different initializations.

Alignments obtained using different methods (such as GS and EM) could also have a complementary nature. Therefore, aside from the above two *homogeneous* combination methods where the components are obtained using the same alignment method, we also investigate *heterogenous* combination where the alignments are obtained using different alignment methods.

Table 3.9 shows the typical stages of a SMT system training pipeline where alignment combination can be applied. We experimented with various combination methods as described below:

Method A: Apply Step 1 separately to each input alignment, then obtain a single alignment by taking the union of all input alignment points. Then continue from step 2 using this combined alignment.

Method B: Apply steps 1 and 2 separately to each input alignment, then combine the set of extracted phrases. Continue from step 3 using this combined phrase list.

Table 3.9. Steps in phrase-based SMT training pipeline where alignment combination can be applied.

Step 1.	Symmetrize word alignments obtained in two directions using *grow-diag-final* heuristic.
Step 2.	Extract phrases that are consistent with the symmetrized word alignment.
Step 3.	Obtain phrase-level features by calculating conditional probability estimates from co-occurrence counts in the training corpus. The output of this step is a so-called *phrase table* with five features for each phrase.
Step 4.	Optimize the weights of these phrase-level features (along with other features such as LM etc.).
Step 5.	Decode a given test sentence using the phrase table and the optimized weights.

Method C: Apply steps 1–3 separately to obtain multiple different phrase tables. These phrase tables may have overlapping phrases, but some of their feature values will be different. Use all these phrase tables in weight tuning and decoding, effectively increasing the number of features.

Method D: Replicate the training corpus multiple times, applying a different input alignment for each replica. Then apply steps 1–5 on this synthetically larger training corpus.

Table 3.10 compares the above four methods against the individual alignments on the Turkish-English (TE) task and sampling settings described in Section 3.4.1. As input to alignment combination, we chose GS-5, GS-80 and GS-N.

Since we obtained the best result with Method D, we applied it also to the other translation directions in Section 3.4.1. Table 3.11 shows the results. For reference, we also report the results with IBM Model 4 alignments (M4) trained in the standard bootstrapping regimen of $1^5H^53^34^3$.

Table 3.10. BLEU scores using different combination methods. C1 uses the default N-best list size of 300 during tuning, C2 uses 1000 due to more number of features in this method.

Method	TE BLEU
GS-N	41.14
GS-5	40.63
GS-80	41.78
A	39.86
B	37.78
C1	40.58
C2	40.91
D	41.32

Table 3.11. BLEU scores for individual and combined alignments from Gibbs sampling.

Method	TE	ET	AE	EA
EM-5	38.91	26.52	15.50	15.17
EM-80	39.19	26.47	15.66	15.02
GS-N	41.14	27.55	14.64	15.89
GS-5	40.63	27.24	16.41	15.82
GS-80	41.78	29.51	15.92	16.02
Comb. Method D	41.32	29.86	15.68	16.62
M4	39.94	27.47	16.46	15.43

An appealing property of alignment combination is that it "smooths out" the occasional big drop in BLEU score and as a result is not worse than EM in any of the cases. Compared to the individual alignments, the combination outperforms the *best* individual alignment in two of the four language directions. More importantly, it outperforms the *worst* individual alignment in *all* four cases.

An additional utility of alignment combination is that, in general, even if the combination method gives a result not significantly outperforming the *best* individual alignment, it removes the problem of choosing/predicting the "best" alignment from the individual alternatives.

Tables 3.10 and 3.11 represent a *multiple-chain homogeneous* alignment combination strategy. We also observed similar benefits from the *single-chain homogeneous* combination and *heterogeneous* combination methods as shown in Figures 3.25–3.27.

3.7.3. Modifications to Minimum Error Rate Training Procedure

In this section we attempt to reduce the variation due to factors other than differences in alignment. Here we address below four such sources of noise:

(i) Randomness in MERT, in particular, the additional random re-starts at each iteration.
(ii) Local optima in MERT, leading to possibly finding different local optima depending on starting point.
(iii) Possible data sparsity in MERT, leading to poor estimates of feature weights.
(iv) Measurement noise stemming from the test set (and the metric).

Figure 3.28 shows the results of our attempts at reducing each type of variation, TE direction at the top and ET direction at the bottom. The leftmost column is the baseline using the standard MERT procedure, which was used in all previous experiments in this setting, e.g., in the rightmost column of Figure 3.24. The mean and standard deviation of BLEU scores (excluding the first 1000 samples) for each

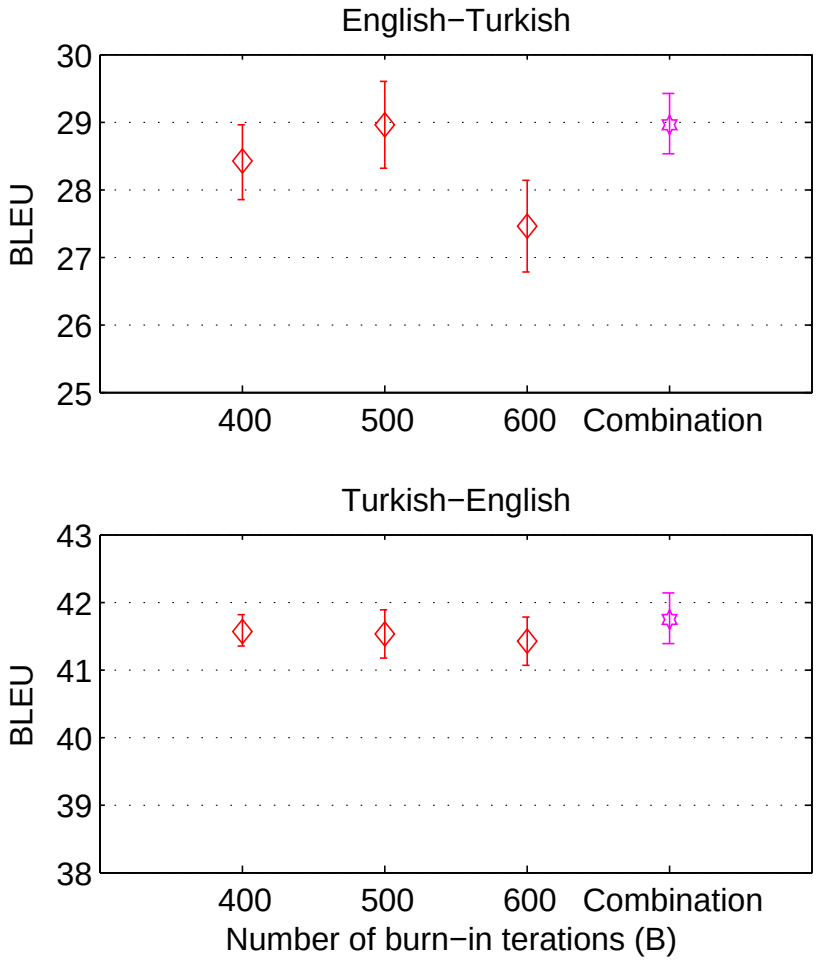

Figure 3.25. GS single-chain homogeneous combination

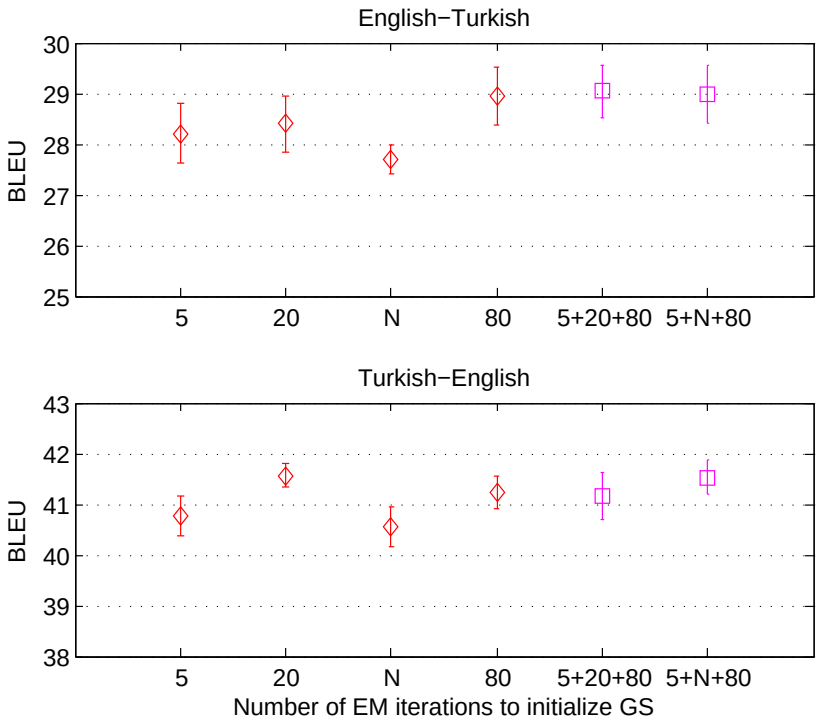

Figure 3.26. GS multi-chain homogeneous combination

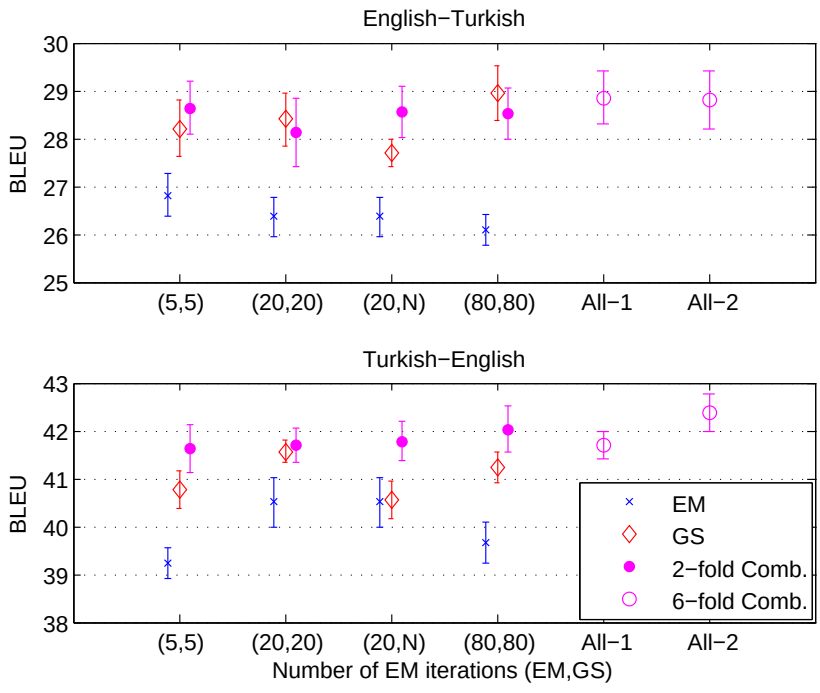

Figure 3.27. EM+GS heterogeneous combination

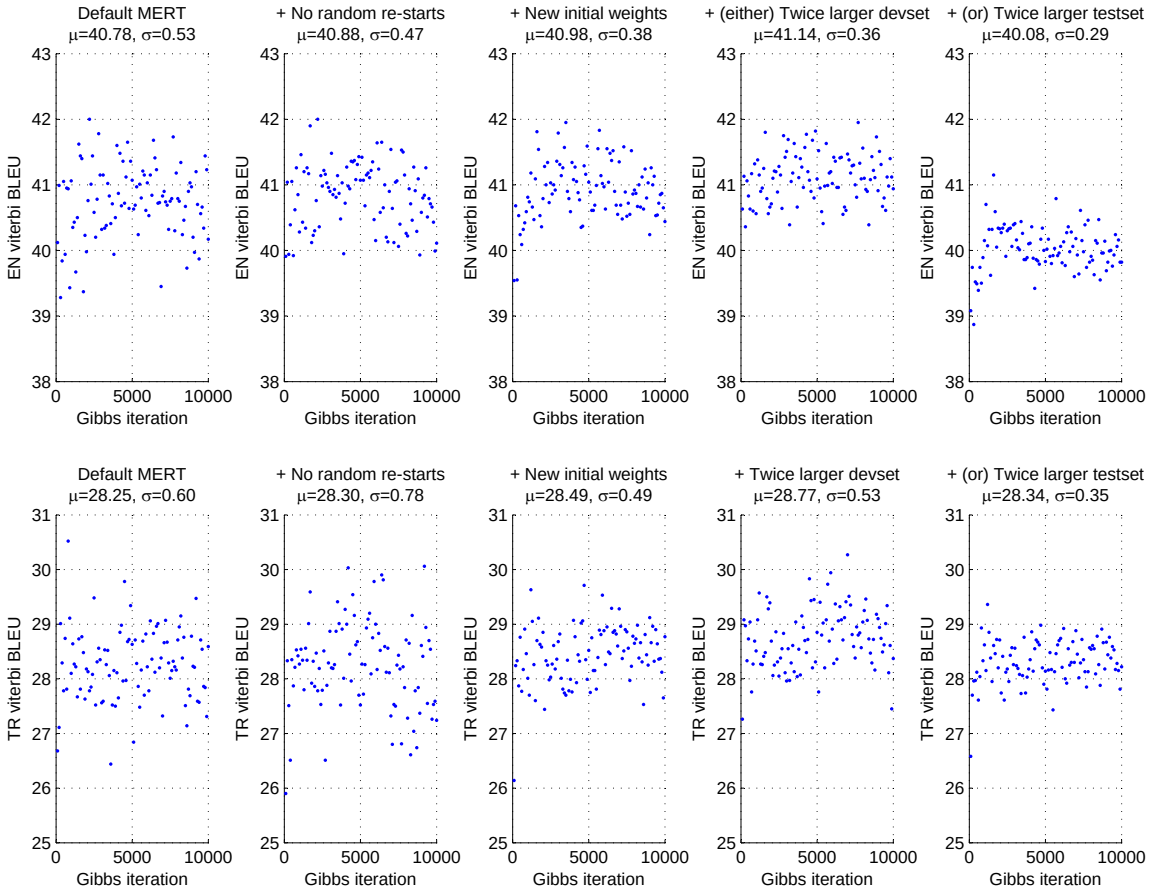

Figure 3.28. BLEU scores obtained applying modifications to the MERT procedure.

graph is also given in the figure.

The second column addresses (i) above, eliminating random re-starts. There is no clear effect on the variation, slightly decreasing in TE and slightly increasing in ET. However, the mean BLEU score does not change, which provides a nice side benefit since we can now use the MERT procedure without additional re-starts, which results in a significant reduction in SMT training time. In this particular case, the 20:1 reduction in line-search optimization resulted in 5:1 reduction in the overall training time.

The third column addresses (ii) above, by using hopefully-better default weights when starting MERT. These weights were determined by the analysis of the MERT output weights in previous experiments in the rightmost column of Figure 3.24, which

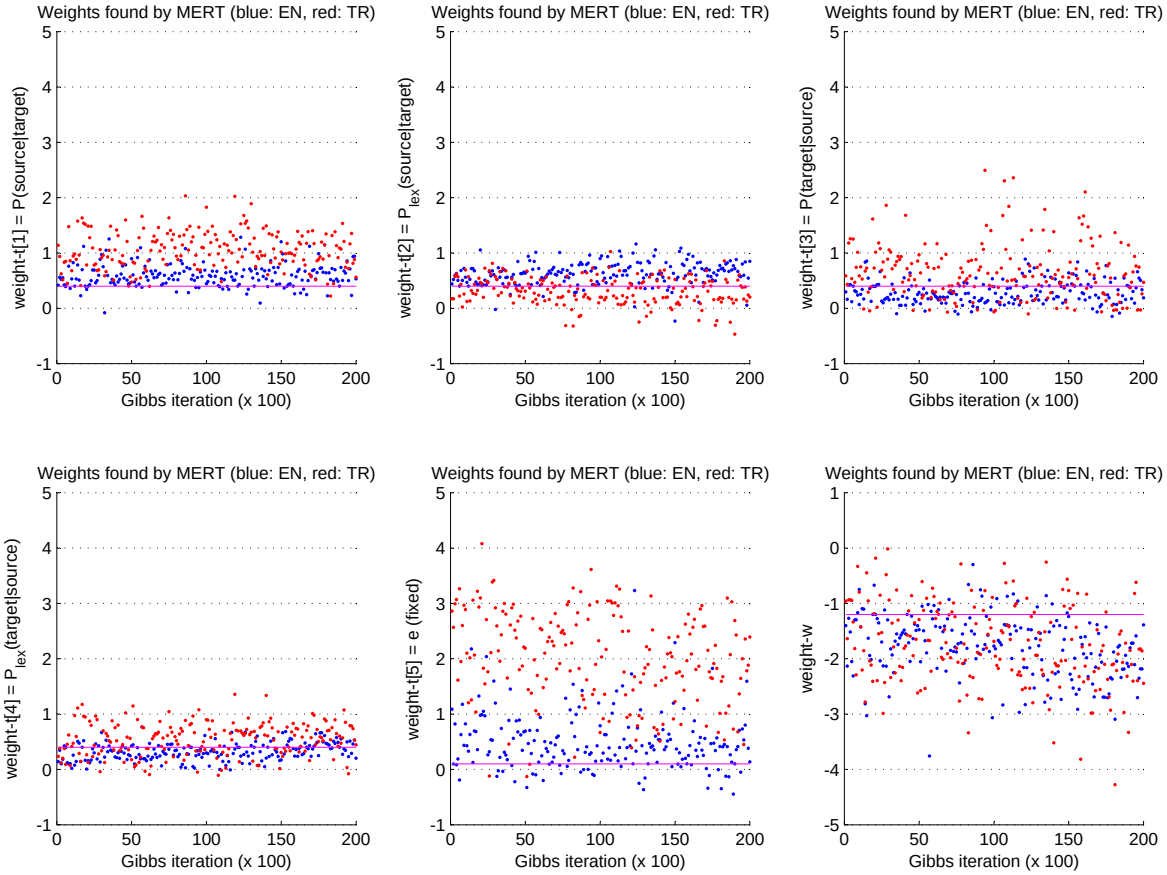

Figure 3.29. Phrase translation and word penalty weights found by the MERT procedure during training of systems from different alignments.

are shown in Figures 3.29 and 3.30. Our hypothesis goes that if we start from a set of initial weights closer to the average previous optimum, those systems that previously could not reach a good optimum due to poor initialization can this time attain better optima, thus hoping to eliminate the lowest BLEU scores from the variations. Figure 3.28 shows that the new weights indeed result in higher average BLEU scores and less variation.

The fourth column addresses (iii) above, by increasing the development set size from 506 to 1012 sentences. The effect on variation is not notable, but a slight increase in average BLEU score is observed, suggesting better weight estimates.

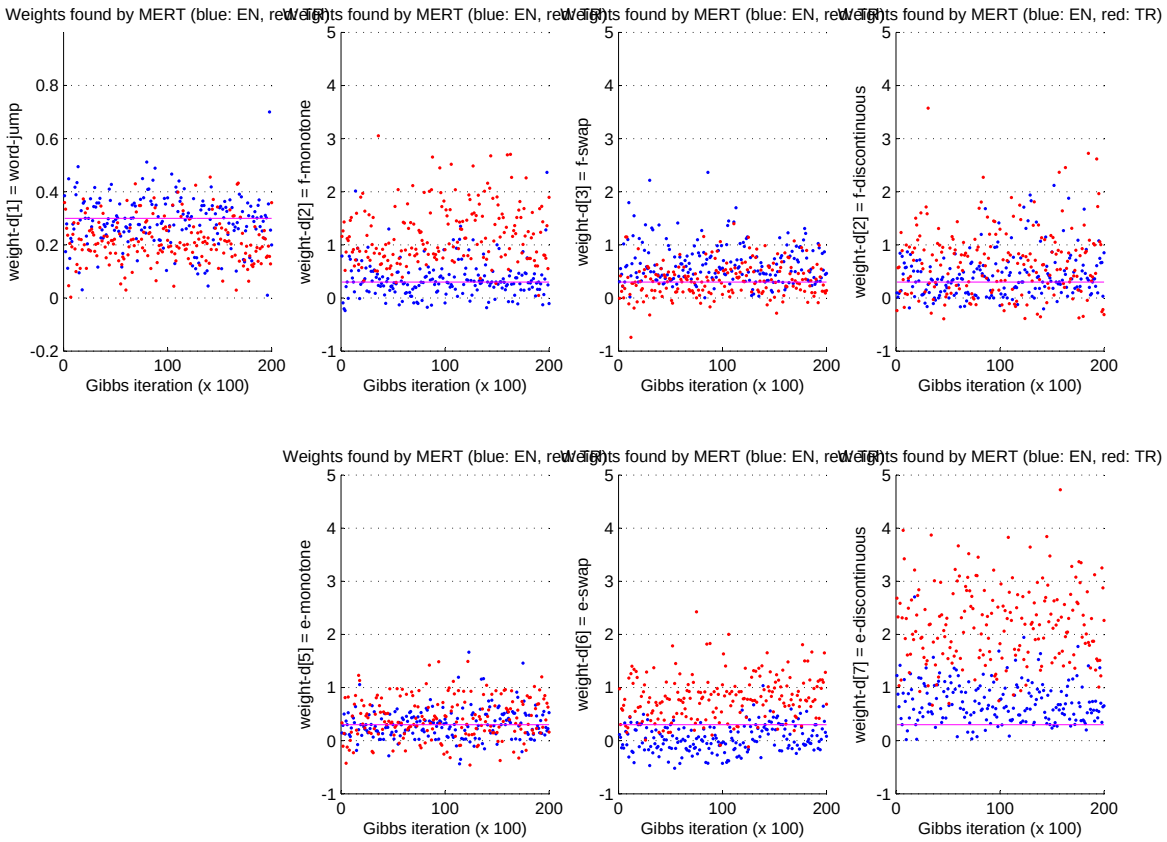

Figure 3.30. Distortion model weights found by the MERT procedure during training of systems from different alignments.

Finally, the fifth column addresses (iv) above, by increasing the test set size from 500 to 1006 sentences. Note that the previous BLEU scores are not directly comparable with this graph since the test set changed. The new sentences might be "harder" or "easier", depending on the sentence lengths, out-of-vocabulary rate, etc. Nevertheless, we observe a significant decrease in standard deviation, suggesting that some of the noise we observe is due to the intricacies of the test set. Increasing the test set size seems to be the single most effective attempt at reducing BLEU variation (and increasing confidence).

Overall, the standard deviation of BLEU scores have dropped from 0.53 to 0.29 in TE direction, and from 0.60 to 0.35 in ET direction. Meanwhile, for the first three modifications, the mean BLEU score increased from 40.78 to 41.14 in TE direction, and from 28.25 to 28.77 in ET direction, without any change in alignment or translation model training.

3.8. Conclusion

We developed a Gibbs sampling-based word alignment inference method for Bayesian IBM Models 1 and 2 and showed that it compares favorably to EM estimation in terms of translation BLEU scores. We observe the largest improvement when data is sparse, e.g., in the cases of smaller corpora and/or more morphological complexity. The proposed method successfully overcomes the well-known "garbage collection" problem of rare words in EM-estimated current models and learns a compact, sparse word translation distribution with more training vocabulary coverage. We also found Gibbs sampling to perform better than variational Bayes inference, which leaves a substantially high portion of source singletons unaligned. Additionally, we utilized alignment combination techniques to further improve the performance and robustness.

Future research avenues include estimation of the hyperparameters from available data or auxiliary sources and utilization of the proposed algorithm in either initialization or inference of more advanced alignment models.

4. JOINT LEARNING OF WORD ALIGNMENT AND MORPHOLOGICAL SEGMENTATION

In this chapter we propose a method for unsupervised determination of the optimal morphological segmentation for statistical machine translation (SMT). We present a segmentation metric that takes into account both sides of the SMT training corpus. We formulate the objective function as the posterior probability of the training corpus according to a generative segmentation-translation model. We describe how the IBM Model-1 translation likelihood can be computed incrementally between adjacent segmentation states for efficient computation. Submerging the proposed segmentation method in a SMT task from morphologically-rich Turkish to English does not exhibit the expected improvement in translation BLEU scores. However, the proposed parallel search algorithm improves the translation performance (as measured by BLEU) compared to the original sequential search algorithm of Morfessor [11].

4.1. Introduction

In SMT, words are normally considered as the building blocks of translation models. However, especially for morphologically complex languages such as Finnish, Turkish, Czech, Arabic etc., it has been shown that using sub-lexical units obtained after morphological preprocessing can improve the machine translation performance over a word-based system [62–64]. The motivation for morphological segmentation can be illustrated with an example parallel corpus given in Table 4.1. Applying the segmentation in Table 4.2 can help the SMT training in learning a more accurate translation model. Morphological segmentation also helps the system cope with words unseen in the training corpus (out-of-vocabulary words). However, the effect of segmentation on translation performance is indirect and difficult to isolate [65].

Many systems apply morphological segmentation before SMT training. But the challenge in designing a sub-lexical SMT system is the decision of what segmentation to

Table 4.1. Word-based alignment problem with an agglutinative language.

English corpus	Turkish corpus
key	anahtar
my key	anahtarım
wallet	cüzdan
my wallet	cüzdanım

Table 4.2. Subword-based alignment problem with an agglutinative language.

English corpus	Turkish corpus
key	anahtar
my key	anahtar +ım
wallet	cüzdan
my wallet	cüzdan +ım

use. This constitutes our main research motivation in this chapter: For the particular language pair and training corpus at hand, what is the optimal sub-word segmentation in terms of translation performance?

Linguistic morphological analysis is intuitive, but it is language-dependent and usually needs disambiguation. Furthermore, the linguistic approach is not necessarily optimal in that (i) manually engineered segmentation schemes can outperform a straightforward linguistic morphological segmentation, e.g., [62], and (ii) linguistic segmentation may result in even worse performance than a word-based system, e.g., [66].

Existing solutions to this problem are predominantly heuristic, language-dependent, and as such are not easily portable to other languages. The optimal degree of segmentation might decrease as the amount of training data increases [62,67]. Also, the optimal segmentation could change when paired with a different language. Therefore, it is desirable to learn the optimal segmentation in an unsupervised manner.

In this chapter, we extend the unsupervised monolingual approach of Creutz and Lagus [68] to take into account bilingual information from the parallel training corpus when making segmentation decisions. As a result, the segmentation learning process is tailored to the particular SMT task via the same parallel corpus that is used in training the statistical translation models.

4.2. Related Work

Most works in SMT-oriented segmentation are supervised in that they consist of manual experimentation to choose the best among a set of segmentation schemes, and are language(pair)-dependent. For Arabic, Sadat and Habash [69] present several morphological preprocessing schemes that entail varying degrees of decomposition and compare the resulting translation performances in an Arabic-to-English task. Shen *et al.* [44] use a subset of the morphology and apply only a few simple rules in segmenting words. Durgar El-Kahlout and Oflazer [66, 70] tackle this problem when translating from English to Turkish, an agglutinative language. They use a morphological analyzer and disambiguation to arrive at morphemes as tokens. However, training the translation models with morphemes actually degrades the translation performance. They outperform the word-based baseline only after some selective morpheme grouping. Bisazza and Federico [64] adopt an approach similar to the Arabic segmentation studies above, this time in a Turkish-to-English translation setting.

Unsupervised segmentation by itself has garnered considerable attention in the computational linguistics literature [68,71–75]. However, only a fraction of works report their performance in a translation task. Virpioja *et al.* [76] used Morfessor [68] to segment both sides of the parallel training corpora in translation between Danish, Finnish, and Swedish, but without a consistent improvement in results. Poon *et al.* [71] and Luong *et al.* [77] propose unsupervised segmentation methods for the purpose of machine translation. However, the segmentation learning in these works does not have input from the translation model.

Morfessor, which gives state of the art results in many tests [78], uses only monolingual information in its objective function. It is conceivable that we can achieve a better segmentation for translation by considering not one but both sides of the parallel corpus. A possible choice is the post-segmentation alignment accuracy. However, Elming et al. [79] show that optimizing segmentation with respect to alignment error rate (AER) does not improve and even degrades machine translation performance. Snyder and Barzilay [72] use bilingual information but the segmentation is learned independently from translation modeling.

In the work by Chang *et al.* [80], the granularity of the Chinese word segmentation is optimized by training SMT systems for several values of a granularity bias parameter and it is found that the value that maximizes translation performance (as measured by BLEU) is different than the value that maximizes segmentation accuracy (as measured by precision and recall).

One motivation in morphological preprocessing before translation modeling is "morphology matching" as in the work by Lee [67] and in the scheme "EN" of Habash and Sadat [62]. In [67], the goal is to match the lexical granularities of the two languages by starting with a fine-grained segmentation of the Arabic side of the corpus and then merging or deleting Arabic morphemes using alignments with a part-of-speech tagged English corpus. But this method is not completely unsupervised since it requires external linguistic resources in initializing the segmentation with the output of a morphological analyzer and disambiguator. Talbot and Osborne [81] tackle a special case of morphology matching by identifying redundant distinctions in the morphology of one language compared to another.

Xu *et al.* [28] and Nguyen *et al.* [29] present unsupervised methods that jointly learn segmentations and alignments. However, they do not report evaluations on agglutinative languages such as Turkish and Finnish.

4.3. Proposed Method

4.3.1. Monolingual Model

To model the segmentation process, the generative model of Morfessor [68, 82] introduces an auxiliary variable $M_{\mathbf{F}}$ that represents the lexicon of morphemes that make up the words in the monolingual corpus $\mathbf{F}$. Then the problem of finding the maximum *a posteriori* (MAP) segmentation can be written as:

$$\begin{aligned} \hat{M}_{\mathbf{F}} &= \arg\max_{M_{\mathbf{F}}} P(M_{\mathbf{F}}|\mathbf{F}) && (4.1) \\ &= \arg\max_{M_{\mathbf{F}}} \frac{P(M_{\mathbf{F}}, \mathbf{F})}{P(\mathbf{F})} && (4.2) \\ &= \arg\max_{M_{\mathbf{F}}} P(M_{\mathbf{F}}, \mathbf{F}) && (4.3) \end{aligned}$$

Since there can be several valid segmentations of $\mathbf{F}$ given $M_{\mathbf{F}}$, for clarity we introduce a hidden variable $\mathbf{F}_{seg}$ that represents the segmented version of $\mathbf{F}$ according to $M_{\mathbf{F}}$. Now the monolingual generative model can be written as:

$$P(M_{\mathbf{F}}, \mathbf{F}) = \sum_{\mathbf{F}_{seg}} P(M_{\mathbf{F}}) P(\mathbf{F}_{seg}|M_{\mathbf{F}}) P(\mathbf{F}|\mathbf{F}_{seg}) \quad (4.4)$$

where $P(\mathbf{F}|\mathbf{F}_{seg})$ is either 1 or 0 indicating legal segmentations of $\mathbf{F}$ according to $M_{\mathbf{F}}$.

In searching for the MAP segmentation model $\hat{M}_{\mathbf{F}}$, the summation is approximated with the $max()$ operation so that (4.3) becomes:

$$\hat{M}_{\mathbf{F}} \approx \arg\max_{M_{\mathbf{F}}} P(M_{\mathbf{F}}) P(\mathbf{F}_{seg}|M_{\mathbf{F}}) \quad (4.5)$$

In (4.5), the prior $P(M_{\mathbf{F}})$ on the lexicon is assumed to only depend on the frequencies and lengths of the individual morphs, which are also assumed to be independent. The likelihood of the segmented corpus $P(\mathbf{F}_{seg}|M_{\mathbf{F}})$ is computed as the product of

morph probabilities estimated from their frequencies in the corpus.

4.3.2. Bilingual Model

We re-formulate the monolingual MAP problem in (4.1) so as to take into account both sides of the parallel training corpus as following:

$$\begin{aligned} \hat{M}_{\mathbf{F}} &= \arg\max_{M_{\mathbf{F}}} P(M_{\mathbf{F}}|\mathbf{E},\mathbf{F}) && (4.6)\\ &= \arg\max_{M_{\mathbf{F}}} P(M_{\mathbf{F}},\mathbf{E},\mathbf{F}) && (4.7)\\ &= \approx \arg\max_{M_{\mathbf{F}}} P(M_{\mathbf{F}})P(\mathbf{F}_{seg}|M_{\mathbf{F}})P(\mathbf{E}|\mathbf{F}_{seg}) && (4.8) \end{aligned}$$

This proposed segmentation model takes into account the likelihood of both sides of the parallel corpus while searching for the optimal segmentation. The joint likelihood is decomposed into a prior, a monolingual likelihood, and a translation likelihood, as shown in (4.8). We model the first two components as in the monolingual case while for the translation component $P(\mathbf{E}|\mathbf{F}_{seg})$ we use IBM Model 1, which is presented in Section 3.3. The translation likelihood of an individual sentence pair ($\mathbf{e}$,$\mathbf{f}$) according to IBM Model 1 is given by [5]:[5]

$$P(\mathbf{f}|\mathbf{e}) = \frac{P(J|\mathbf{e})}{(I+1)^J}\prod_{j=1}^{J}\sum_{i=0}^{I} t_{e_i,f_j}. \quad (4.9)$$

The sentence length probability distribution $P(J|\mathbf{e})$ is assumed to be Poisson with the expected sentence length equal to I.

The role of the bilingual component $P(\mathbf{E}|\mathbf{F}_{seg})$ in (4.8) can be motivated with a simple example as follows. Consider an occurrence of two phrase pairs in a Turkish-English parallel corpus and the two hypothesized sets of segmentations for the Turkish phrases as shown in Table 4.3. Without access to the English side of the corpus, a

[5]For coherence with the SMT literature, where the derivations are in the form of $P(f|e)$, we switch the notation of the source and target language corpus labels from here to the end of Section 4.3.3, without loss of generalization.

monolingual segmenter can quite possibly score Segmentation #1 higher than Segmentation #2 (e.g., due to the high frequency of the observed morph "+m"). On the other hand, a bilingual segmenter is expected to assign a higher alignment probability $P(\mathbf{E}|\mathbf{F})$ to Segmentation #2 than Segmentation #1, because of the aligned words key||anahtar, therefore ranking Segmentation #2 higher.

Table 4.3. Example segmentation hypotheses.

	Phrase #1	Phrase #2
Turkish phrase	anahtar	anahtarım
English phrase	key	my key
Segmentation #1	anahtar	anahtarı +m
Segmentation #2	anahtar	anahtar +ım

4.3.3. Incremental Computation of Model-1 Likelihood

During search through possible segmentations, the translation likelihood $P(\mathbf{f}|\mathbf{e})$ needs to be calculated according to (4.9) for every hypothesized segmentation. In order to compute (4.9), we need to have at hand the individual morph translation probabilities t_{e_i,f_j}. These can be estimated using the EM algorithm given by [5], which is guaranteed to converge to a global maximum of the likelihood for Model 1. However, running the EM algorithm to optimization for each considered segmentation model can be computationally expensive, and can result in overtraining. Therefore, in this work we used the likelihood computed after the first EM iteration, which we show to also have the nice property that $P(\mathbf{f}|\mathbf{e})$ can be computed incrementally from one segmentation hypothesis to the next.

The incremental updates are derived from the equations for the count collection and probability estimation steps of the EM algorithm as follows. In the count collection step, in the first iteration, we need to compute the fractional counts $c(f_j|e_i)$ [5]:

$$c(f_j|e_i) = \frac{1}{I+1}(\#f_j)(\#e_i), \tag{4.10}$$

where $(\#f_j)$ and $(\#e_i)$ denote the number of occurrences of f_j in $\mathbf{f}$ and e_i in $\mathbf{e}$, respectively.

Let f_k denote the word hypothesized to be segmented. Let the resulting two sub-words be f_p and f_q, any of which may or may not previously exist in the vocabulary. Then, according to (4.10), as a result of the segmentation no update is needed for $c(f_j|e_i)$ for $j = 1 \dots V_F$, $j \neq p, q$, $i = 1 \dots V_E$ (note that f_k no longer exists); and the necessary updates $\Delta c(f_j|e_i)$ for $c(f_j|e_i)$, where $j = p, q$; $i = 1 \dots V_E$ are given by:

$$\Delta c(f_j|e_i) = \frac{1}{I+1}(\#f_k)(\#e_i). \tag{4.11}$$

Note that (4.11) is nothing but the previous count value for the segmented word, $c(f_k|e_i)$. So, all needed in the count collection step is to copy the set of values $c(f_k|e_i)$ to $c(f_p|e_i)$ and $c(f_q|e_i)$, adding if they already exist.

Then in the probability estimation step, the normalization is performed including the newly added fractional counts.

4.3.4. Parallel Search and Stochastic Search

The original search algorithm of Morfessor [82] is a greedy algorithm where the costs of the following search points are affected by the decision in the current step. This leads to a sequential search and does not lend itself to parallelization. Specifically, in an iteration of the algorithm, all words in the vocabulary are processed one-by-one (preferably in random order), computing for each word the posterior probability of the generative model after each possible binary segmentation ("splitting") of the word. If the highest-scoring split increases the posterior probability compared to not splitting, that split is accepted (for all occurrences of the word) and the resulting sub-words are explored recursively for further segmentations. This process is repeated until an iteration no more results in a significant increase in the posterior probability.

In the first proposed search alternative [11], which we call "batch-update", the segmentation decisions for individual words are stored but are not applied until the end of an iteration. In this way, all cost calculations can be performed independently and in parallel. Since the model is not updated at every decision, the search path generally differs from that in the sequential search and hence results in a different final segmentation.

The second proposed alternative search strategy [12], which we call "stochastic search", is an application of Gibbs sampling. Instead of the greedy model updates at each processed word, the segmentation decision for a word is sampled from the distribution proportional to the posterior probability of the model given the existing state of segmentation for the rest of the words.

Note that these two proposed methods are not mutually exclusive and they can co-exist in a segmentation scheme.

4.4. Results

We performed *in vivo* testing of the segmentation algorithm on the Turkish side of a Turkish-to-English task. We compared the segmentations produced by Morfessor, Morfessor modified for parallel search (Morfessor-p), and Morfessor with bilingual cost (Morfessor-bi) against the word-based performance. We used the ATR Basic Travel Expression Corpus (BTEC) [35], which contains travel conversation sentences similar to those in phrase-books for tourists traveling abroad. The training corpus contained 19,972 sentences with average sentence length 5.6 and 7.7 words for Turkish and English, respectively. The test corpus consisted of 1,512 sentences with 16 reference translations. We used GIZA++ [15] for post-segmentation token alignments and the Moses toolkit [39] with default parameters for phrase-based translation model generation and decoding. Target language models were trained on the English side of the training corpus using the SRILM toolkit [40]. The BLEU metric [42] was used for translation evaluation.

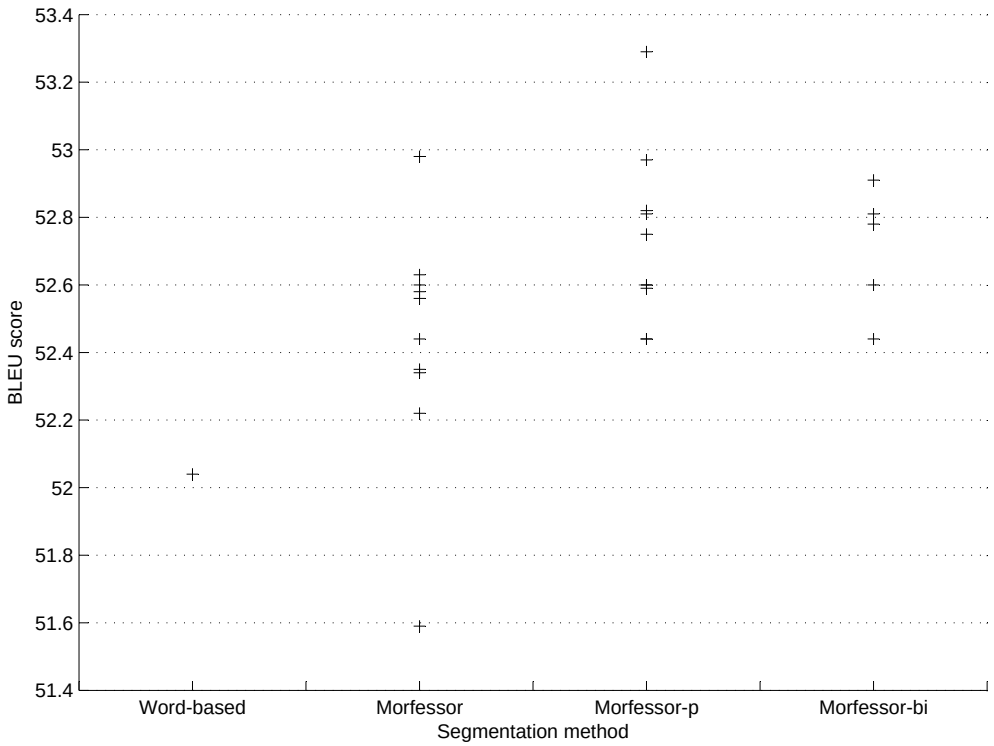

Figure 4.1. BLEU scores obtained with different segmentation methods. Multiple data points for a system correspond to different random orders in processing the data.

Figure 4.1 compares the translation performance obtained using the described segmentation methods. All segmentation methods generally improve the translation performance (Morfessor and Morfessor-p) compared to the word-based models. However, Morfessor-bi, which utilizes both sides of the parallel corpus in segmenting, does not convincingly outperform the monolingual methods.

In order to investigate whether the proposed bilingual segmentation cost correlates any better than the monolingual segmentation cost of Morfessor, we show several cost-BLEU pairs obtained from the final and intermediate segmentations of Morfessor and Morfessor-bi in Figure 4.2. The correlation coefficients show that the proposed bilingual metric is somewhat predictive of the translation performance as measured by BLEU, while the monolingual Morfessor cost metric has almost no correlation. Yet, the strong noise in the BLEU scores (vertical variation in Figure 4.2) diminishes the effect of this correlation, which explains the inconsistency of the results in Figure 4.1.

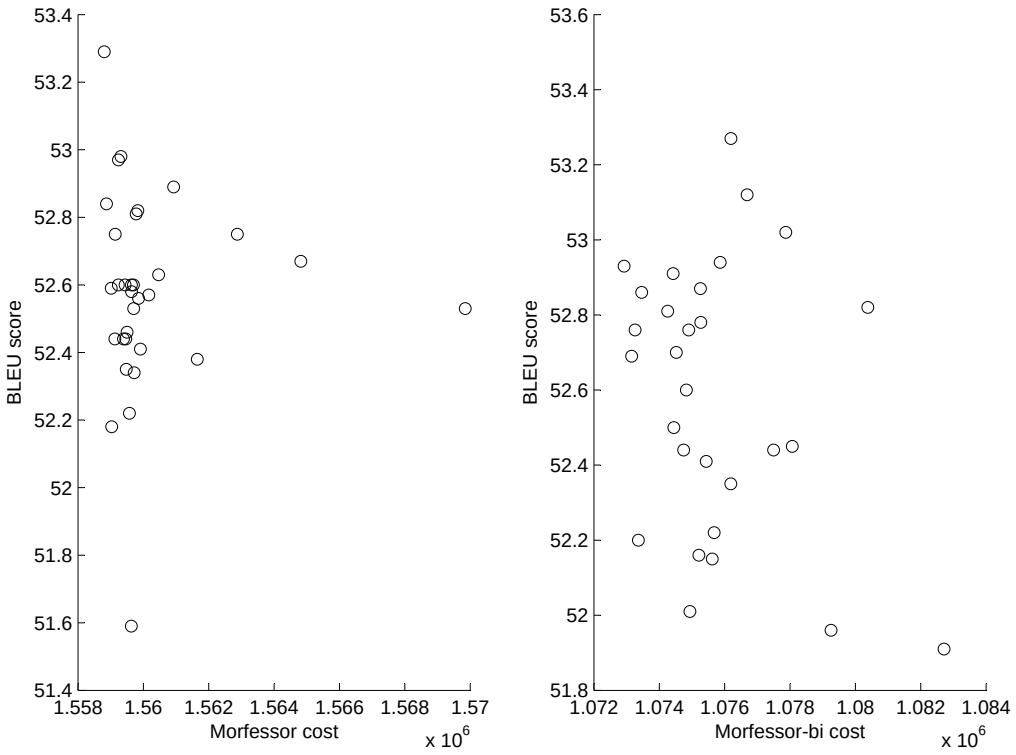

Figure 4.2. Cost-BLEU plots of Morfessor and Morfessor-bi. Correlation coefficients are -0.005 and -0.279, respectively.

Indeed, in our experiments even though the total cost kept decreasing at each iteration of the search algorithm, the BLEU scores obtained by those intermediate segmentations fluctuated without any consistent improvement.

Table 4.4 displays sample segmentations produced by both the monolingual and bilingual segmentation algorithms. We can observe that utilizing the English side of the corpus enabled Morfessor-bi: (i) to consistently identify the root word "anahtar" (top portion), and (ii) to match the English plural word form "games" with the Turkish plural word form "oyunlar" (bottom portion). Monolingual Morfessor is unaware of the target segmentation, and hence it is up to the subsequent translation model training to learn that "oyun" is sometimes translated as "game" and sometimes as "games" in the segmented training corpus.

Stochastic search is able to find better segmentations with lower model costs

Table 4.4. Sample segmentations produced by Morfessor and Morfessor-bi.

Count	Morfessor	Morfessor-bi	English Gloss
7	anahtar	anahtar	(the) key
6	anahtar + ımı	anahtar + ımı	my key (*ACC.*)
5	anahtarla	anahtar + la	with (the) key
4	anahtarı	anahtar + ı	[1](the) key (*ACC.*); [2]his/her key
3	anahtarı + m	anahtar + ım	my key
3	anahtarı + n	anahtar + ın	[1]your key; [2]of (the) key
1	anahtarı + nız	anahtar + ınız	your (*pl.*) key
1	anahtarı + nı	anahtar + ını	[1]your key (*ACC.*); [2]his/her key (*ACC.*)
1	anahtar + ınızı	anahtar + ınızı	your (*pl.*) key (*ACC.*)
1	oyun + lar	oyunlar	(the) games
2	oyun + ları	oyunlar + ı	[1](the) games (*ACC.*); [2]his/her games; [3]their game(s)
1	oyun + ların	oyunlar + ı + n	[1]of (the) games; [2]your games
1	oyun + larınızı	oyunlar + ı + n + ızı	your (*pl.*) games (*ACC.*)

compared to the original greedy search as shown in Table 4.5 for the monolingual Morfessor. However, this search improvement does not translate over to translation performance in terms of BLEU score (Table 4.6) in the IWSLT 2010 task [83]. This suggests a model mismatch, which can be expected in this case since the segmentation model uses only monolingual observations. Table 4.6 also shows that the batch-update search, while enabling parallel computation, results in lower test set performance in this task.

Table 4.5. Segmentation model scores (in negative log probability) obtained by greedy search with three different random vocabulary scan orders and by stochastic search with 2000 iterations over the vocabulary.

Search	Model score
Original	1559831
(greedy)	1559315
	1559527
Stochastic	1554433

Table 4.6. Comparison of %BLEU scores with different segmentation search algorithms in the IWSLT 2010 task.

Search algorithm	Tuning	Test	iwslt09	iwslt10
Original	59.41	54.42	52.15	49.83
Batch-update	59.22	53.61	50.68	48.55
Stochastic	59.09	54.55	51.90	48.60

4.5. Analysis and Further Experiments

4.5.1. Utilizing Allomorphy

Morfessor does not use any linguistic knowledge in its model. However, by incorporating minimal linguistic knowledge in the form of allomorphy (the same lexical morpheme appearing in different surface forms depending on the stem it is attaching to), one might expect to improve the translation performance. To test this hypothesis, we used the following setup: The segmentation model is trained and the corpus segmented as before using Morfessor. Then, all the allomorphic letters in all the suffixes are mapped to their base letter, (e.g., [ı, i, u, ü] are all mapped to H etc.), hoping that equivalences between variants of the same lexical morphemes are in this way captured. This postprocessing is not applied to the stems. The resulting corpus is fed to the SMT training (or decoding) phase.

Table 4.7 shows that, even though small improvements on development sets were observed, we did not obtain the expected improvements on the test sets. It is possible that imposing allomorphy externally after the segmentation is learned has a negative effect on the performance. A possible future research avenue could be to use this linguistic knowledge during segmentation learning inside Morfessor (though the new segmentation method would no longer be truly unsupervised).

Table 4.7. Comparison of %BLEU scores with and without postprocessing allomorphs in Morfessor output in the IWSLT 2010 task.

Representation	dev1	dev2	iwslt09	iwslt10
Surface forms	59.41	54.42	52.15	49.83
Allomorphs	59.53	55.28	51.57	48.93

4.5.2. Segmentation Training with Monolingual Out-of-Domain Corpus

In this section, we explore whether using a large monolingual corpus can reduce data sparsity of Turkish word forms and hence improve the segmentation. We experiment with using a large Turkish monolingual corpus to see whether a better segmentation can be learned. The additional corpus, which consists of about 40 Mwords with a vocabulary size of about 500 K, is gathered from Turkish news sites on the web, so it is out of domain for the BTEC corpus in the IWSLT task.

In the first experiment (named here as "+mono"), we simply merge the BTEC corpus with the additional monolingual corpus and train Morfessor. In the second experiment (named here as "+mono(flat)"), we set the frequencies of all the words in the vocabulary to 1. This latter method results in more satisfactory segmentations in some applications [68], mainly because on large corpora, frequently occurring complex words are not segmented by Morfessor. As a result, training Morfessor on "types" rather than on "tokens" is found to match linguistic segmentation more closely. Since our additional monolingual corpus is quite large, we also experimented with this flat-vocabulary method. But we first cut-off the singletons in the out-of-domain corpus before merging the two vocabularies, mainly for text noise reduction.

The results are shown in Table 4.8. Using an out-of-domain monolingual corpus did not help the translation performance in our experiments, though training on types is found to be more effective than training on tokens in this case.

Table 4.8. %BLEU scores with and without added monolingual out-of-domain corpus for segmentation training.

Corpus	tuning	dev1	dev2	iwslt09	iwslt10
btec	dev1	59.41	54.42	52.15	49.83
+mono	dev1	55.88	50.49	49.17	46.09
+mono(flat)	dev1	58.98	53.53	50.69	48.87
+mono	dev2	53.60	53.46	50.31	47.01
+mono(flat)	dev2	56.89	56.54	51.08	49.66

4.5.3. Experiments with Morfessor Categories-MAP

Up to here, the unsupervised segmentation experiments are conducted using Morfessor-baseline, which employs a fairly simple segmentation model where the induced morphs are assumed to be independent of their context. A more advanced model called Morfessor Categories-MAP [68] probabilistically assigns each induced morph to one of prefix, stem, or suffix classes. In an observed corpus of words segmented into morphs, the transitions between classes and the emissions of morphs from a given class are modeled in a hidden Markov model (HMM) framework.

The performance of this segmentation model, named here as "Morfessor-catmap", is compared in Table 4.9. It exceeds the performance of Morfessor-baseline, but still falls short of supervised segmentation.

4.6. Conclusions

We have presented a method for determining optimal sub-word translation units automatically from a parallel corpus. We have also showed a method of incrementally computing the first iteration parameters of IBM Model-1 between segmentation hypotheses. The proposed parallel search algorithm improved the translation performance compared to the original sequential search algorithm. Being language-independent, the

Table 4.9. %BLEU scores of the developed Turkish-English systems each tuned on devset1.

Segmentation	dev1	dev2	iwslt09	iwslt10
Word-based	56.65	51.40	49.48	47.49
Morfessor-baseline	59.41	54.42	52.15	49.83
Morfessor-catmap	62.69	54.78	53.03	50.91
Linguistic+manual	64.62	59.46	56.40	53.32

proposed algorithm can be added as a one-time preprocessing step prior to training in a SMT system without requiring any additional data/linguistic resources. The experiments show that the translation units learned by the proposed algorithm improves on the word-based baseline in a Turkish-to-English translation task. However, the addition of bilingual information in the model did not yield a noticeable effect, suggesting more work needs to be done to more effectively utilize the information in the parallel corpus in guiding the segmentation decisions.

Overall, experimental results show that while unsupervised segmentation improves translation BLEU scores over the word-based baseline for this task, it does not (yet) reach the performance of task-optimized supervised segmentation (Table 4.9). Even though up to now we have tested our results on Turkish, the applied methods are entirely language-independent (save affixation) and we expect them to be applicable particularly to other agglutinative languages as well.

Possible future research avenues include improving the model (e.g., incorporating the HMM morpheme generation model of Morfessor Categories-MAP [68]), improving the search method and evaluating on other morphologically-rich languages.

5. CONCLUSION

In this dissertation, we proposed unsupervised solutions to two prominent problems in statistical machine translation:

In Chapter3, we propose a Bayesian approach to word alignment inference in IBM Models 1 and 2. In the proposed approach, the model parameters are treated as random variables with a prior and are integrated out during inference. We compare the inferred word alignments against EM and variational Bayes inference in terms of their end-to-end translation performance on several language pairs and types of corpora up to 15 million sentence pairs. We show that Bayesian inference outperforms both EM and VB in the majority of test cases. We also propose several metrics to measure the effectiveness of an alignment algorithm. Our analysis reveals that the proposed method effectively addresses the high-fertility rare word problem in EM and unaligned rare word problem in VB, achieves higher agreement and vocabulary coverage rates than both, and leads to smaller phrase tables.

In Chapter 4, we tackle the problem of unsupervised determination of the optimal morphological segmentation for SMT and propose a segmentation metric that takes into account both sides of the parallel training corpus. We formulate the objective function as the posterior probability of the training corpus according to a generative segmentation-translation model. We describe how the IBM Model-1 translation likelihood can be computed incrementally between adjacent segmentation states for efficient computation. We also propose a parallelizable search algorithm, which improves the search performance of the monolingual segmentation as well.

5.1. Future Work

Model 1 assumes that all alignments are equally likely, i.e., a target word can be aligned with any word in its source sentence with equal probability. However, for morphologically imbalanced language pairs such as Turkish-English, it can be expected

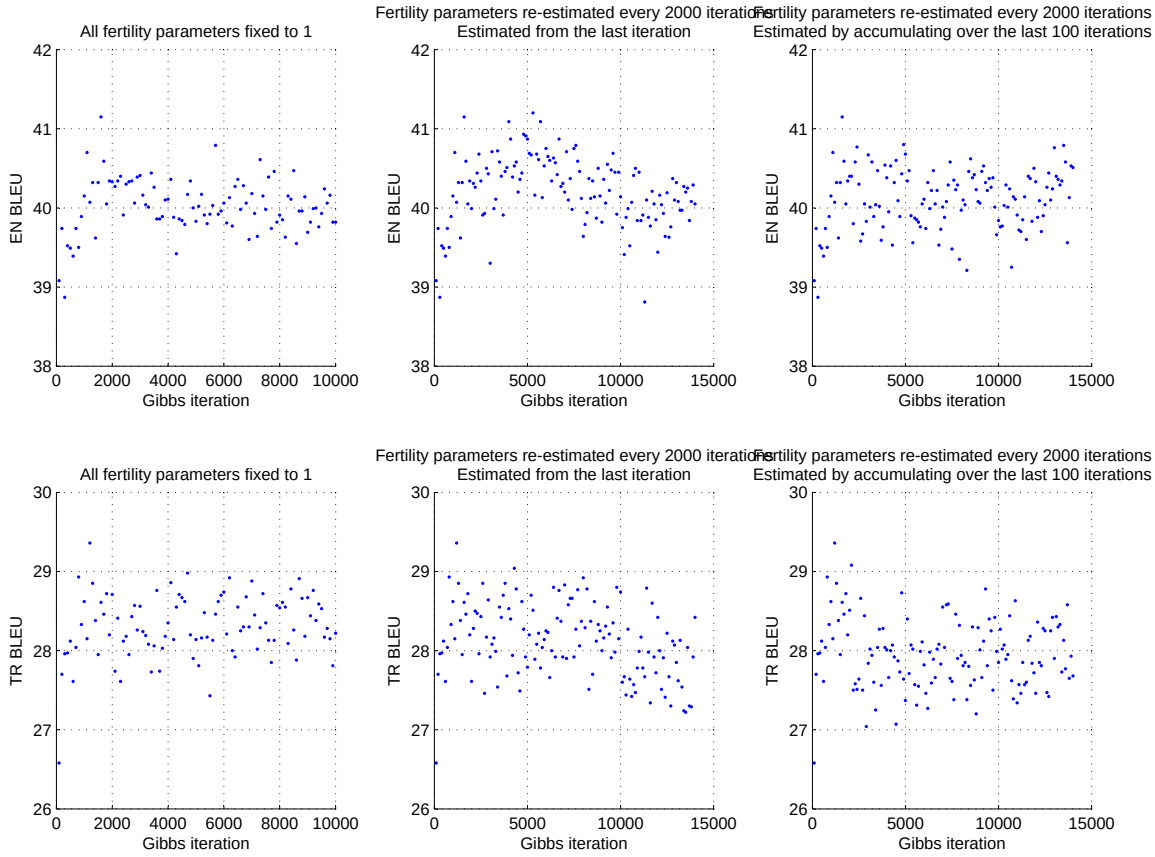

Figure 5.1. BLEU scores obtained by standard Model 1 and its fertility extensions.

that some Turkish word types consistently generate more target words than some other Turkish word types. This can be incorporated in our model by replacing

$$a_j|\mathbf{e} \sim \text{Uniform}(a_j; |\mathbf{e}| + 1) \tag{5.1}$$

with the following:

$$a_j|\mathbf{e} \sim \text{Multinomial}(a_j; \mathbf{k}) \tag{5.2}$$

where $\mathbf{k}$ is a vector of parameters specifying the *expected fertility* of each source word type. Note that this is a more general model than Model 1, which is a special case where $k_e = \text{const.}$ for all e in the source vocabulary.

In contrast to our earlier Bayesian model, where the translation parameters were integrated out from inference equations, we now face the task of estimating $\mathbf{k}$ from data as well. Our implementation of a MCEM procedure starts with $k_e = 1.0$ and re-estimates k_e at certain intervals from maximum-likelihood estimation from the data (hidden + observed variables, whose samples are available as output of Gibbs sampling).

Figure 5.1 shows the results for TE (top) and ET (bottom) directions. The graphs on the left are from Figure 3.24 as reference. Note that the first 2000 iterations of all three columns are identical, since fertility parameters have not yet been updated. The results are very recent and need more analysis. We would expect better alignments especially where the source is Turkish and target is English, so evaluating the alignments and not the systems as in Figure 5.1 would be a good starting point.

5.2. Application to Neural Machine Translation

The initial motivation for the alignment and segmentation models proposed in this thesis was to improve the performance of SMT. However, another machine translation paradigm called Neural Machine Translation (NMT) recently has become popular and achieved success both in machine translation evaluations [84] and deployment [85]. In the following, we discuss how the contributions in this thesis relate to NMT and mention possible applications of the presented work in an NMT setting.

NMT systems usually have an encoder-decoder architecture [86–88], where one neural network (the encoder) maps the input sequence of words to a sequence of real-valued vectors, which are then fed to a separate neural network (the decoder) to produce the output sequence of words. The decoder utilizes a third network called attention which computes an additional context vector for the current output word position by weighting the encoder outputs for all input positions, approximating a soft alignment. The network parameters are trained end-to-end with the objective to maximize the conditional probability of the target sentences in the training set given their source sentences. The main advantages of NMT over SMT include end-to-end training resulting

in all model parameters being optimized simultaneously, distributed representations of words that capture and exploit similarities between words, and utilizing a much larger context resulting in sentence-level fluent output.

The word alignments inferred by the Bayesian word alignment method presented in Chapter 3 can be used in NMT for several purposes, e.g., for supervising the training by incorporating the alignments in the cost function (guided alignment training, [89–91]), for bootstrapping the training by adding sub-sentence pairs extracted from the training corpus using standard SMT [89,92], for constraining the set of target vocabularies for the decoder [91], for tracking the origin of the output words (e.g., within a computer-aided translation tool), or for overriding the NMT decisions for words that are usually better handled by other (e.g., rule-based) methods such as numbers, dates, certain terminology etc. Furthermore NMT does not cope well with rare or unknown words, out-of-domain input, and low-resource conditions, in such cases completely sacrificing adequacy in favor of fluency [93]; therefore under such conditions SMT could be preferred over NMT or they could be used in combination.

The research problem in Chapter 4, unsupervised determination of the optimal segmentation for a particular machine translation task, is also relevant in NMT since most NMT systems apply sub-word segmentation as a preprocessing step. The motivation for segmentation in NMT is generally twofold: to reduce the vocabulary size due to computational limitations, and to decrease the out-of-vocabulary and rare input words since they severely degrade the NMT output quality [86,93,94]. Sub-word representation [75] achieves both goals. A third motivation in the case of morphologically-rich languages could be computing real-valued representations and attention weights at the morpheme level in order to better model the translation process. Currently the most commonly used sub-word segmentation methods such as byte-pair encoding [95] and the wordpiece model [85] utilize only monolingual information while making their sub-word boundary decisions. Our approach in Chapter 4 of using bilingual information during segmentation decisions could potentially lead to better segmentations in terms of NMT performance.

REFERENCES

1. Koehn, P., F. J. Och and D. Marcu, "Statistical Phrase-Based Translation", *Proceedings of HLT-NAACL*, pp. 48–54, Edmonton, May-June 2003.

2. Chiang, D., "Hierarchical phrase-based translation", *Computational Linguistics*, Vol. 33, No. 2, pp. 201–228, 2007.

3. Galley, M., J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang and I. Thayer, "Scalable Inference and Training of Context-Rich Syntactic Translation Models", *Proceedings of ACL-COLING*, pp. 961–968, Sydney, Australia, July 2006.

4. Koehn, P., *Statistical Machine Translation*, Cambridge University Press, 2010.

5. Brown, P. F., V. J. Della Pietra, S. A. Della Pietra and R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation", *Computational Linguistics*, Vol. 19, No. 2, pp. 263–311, 1993.

6. Vogel, S., H. Ney and C. Tillmann, "HMM-based word alignment in statistical translation", *Proceedings of COLING*, pp. 836–841, 1996.

7. Liang, P., B. Taskar and D. Klein, "Alignment by Agreement", *Proceedings of HLT-NAACL*, pp. 104–111, New York City, New York, June 2006.

8. Dempster, A., N. Laird and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society, Series B*, Vol. 39, No. 1, pp. 1–38, 1977.

9. Mermer, C. and M. Saraclar, "Bayesian Word Alignment for Statistical Machine Translation", *Proceedings of ACL-HLT: Short Papers*, pp. 182–187, Portland, Oregon, June 2011.

10. Mermer, C., M. Saraclar and R. Sarikaya, "Improving Statistical Machine Translation Using Bayesian Word Alignment and Gibbs Sampling", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 21, No. 5, pp. 1090–1101, May 2013.

11. Mermer, C. and A. A. Akın, "Unsupervised Search for the Optimal Segmentation for Statistical Machine Translation", *Proceedings of ACL: Student Research Workshop*, pp. 31–36, Uppsala, Sweden, July 2010.

12. Mermer, C. and M. Saraclar, *Unsupervised Turkish Morphological Segmentation for Statistical Machine Translation*, 2011, `http://cl.haifa.ac.il/MT/abstracts/mermer.pdf`, accessed at March 2019.

13. Och, F. J., "Minimum Error Rate Training in Statistical Machine Translation", *Proceedings of ACL*, pp. 160–167, Sapporo, Japan, July 2003.

14. Brown, P. F., S. A. D. Pietra, V. J. D. Pietra, M. J. Goldsmith, J. Hajic, R. L. Mercer and S. Mohanty, "But Dictionaries Are Data Too", *Proceedings of HLT*, pp. 202–205, Plainsboro, New Jersey, 1993.

15. Och, F. J. and H. Ney, "A systematic comparison of various statistical alignment models", *Computational Linguistics*, Vol. 29, No. 1, pp. 19–51, 2003.

16. Moore, R. C., "Improving IBM Word Alignment Model 1", *Proceedings of ACL*, pp. 518–525, Barcelona, Spain, July 2004.

17. Zhao, B. and E. P. Xing, "BiTAM: Bilingual Topic AdMixture Models for Word Alignment", *Proceedings of COLING-ACL: Poster Sessions*, pp. 969–976, Sydney, Australia, July 2006.

18. Vaswani, A., L. Huang and D. Chiang, "Smaller Alignment Models for Better Translations: Unsupervised Word Alignment with the L_0-norm", *Proceedings of ACL*, pp. 311–319, 2012.

19. Zhao, S. and D. Gildea, "A Fast Fertility Hidden Markov Model for Word Alignment Using MCMC", *Proceedings of EMNLP*, pp. 596–605, Cambridge, Massachusetts, October 2010.

20. Goldwater, S. and T. Griffiths, "A fully Bayesian approach to unsupervised part-of-speech tagging", *Proceedings of ACL*, pp. 744–751, Prague, Czech Republic, June 2007.

21. Gao, J. and M. Johnson, "A comparison of Bayesian estimators for unsupervised hidden Markov model POS taggers", *Proceedings of EMNLP*, pp. 344–352, Honolulu, Hawaii, October 2008.

22. Goldwater, S., T. L. Griffiths and M. Johnson, "Contextual Dependencies in Unsupervised Word Segmentation", *Proceedings of ACL-COLING*, pp. 673–680, Sydney, Australia, July 2006.

23. Mochihashi, D., T. Yamada and N. Ueda, "Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling", *Proceedings of ACL-AJCNLP*, pp. 100–108, Suntec, Singapore, August 2009.

24. Johnson, M., T. L. Griffiths and S. Goldwater, "Bayesian inference for PCFGs via Markov Chain Monte Carlo", *Proceedings of NAACL-HLT*, pp. 139–146, Rochester, New York, April 2007.

25. Chiang, D., J. Graehl, K. Knight, A. Pauls and S. Ravi, "Bayesian Inference for Finite-State Transducers", *Proceedings of NAACL-HLT*, pp. 447–455, Los Angeles, California, June 2010.

26. Blunsom, P., T. Cohn, C. Dyer and M. Osborne, "A Gibbs Sampler for Phrasal Synchronous Grammar Induction", *Proceedings of ACL-AJCNLP*, pp. 782–790, Suntec, Singapore, August 2009.

27. DeNero, J., A. Bouchard-Côté and D. Klein, "Sampling Alignment Structure under

a Bayesian Translation Model", *Proceedings of EMNLP*, pp. 314–323, Honolulu, Hawaii, October 2008.

28. Xu, J., J. Gao, K. Toutanova and H. Ney, "Bayesian semi-supervised Chinese word segmentation for statistical machine translation", *Proceedings of COLING*, pp. 1017–1024, Manchester, UK, August 2008.

29. Nguyen, T., S. Vogel and N. A. Smith, "Nonparametric Word Segmentation for Machine Translation", *Proceedings of COLING*, pp. 815–823, 2010.

30. Chung, T. and D. Gildea, "Unsupervised Tokenization for Machine Translation", *Proceedings of EMNLP*, pp. 718–726, Singapore, August 2009.

31. Riley, D. and D. Gildea, "Improving the IBM Alignment Models Using Variational Bayes", *Proceedings of ACL: Short Papers*, pp. 306–310, 2012.

32. Geman, S. and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 6, No. 6, pp. 721–741, 1984.

33. Och, F. J. and H. Ney, "A comparison of alignment models for statistical machine translation", *Proceedings of COLING*, pp. 1086–1090, 2000.

34. Newman, D., A. U. Asuncion, P. Smyth and M. Welling, "Distributed Algorithms for Topic Models", *Journal of Machine Learning Research*, Vol. 10, pp. 1801–1828, 2009.

35. Kikui, G., S. Yamamoto, T. Takezawa and E. Sumita, "Comparative study on corpora for speech translation", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14, No. 5, pp. 1674–1682, 2006.

36. Paul, M., M. Federico and S. Stücker, "Overview of the IWSLT 2010 Evaluation Campaign", *Proceedings of IWSLT*, pp. 3–27, December 2010.

37. Chen, S. F. and J. Goodman, "An empirical study of smoothing techniques for language modeling", *Computer Speech and Language*, Vol. 13, pp. 359–394, 1999.

38. Mermer, C., *gibbs_ibm-model-1_collapsed_v1.00.perl*, 2011, `http://anthology.aclweb.org/attachments/P/P11/P11-2032.Software.txt`, accessed at March 2019.

39. Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and E. Herbst, "Moses: open source toolkit for statistical machine translation", *Proceedings of ACL: Demo and Poster Sessions*, pp. 177–180, Prague, Czech Republic, June 2007.

40. Stolcke, A., "SRILM – an extensible language modeling toolkit", *Proceedings of ICSLP*, Vol. 3, 2002.

41. Zaidan, O. F., "Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems", *The Prague Bulletin of Mathematical Linguistics*, Vol. 91, No. 1, pp. 79–88, 2009.

42. Papineni, K., S. Roukos, T. Ward and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation", *Proceedings of ACL*, pp. 311–318, Philadelphia, Pennsylvania, July 2002.

43. Clark, J. H., C. Dyer, A. Lavie and N. A. Smith, "Better Hypothesis Testing for Statistical Machine Translation: Controlling for Optimizer Instability", *Proceedings of ACL:HLT*, pp. 176–181, Portland, Oregon, 2011.

44. Shen, W., B. Delaney, T. Anderson and R. Slyh, "The MIT-LL/AFRL IWSLT-2007 MT system", *Proceedings of IWSLT*, Trento, Italy, 2007.

45. Dyer, C., J. H. Clark, A. Lavie and N. A. Smith, "Unsupervised Word Alignment with Arbitrary Features", *Proceedings of ACL:HLT*, pp. 409–419, Portland,

Oregon, June 2011.

46. Riley, D. and D. Gildea, *Improving the performance of GIZA++ using variational Bayes*, Tech. Rep. 963, The University of Rochester, Computer Science Department, December 2010.

47. Oflazer, K., "Two-level description of Turkish morphology", *Literary and Linguistic Computing*, Vol. 9, No. 2, 1994.

48. Eisele, A. and Y. Chen, "MultiUN: A Multilingual Corpus from United Nation Documents", *Proceedings of LREC*, pp. 2868–2872, 2010.

49. Nizar Habash, O. R. and R. Roth, "MADA+TOKAN: A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization", *Proceedings of Second International Conference on Arabic Language Resources and Tools*, 2009.

50. Callison-Burch, C., P. Koehn, C. Monz, M. Post, R. Soricut and L. Specia, "Findings of the 2012 Workshop on Statistical Machine Translation", *Proceedings of WMT*, pp. 10–51, 2012.

51. Bojar, O., Z. Žabokrtský, O. Dušek, P. Galuščáková, M. Majliš, D. Mareček, J. Maršík, M. Novák, M. Popel and A. Tamchyna, "The Joy of Parallelism with CzEng 1.0", *Proceedings of LREC*, 2012.

52. Chatterjee, S. and N. Cancedda, "Minimum Error Rate Training by Sampling the Translation Lattice", *Proceedings of EMNLP*, pp. 606–615, 2010.

53. Cer, D., D. Jurafsky and C. D. Manning, "Regularization and Search for Minimum Error Rate Training", *Proceedings of WMT*, pp. 26–34, 2008.

54. Kumar, S. and W. Byrne, "Minimum Bayes-Risk Decoding for Statistical Machine Translation", *Proceedings of HLT-NAACL*, pp. 169–176, 2004.

55. Snover, M., B. Dorr, R. Schwartz, L. Micciulla and J. Makhoul, "A study of translation edit rate with targeted human annotation", *Proceedings of AMTA*, pp. 223–231, 2006.

56. Bodrumlu, T., K. Knight and S. Ravi, "A New Objective Function for Word Alignment", *Proceedings of NAACL-HLT Workshop on Integer Linear Programming for Natural Language Processing*, pp. 28–35, Boulder, Colorado, June 2009.

57. Schoenemann, T., "Probabilistic Word Alignment under the L_0-norm", *Proceedings of CoNLL*, pp. 172–180, 2011.

58. Guzman, F., Q. Gao and S. Vogel, "Reassessment of the role of phrase extraction in PBSMT", *Proceedings of MT Summit*, 2009.

59. Bojar, O. and M. Prokopová, "Czech-English Word Alignment", *Proceedings of LREC*, pp. 1236–1239, 2006.

60. Fraser, A. and D. Marcu, "Measuring Word Alignment Quality for Statistical Machine Translation", *Computational Linguistics*, Vol. 33, No. 3, pp. 293–303, 2007.

61. Gao, Q. and S. Vogel, "Parallel Implementations of Word Alignment Tool", *Proceedings of ACL-HLT Workshop on Software Engineering, Testing, and Quality Assurance for NLP*, pp. 49–57, 2008.

62. Habash, N. and F. Sadat, "Arabic Preprocessing Schemes for Statistical Machine Translation", *Proceedings of NAACL-HLT: Short Papers*, pp. 49–52, New York City, New York, June 2006.

63. Oflazer, K. and İ. Durgar El-Kahlout, "Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation", *Proceedings of Second Workshop on Statistical Machine Translation*, pp. 25–32, Prague, Czech Republic, June 2007.

64. Bisazza, A. and M. Federico, "Morphological Pre-Processing for Turkish to English Statistical Machine Translation", *Proceedings of IWSLT*, pp. 129–135, Tokyo, Japan, 2009.

65. Lopez, A. and P. Resnik, "Word-based alignment, phrase-based translation: What's the link?", *Proceedings of 7th Conference of the Association for Machine Translation in the Americas (AMTA-06)*, pp. 90–99, 2006.

66. Durgar El-Kahlout, İ. and K. Oflazer, "Initial explorations in English to Turkish statistical machine translation", *Proceedings of WMT*, pp. 7–14, New York City, New York, June 2006.

67. Lee, Y.-S., "Morphological Analysis for Statistical Machine Translation", *Proceedings of HLT-NAACL: Short Papers*, pp. 57–60, Boston, Massachusetts, May 2004.

68. Creutz, M. and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning", *ACM Transactions on Speech and Language Processing*, Vol. 4, No. 1, pp. 1–34, 2007.

69. Sadat, F. and N. Habash, "Combination of Arabic Preprocessing Schemes for Statistical Machine Translation", *Proceedings of COLING-ACL*, pp. 1–8, Sydney, Australia, July 2006.

70. Durgar El-Kahlout, I. and K. Oflazer, "Exploiting morphology and local word reordering in English-to-Turkish phrase-based statistical machine translation", *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 18, No. 6, pp. 1313–1322, August 2010.

71. Poon, H., C. Cherry and K. Toutanova, "Unsupervised Morphological Segmentation with Log-Linear Models", *Proceedings of HLT-NAACL*, pp. 209–217, Boulder, Colorado, June 2009.

72. Snyder, B. and R. Barzilay, "Unsupervised Multilingual Learning for Morphological

Segmentation", *Proceedings of ACL-HLT*, pp. 737–745, Columbus, Ohio, June 2008.

73. Dasgupta, S. and V. Ng, "High-Performance, Language-Independent Morphological Segmentation", *Proceedings of NAACL-HLT*, pp. 155–163, Rochester, New York, April 2007.

74. Brent, M. R., "An efficient, probabilistically sound algorithm for segmentation and word discovery", *Machine Learning*, Vol. 34, No. 1, pp. 71–105, 1999.

75. Arısoy, E., *Statistical and Discriminative Language Modeling for Turkish Large Vocabulary Continuous Speech Recognition*, Ph.D. Thesis, Boğaziçi University, 2009.

76. Virpioja, S., J. J. Väyrynen, M. Creutz and M. Sadeniemi, "Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner", *Proceedings of MT Summit XI*, pp. 491–498, Copenhagen, Denmark, 2007.

77. Luong, M.-T., P. Nakov and M.-Y. Kan, "A Hybrid Morpheme-Word Representation for Machine Translation of Morphologically Rich Languages", *Proceedings of 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 148–157, Cambridge, MA, October 2010.

78. Kurimo, M., S. Virpioja, V. T. Turunen, G. W. Blackwood and W. Byrne, "Overview and Results of Morpho Challenge 2009", *Working notes of the CLEF workshop*, 2009.

79. Elming, J., N. Habash and J. M. Crego, "Combination of statistical word alignments based on multiple preprocessing schemes", C. Goutte, N. Cancedda, M. Dymetman and G. Foster (Editors), *Learning Machine Translation*, chap. 5, pp. 93–110, MIT Press, 2009.

80. Chang, P.-C., M. Galley and C. D. Manning, "Optimizing Chinese word segmentation for machine translation performance", *Proceedings of WMT*, pp. 224–232,

Columbus, Ohio, June 2008.

81. Talbot, D. and M. Osborne, "Modelling Lexical Redundancy for Machine Translation", *Proceedings of COLING-ACL*, pp. 969–976, Sydney, Australia, July 2006.

82. Creutz, M. and K. Lagus, *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*, Publications in Computer and Information Science, Report A81, Helsinki University of Technology, March 2005.

83. Mermer, C., H. Kaya and M. U. Doğan, "The TÜBİTAK-UEKAE Statistical Machine Translation System for IWSLT 2010", *Proceedings of IWSLT*, pp. 183–188, Paris, France, December 2010.

84. Bojar, O., R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. Jimeno Yepes, P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Neveol, M. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor and M. Zampieri, "Findings of the 2016 Conference on Machine Translation", *Proceedings of the First Conference on Machine Translation*, pp. 131–198, Association for Computational Linguistics, Berlin, Germany, August 2016.

85. Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation", *CoRR*, Vol. abs/1609.08144, 2016.

86. Sutskever, I., O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks", *Advances in neural information processing systems*, pp. 3104–3112, 2014.

87. Bahdanau, D., K. Cho and Y. Bengio, "Neural machine translation by jointly

learning to align and translate", *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, 2015.

88. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention is All you Need", I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett (Editors), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008, Curran Associates, Inc., December 2017.

89. Chen, W., E. Matusov, S. Khadivi and J. Peter, "Guided Alignment Training for Topic-Aware Neural Machine Translation", *Proceedings of AMTA 2016, vol.1: MT Researchers' Track*, pp. 121–134, Austin, October 2016.

90. Liu, L., M. Utiyama, A. M. Finch and E. Sumita, "Neural Machine Translation with Supervised Attention", *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pp. 3093–3102, 2016.

91. Mi, H., Z. Wang and A. Ittycheriah, "Supervised Attentions for Neural Machine Translation", *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 2283–2288, 2016.

92. Denkowski, M. and G. Neubig, "Stronger Baselines for Trustable Results in Neural Machine Translation", *Proceedings of the First Workshop on Neural Machine Translation*, pp. 18–27, Association for Computational Linguistics, Vancouver, August 2017.

93. Koehn, P. and R. Knowles, "Six Challenges for Neural Machine Translation", *Proceedings of the First Workshop on Neural Machine Translation*, pp. 28–39, Association for Computational Linguistics, 2017.

94. Luong, T., I. Sutskever, Q. Le, O. Vinyals and W. Zaremba, "Addressing the Rare

Word Problem in Neural Machine Translation", *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 11–19, Association for Computational Linguistics, 2015.

95. Sennrich, R., B. Haddow and A. Birch, "Neural Machine Translation of Rare Words with Subword Units", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Association for Computational Linguistics, 2016.

APPENDIX A: DERIVATION OF THE GIBBS SAMPLING FORMULA

Here we describe the derivation of the Gibbs sampler for IBM Model 2 given in Equation (3.21). Since IBM Model 1 is a special case of Model 2 where $\mathbf{d}$ is fixed (Section 3.3.5), the derivation of the sampler for Model 1 given in Equation (3.6) would follow exactly the same steps, except that there would be no prior $P(\mathbf{d}, \mathbf{\Phi})$ and the related terms.

A.1. The Dirichlet Priors

We choose a simple prior for the parameters $\mathbf{T}$ where each $\mathbf{t}_e$ has an independent[6] Dirichlet prior with hyperparameters $\mathbf{\Theta}_e$ (Section 3.3.2):

$$P(\mathbf{t}_e; \mathbf{\Theta}_e) = \frac{1}{B(\mathbf{\Theta}_e)} \prod_{f=1}^{V_F} (t_{e,f})^{\theta_{e,f}-1}, \tag{A.1}$$

where $\theta_{e,f} > 0 \ \forall\{e, f\}$ and

$$B(\mathbf{\Theta}_e) \stackrel{def}{=} \frac{\prod_{f=1}^{V_F} \Gamma(\theta_{e,f})}{\Gamma(\sum_{f=1}^{V_F} \theta_{e,f})}. \tag{A.2}$$

Hence, the complete prior for $\mathbf{T}$ is given by:

$$P(\mathbf{T}; \mathbf{\Theta}) = \prod_{e=1}^{V_E} \frac{1}{B(\mathbf{\Theta}_e)} \prod_{f=1}^{V_F} (t_{e,f})^{\theta_{e,f}-1}. \tag{A.3}$$

[6]While the prior knowledge about $\mathbf{T}$ could have been possibly expressed as a more refined, correlated distribution; we show that a simple, independent prior is also successful in biasing the parameters away from flat distributions.

Similarly, from Section 3.3.5:

$$P(\mathbf{d};\mathbf{\Phi}) = \frac{1}{B(\mathbf{\Phi})} \prod_{r=-\max_s I}^{\max_s I} (d_r)^{\phi_r - 1}. \tag{A.4}$$

We further define the priors for the translation and distortion parameters to be independent so that $P(\mathbf{T},\mathbf{d}) = P(\mathbf{T})P(\mathbf{d})$.

A.2. The Complete Distribution

Since we are only interested in inferring $\mathbf{A}$, we integrate out the unknowns $\mathbf{T}$ and $\mathbf{d}$ in (3.20) using (A.3) and (A.4):

$$\begin{aligned}
&P(\mathbf{F},\mathbf{A}|\mathbf{E};\mathbf{\Theta},\mathbf{\Phi}) \\
&\quad = \int_{\mathbf{T}}\int_{\mathbf{d}} P(\mathbf{T};\mathbf{\Theta})P(\mathbf{d};\mathbf{\Phi})P(\mathbf{F},\mathbf{A}|\mathbf{E},\mathbf{T},\mathbf{d}) && \text{(A.5)} \\
&\quad = \int_{\mathbf{T}} \prod_{e=1}^{V_E} \frac{1}{B(\mathbf{\Theta}_e)} \prod_{f=1}^{V_F} (t_{e,f})^{N_{e,f}+\theta_{e,f}-1} \\
&\qquad \cdot \int_{\mathbf{d}} \frac{1}{B(\mathbf{\Phi})} \prod_{r=-\max_s I}^{\max_s I} (d_r)^{C_r+\phi_r-1} && \text{(A.6)} \\
&\quad = \prod_{e=1}^{V_E} \frac{1}{B(\mathbf{\Theta}_e)} \int_{\mathbf{t}_e} \prod_{f=1}^{V_F} (t_{e,f})^{N_{e,f}+\theta_{e,f}-1} \\
&\qquad \cdot \frac{1}{B(\mathbf{\Phi})} \int_{\mathbf{d}} \prod_{r=-\max_s I}^{\max_s I} (d_r)^{C_r+\phi_r-1}. && \text{(A.7)}
\end{aligned}$$

As a result of choosing conjugate priors, the integrands with respect to $\mathbf{t}_e$ and $\mathbf{d}$ in (A.7) can be recognized to be in the same form as the priors (i.e., Dirichlet distributions) with new sets of parameters $\mathbf{N}_e + \mathbf{\Theta}_e$ and $\mathbf{C} + \mathbf{\Phi}$, respectively, where we have defined $\mathbf{N}_e = N_{e,1} \cdots N_{e,V_F}$ and $\mathbf{C} = C_{-\max_s I} \cdots C_{\max_s I}$. Since the integral of a probability distribution is equal to 1, we obtain the closed-form expression:

$$P(\mathbf{F},\mathbf{A}|\mathbf{E};\mathbf{\Theta},\mathbf{\Phi}) = \prod_{e=1}^{V_E} \frac{B(\mathbf{N}_e + \mathbf{\Theta}_e)}{B(\mathbf{\Theta}_e)} \cdot \frac{B(\mathbf{C}+\mathbf{\Phi})}{B(\mathbf{\Phi})}. \tag{A.8}$$

A.3. Gibbs Sampler Derivation

Given the complete distribution in (A.8), the Gibbs sampling formula $P(z_j|\mathbf{z}^{\neg j})$ (Section 3.3.3) can be derived as:

$$
\begin{aligned}
&P(a_j|\mathbf{E},\mathbf{F},\mathbf{A}^{\neg j};\boldsymbol{\Theta},\boldsymbol{\Phi}) \\
&= \frac{P(\mathbf{F},\mathbf{A}|\mathbf{E};\boldsymbol{\Theta},\boldsymbol{\Phi})}{P(\mathbf{F},\mathbf{A}^{\neg j}|\mathbf{E};\boldsymbol{\Theta},\boldsymbol{\Phi})} && \text{(A.9)} \\
&\propto \frac{P(\mathbf{F},\mathbf{A}|\mathbf{E};\boldsymbol{\Theta},\boldsymbol{\Phi})}{P(\mathbf{F}^{\neg j},\mathbf{A}^{\neg j}|\mathbf{E};\boldsymbol{\Theta},\boldsymbol{\Phi})} && \text{(A.10)} \\
&= \prod_{e=1}^{V_E} \frac{B(\mathbf{N}_e+\boldsymbol{\Theta}_e)}{B(\mathbf{N}_e^{\neg j}+\boldsymbol{\Theta}_e)} \cdot \frac{B(\mathbf{C}+\boldsymbol{\Phi})}{B(\mathbf{C}^{\neg j}+\boldsymbol{\Phi})} && \text{(A.11)} \\
&= \frac{B(\mathbf{N}_{e_{a_j}}+\boldsymbol{\Theta}_{e_{a_j}})}{B(\mathbf{N}_{e_{a_j}}^{\neg j}+\boldsymbol{\Theta}_{e_{a_j}})} \cdot \frac{B(\mathbf{C}+\boldsymbol{\Phi})}{B(\mathbf{C}^{\neg j}+\boldsymbol{\Phi})} && \text{(A.12)} \\
&= \frac{\prod_{f=1}^{V_F}\Gamma\left(N_{e_{a_j},f}+\theta_{e_{a_j},f}\right)}{\prod_{f=1}^{V_F}\Gamma\left(N_{e_{a_j},f}^{\neg j}+\theta_{e_{a_j},f}\right)} \\
&\quad \cdot \frac{\Gamma\left(\sum_{f=1}^{V_F}\left(N_{e_{a_j},f}^{\neg j}+\theta_{e_{a_j},f}\right)\right)}{\Gamma\left(\sum_{f=1}^{V_F}\left(N_{e_{a_j},f}+\theta_{e_{a_j},f}\right)\right)} \\
&\quad \cdot \frac{\prod_{r=-\max_s I}^{\max_s I}\Gamma\left(C_r+\phi_r\right)}{\prod_{r=-\max_s I}^{\max_s I}\Gamma\left(C_r^{\neg j}+\phi_r\right)} \\
&\quad \cdot \frac{\Gamma\left(\sum_{r=-\max_s I}^{\max_s I}\left(C_r^{\neg j}+\phi_r\right)\right)}{\Gamma\left(\sum_{r=-\max_s I}^{\max_s I}\left(C_r+\phi_r\right)\right)} && \text{(A.13)} \\
&= \frac{\Gamma\left(N_{e_{a_j},f_j}+\theta_{e_{a_j},f_j}\right)}{\Gamma\left(N_{e_{a_j},f_j}^{\neg j}+\theta_{e_{a_j},f_j}\right)} \\
&\quad \cdot \frac{\Gamma\left(\sum_{f=1}^{V_F}\left(N_{e_{a_j},f}^{\neg j}+\theta_{e_{a_j},f}\right)\right)}{\Gamma\left(\sum_{f=1}^{V_F}\left(N_{e_{a_j},f}+\theta_{e_{a_j},f}\right)\right)} \\
&\quad \cdot \frac{\Gamma\left(C_{a_j-\lfloor j\frac{I}{J}\rfloor}+\phi_{a_j-\lfloor j\frac{I}{J}\rfloor}\right)}{\Gamma\left(C_{a_j-\lfloor j\frac{I}{J}\rfloor}^{\neg j}+\phi_{a_j-\lfloor j\frac{I}{J}\rfloor}\right)} \\
&\quad \cdot \frac{\Gamma\left(\sum_{r=-\max_s I}^{\max_s I}\left(C_r^{\neg j}+\phi_r\right)\right)}{\Gamma\left(\sum_{r=-\max_s I}^{\max_s I}\left(C_r+\phi_r\right)\right)} && \text{(A.14)}
\end{aligned}
$$

$$
\begin{aligned}
&P(a_j|\mathbf{E},\mathbf{F},\mathbf{A}^{\neg j};\boldsymbol{\Theta},\boldsymbol{\Phi}) \\
&= \left(N^{\neg j}_{e_{a_j},f_j} + \theta_{e_{a_j},f_j}\right) \cdot \frac{1}{\sum_{f=1}^{V_F}\left(N^{\neg j}_{e_{a_j},f} + \theta_{e_{a_j},f}\right)} \\
&\quad \cdot \left(C^{\neg j}_{a_j-\lfloor j\frac{I}{J}\rfloor} + \phi_{a_j-\lfloor j\frac{I}{J}\rfloor}\right) \cdot \frac{1}{\sum_{r=-\max_s I}^{\max_s I}\left(C^{\neg j}_{r} + \phi_{r}\right)} \qquad \text{(A.15)} \\
&\propto \frac{N^{\neg j}_{e_{a_j},f_j} + \theta_{e_{a_j},f_j}}{\sum_{f=1}^{V_F} N^{\neg j}_{e_{a_j},f} + \sum_{f=1}^{V_F}\theta_{e_{a_j},f}} \\
&\quad \cdot \left(C^{\neg j}_{a_j-\lfloor j\frac{I}{J}\rfloor} + \phi_{a_j-\lfloor j\frac{I}{J}\rfloor}\right), \qquad \text{(A.16)}
\end{aligned}
$$

where (A.10) follows since $P(f_j|\mathbf{A}^{\neg j},\mathbf{E};\boldsymbol{\Theta})$ is independent of a_j, in (A.11) we used (A.8), in (A.13) we used (A.2) and grouped similar factors, in (A.15) each fraction is simplified using the property of the gamma function $\Gamma(x+1) = x\Gamma(x)$, and in (A.16) the proportionality comes from the omission of the last term in (A.15), which is constant for all values of a_j.