Improvements in Hierarchical Phrase-based Statistical Machine Translation

by

Baskaran Sankaran
M.S. (Research), Anna University, 2002
B.E., Madurai Kamaraj University, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Sciences

© Baskaran Sankaran 2013
SIMON FRASER UNIVERSITY
Fall 2013

All rights reserved.
However, in accordance with the *Copyright Act of Canada*, this work may be reproduced without authorization under the conditions for "Fair Dealing." Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.

APPROVAL

Name: Baskaran Sankaran

Degree: Doctor of Philosophy

Title of Thesis: Improvements in Hierarchical Phrase-based Statistical Machine Translation

Examining Committee: Dr. Arrvindh Shriraman, Assistant Professor
Chair

Dr. Anoop Sarkar, Associate Professor
Senior Supervisor

Dr. Greg Mori, Associate Professor
Supervisor

Dr. Gholamreza Haffari, Lecturer
Information Technology, Monash University
Supervisor

Dr. Fred Popowich, Professor
Internal Examiner

Dr. David Chiang, Research Assistant Professor
Computer Science, University of Southern California
External Examiner

Date Approved: December 19th, 2013

Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the non-exclusive, royalty-free right to include a digital copy of this thesis, project or extended essay[s] and associated supplemental files ("Work") (title[s] below) in Summit, the Institutional Research Repository at SFU. SFU may also make copies of the Work for purposes of a scholarly or research nature; for users of the SFU Library; or in response to a request from another library, or educational institution, on SFU's own behalf or for one of its users. Distribution may be in any form.

The author has further agreed that SFU may keep more than one copy of the Work for purposes of back-up and security; and that SFU may, without changing the content, translate, if technically possible, the Work to any medium or format for the purpose of preserving the Work and facilitating the exercise of SFU's rights under this licence.

It is understood that copying, publication, or public performance of the Work for commercial purposes shall not be allowed without the author's written permission.

While granting the above uses to SFU, the author retains copyright ownership and moral rights in the Work, and may deal with the copyright in the Work in any way consistent with the terms of this licence, including the right to change the Work for subsequent purposes, including editing and publishing the Work in whole or in part, and licensing the content to other parties as the author may desire.

The author represents and warrants that he/she has the right to grant the rights contained in this licence and that the Work does not, to the best of the author's knowledge, infringe upon anyone's copyright. The author has obtained written copyright permission, where required, for the use of any third-party copyrighted material contained in the Work. The author represents and warrants that the Work is his/her own original work and that he/she has not previously assigned or relinquished the rights conferred in this licence.

Simon Fraser University Library
Burnaby, British Columbia, Canada

revised Fall 2013

Abstract

Hierarchical phrase-based translation (Hiero) is a statistical machine translation (SMT) model that encodes translation as a synchronous context-free grammar derivation between source and target language strings (Chiang, 2005; Chiang, 2007). Hiero models are more powerful than phrase-based models in capturing complex source-target reordering as well as discontiguous phrases, while being easier to estimate and decode with compared to their full syntax-based counterparts.

In this thesis, we propose improvements to two broad aspects of the Hiero translation pipeline: i) learning Hiero translation model and estimating their parameters and ii) parameter tuning for discriminative log-linear models that are used to decode with such features. We use our own open-source implementation of Hiero called *Kriya* (Sankaran et al., 2012b) for all the experiments in this thesis. This thesis contains the following specific contributions:

- We propose a Bayesian model for learning Hiero grammars as an alternative to the heuristic method usually used in Hiero. Our model learns a peaked distribution of grammars, which consistently performs better than the heuristically extracted grammars across several language pairs (Sankaran et al., 2013a).

- We propose a novel *unified-cascade framework* for jointly learning alignments and the Hiero translation rules by removing the disconnect between the alignments and extracted synchronous context-free grammar. This is the first time a joint training framework is being proposed for Hiero, where we iterate the two step inference so that it learns in alternate iterations the phrase alignments and then the Hiero rules that are consistent with alignments.

- We extend our Bayesian model for extracting *compact* Hiero translation rules using arity-1 grammars, resulting in up to 57% reduction in model size while retaining the translation performance (Sankaran et al., 2011; Sankaran et al., 2012a).

- We propose several novel approaches for parameter tuning of discriminative log-linear models

for SMT which can be used for jointly optimizing towards multiple evaluation metrics. We show that our methods for multi-objective tuning for SMT yield substantial gains in translation quality measured through automatic as well as human evaluations (Sankaran et al., 2013b; Duh et al., 2013).

This thesis is dedicated to the One

Acknowledgements

I would like to record my sincere gratitude to my thesis advisor Prof. Anoop Sarkar for his guidance and consistent support during my PhD research. He kindled my interests in Statistical Machine Translation and also imparted me valuable skills in research as well as in teaching. His classes on natural language processing topics has always been both enjoyable and thought provoking. He has also been a great source of strength at a personal level, providing me support and inspiration.

I would like to thank my thesis committee: Prof. Greg Mori, Dr. Gholamreza Haffari, Prof. Fred Popowich and Prof. David Chiang. I have hugely benefitted from Prof. Mori's questions and insightful feedback on my thesis as also from his machine learning course. Dr. Haffari has been a good friend as well apart from being a collaborator in some of my works through which I learned a great deal in Bayesian modelling. Prof. Popowich has amazed with his comments on minutest details in research and I also enjoyed working with him in some projects. I am honoured to have Prof. David Chiang as the external examiner for my thesis defence and his particular work on Hiero among the multitude of his research has been a foundation for this thesis.

I have had the fortune of working with these wonderful people during my PhD or before that, who taught me various things ranging from linguistics to Map-Reduce to NLP: Kevin Duh, Vijay K. Shanker, S. Rajendran, Miles Osborne, L. Sobha, K. V. Subbarao, Maite Taboada and Raghavendra Udupa. Interacting with researchers at conferences and other venues is both fun and rewarding; I have enjoyed discussing with some of them: Srinivas Bangalore, Chris Callison-Burch, Jonathan H Clark, Chris Dyer, Philipp Koehn, Adam Lopez, Graham Neubig, Avneesh Saluja, Taro Watanabe, Dekai Wu, and Omar F. Zaidan.

I am very glad to have worked/ interacted with several current and alumni members of the Natural Language Lab with which I was affiliated: Diptesh Chatterjee, Ann Clifton, Rohit Dholakia, Manaal Faruqui, Ajeet Grewal, Willem (Bruce) Krayenhoff, Young-Chan Kim, Yudong Liu, Porus Patell, Marzieh Razavi, Majid Razmara, Maxim Roy, Maryam Siahbani, Milan Tofiloski, Ravikiran

Vadlapudi, Max Whitney and Winona Wu.

I also had the amazing company of several friends during my life in Vancouver. I particularly enjoyed the time I spent with Tharini Adithan, Usman Alim, Ram Shakthi Bhattar, Ankit Gupta, Ajit Khosla, Manu Pallappa, Ashok Raja, Sunder Rajan, Thara Rajasekaran, Pradeep Reddy Ramana, Malathi Ravishankar, Ananthanarayan Sankaranarayanan, Somsubhra Sharangi, Arrvindh Shriraman, Shanker Shyam Sundhar and several others.

I am deeply indebted to my parents Krishnambal and Sankaran along with my sister Anuradha, who gave me the wonderful background that shaped me to what I am today. I am also thankful to my parents for giving me freedom to pursue my interests and for encouraging my exploration. And finally, but also importantly I would like to thank my wife Vasudha for her support and patience throughout my PhD studies. This thesis for sure would not have been possible without her. And I am very pleased to recount that, she was also preparing to welcome Aniruddha, our son - who was to be born three days after my defense, while supporting me through the final stages of my PhD.

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols - Generic

$\mathbf{f}, \mathbf{e}$	Set of source and target sentences in parallel corpus
f, e	Specific source and target sentence pair (also for a phrase pair)
A	Word alignments of a sentence/ phrase pair
$D(f)$	Set of derivations for a given source sentence f
d	A specific derivation in $D(f)$
$P(d)$	Probability of a derivation under the log-linear model
h	A translation hypothesis for a source sentence f
$\mathscr{Y}_e$	A function over the target-side yields for a given sentence f
ϕ	Features of the log-linear statistical model
w	Weights of the log-linear model feature functions
h_{phi}	Feature vector for a specific hypothesis h
X	Non-terminal symbol used in Hiero
S	Top-level non-terminal for denoting sentence productions
T	Set of terminals in source and target languages
γ, α	Sequence of terminals and non-terminals in source and target sides respectively
$\sim$	One-to-one correspondence between the non-terminals in the right-hand side of Hiero rules
$\otimes$	Mixture operation used by the ensemble decoder

List of Symbols - Hiero Grammar Extraction

$\mathbf{f}$, $\mathbf{e}$	Set of source and target sentences in parallel corpus
f, e	Specific source and target sentence pair (also for a phrase pair)
A	Word alignments of a sentence/ phrase pair
$\mathcal{X}$	Set of initial phrase-pairs from Och and Ney (2004)
Φ	Set of latent derivations of $\mathcal{X}$
$\mathcal{G}$	Hiero grammar
G_m	Minimal Hiero grammar
θ	Posterior parameters of the inferred Hiero grammar $\mathcal{G}$
x	A specific phrase-pair, i.e $x \in \mathcal{X}$
ϕ_x	Set of derivations for the phrase-pair x
d	A specific derivation, i.e. $d \in \phi_x$
$\mathbf{r}$	Rules contained in the derivation d
$\mathcal{T}_G$	Tripartite graph of three types of nodes, viz. $\mathcal{X}$, Φ and $\mathcal{G}$
v_x	Individual phrase-pair vertex in $\mathcal{T}_G$
$v_{d,x}$	Set of derivation vertices in $\mathcal{T}_G$ corresponding to x
v_r	Individual rule vertex (part of grammar nodes) in $\mathcal{T}_G$
z_d	Derivation type parameter in Bayesian model
γ, ϕ^z	Priors for deciding derivation type z_d under Bernoulli and Dirichlet distributions respectively
α_h, $\boldsymbol{\alpha_z}$	Hyperparameters (possibly a vector) of a Dirichlet dist or DP
P_0, p_0	Base measure
l_x	Lexical alignment probability for x
lf_x, lb_x	Forward and reverse lexical alignment probabilities for x
$c_d(r)$	Frequency of rule r observed in derivation d
u, π	Parameters of the Variational distributions
t	Time steps in Bayesian inference

List of Symbols - Multi-metric Optimization

$\mathbf{f}, \mathbf{e}$	Source and target sides of a tuning set
H	Hypotheses set used for tuning (could simply be the N-best list)
H_{ens}	Hypotheses set produced by ensemble decoder
$\mathcal{N}$	Decoding function to generate H given $(\mathbf{f}, \mathbf{e})$
$\otimes$	Mixture operation employed by the ensemble decoder
$\mathcal{N}_{ens}$	Ensemble-decoding function to generate H_{ens} given $(\mathbf{f}, \mathbf{e})$, $\mathbf{w}$, λ and $\otimes$
M	Evaluation metric that is being optimized under MMO
w	Weights of the log-linear model feature functions
$\mathbf{w}$	Set of weights w for all the ensemble components
λ	Meta-weight applied of the evaluation metric M
g	Combining function for the MMO (instantiations vary for the different MMO methods)
p_s	Pareto-optimal solution
h	A specific hypothesis from H
$\{M(H)\}$	Metric scores for hypotheses in given H
f	Function for computing the approximate Pareto-frontier
$\{\mathcal{F}\}$	Pareto-frontier approximation for a given H, $\{M(H)\}$ and λ
$\mathcal{T}$	Labelled training data for PRO classifier
j	Tuning iteration run

Chapter 1

Introduction

Machine Translation (MT) has been an important area of Artificial Intelligence (AI) that has fascinated the researchers since early days. The introduction of statistical alignment models by Brown et al. (1993) shifted the research from the rule-based translation systems towards Statistical Machine Translation (SMT). This shift was also made possible by the availability of fairly large quantities of parallel texts as well as by the advent of increasingly powerful computers.

The initial *word*-based (Brown et al., 1993) SMT models gave way to the subsequent phrase-based models (Marcu and Wong, 2002; Och and Ney, 2002; Och and Ney, 2004), where the basic translation units are arbitrarily long *phrases* that do not need to have any syntactic validity.

Hierarchical phrase-based machine translation, popularly known as *Hiero* (Chiang, 2005; Chiang, 2007) is still another prominent approach for SMT. Hiero improves upon the phrase-based models by capturing cross-lingual translation phenomena such as discontiguous phrasal correspondences and long-distance reordering.

In contrast to the phrase-based models that employ flat translation rules, Hiero encodes translations through nested phrase-pairs having non-terminals that are co-indexed on both source and target sides. Each linked non-terminal pair act as substitution site for nesting another phrase-pair. This notion of hierarchy allows the Hiero models to capture the long-distance reordering between source and target languages in a way that is better than the phrase-based models. Additionally they also model discontiguous translations as exemplified by the canonical example of translating the English word *not* as *ne* ___ *pas* in French (with an appropriate verb form inserted between *ne* and *pas*).

Formally Hiero grammar is a form of synchronous context-tree grammar (SCFG), whereby it generates the source and target sentences simultaneously in a derivation process. The rules of the

Hiero grammar are learned using a heuristic approach from the word-aligned phrase-pairs. The decoding is performed with a CKY-style chart-parsing decoder in conjunction with beam search and cube pruning for efficient search as explained in Chiang (2007). We hold more detailed exposition of Hiero models for chapter 2 that provides the background for various topics in this thesis.

The syntax-based models represent the evolutionary next step in SMT. They utilize syntactic structure of source and/ or target sides for translation. Researchers have proposed myriad models using different combinations such as string-to-tree, tree-to-tree and so on (Yamada and Knight, 2001; Quirk et al., 2005; Galley et al., 2006, *inter alia*). A major limitation of these models is their requirement for a syntactic parser for at least one language, which limits its application to a few high resource languages. Unlike the full syntax-based models, Hiero does not require any linguistic parser making it an attractive choice for wide variety of languages.

Hiero models are shown to attain competitive performance with other statistical frameworks (Zollmann et al., 2008; Callison-Burch et al., 2012) for several language pairs as well as for large-scale corpora settings that are common in SMT. They are also shown to outperform the phrase-based models for language-pairs involving complex reordering such as Chinese-English (Chiang, 2007).

1.1 Summary of Contributions

In this thesis we propose several improvements to the Hiero model, broadly focusing on the following two aspects of the machine translation pipeline i) in the model training phase for learning translation grammar and ii) in the parameter optimization step for tuning the feature weights.

Hiero uses a heuristic approach for learning the translation grammar from the word-aligned sentence pairs. This approach suffers from the disadvantage of poor parameter estimation among others. We present an alternative approach for learning Hiero grammars under a Bayesian framework that leads to better parameter estimates. We assume the same setting as the heuristic rule extraction and use the initial phrase-pairs obtained by symmetrizing the word-alignments as the input to our Bayesian model in order to extract a grammar, which is directly used for translation. Our model is motivated by a novel prior based on Model-1 alignments and reasons over the space of derivation trees in learning the posterior distribution of the grammar.

We then broaden our scope and consider the larger part of the SMT training pipeline. SMT systems typically follow a standard pipeline of extracting word alignments, phrase alignments and translation rules in a series of steps, where some of the steps employ heuristic methods. The disconnect between each of these steps has led to a situation, where for example the word alignments are not optimal for the final goal, i.e translation (DeNero and Klein, 2010). Additionally in the case of Hiero, this disconnect is greater because the rules in the Hiero grammar are structurally different from the flat phrase alignments extracted in the previous step. While joint inference techniques have been proposed in recent years for phrase-based models (Burkett et al., 2010; Neubig et al., 2011), none exist for Hiero.

We present a novel *unified-cascade framework* for iteratively learning the alignment structure and the derivation trees in separate steps, where each step infers one by fixing the other. Our primary goal is to remove the disconnect between the different stages of the pipeline and the iterative setting encourages the alignments and grammar to be consistent with each other. At the same time the two steps employ distinct models to infer the respective posteriors by treating them to be conditionally independent. This separation further reduces the computational complexity of the framework.

Given the set of aligned phrases, the original heuristic rule extraction method extracts all possible SCFG rules that are consistent with the word alignments. While this approach is effective in practice, the resulting translation model is several times larger than a phrase-based model trained on the same corpus. This is because the non-terminal can be inserted are overlapping. The larger model size impacts the decoding speed and also leads to *overgeneration* and search errors. Existing approaches in the literature address these issues by employing various pruning techniques to reduce the grammar size or by restricting the depth of nesting to mitigate overgeneration and to reduce decoding complexity. However they rely on the heuristic approach in the first place to extract the grammar. In contrast we are motivated to extract a sparser grammar such that the model inherently rewards the good rules, while penalizing poor ones. We show that unary Hiero model (where rules can have only one non-terminal) in combination with our Bayesian rule extraction can be effective in controlling the grammar size without reducing the translation performance for several close language pairs. Thus our Bayesian model for unary Hiero grammars could be extremely useful in situations where the memory and processing capabilities are severely limited as in the case of smartphones and other mobile devices.

In the final part of the thesis, we explore parameter tuning for machine translation. Typical SMT systems use a log-linear framework with some standard set of features, whose weights are optimized to improve the translation quality as measured by a specific MT evaluation metric (Och, 2003; Hopkins and May, 2011). As opposed to the single-metric tuning, we propose multi-metric optimization (MMO) and introduce a new class of approaches for training feature weights. We draw inspiration from diverse areas for the proposed approaches, wherein the main idea is to jointly optimize towards multiple evaluation metrics. It should be noted that our MMO approaches can be employed for diverse tuning regimes apart from being applied to SMT frameworks other than Hiero.

We thus make contributions to *all* parts of the Hiero translation pipeline. The main contributions of this thesis are summarized below.

- We present a Bayesian model for extracting translation grammars for Hiero as an alternative to the traditional heuristic method. The key ideas are to sample an entire derivation tree instead of individual rules, as well as globally re-weighting the rule counts. This results in a sparse Hiero grammar having better probability estimates helping the decoder to better discriminate competing hypotheses during decoding. We further present an extension of our model in a distributed setting that allows us to learn grammars for large parallel corpora that are typical in SMT.

- We introduce a *unified-cascade* framework for jointly learning the alignments as well as the Hiero grammar. Unlike the joint approaches proposed earlier for phrase-based systems, our approach uses two distinct models for alignments and grammar (in a cascade), while additionally having an external (unifying) layer that iterates between the two. The iteration of alignment and rule extraction steps encourage the alignments to be optimized for the translation grammar and vice versa. Our approach has the advantage that it completely obviates the need for any heuristic component in the pipeline. We validate the proposed framework by employing a simple alignment model together with our Variational-Bayes grammar extraction.

- We then propose a variant of our Bayesian model for learning compact Hiero grammars employing unary Hiero grammars. We show the unary grammars to be sufficient for several close language pairs. The grammars extracted by our model are more appropriate for further pruning unlike the heuristic extraction method as they substantially reduce the model size while achieving competitive BLEU scores as the full unpruned model.

- In the context of parameter optimization for SMT, we explore multi-metric optimization (MMO) with the goal of improving the translation performance for several MT evaluation metrics. We propose several approaches for jointly optimizing multiple metrics, drawing ideas from Economics and Engineering; some of our important approaches are *ensemble combination*, *lateen* and *ensemble tuning*. Our results show substantial gains over the classical single-metric optimization apart from statistically significant gains for BLEU. Our human evaluation experiments also show statistically significant reduction in the human effort for post-editing the translations from a multi-metric optimized system. We also present detailed analysis about the interplay between individual metrics as also their sensitivity to translation changes for different language-pairs. We finally propose some future extensions by formulating multi-metric optimization in the Game theoretic perspective employing *collective bargaining* strategies.

- Finally, we have released Kriya[1] - our Python-based Hiero implementation, for machine translation research. We will also be releasing the code for multi-metric optimization, which has been integrated into Kriya.

1.2 Thesis Overview

The organization of the different chapters of this thesis is explained below.

Chapter 2 discusses the background materials that are relevant for this thesis. We first explain the Hiero translation framework followed by a description of the novel features of *Kriya*, which is used for all the experiments reported in this thesis. Finally this chapter covers the background information for parameter tuning in SMT particularly covering topics such as ensemble decoding (Razmara et al., 2012) and the pairwise-ranking (Hopkins and May, 2011).

I present a Bayesian model for learning Hiero grammar in **Chapter 3** that leads to better estimation of rule parameters compared to the heuristic method. The key idea in this is to globally re-weight the rules based on evidence from the entire set of phrase pairs. I also discuss a distributed version of the Variational Bayes inference, which makes the inference practical for large corpora that are common in SMT. I also analyze the characteristics of the grammar learned by the Bayesian model by contrasting it with the heuristically extracted grammar.

Having presented the Bayesian model for Hiero grammar extraction in the previous chapter, I take a step back and look into the issue of alignment and grammar extraction steps being disjoint. I devote **Chapter 4** to introduce a *unified-cascade* framework that elegantly combines these two steps. The framework allows joint inference of alignments and Hiero grammar through two distinct models alternating in an iterative setup. We demonstrate the viability of this framework using the alignment model proposed by Neubig et al. (2011) and our Bayesian model for grammar extraction (Sankaran et al., 2012a).

[1] https://github.com/sfu-natlang/Kriya

I focus on the task of learning a compact unary Hiero grammar in **Chapter 5** and present two different approaches. Both approaches extract a derivation tree for each phrase pair along with its component rules (as opposed extracting rules independently). First I describe a combinatorial optimization approach that (loosely) formulates Hiero rule extraction as a minimum set cover problem over a tripartite graph of phrase pairs, derivations and component rules. Under this formulation I propose a greedy algorithm for solving the optimization problem. For the second approach, I present a variant of the Bayesian model proposed earlier and experiment with two different inference methods.

Changing gears, I move on to explore the multi-metric optimization (MMO) in the context of parameter tuning for SMT in **Chapter 6**. We seek to optimize the parameters of a translation system to improve its performance across several MT evaluation metrics instead of the conventional single-metric optimization. I discuss our MMO approaches and then present the results showing significant gains in both automatic and human evaluations. I finally present an analysis of different evaluation metrics, specifically about their interplay with other metrics and their sensitivity to the translation changes.

I close the thesis with a summary and a discussion on future directions in **Chapter 7**.

Chapter 2

Background

This chapter covers the background material for the topics that are relevant for this thesis. We start by providing a formal definition of Hiero grammars and then briefly explain the training process in Hiero and also its chart-parsing decoder. Following this, we explain the novel language model integration approach used in Kriya and summarize its features. We then describe ensemble decoding and close this chapter with a discussion on pairwise ranking optimization (PRO).

2.1 Hiero Grammar: Formal Definition

Hiero is based on synchronous context-free grammar (SCFG) formalism that generates the source and target sentences by successively rewriting the non-terminals in the production rules starting from a top-level rule rooted at S.

Formally a grammar G in Hiero is a special case of SCFG and is defined as a 4-tuple: $G = (T, N, R, R_g)$, where T and N are the set of terminals and non-terminals in G. Hiero grammars typically use two types of non-terminals X and S, where S is the special start symbol. R is a set of production rules of the form:

$$X \rightarrow <\gamma,\, \alpha,\, \sim>,\ \gamma, \alpha \in \{X \cup T^{+}\} \tag{2.1}$$

γ and α are sequences of terminals and non-terminals in source and target sides respectively. The $\sim$ denote the alignment of non-terminals in the source and target sides, such that the co-indexed non-terminal pair is rewritten synchronously. These production rules are combined to derive the top symbol S by using the *glue* rules R_g. Hiero uses two types of glue rules:

$$S \rightarrow <X_1,\ X_1> \tag{2.2}$$

$$S \rightarrow <S_1 X_2,\ S_1 X_2> \tag{2.3}$$

Here again the non-terminal indices indicate synchronous rewriting of the source and target non-terminals having the same index. The second glue rule is additionally useful for translating longer spans (beyond the length of production rules) by concatenating smaller ones.

2.2 Heuristic Rule Extraction

Hiero uses a heuristic approach for extracting rules from phrase-pairs as explained below. Hiero training shares the initial steps as the training of phrase-based models beginning from word alignments until the generation of aligned phrase-pairs.

Given a parallel corpus, the training first obtains the source-target and target-source alignments by running an aligner, for example Giza++ (Och and Ney, 2000). The bidirectional alignments are then symmetrized using some heuristic alignment strategy (Och and Ney, 2003), such as union or intersection. Finally, it extracts the aligned phrase-pairs using the alignment template approach (Och and Ney, 2004), such that extracted phrase-pairs are consistent with the word alignments. In other words the phrase-pairs are constrained by the source-target alignments such that all the alignment links from the source (target) words are connected to the target (source) words *within* the phrase.

For example, consider a word-aligned sentence pair $\langle f_1^J, e_1^I, A\rangle$, where f_1^J and e_1^I indicate the source and target sentences of length J and I having word alignments A. Then a source-target sequence pair $\langle f_i^j, e_{i'}^{j'}\rangle$ can be a phrase-pair *iff* the following alignment constraints are satisfied.

$$(k, k') \in A \quad \text{where, } k \in [i, j] \text{ and } k' \in [i', j'] \tag{2.4}$$

$$(k, k') \notin A \quad \text{where, } k \in [i, j] \text{ and } k' \notin [i', j'] \tag{2.5}$$

$$(k, k') \notin A \quad \text{where, } k \notin [i, j] \text{ and } k' \in [i', j'] \tag{2.6}$$

Additionally it is also common for Hiero systems to restrict the extraction to tighter phrase-pairs, so that any phrase-pair having an unaligned boundary word(s) is ignored. This has the effect of controlling the number of extracted Hiero rules, which would otherwise be substantially higher.

The tighter phrase-pairs constraint can be written as:

$$\begin{aligned}
(k, k') \in A \quad & \text{where, } k = i \text{ and } k' = [i', j'] \\
(k, k') \in A \quad & \text{where, } k = j \text{ and } k' = [i', j'] \\
(k, k') \in A \quad & \text{where, } k = [i, j] \text{ and } k' = i' \\
(k, k') \in A \quad & \text{where, } k = [i, j] \text{ and } k' = j'
\end{aligned}$$

After extracting the initial phrase-pair the heuristic algorithm for extracting Hiero rules proceeds as below. Let $x = \langle f_i^j, e_{i'}^{j'} \rangle$ be an initial phrase-pair, then

$$X \rightarrow \langle f_i^j, e_{i'}^{j'} \rangle \tag{2.7}$$

is a rule. Now let $x' = \langle f', e' \rangle$ be a sub phrase of the above rule in (2.7), such that $f_i^j = f_p f' f_s$ and $e_{i'}^{j'} = e_p e' e_s$ (the alignments between f' and e' should satisfy the alignment constraints (2.4) through (2.6)). It then creates a new rule from this rule by introducing a non-terminal X in both source and target sides covering the spans of the sub-phrase x' yielding the rule:

$$X \rightarrow \langle f_p X_1 f_s, e_p X_1 e_s \rangle \tag{2.8}$$

Notice that the non-terminals on the right-side of the rule are co-indexed allowing them to rewritten synchronously.

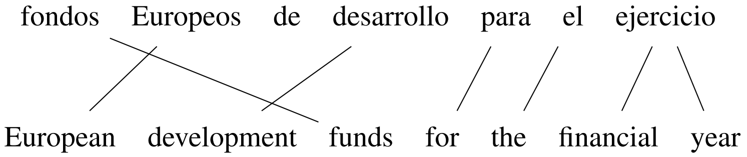


Figure 2.1: An example Spanish-English *phrase-pair* with word alignments

As an example, consider the following phrase-pair in Figure 2.1 shown with the word alignments. Given this initial phrase pair, the Hiero rule extraction would first create the rule:

$$\begin{aligned}
X \;\rightarrow\; & \langle \text{fondos Europeos de desarrollo para el ejercicio,} \\
& \quad \text{European development funds for the financial year} \rangle
\end{aligned} \tag{2.9}$$

It then considers the sub-phrase ⟨fondos Europeos de desarrollo, European development funds⟩. By substituting the non-terminal symbol X in the corresponding source and target side spans of the larger phrase it creates the new rule.

$$X \rightarrow \langle X_1 \text{ para el ejercicio, } X_1 \text{ for the financial year}\rangle \tag{2.10}$$

Figure 2.2 lists some of the SCFG rules extracted by the heuristic method for this phrase-pair in Figure 2.1.

$$\begin{aligned}
X &\rightarrow \langle \text{fondos Europeos de desarrollo } X_1\text{, European development funds } X_1\rangle\\
X &\rightarrow \langle \text{fondos } X_1 \text{ para } X_2\text{, } X_1 \text{ funds for } X_2\rangle\\
X &\rightarrow \langle X_1 \text{ para el ejercicio, } X_1 \text{ for the financial year}\rangle\\
X &\rightarrow \langle X_1 \text{ para el } X_2\text{, } X_1 \text{ for the } X_2\rangle\\
X &\rightarrow \langle \text{fondos Europeos de desarrollo, European development funds}\rangle\\
X &\rightarrow \langle \text{fondos } X_1 \text{ de desarrollo, } X_1 \text{ development funds}\rangle\\
X &\rightarrow \langle \text{fondos Europeos de desarrollo para el ejercicio,}\\
&\qquad \text{European development funds for the financial year}\rangle
\end{aligned}$$

Figure 2.2: Hiero rules extracted from the phrase-pair in Figure 2.1

Hiero imposes several constraints on the extracted rules in order to limit the grammar size and to reduce the decoding complexity. The extracted rules are filtered to remove those violating any of these constraints.

1. Initial phrase-pairs must not have any unaligned word in the source or target phrase boundaries (only *tight* phrase-pairs are allowed).

2. The biphrases can have up to 10 words on either sides and the extracted rules are limited to 5 tokens (terminals and non-terminals)[1] on source side.

3. Rules can have at most two non-terminals, i.e Maximum rule arity is restricted to two.

4. No adjacent non-terminals are allowed in the source side. This avoids the *spurious* ambiguities during decoding, which is characterized by distinct derivations having the same translation yield with identical values for the feature functions.

[1] Though the original Hiero extraction limits the rules to contain up to 5 tokens (terminals and non-terminals combined), it is useful to set this bigger (for example 7), which captures longer context and possibly better reordering. We fix the maximum number of terminals and non-terminals in Hiero rules to be 7 for all the experiments reported in this thesis.

5. The rule must be lexicalized with at least one aligned source-target word pair so that the translation rule is backed by lexical evidence.

2.2.1 Learning Rule Parameters

In order to decode with the extracted grammar, we need to learn the rule parameters such as conditional translation probabilities $p(e|f)$ and $p(f|e)$, which in turn requires rule counts be known. However each sentence pair in the corpus could be obtained through several derivations, which are never observed. As the maximum likelihood estimates of rule frequencies could not be computed, Chiang (2007) use heuristics to estimate a rule distribution.

It assumes a unit count for each phrase-pair and distributes this count equally to all the rules that are extracted from the phrase-pair. The rule counts $c(f, e)$ are then aggregated across all phrase-pairs in the training corpus. The conditional translation probabilities $p(e|f)$ and $p(f|e)$ are then computed by *relative frequency estimation* of the counts (weights).

The heuristic estimator was originally proposed in the context of *data oriented parsing* (DOP) by Bod (1998) for estimating the parameters of the probabilistic tree substitution grammars (PTSG) for parsing. This was later adapted for the phrase-based model (Och and Ney, 2004) and successfully used in several SMT models (Koehn et al., 2003; Quirk et al., 2005; Galley et al., 2006, *inter alia*) including Hiero. We will discuss some of the issues associated with this heuristic estimator in Section 3.1.

2.2.2 Hiero: Standard Features

Following the standard statistical model of SMT, Hiero uses a log-linear model (Och and Ney, 2002) for translation. Under this the probability of a Hiero derivation can be written in terms of different feature functions (ϕ) as:

$$P(d) \propto \prod_{i=1}^{k} {\phi_i}^{w_i} \tag{2.11}$$

where k is the total number of features and w denote the weights of the feature functions. Hiero uses the following standard feature functions: conditional translation probabilities $p(e|f)$ and $p(f|e)$, conditional lexical weights $p_{lex}(e|f)$ and $p_{lex}(f|e)$, phrase penalty, word penalty, glue rule weight and language model. Unless we note otherwise, most of the experiments reported in this thesis use these standard features.

2.3 Hiero Decoding

Hiero uses a CKY-style (Cocke, 1969; Kasami, 1965; Younger, 1967) algorithm for decoding. Given a source sentence f, the decoder finds the target side yield $\mathscr{Y}_e$ of the best scoring derivation obtained by applying rules in the synchronous context-free grammar.

$$\hat{e} = \mathscr{Y}_e \left(\arg\max_{d \in D(f)} P(d) \right) \tag{2.12}$$

where, $D(f)$ is the set of derivations attainable from the learned grammar for the source sentence f.

The decoder parses the source sentence with a modified version of CKY parser with the target side of corresponding derivations simultaneously yielding the candidate translations. The rule parameters and other features are used to score the derivations along with the language model score of the target translation as in Equation 2.13.

The derivation starts from the leaf cells of the CKY chart corresponding to the source side tokens and proceeds bottom-up. For each cell in the CKY chart, the decoder identifies the applicable rules and analogous to monolingual parsing, the non-terminals in these rules should have corresponding entries in the respective antecedent cells. The target side of the production rules yield the translation for the source span and the translations in the top-most cell correspond to the entire sentence.

The log-linear model over derivations $P(d)$ can be factorized to separate the language model (LM) feature from other features. The LM feature scores the target yield as $P_{lm}(e)$ usually with a n-gram model trained separately. The model can be written by factorizing derivation d into its component rules R_d as below.

$$P(d) \propto \left(\prod_{i=1}^{k-1} \prod_{r \in R_d} \phi_i(r)^{w_i} \right) P_{lm}(e)^{w_{lm}} \tag{2.13}$$

where, w_i is the corresponding weight of the feature ϕ_i. The feature weights w_i are optimized by minimizing a loss (Och, 2003) or by comparing pairwise rankings (Hopkins and May, 2011) with respect to some evaluation metric, usually BLEU (Papineni et al., 2002).

2.4 Kriya

We now explain *Kriya* - our Hiero implementation that includes both rule extraction module and a chart-parsing decoder. Kriya is similar to other implementations of Hiero-style systems, but has several distinguishing features. We introduce an unique approach in Kriya for computing the language model heuristic (see Section 2.4.1) in the *cube pruning* (Chiang, 2007) step that yields a small but consistent improvement (Sankaran et al., 2012b) in BLEU score. Kriya supports shallow-n decoding (de Gispert et al., 2010a) that speeds up the decoder by restricting the number of hierarchical nestings. This approach is especially helpful for close language pairs such as Arabic-English as the BLEU scores are not negatively impacted (Sankaran and Sarkar, 2012). As part of the training pipeline, Kriya further supports different phrase-table pruning techniques (Iglesias et al., 2009; He et al., 2009; Yang and Zheng, 2009) to prune the extracted grammars.

We use Kriya decoder for most of out experiments in this thesis. While our improvements are directly incorporated into Kriya, the ideas are generally applicable to any other Hiero-style system.

2.4.1 Language Model Integration in Kriya

The traditional phrase-based decoders such as Moses (Koehn et al., 2007) use beam search to generate the target hypotheses in the left-to-right order. In contrast, CKY decoders in Hiero-style systems can freely expand target hypotheses generated in intermediate cells or either sides in the higher cells. Thus the generation of the target hypotheses is fragmented and out of order in Hiero in contrast to the left to right order preferred by n-gram language models.

This leads to challenges in the estimation of language model scores for partial target hypotheses, which is addressed in different ways in the existing Hiero-style systems. Some systems add a sentence initial marker (<s>) to the beginning of each path and some others have the sentence boundary markers (<s> and </s>) implicitly in the derivation through the translation models. Thus the language model score for a partial hypothesis in an intermediate cell is approximated and the exact language model score (taking sentence boundaries into account) is computed only in the last cell after the entire target hypothesis is generated.

We introduce a novel improvement in computing the language model scores: for each of the target hypothesis fragment, our approach finds the best position for the fragment in the final sentence and uses the corresponding score. We compute three different scores corresponding to the three positions where the fragment can end up in the final sentence, viz. sentence initial, middle and final: and choose the best score. As an example for fragment t_f consisting of a sequence of target tokens, we compute LM scores for i) <s> t_f, ii) t_f and iii) t_f </s> and use the best score for pruning the search space[2].

[2] This ensures that the LM score estimates are never underestimated for pruning. We retain the actual LM score for fragment (case ii) for computing the exact LM score for the full candidate sentence in the last cell.

Language Pair	Moses	Kriya
English-Spanish	28.12	**28.19**
English-French	23.48	**23.54**
French-English	26.15	***26.63***
Arabic-English	37.31	***37.74***
Chinese-English	24.48	***25.96***

Table 2.1: Moses (Phrase-based) vs. Kriya (Hiero) - BLEU scores. Bold face indicates best BLEU score for each language pair and italicized figures point to statistically significant improvements assuming significance level $\alpha = 0.1$.

This improvement significantly reduces the search errors in the cube pruning step at the cost of additional language model queries. For example, a partial candidate covering a non-final source span might be better reordered to the end of the final candidate translation. If we compute the LM score for the partial target fragment in the naive way, the hypothesis might get pruned early on, before being reordered subsequently by a production rule. In contrast our approach would compute three LM scores (as above) and would correctly use the last LM score (case iii). Thus the hypothesis is less likely to be pruned due to its high score.

2.4.2 Kriya Performance

We now benchmark the performance of Kriya by comparing it with phrase-based Moses - a well-known open source SMT toolkit. Table 2.1 shows the BLEU scores for Moses and Kriya for a variety of typologically different languages. For each language pair both the systems were trained and tested on identical datasets (Sankaran et al., 2012b).

We now compare the results of the simpler unary Hiero having at most one non-terminal per rule with the conventional model that allows two non-terminals. The unary model is competitive to the full model for a wide variety of close language pairs such as French-English and Arabic-English as shown in Table 2.2. We do see a reduction in the BLEU score for Chinese-English as has also been found by (Zollmann et al., 2008). We thus hypothesize that unary models have the same expressive power as the regular Hiero models, at least for languages with little syntactic divergence. They also reduce the model size almost by half achieving a highest reduction of 51% for Arabic-English.

Kriya also achieved competitive scores to Joshua- a well-known Hiero system, in terms of both BLEU score (French-English) as well as human evaluation (French-English and English-Czech) in

Language Pair	**Regular Hiero**	**Unary Hiero**	
	BLEU	BLEU	Model size
English-Spanish	28.19	28.15	351.3 (55.5%)
English-French	23.54	23.48	290.3 (55.8%)
French-English	26.63	26.66	248.2 (56.4%)
Arabic-English	37.74	37.71	161.4 (49.0%)
Chinese-English	25.96	*25.25*	154.2 (53.9%)

Table 2.2: Regular Hiero vs. Unary Hiero model. Statistically significant BLEU differences in the unary Hiero are italicized. Model size of the unary grammar is in millions of rules and the numbers within the brackets denote the $\%$ model size compared to the binary Hiero grammar.

the recent machine translation shared task (Callison-Burch et al., 2012).

2.5 Ensemble Decoding

We now briefly review ensemble decoding (Razmara et al., 2012) which is used in some of our multi-metric optimization approaches we present in Chapter 6. Under the log-linear model for SMT the posterior probability of a candidate translation is expressed as:

$$p(e|f) \propto \exp\Big(w \cdot \phi\Big) \tag{2.14}$$

where ϕ represents the individual features

$$p(e|f) \propto \exp\Big(\sum_i w_i \phi_i(e, f)\Big) \tag{2.15}$$

The idea of ensemble decoding is to combine several models dynamically at decode time. Given multiple models, the scores are combined for each partial hypothesis across the different models during decoding using a user-defined mixture operation $\otimes$.

$$p(e|f) \propto \exp\Big(w_1 \cdot \phi_1 \otimes w_2 \cdot \phi_2 \otimes \ldots\Big) \tag{2.16}$$

Razmara et al. (2012) propose several mixture operations, such as *log-wsum* (simple linear mixture), *wsum* (log-linear mixture) and *max* (choose locally best model) among others. The different mixture operations allow the user to encode the beliefs about the relative strengths of the models. Thus ensemble decoding exploits the strengths of different models in translating different parts of

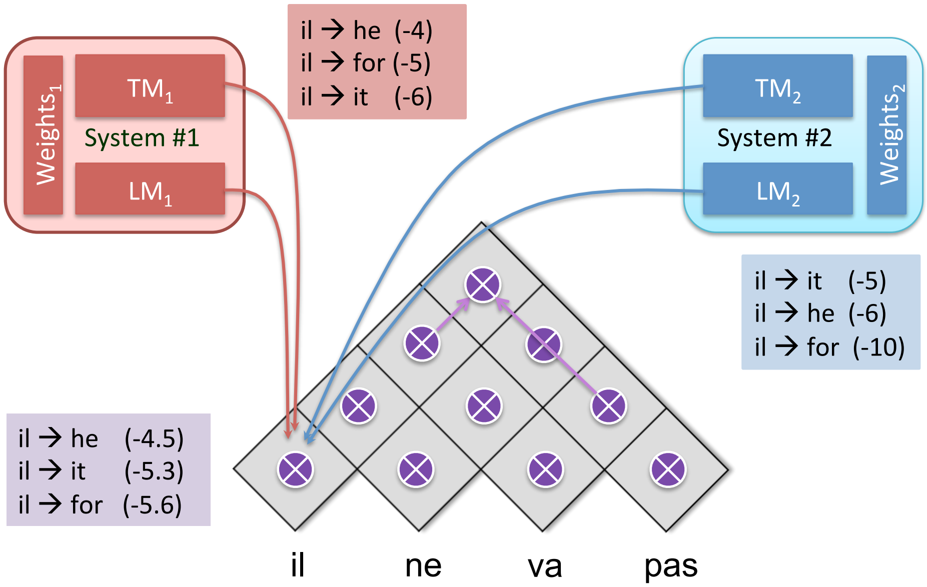


Figure 2.3: Ensemble decoding with two distinct models

the sentence. Unlike the mixture models (Foster and Kuhn, 2007; Foster et al., 2010) that combine the models statically, ensemble decoding combines them dynamically during decoding.

Figure 2.3 illustrates the ensemble decoding with two distinct sets of models coloured differently. The ensemble decoder collects translation options from both sets of models and combines them using the mixture operation shown in each chart cell as $\otimes$.

2.6 Pairwise-Ranking Optimization

The feature weights of the log-linear model of a translation system are typically optimized on a held-out set. The popular MERT (Och, 2003) approach searches for weights that minimize a loss function defined with respect to some evaluation metric. Given a metric like BLEU (Papineni et al., 2002) this corresponds to maximizing the BLEU score. The actual search for weights is carried through a modified Powell's line search (Och, 2003) algorithm. However one major problem with MERT is its inability to scale to a large set of features (Hopkins and May, 2011).

The pairwise ranking optimization (Hopkins and May, 2011) addresses this issue so that the optimization scales to a high-dimensional space of features, which has been gaining interest within SMT in recent years (Chiang et al., 2009; Cherry, 2013). The pairwise ranking optimization (PRO) method formulates tuning as a ranking problem and trains a binary classifier based on the ranking of pairs of hypotheses (high and low ranking candidates according to say BLEU), which yields a weight vector separating the good hypotheses from bad ones.

PRO mainly differs from MERT in the way it generates candidate hypotheses for optimization. While MERT considers all the hypotheses in the N-best list for each source sentence, PRO samples random pairs of hypotheses and computes the differences in their respective scores according to BLEU. After sampling a large number of such candidate pairs (say 2500), it chooses a fixed number of candidate pairs (say 50) having highest metric score differential for each sentence.

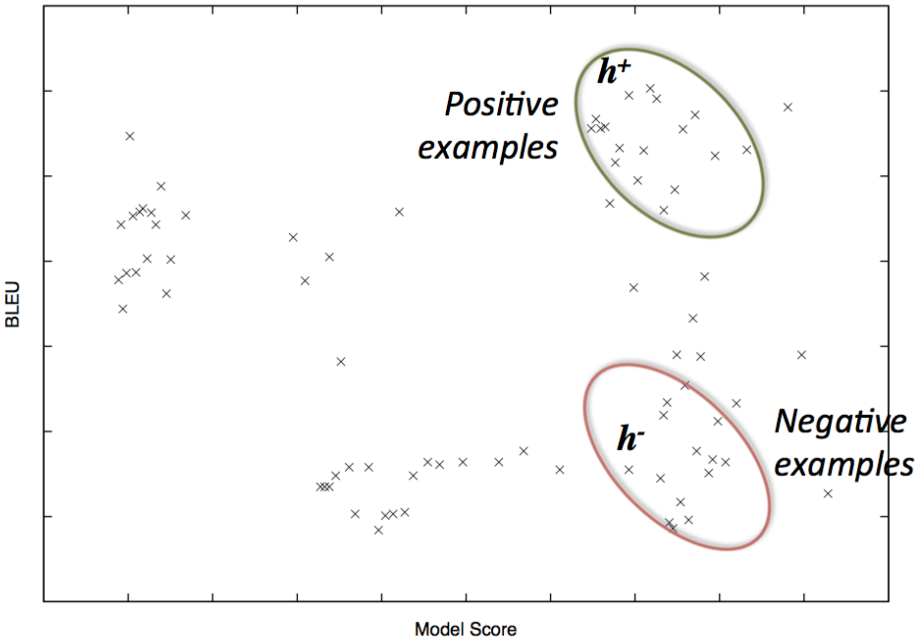


Figure 2.4: Simplified illustration of Pairwise ranking optimization (Hopkins and May, 2011). The translations in the N-best for a single source sentence are plotted along the model-score and metric-score dimensions (without loss of generality assume high values to be better for both axes). Positive examples correspond to high ranking candidates that are ranked high by the metric as well as the model. Negative examples are low ranking candidates that get poor model scores. PRO chooses pairs of candidates that have highest difference in their metric scores (for example h^+ and h^-).

Let h^+ and h^- be high and low ranking candidates respectively. Assuming $\boldsymbol{h_\phi}$ to be the vector of different feature values for a hypothesis h, PRO creates labelled pairs of training examples given by the label (1 or 0) and the difference of the feature vectors of the candidates in the pair. It adds both possible differences to ensure balance in the training data.

$$\begin{array}{ll} 1 & \{\boldsymbol{h}_{\phi}^{+} - \boldsymbol{h}_{\phi}^{-}\} \\ 0 & \{\boldsymbol{h}_{\phi}^{-} - \boldsymbol{h}_{\phi}^{+}\} \end{array}$$

Positive (high ranking) candidates are assigned a label 1, which low ranking candidates are given 0. It then trains a binary classifier to learn the optimal weights that would better separate positive examples from the negative ones. While PRO used the MegaM[3] classifier (Daumé, 2004), we use SVMRank[4] (Joachims, 2006) for the inner classification routine during optimization. The latter classifier has the advantage that it implicitly considers all possible pairs instead of only a few pairs. Secondly the SVMRank can factor in the actual ranking difference (i.e. difference in the metric scores), instead of binary (0/1) labels and this allows the classifier to consider the natural gradations among the different candidates.

2.7 Lateen Optimization

Lateen optimization was first introduced in the context of inducing dependency grammars employing multiple objectives (Spitkovsky et al., 2011). The word *lateen* refers to the triangular sails used in small marine vessels; these triangular sails help the vessel to effectively sail in any direction by using the *tacking* maneuver even in hostile wind conditions.

Lateen-EM uses this analogy in non-convex likelihood optimization to maneuver between the gradients of *soft* and *hard* EM. As a specific example, the optimizer can escape a local optima by switching to the hard EM objective from soft EM or vice versa. Several lateen strategies have been introduced based on when and how the secondary objective is used. For example the 'simple lateen EM' involves alternating between the optimization between two objectives and shallow lateen and the 'early stopping lateen EM' stops the optimization when the performance degrades for the secondary objective. The different lateen strategies have been shown to either speed up the EM training or to improve the accuracy for the unsupervised dependency parsing setting (Spitkovsky et al., 2011).

[3] http://www.umiacs.umd.edu/~hal/megam/

[4] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

Chapter 3

Better Estimation for Hiero Grammars

We explained the heuristic rule extraction (Chiang, 2007) method used by Hiero in Section 2.2. The heuristic extraction suffers from issues such as poor estimation of rule parameters, and excessively larger models. The latter problem arises as the heuristic method induces an exhaustive set of overlapping rules for each phrase-pair by allowing non-terminal substitutions at every potential (source-target) phrase alignments. We deal with this problem later in Chapter 5, where we focus on extracting compact grammars. We examine the former problem in this Chapter and propose a new model to improve the estimation of Hiero rule parameters.

3.1 Heuristic estimation: An Analysis

We first present a brief analysis of the heuristic estimator used by Hiero starting with a specific example to illustrate the weakness of the heuristic estimator.

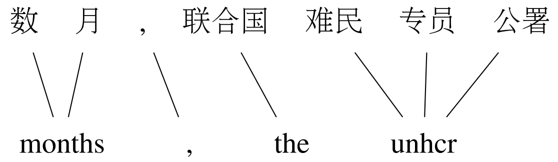


Figure 3.1: Chinese-English *phrase-pair* with alignments

Consider the word aligned Chinese-English phrase pair shown in Figure 3.1. The heuristic approach extracts 32 rules, some of which are shown in Figure 3.2 and then distributes counts among them.

$^*X \rightarrow \langle$联合国 ||| the$\rangle$

$^*X \rightarrow \langle$数 月 , 联合国 ||| months , the$\rangle$

$^*X \rightarrow \langle$联合国 X_1 ||| the $X_1\rangle$

$X \rightarrow \langle$数 月 , X_1 ||| months , $X_1\rangle$

$X \rightarrow \langle$联合国 难民 专员 公署 ||| the unhcr$\rangle$

$X \rightarrow \langle X_1$ 联合国 X_2 ||| X_1 the $X_2\rangle$

$X \rightarrow \langle X_1$, X_2 难民 专员 公署 ||| X_1 , X_2 unhcr$\rangle$

Figure 3.2: Rules extracted for the example phrase-pair in Figure 3.1

<table>
<tr><th><i>Translation rule</i></th><th>$P_{Heu}(e|f)$</th><th>$P_{VB}(e|f)$</th></tr>
<tr><td>$^*X \rightarrow \langle$在 中 南海 | at chong nan hai$\rangle$</td><td>0.333</td><td>0.003</td></tr>
<tr><td>$^*X \rightarrow \langle$在 中 南海 | at zhong nan hai$\rangle$</td><td>0.333</td><td>0.008</td></tr>
<tr><td>$X \rightarrow \langle$在 中 南海 | at zhongnanhai$\rangle$</td><td>0.333</td><td>$\mathbf{0.988}$</td></tr>
</table>

Figure 3.3: Different translations of the Chinese phrase 在 中 南海. The probabilities are shown for grammar extracted from heuristic as well our proposed method. The least preferred translations are marked with *. Our Variational-Bayes method extracts a grammar having a peaked distribution as shown.

As we mentioned in Section 2.2.1, the heuristic approach proposed by Chiang (2007) treats each phrase-pair to be independent item initializing it with unit count. It then assigns fractional pseudo-counts[1] uniformly (from the unit count) to all the rules extracted from the given phrase-pair. These counts are assigned locally to rules extracted from respective phrase-pairs, and then are aggregated across the phrase-pairs. The conditional translation probabilities are then estimated using relative frequency estimation.

A major problem with this heuristic rule extraction method is the lack of *global re-weighting* of the pseudo-counts beyond their local assignments. By assigning uniform weight to the rules (extracted from a given phrase-pair), the heuristic method assumes all such rules to be equally good,

[1] Even though the different derivations of the initial phrase-pairs are not seen, the heuristic approach assumes a hypothetical probability distribution over rules as though they were actually observed.

at least locally within a phrase pair. However, some rules might be better than others in terms of generalization, for capturing a syntactic phrase-pair, or being a semantically coherent unit of translation.

Due to this uniform treatment of good and poor translations, the probability mass is wasted on poor translation candidates. For example the phrase-pair in Fig 3.1 would generate several poor translation rules (marked as * in Fig. 3.2). This is due to the incorrect alignment link between 联合国 and *the* (note that the word *the* is typically aligned with a large number of words due to its frequency). The heuristic extraction method simply assigns uniform count to all translations and as a result the first translation in Fig. 3.2 becomes the fourth best translation for this source phrase.

In (Chiang, 2007) the rule extraction algorithm produces a fairly flat distribution over rules. For example the different translation options of the Chinese phrase 在 中 南海 (*at zhongnanhai*) all have the same $p(e|f)$ probability as shown in Figure 3.3. In contrast, our method produces a peaked distribution and shifts the probability mass towards *at zhongnanhai*, which is the preferred translation.

These problems are similar to the original DOP model (Bod, 1998) that first proposed the idea of hypothesizing a distribution over tree fragments, when the true distribution of these fragments in different derivations is not observed. The estimator was subsequently shown to be biased and inconsistent (Johnson, 2002). However it has been recognized that a biased estimator is not much of a concern (Johnson, 2002) and that it will have better generalization property (Prescher et al., 2004). Going further Shen (2011) provides a theoretical justification for the different instantiations of the DOP estimator in the context of machine translation using a simplified monotonic translation setting. Formalizing this as *exhaustive pattern learning* (EPL), he shows the probability distribution learned by these overlapping structures to be a constant-factor approximation of an ensemble model that combines models from different views of the dataset. Additionally this heuristic approach has been successfully applied in the machine translation context several times (Koehn et al., 2003; Quirk et al., 2005; Galley et al., 2006; Chiang, 2007) for inducing translation rules.

3.2 Derivations of a Phrase-pair

Our model uses the notion of a *derivation*: the set of rules that fully derive an aligned phrase pair, and learns the estimates for the rules contained in the derivations. Before proceeding to describe our model, we explain the notion of phrase-pair derivation, which is a key aspect of our models.

As we outlined in the last chapter in Section 2.2, the heuristic rule extraction approach considers the rules to be independent of each other and extracts all the rules that doesn't violate the constraints. However, this approach misses the fact that a set of rules could be combined coherently to generate a phrase-pair. We illustrate this with a Korean-English initial phrase pair shown along with its word alignments in Figure 3.4.

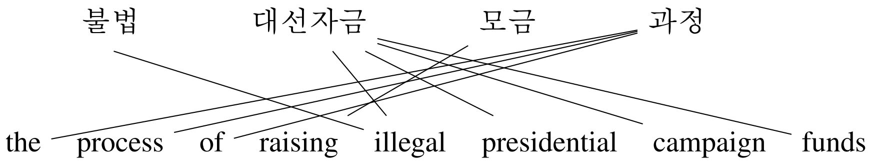


Figure 3.4: Korean-English phrase-pair shown with word alignments

Given this phrase-pair, the rule extraction step would consider the following translation rules (among others) for extraction. The first rule marked ** violates a Hiero constraint that no rule can have adjacent non-terminals on the source side and hence will be ignored by the heuristic method.

** $X \rightarrow \langle$불법 대선자금 $X_1 X_2$, X_2 raising $X_1 \rangle$

$X \rightarrow \langle$모금, raising$\rangle$

$X \rightarrow \langle$과정, the process of$\rangle$

$X \rightarrow \langle$불법 대선자금 X_1, X_1 illegal presidential campaign funds$\rangle$

$X \rightarrow \langle$모금 과정, the process of raising$\rangle$

$X \rightarrow \langle X_1$ 모금 X_2, X_2 raising $X_1 \rangle$

$X \rightarrow \langle$불법 대선자금, illegal presidential campaign funds$\rangle$

Figure 3.5: Example rules considered by the heuristic approach while extracting rules for the phrase-pair in Fig 3.4. The rule marked with ** would be ignored since it has adjacent non-terminals on the source side. However the second and third rules - rewriting X_1 and X_2 in the first rule, would be assigned counts. The heuristic method thus over counts the evidence for these two rules.

However the second (#2) and third (#3) rules which rewrites the two non-terminals X_1 and X_2 will be included by the heuristic method. The counts are then assigned uniformly to each rule proportional to the number of times the rule is independently extracted from a phrase pair. Under this strategy rule #3 will be get count estimates as part of the derivation containing the ignored rule (#1) as well as some other derivation (for example the derivation containing rules #3, #6 and #7).

The rule #2 will similarly get inflated count estimates.[2] Apart from the issue of inflated estimates, the over-counting also penalizes the other rules because the probability mass is actually taken away from these rules.

Finally in addition to the over-counting issue, the extracted rule might not even generate the phrase-pair from which it was extracted in the first place. This happens if the other partner rule which is required for completing the generation is ignored by the heuristic approach. In contrast our approach groups the rules extracted from a phrase pair according to the derivations they belong to. It then chooses rules from one or a few derivations based on their coverage over *all* phrase pairs.

Figure 3.6 shows some possible derivations for the phrase-pair example from Figure 3.4. We distinguish two types of derivations: (i) Terminal derivation (TERM) which directly rewrites a phrase pair as a terminal rule (the first derivation in Figure 3.6), (ii) Arity-1 hierarchical derivation (HIER-A1) consisting of a pair of rules (derivation #2) and (iii) Arity-2 hierarchical derivation (HIER-A2) consisting of three rules (derivation #3 in Figure 3.6).

#1 $X \rightarrow \langle$불법 대선자금 모금 과정, the process of raising illegal presidential campaign funds$\rangle$

#2 $X \rightarrow \langle$불법 대선자금 X_1, X_1 illegal presidential campaign funds$\rangle$
$X \rightarrow \langle$모금 과정, the process of raising$\rangle$

#3 $X \rightarrow \langle X_1$ 모금 X_2, X_2 raising $X_1\rangle$
$X \rightarrow \langle$불법 대선자금, illegal presidential campaign funds$\rangle$
$X \rightarrow \langle$과정, the process of$\rangle$

Figure 3.6: Three possible derivations (among many others) of the phrase-pair in Fig 3.4

In this chapter, we present a Bayesian rule extraction model for Hiero as an alternative to the heuristic approach. Our model, called binary Hiero model explains the generation of a given initial phrase-pair through a small number of derivations along with their constituent rules. Unlike the heuristic approach, our model seeks to distribute the probability mass among the rules (generated from a phrase-pair) based on their contribution in explaining the collection of all phrase-pairs in

[2] Alternately, the heuristic method could be forced to assign uniform counts only to the *unique* rules as is done in the heuristic method originally proposed (Chiang, 2007). In that case, some frequent translations that are seen in several derivations would be penalized with poor probability estimates due to under-counting.

a global manner. We resort to approximate inference employing Variational Bayesian method in order to estimate the probabilities of the Hiero grammar. This difference in estimation methods can lead to a peaked distribution of rule probabilities and we demonstrate using three different language pairs that our Bayesian rule extraction model at best improves or at least retains the translation performance as the heuristic method. Secondly we also extend it using a distributed framework that enables rule extraction on large datasets. Finally, we also present a detailed qualitative analysis about the probability distributions of the grammar estimated from our Bayesian model vis-a-vis heuristic grammar.

3.3 Binary Hiero Model

As mentioned earlier, our model replaces the heuristic rule extraction step in Hiero pipeline. Consequently our model assumes the existence of *initial* phrase-pairs obtained from bidirectional symmetrization of word alignments. We use the following two-step generative story to create an aligned phrase pair from the Hiero rules.

1. First decide the derivation type z_d for generating the aligned phrase pair x. It can either be a terminal derivation or hierarchical derivation with one/two gaps,[3] i.e. $z_d = \{\textsc{Term}, \textsc{Hier-A}1, \textsc{Hier-A}2\}$.
2. Then identify the constituent rules $\mathbf{r}$ in the derivation to generate the phrase pair.

$\phi^z \sim \text{Dirichlet}(\boldsymbol{\alpha_z})$	[draw derivation type parameters]
$\theta \sim \text{Dirichlet}(\alpha_h p_0)$	[draw rule parameters]
$z_d \sim \text{Multinomial}(\phi^z)$	[decide the derivation type]
$r \vert \mathbf{r} \in d_x \sim \text{Multinomial}(\theta)$	[generate rules deriving phrase-pair x]

Figure 3.7: Definition of the proposed binary Hiero model

Under this model the probability of a particular derivation $d \in \phi_x$ for a given phrase pair x can be expressed as:

$$p(d) \propto p(z_d) \prod_{r \in d} p(r|\mathcal{G}, \theta) \tag{3.1}$$

[3] This refers to the maximum arity of a rule involved in the derivation.

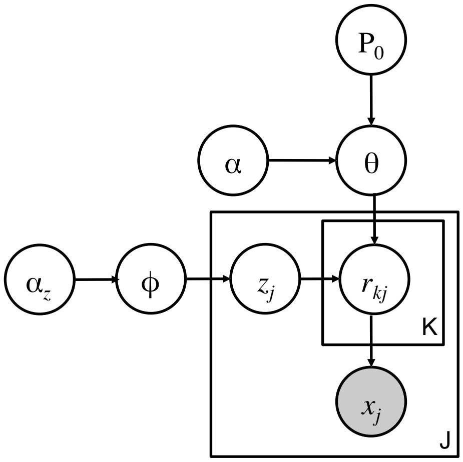


Figure 3.8: Graphical model representation for Binary Hiero model. The generative process first decides the derivation type z_j from a Multinomial parametrized by ϕ. It then generates the rules r_{kj} in the derivation by using a Dirichlet distribution θ with base measure P_0 and concentration parameter α. There are K rules in the derivation which yields the phrase-pair x_j.

where r is a rule in grammar $\mathcal{G}$ and θ is the grammar parameter.

Figure 3.7 depicts the generative story of our generative model while the equivalent graphical representation is shown in Figure 3.8. The derivation-type z_d is sampled from a multinomial distribution parameterized by ϕ^z, where ϕ^z is distributed itself by a Dirichlet distribution with hyperparameter $\boldsymbol{\alpha_z}$. The grammar rules are generated from a multinomial distribution parameterized by θ, where θ itself is distributed according to a Dirichlet distribution parameterized by a concentration parameter α_h and a base distribution p_0. For the base distribution, we use a simple but yet informative prior based on geometric mean of the bidirectional alignment scores. This allows us to only explore the rules that would be consistent with the underlying word alignments.[4] Thus our setting closely resembles that of the Hiero heuristic rule extraction.

Our goal is thus to infer the joint posterior $p(\theta, \Phi | \alpha_h, p_0, \boldsymbol{\alpha_z}, \mathcal{X})$, where θ are the model parameters and Φ the latent derivations over all the phrase pairs.

[4] While a non-parametric prior would be better from a Bayesian perspective, we leave it for future consideration.

3.4 Variational Bayesian Inference

Variational inference (Ghahramani and Beal, 2000; Attias, 2000) is an approximation technique typically used in Bayesian settings. It is used for approximating an intractable posterior distribution $p(\Phi; \theta)$ by finding a tractable variational distribution $q(\Phi; \theta)$ over the latent variables Φ and parameters θ.

Unlike the maximum likelihood (ML) or maximum a posteriori (MAP) which learn a point estimate, VB learns a *distribution* over parameters by minimizing a measure of divergence between q and p, such as KL($q \parallel p$). Assuming a factorization $p(\Phi, \theta) \approx q(\Phi)q(\theta)$ enables $q(\Phi, \theta)$ to be estimated by alternately updating $q(\Phi)$ and $q(\theta)$ in an iterative setting similar to EM.

Variational Bayes has been used earlier for inducing probabilistic context-free grammars (PCFG) (Kurihara and Sato, 2006). Unlike theirs, we use VB for learning a Hiero-style (translation) grammar. Additionally, we do not use the free energy criterion for *model selection* as done in their work. Instead we use an informative prior for $q(\theta)$, which together with an appropriate concentration parameter α_h, pushes the grammar towards sparsity.

3.4.1 Binary model: Training

For inference we resort to a variational approximation and factorize the posterior distributions over grammar parameters θ and latent derivations Φ as:

$$p(\theta, \Phi | \alpha_h, p_0, \boldsymbol{\alpha_z}, \mathcal{X}) \approx q(\theta|\mathbf{u})q(\Phi|\pi)$$

where $\mathbf{u}$ and π are the parameters of the variational distributions.

The inference is then performed in an EM-style algorithm (shown in Algorithm 1) - iteratively updating the parameters $\mathbf{u}$ and π. We initialize $\mathbf{u}^0 := \alpha_h p_0$, which is then updated with expected rule counts in subsequent iterations. The expected count for a rule r at time-step t can be written as:

$$\mathbb{E}[r^t] = \sum_{d \in \phi_x} p(d|\pi^{t-1}, x) f_d(r) \tag{3.2}$$

where $p(d|\pi^{t-1}, x)$ is the probability of the derivation d for the phrase pair x and $f_d(r)$ is the frequency of the rule r in derivation d. The $p(d|.)$ term in Equation 3.2 can then be written in terms of π as:

$$p(d|\pi^{t-1}, x) \propto p(z_d) \prod_{r \in d} \pi_r^{t-1} \tag{3.3}$$

Algorithm 1 Variational-Bayes Inference for binary Hiero grammar

```
Input: Set of aligned phrase-pairs $\mathcal{X}$
Get prior distribution $\mathbf{u} = \{u_r = \alpha_h p_0(r) | r \in \mathcal{G}\}$
Set $\mathbf{u}^0 = \mathbf{u}$
for time-step $t = 1, 2, \ldots$ do
  for $z_d \in Z$ do
    $p(z_d) \leftarrow \exp\left(\psi(\alpha^{t-1}_{z_d}) - \psi(\sum_{z_d} \alpha^{t-1}_{z_d})\right)$
  for $r \in \mathcal{G}$ do
    $\pi^{t-1}_r \leftarrow \exp\left(\psi(u^{t-1}_r) - \psi(\sum_r u^{t-1}_r)\right)$
  for $x \in \mathcal{X}$ do
    for $d \in \phi_x$ do
      Compute $p(d|\pi^{t-1}, x)$ as in (3.3)
      $\mathbb{E}[z^t_d] \leftarrow \mathbb{E}[z^t_d] + p(d|\pi^{t-1}, x)$
      for $r \in d$ do
        $\mathbb{E}[r^t] \leftarrow \mathbb{E}[r^t] + p(d|\pi^{t-1}, x)\, f_d(r)$
  for $z_d \in Z$ do
    Estimate $\alpha^t_{z_d} \leftarrow \alpha^0_{z_d} + \mathbb{E}[z^t_d]$
  for $r \in \mathcal{G}$ do
    Estimate posterior $\mathbf{u}^t$ :   $u^t_r \leftarrow u^0_r + \mathbb{E}[r^t]$
Output: Posterior distribution $\mathbf{u}^t$
```

The $p(d|.)$ are normalized across all the derivations of a given phrase pair to yield probabilities. For each *derivation type* z_d, its expected count (at time t) is the sum of the probabilities of all the derivations of its type.

$$\mathbb{E}[z^t_d] = \sum_x \sum_{\{z_d = z_{d'} | d' \in \phi_x\}} p(d'|\pi^{t-1}, x) \tag{3.4}$$

We initialize the Dirichlet hyperparameters α_{z_d} using a Gamma prior ranging between 10^{-1} and 10^3: $\alpha_{z_d} \sim \text{Gamma}(10^{-1}, 10^3)$.

We run inference for a fixed number of iterations[5] and use the grammar along with their posterior counts from the last iteration for the translation table.

3.4.2 Enumerating the Phrase-pair Derivations

The model explores the set of possible derivations that are consistent with the word alignments of each phrase-pair. In order to keep the sampling fast, we need an efficient way to enumerate the

[5] In our experiments, we set the number of iterations to 10.

derivations for a phrase pair, which additionally must satisfy the alignment constraints. We use the decomposition tree algorithm (Zhang et al., 2008a) that maximally decomposes a word-aligned phrase pair, so as to encode it as a compact alignment tree.

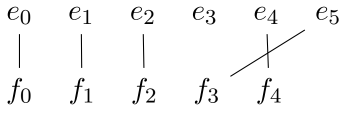

e_0 e_1 e_2 e_3 e_4 e_5
f_0 f_1 f_2 f_3 f_4

Figure 3.9: Example phrase pair with alignments.

For a phrase-pair with a given alignment as shown in Figure 3.9, Zhang et al. (2008a) generalize the $\mathcal{O}(n + K)$ time algorithm for computing all K common intervals of two different permutations of length n. The contiguous blocks of the alignment are captured as the nodes in the alignment tree and the tree structure for the example phrase pair in Figure 3.9 is shown in Figure 3.10. The italicized nodes form a left-branching chain in the alignment tree and the sub-spans of this chain also lead to alignment nodes that are not explicitly captured in the tree (Please refer to Zhang et al. (2008a) for details). In our work, each node in the tree (and also each sub-span in the left-branching chain) corresponds to an *aligned source-target sub-span* within the phrase-pair, and is a potential site for introducing the non-terminal X to generate hierarchical rules.

Given this alignment tree for a phrase pair, a derivation can be obtained by introducing a non-terminal at some node n_d in the tree and re-writing the span rooted at n_d as a separate rule. As mentioned earlier, we compute the derivation probability as a product of the probabilities of the component rules, which are computed using the Equation 3.3.

3.4.3 Distributing Inference

While the above training procedure works well for smaller datasets, it does not scale well for the realistic MT datasets (which have millions of sentence pairs) due to greater memory and time requirements. To address this shortcoming, we distribute the training using a Map-Reduce style framework, where each node works on the local dataset in computing the required statistics and then communicates the statistics to a central aggregate/ reduce node.

Distributed inference for Expectation Maximization algorithm was studied in (Wolfe et al., 2008). They used three different topologies in terms of computation time, bandwidth requirement and so on. While Map-Reduce is substantially slower than the All-pairs and Junction-tree topologies, it takes much lesser bandwidth than the other two apart from being much easier to implement. Furthermore our choice of the Variational inference naturally lends itself to distributed training.

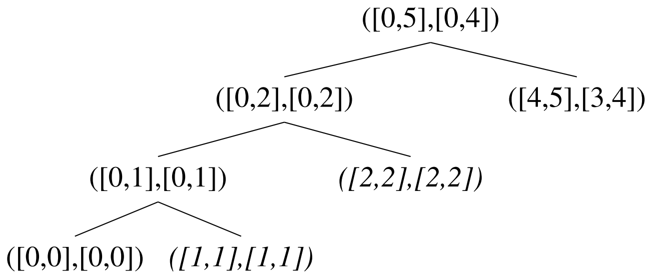


Figure 3.10: Decomposed alignment tree for the phrase-pair in Fig. 3.9. The italicized nodes result in a left-branching chain in the alignment tree, which gives rise to additional spans not captured by the alignment tree representation.

We simply shard the set of aligned phrase pairs and parallelize the training steps for the shards across different nodes. After each node completes the processing of the assigned shards and collects the statistics (expected rule counts for example), we need to aggregate the statistics to get a global view, which will then be used in the next iteration/training step. We parallelize this aggregation across several nodes in one or two reduce steps as required. At the end of aggregation we communicate the updated statistics to each node on a need basis.[6]

3.5 Binary Model: Experiments

We evaluate our model on three different language pairs in two distinct settings. We first have a small data setting and consider the Korean-English language pair for our experiments. This is to assess if our Bayesian model can result in better parameter estimation than the heuristic approach with limited amount of training data. The second one is a large data setting, where we experiment with two different language pairs having more than 1M sentence pairs. We choose Chinese-English and Arabic-English language pairs in order to evaluate the strength of our rule extraction model under varying degree of reordering complexity.

We use three datasets of varying sizes for our experiments. We use the University of Rochester Korean-English dataset consisting of almost 60K sentence pairs for the small data setting. For moderate and large datasets we use Arabic-English (ISI parallel corpus) and Chinese-English (Hong Kong parallel text and GALE phase-1) corpora. We use the MTC dataset having 4 references for tuning and testing for our Chinese-English experiments. The statistics of the corpora used in our

[6] We simulate the Map-Reduce style of computation using a regular high-performance cluster using a mounted filesystem rather than a Hadoop cluster with a distributed filesystem.

experiments are summarized in Table 3.1.

Lang.	Training Corpus	Train/ Tune/ Test
Korean-English	University of Rochester corpus	59218/ 1118/ 1118
Arabic-English	ISI web-crawled parallel corpus	1.1 M/ 1982/ 987
Chinese-English	HK parallel text + GALE phase-1	2.3 M/ 1928/ 919

Table 3.1: Hiero binary grammar extraction: Corpus Statistics in # of sentences

We follow the standard MT practice and use GIZA++ (Och and Ney, 2003) for word aligning the parallel corpus. We then use the heuristic step that symmetrizes the bidirectional alignments (Och et al., 1999) to extract the initial phrase-pairs up to a certain length, consistent with the word alignments. Finally we employ our proposed Variational-Bayes training to learn rules for Hiero. As a baseline Hiero model, we use the heuristic rule extraction (Chiang, 2007) approach to extract the rules. In both cases the parameters are estimated by the relative frequency estimation.

For decoding we use the decoder from Kriya (Sankaran et al., 2012b) - our in-house implementation of hierarchical phrase-based model. We use the following 8 standard features for the log-linear model: translation probabilities ($p(e|f)$ and $p(f|e)$), lexical probabilities ($p_l(e|f)$ and $p_l(f|e)$), phrase and work penalties, language model and glue rule penalty.

3.5.1 Binary Model: Results

The main BLEU score results are summarized in Table 3.2 and the key aspects are summarized below.

- **Higher BLEU scores**: Our Bayesian model performs better than the baseline heuristic rule extractor for Korean-English. Furthermore, the improvement of 0.5 BLEU is statistically significant at p-value of 0.01.

- **Large corpora**: Our distributed inference model easily scales to the large corpora and the inference completes in less than a day for Chinese-English. It also retains BLEU scores in the same level as the baseline models for both Arabic-English and Chinese-English.

3.5.2 Analysis

We now compare the probability distributions of the two grammars at the level of individual rules to understand the differences between them. We considered a set of source phrases that are common

Grammar	Ko-En	Ar-En	Zh-En
Heuristic baseline	7.18	**37.82**	**28.58**
Variational-Bayes	***7.68***	37.76	28.40

Table 3.2: BLEU scores for baseline heuristic extraction and the proposed Variational-Bayes model. Best scores are in **boldface** and statistically significant differences are *italicized*.

in both grammars and analyzed their probability distributions over the translation options.

Specifically we use the Q-Q plot to study the behaviour of two probability distributions as explained below considering the Chinese phrase 联合国 (*united nations*) as a representative example. The Q-Q plot in Figure 3.11 plots the $p(e|f)$ probabilities (sorted for the baseline grammar) for different translations of the source phrase. The translations from the baseline grammar are then paired off with the points in the sorted VB curve and the corresponding probabilities are plotted in the same order as the baseline translations. The following conclusions can be drawn from this plot:

- **Penalize poor translations**: Among the low-probability translations, majority of the translations in the VB-grammar have probability less than the corresponding baseline translations. This has the desired property of potentially shifting the probability mass away from poor translations.

- **Reward good ones**: VB-grammar rewards some translations that were deemed to be poor by the heuristic method, by assigning a slightly higher probability than the heuristic grammar. A manual inspection showed that the rules with higher probabilities were objectively better translation rules. For example Table 3.3 contrasts the probabilities assigned by the two methods for the first four translation options in Fig. 3.11.

- **Uniform probability is *not* informative**: The heuristic extraction method tends to assign a uniform probability for groups of translations and this is evident in the flat segments of the baseline curve in Figure 3.11. While this behaviour is dominant in the low-probability region, we also notice this in high probability region. The VB-grammar estimates do not suffer this degeneracy and it assigns varying probabilities and helps the decoder to choose better translation options among several different options.

We also observe a similar trend for several source phrases in both Arabic-English and Chinese-English corpora.

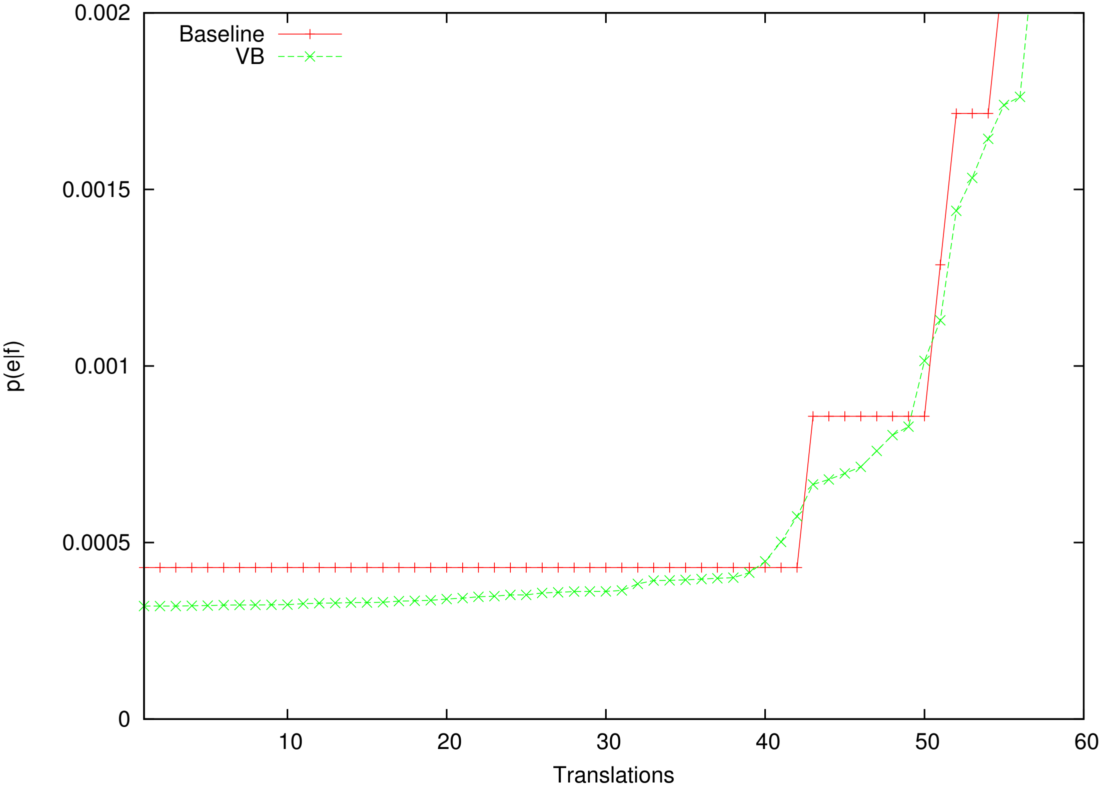


Figure 3.11: Q-Q plot comparing the $p(e|f)$ distributions of the baseline and VB-grammars for Chinese phrase 联合国 (*united nations*). The points on the two curves represent distinct target translations (the numbers on the x-axis indicate the indices) and the points are sorted according to VB translation probabilities against the paired-off translation probabilities from the baseline grammar. The y-axis is clipped to highlight the variations in the low-probability range.

At the macro level, we compare the sizes of the different types of rules in the heuristic and the Variational-Bayes grammar. The baseline grammar extracts a small number of additional arity-1 rules than the grammar extracted by our model (see Figure 5.9). Our model extracts rules used in a *derivation* of a phrase-pair, only if *all* its constituent rules are consistent with the Hiero rule constraints (such as restriction on the total number of terminals and non-terminals in the rule). However the heuristic method extracts all the consistent rules and does *not* consider the derivations. While this is a more stricter constraint, the VB model extracts slightly more (about 170K) arity-2 rules as we allow the unaligned words to be attached to different levels of hierarchical rules during the construction of the decomposition tree. This extracts translation rules that are beyond the purview of the heuristic method, since the Viterbi alignments cannot capture them.

Translations	**Heu** $p(e\|f)$	**VB** $p(e\|f)$
alert the united nations	4.28e-04	3.46e-04
during	4.28e-04	3.20e-04
for un	4.28e-04	*5.02e-04*
human	4.28e-04	3.20e-04

Table 3.3: Probabilities assigned by the two methods for the first four translations (#1 through #4 on x-axis) in Fig. 3.11. The better translation among four and the higher probability assigned by our model are *italicized*.

As the final part of the analysis, we also present the 100 high probability lexical phrases extracted by both rule extraction methods for the Arabic-English corpus in Figure 3.12. As seen, the heuristic grammar assigns high probability to the rules translating proper nouns and short phrases, whereas the VB method assigns high probability to more generic translations.

3.6 Related Work

The notion of modelling many equivalent translations by considering different derivations has been used earlier in the context of discriminative machine translation, particularly for the translation model (Blunsom et al., 2008b). Unlike the heuristic method, this approach trains the model by maximizing the conditional likelihood $p(e|f)$. This was later extended to support language model integration by approximating the partition function using Monte-Carlo sampling (Blunsom and Osborne, 2008). Here, we use the different derivations to perform estimation in a generative setting.

Several approaches have been proposed earlier for improving the estimation of Hiero grammars and as an alternative to the heuristic method. Some of the early works have employed Bayesian techniques for inducing Hiero-style rules directly from the bitext. DeNero et al. (2008) use a Maximum likelihood model of learning phrase pairs (Marcu and Wong, 2002), but use sampling to compute the expected counts of the phrase pairs for the E-step. Blunsom et al. (2008a) proposed a generative model for deriving a sentence pair through a series of terminal and ITG-style non-terminal rules and used Variational Bayes for learning the SCFG rules.

A non-parametric Bayesian model using a Gibbs sampler to reason over the space of derivations (Blunsom et al., 2009) use priors to bias the grammar to be small, however they do not compare the resulting grammar size. Additionally, the model suffered from weaker reordering ability. However these approaches use small datasets that range between 33K-300K sentence pairs and hence their efficacy on realistic SMT datasets is unclear. In contrast, our experiments scale well for large datasets as we demonstrate for two language pairs.

Recently Levenberg et al. (2012) proposed a Bayesian model employing a Pitman-Yor Process (PYP) prior for learning SCFG rules. While this approach is able scale well for large corpora (by distributing the sampler across several threads), this allows the SCFG rules have unrestricted number of non-terminals. Broadly speaking, all these generative approaches learn a posterior over parallel tree structures over the entire sentence pairs.

However, none of these models actually extract the Hiero grammar, which can directly be used by the decoder for translation. Instead the SCFG rules extracted from these models are used for the limited purpose of obtaining the phrase alignments. In other words these models essentially use the SCFG rules for learning the hierarchical alignments between source and target sentences. Because of this limitation, these models involve an additional step for extracting the actual translation rules by using the *heuristic* extraction method (Chiang, 2007) from the hierarchical alignments. Thus strictly speaking, these models are only Bayesian alignment models whereas our model is a Bayesian rule extraction model.

Separately a Bayesian ITG model was proposed for learning the phrase translation correspondences directly from the data without the word alignments (Cohn and Haffari, 2013). This approach is however limited to learning flat phrases (and not hiero rules); which it does by sampling the ITG derivations of the entire sentence pairs.

We differ from earlier Bayesian approaches in that our model is guided by the word alignments to reason over the space of the SCFG rules and this restricts the search space of our model. We believe the word alignments to encode information, useful for identifying the good phrase-pairs. Our model exploits the word alignment information in the form of lexical alignment probability in order to construct an informative prior over SCFG rules and it moves away from a heuristic framework, instead using a Bayesian model to infer a high-quality grammar from the data.

Separately, Variational Bayes has been successfully used for learning minimal non-compositional phrase-pairs using bi-parsing (Zhang et al., 2008b). This approach proposes a combined model for learning word and phrase alignments by using a phrasal ITG framework, whose parameters are estimated by Variational Bayes biased with a sparse prior to control the over fitting issue (Marcu and Wong, 2002). In contrast our VB model induces a sparse Hiero grammar as opposed to their goal of learning phrase alignments for a phrase-based model.

Variational approximation has also be employed in the context of MT decoding for finding the best scoring translation as opposed to just finding the best scoring derivation (Li et al., 2009). They show that decoding for best scoring translation can lead to significant improvements in the BLEU scores over the traditional Viterbi decoding by means of merging the spuriously ambiguous derivations yielding the same string.

3.7 Binary Model: Summary

In this chapter, we presented a Bayesian model for training the binary Hiero grammar as an alternative to the heuristic rule extraction approach. For inference, we use Variational-EM and extended it in a Map-Reduce style framework for distributing the training process across multiple processors. This allowed us to efficiently train the model for large corpora without increasing time complexity. Our model leads to better probability estimates for the rules in the extracted grammar, arising from a sparse distribution of the rules induced by our model that extracts derivations of phrase-pairs instead of individual rules. We demonstrate that the better probability estimates can actually lead to improved performance in small-data setting (Korean-English), while matching the translation performance of the heuristic estimator in large-data settings of Arabic-English and Chinese-English. We finally provided quantitative results on three language pairs apart from presenting a detailed qualitative analysis to explain the superiority of the grammar inferred by our approach.

Phrases from Heuristic Grammar

major news/فيما يلى العناوين الرئيسية فى b share index :/: " ب " مؤشر الاسهم hugo/هوجو million shares/مليون سهم
b share index/" ب " مؤشر الاسهم bayer/باير igor/ايجور yuan (/) يوان , indonesia , laos ,/واندونيسيا ولاوس items in
مانموهان masood khan/مسعود خان , according to the bank of/وفقا لما ذكره بنك ، jack straw/جاك استرو tony/تونى
daniel/دانييل nanjing/نانجينغ celta/سلتا national security/الأمن القومى khan jamali/خان جمالى manmohan singh/سينغ
china (hong kong/الصين (هونج كونج jose luis/خوسيه لويس china (hong kong)/(الصين (هونج كونج kong)/(كونج
a tael/للتايل vladimir/فلاديمير pradesh/براديش (hong kong)/(هونج كونج) mwai kibaki/مواى كيباكى athletic/اتلتيك
john howard/جون هوارد social stability/الاستقرار الاجتماعى shares .// نهاية الخبر / . سهم carlo/كارلو daniel/دانيل
(/الناتو / president hu/الرئيس هو جين (hong kong/هونج كونج) cambodia , indonesia , laos ,/وكمبوديا واندونيسيا ولاوس
eduardo duhalde/ادواردو دوهالدى president chandrika/الرئيسة تشاندريكا changchun/تشانغ تشون toledo/توليدو nato
jose/خوسيه yashwant/ياشوانت glafcos/جلافكوس visiting/الزائرة everton/ايفرتون ramirez/راميريز herve de/هيرفيه دو
tanzanian/الرئيس التنزانى yen/ينات dos santos/دوس سانتوس yoriko kawaguchi/يوريكو كاواجوتشى junichiro/جونتشيرو
kong/كونج state warren/الخارجية الاميركي وارن yen/ينا lake victoria/بحيرة فيكتوريا nabil abu/نبيل ابو manh/مانه president
اللانمطى tokyo stock price/اسعار بورصة طوكيو turnover :/: قيمة التداول kufuor/كوفور sese/سيسي march 31/مارس 31
الخارجية jose/جوزيه : 50/50 : his palestinian counterpart/نظيره الفلسطينى yellow/الاصفر syndrome (sars/سارس /
gold dealers in hong/كبار المتعاملين فى الذهب فى هونج abdullah gul/عبدالله جول lithuania/وليتوانيا iraqi foreign/العراقى
one of the major gold deal-/احد كبار المتعاملين فى الذهب فى هونج yen/ين international humanitarian/الانسانى الدولى
تشاو yen (// ين west bank/بالضفة الغربية atletico/اتلتيكو racing/راسينغ sao tome/ساو تومى fidel/فيديل ers in hong
2 1 1/- 1 1 2 visiting chinese/الصينى الزائر dollars)/(امريكى fernando/فرناندو werder/فيردر silva/سيلفا zhaoxing/شينغ
president museveni/الرئيس موسيفينى baghdad –/-- بغداد in 1996 ./. فى عام 1996 nicholas/الخارجية الاميركية نيكولاس 0
yen to/ين الى yang liwei/يانغ لى وى zoran/زوران hong/هونج the major/أكبر المتعاملين president vladimir/الرئيس فلاديمير

Phrases from Variational-Bayes Grammar

from/77 من scott mcclellan/البيت الابيض سكوت مكليلان 100 tons/طن 100 sheikh ali/الشيخ علي june 21/يونيو 21
donald/الأمريكى دونالد electronic information/المعلومات الالكترونية jiangsu// جيانغسو dominique de/دومنيك دو 77
south africa after/جنوب افريقيا بعد economic research/الابحاث الاقتصادية french tourists/سائحا فرنسيا africa 1/1 افريقيا
hong kong) , one/أحد) هونج كونج legitimate concerns/المخاوف المشروعة border with egypt/الحدودي مع مصر
moshe/الاسرائيلي موشيه كاتساف in jerusalem/في مدينة القدس 150 million people/مليون شخص 150 1961/1961 العام
jiang zemin as chairman/جيانغ تسه مين كرئيس generation to/جيل الى international pressure/ضغط دولي katsav
بين amid reports/وسط تقارير in gabon/في الغابون 86 billion/مليار 86 alaves 33/33 الافيس police patrol/دورية الشرطة
north of basra/شمال البصرة bulgaria 2/2 بلغاريا 7 15 8/8 15 7 nicholas/الخارجية الاميركية نيكولاس between berlin/برلين
argentina and iran/الارجنتين وايران iran supports/ايران تساند dialogue with israel/الحوار مع اسرائيل russia 5/5 روسيا
members/عناصر من هذه said/ان صرح peter struck/بيتر ستروك so far this/حتى الان من هذا million pounds/ملايين جنيه
coalition spokesman/اعلن متحدث باسم قوات التحالف immediate release of/الافراج الفورى zeev/الاسرائيلي زئيف of the
many european/أوربية كثيرة palestinian doctors/اطباء فلسطينيون accept without/توافق بدون with rwanda/مع رواندا said
1.4/مليار 4 ، 1 witnesses/اكد شهود saturday ,/السبت مما sudan , swaziland ,/والسودان وسوازيلاند ha 'aretz/هاآرتس / /
على بعد 200 billion/مئتي مليار from muzaffarabad/من مظفر اباد or without/او بدونها billion yuan (/) مليارات يوان billion
tajikistan/طاجيكستان وقازاقستان abbas to visit/عباس لزيارة examination and approval/الفحص والموافقة kilometres/حوالي
agreed/موافقته المبدئية reconciliation and democracy/المصالحة والديمقراطية on all fronts/على كل الجبهات , kazakhstan
warning radar/رادار للانذار liberians to/ليبيرى على fares/اللبناني فارس sunshine policy/سياسة الشمس المشرقة in principle
josep pique , whose/جوزيب بيكيه الذي but peres/لكن بيريز president didier/الرئيس ديدييه on the level/حول مستوى
to 96/96 الى geoff hoon/جوف هون crowded/مزدحمة بالركاب a total of 63/63 اجمالى partial elections/انتخابات جزئية
امن الدولة apache/نوع اباتشي and experience in/والخبرات في remaining obstacles/العقبات المتبقية give details/يعط تفاصيل
مستوى only lead/الا يقود previously/كان سابقا ahmet/التركي احمد saudi cooperation/تعاون سعودي 's state security/في
مساجد from haifa/من حيفا eu and china/الاوروبى والصين maarib , demanding/مأرب مطالبين moussa/كان موسى level/لها
مصر new york :/: - نيويورك mutual support/الدعم المتبادل non-food/غير الغذائية mosques and hospitals/ومستشفيات
international intervention/تدخل دولية andrew smith/اندرو سميث be speaking for/انه يتحدث باسم egypt for/لاجراء

Figure 3.12: Heuristic vs. VB grammars: Top-100 high probability lexical rules extracted by heuristic as well our proposed method for Arabic-English dataset.

Chapter 4

A Unified-cascade Framework for Learning Alignments and Translation Rules

In the previous chapter, we focused on learning Hiero grammars using a Bayesian framework as an alternative to the heuristic approach followed by Hiero. We now step back and examine the larger part of the Hiero training pipeline. As we alluded to in the introductory chapter, the Hiero translation pipeline consists of a series of steps that are independent of each other. Several approaches have been proposed to address this disconnect in the context of phrase-based models (DeNero and Klein, 2010; Neubig et al., 2011), but not for Hiero models. In this chapter we present a unified-cascade framework for learning alignments and grammar in an iterative setup.

4.1 Motivation and Related work

As we pointed out earlier in Chapter 3, the heuristic models for extracting many-to-many alignments have been highly successful in MT. On the other hand principled models for extracting phrase alignments have faced challenges ranging from poor generalization, computational intractability (Marcu and Wong, 2002), computational complexity (DeNero et al., 2008; Zhang et al., 2008b) or their reliance on a heuristic extractor to improve the many-to-many alignments from the extracted alignments (Cohn and Haffari, 2013). While these approaches work with phrase-based models, some of the other works learn phrase alignments from synchronous derivations using ITG-style (Blunsom et al., 2008a) or Hiero-style rules (Blunsom et al., 2009; Levenberg et al., 2012) and apply them for hierarchical phrase-based models. However they go through the heuristic step for extracting the Hiero grammar which are finally used by the decoder.

In a different track several works have exploited word alignments to improving the performance of parsing (Burkett and Klein, 2008; Snyder et al., 2009) outside the machine translation setting. In the reverse direction syntactic parsing has been used to get better alignments (May and Knight, 2007; DeNero and Klein, 2007; Fossum et al., 2008) in the context of machine translations.

Joint models for learning alignments and translation rules have been a fairly recent direction. A joint model using two syntactic parsers and combined with an ITG derivation to model alignments, enables the trees to diverge if required and otherwise encouraging the derivation to synchronize with the trees (Burkett et al., 2010). However it requires a parallel treebank and gold alignments to train on in addition to parsers for the source and target languages, thus severely limiting its applicability. DeNero and Klein (2010) proposed a supervised model for extracting all overlapping bispans called *extraction sets* under a discriminative model by using phrase-level features in addition to the one-to-one alignments.

In contrast to the two supervised models, a recent unsupervised model set on a Bayesian framework uses synchronous ITG derivations together with a hierarchical model to jointly extract phrase alignments and translation rules (Neubig et al., 2011). The extracted translation rules are then directly used in a phrase-based decoder. Very recently, Saers et al. (2013a; 2013b; 2013c, *inter alia*) proposed a Bayesian maximum *a posteriori* (MAP) driven model for extracting bracketing inversion transduction grammar. This approach aims to improve coherence and model consistency between the training and test. This approach differs from Neubig et al. (2011) in the obvious way that the latter has the same model in both training and testing unlike the former that trains a ITG model which is then used as a flat translation lexicon in a phrase-based decoder. In addition to this, this approach replaces the complex MT training pipeline consisting of heuristic and often incompatible components with a single yet cleaner model.

As opposed to the approach by Saers et al., our approach involves a cascade model consisting of two steps that seek to jointly learn the alignments and the translation (ITG) rules for Hiero. While we might use ITG for learning the alignments, we do not use this to obtain the model structure. We extract the model structures from the flat alignments in the second step. Thus our joint inference model learns in the first step, the many-to-many correspondences that are overlapping, and learns the segmentation for extracting hierarchical rules in the second.

4.2 Unified-cascade Framework

We now explain the intuitive idea behind our framework in simple terms and then present an experimental setting in order to validate our framework. We finally propose a plan for using a new alignment model, which we leave for future experimentation.

The key idea of the framework is to separate the inference of alignments and Hiero grammar in two successive steps and then enclose the two steps in an iterative setup. Given the dissimilarity between the alignments and Hiero rules this separation makes it easier for the models to handle the two structures at different steps. Thus the first phase reasons over the sentence pairs to find overlapping alignments in them, naturally yielding a segmentation for the sentence pairs, i.e. *biphrases*. Subsequently the second phase, searches over the space of derivations (of the phrase-pairs) in order to learn the optimal ones leading to better grammar.

Intuitively, our framework could be explained as a two-step generative process for generating sentence pairs from the Hiero grammar. The first step uses the rules in the grammar to derive smaller overlapping phrase-pairs, which are then appropriately tiled in the second step to generate sentences. This is loosely analogous to how the phrase-structure rules generate (monolingual) sentences by going through the smaller syntactic phrases.

The framework consists of two steps, viz. i) generating phrase alignments of different granularities and ii) extracting Hiero rules that are consistent with the alignments. The two phases of the unified-cascade framework are repeated in an iterative setup.

1. Run the alignment model for a fixed burn-in period and collect alignment samples at every k iterations after the burn-in period. (Phase-1)

2. Use the alignment samples from the Gibbs sampler run as initial phrase-pairs and perform Variational EM for extracting grammar rules. Variational EM is run for a fixed number n_2 (set to 10) of iterations. (Phase-2)

3. Repeat steps 1 and 2 for N times and at each iteration collect the samples independently.

4.2.1 Experimental Setting

The formulation of the framework allows us to easily experiment with existing models for alignment and Hiero rule extraction. This would also help us quickly validate the effectiveness of our framework. We use pialign (Neubig et al., 2011) for the first phase and our binary model (Chapter 3) for the second phase.

The joint model proposed by (Neubig et al., 2011) uses a phrasal-ITG based hierarchical model with a Pitman-Yor Process prior. Unlike the earlier models (DeNero et al., 2008; Zhang et al., 2008b) that extract minimal many-to-many phrase alignments, Neubig et al.'s model extracts phrases of varying granularities. This is achieved by inverting the order to first generate the entire sentence from a phrase distribution following by ITG derivations that overcomes the sparsity problem. For inference it uses a sentence-level block sampler exploring the space of ITG-phrase alignments. In order to reduce the time complexity in sampling, it uses a heuristic beam search approximation that prunes the alignment spans based on a probability threshold.

The many-to-many alignments extracted by pialign is directly fed to our binary model that extracts the Hiero grammar. In the reverse direction, we could parse the sentences in the training set with the extracted Hiero grammar and use the resulting alignments to initialize the aligner in the next iteration. However, we decided to use a simple setting for our initial validation; hence we iterate the two steps of the unified framework without the reverse feedback.

4.3 Experiments

We could evaluate the unified-cascade framework for the three language pairs that we used in our binary Hiero model in the previous chapter. However, we are constrained by the speed of the aligner that we use in the unified-cascade framework. We thus evaluate our unified-cascade framework on two language pairs: Korean-English and Arabic-English. In both language we limit the sentence length of the training set to 60. For Arabic-English, we use a subset consisting of 120K sentence pairs sampled from the ISI parallel-corpus. The statistics of the two corpora are shown in Table 4.1.

Lang.	**Training Corpus**	**Train/ Tune/ Test**	**# Words (Src/ Tgt)**
Korean-English	University of Rochester corpus	52K/ 1118/ 1118	1.5M/ 1.4M
Arabic-English	ISI web-crawled parallel corpus	120K/ 1982/ 987	3.1M/ 3.3M

Table 4.1: Hiero binary grammar extraction: Corpus statistics for unified-cascade experiments. The sentences are restricted to have at most 60 words due to the limitation of the aligner.

For experiments involving pialign, we ran the aligner for 10 iterations with 9 burn-in iterations. The samples were read off from the last iteration. For extracting Hiero grammar we use the initial phrase-pairs obtained by pialign and pass them through either the heuristic extractor or our Variational-Bayes inference. We tuned the feature weights using MERT and decoded the test set with the optimal weights. For language model, we use a 5-gram gigaword model, trained by SRILM with Kneser-Ney smoothing. For the unified-cascade setting, we iterate the two steps of the framework for three runs and do a sample combination to get the final grammar.

Aligner	**Extractor**	**BLEU**
Giza++	Heuristic	7.97
Giza++	Var. Bayes	8.03
Pialign	Heuristic	7.70
Pialign	Var. Bayes	*7.54*
Unified-cascade (3 iters)		**8.19**

Table 4.2: Unified-cascade framework: Korean-English BLEU scores. For the unified-cascade framework we ran Pialign and VB inferences for three iterations and did a sample combination. The BLEU scores that are less than the baseline Moses (Giza++, Heuristic) BLEU of 8.23 by a statistically significant margin are *italicized*. The best BLEU score is in **boldface**.

The results of the unified-cascade inference are summarized in Tables 4.2 and 4.3 for Korean-English and Arabic-English settings respectively. We use four Hiero baselines that arise from different combinations of the aligner and extractor as listed in the tables.

The first two baselines use Giza++ aligner and then use the two different (heuristic and VB) methods for extracting translation rules, which are then tuned/ tested with Kriya. Baselines 3 and 4 differ from the earlier ones in that, these baselines use pialign to generate many-to-many alignments. The last row corresponds to the unified grammar setting, where we run the iterative inference three times and then aggregate the grammars.

In both language pairs, baselines employing pialign perform marginally worse and the first iteration of unified-cascade model in fact results in statistically significant BLEU reduction compared to phrase-based baseline of 8.23. However when we run our cascade framework for three iterations, we see consistent BLEU score improvements ranging between 0.2 and 0.65 as compared to other baselines in the table.

One can also compare these scores to the phrase-based model for the sake of completeness. We consider two phrase-based models one using the regular heuristic training pipeline as Koehn et al. (2003) and the other using pialign. For pialign, we use the phrase table extracted by pialign and directly used it with Moses for tuning and decoding. Note that this baseline uses two additional features including span probability (see Neubig et al. (2011)) that are not used in the standard baseline or in the later models in the tables. The two phrase-based models obtained BLEU scores of 8.23 and 8.30 respectively and these are comparable to the performance of our unified-cascade model.

Aligner	Extractor	BLEU
Giza++	Heuristic	25.13
Giza++	Var. Bayes	25.20
Pialign	Heuristic	24.97
Pialign	Var. Bayes	25.09
Unified-cascade (3 iters)		**25.45**

Table 4.3: Unified-cascade framework: Arabic-English BLEU scores. For the unified-cascade framework we ran Pialign and VB inferences independently for three runs and did a sample combination. The best BLEU score is in **boldface**.

Now turning our attention to the Arabic-English language pair we again notice a very similar behaviour as we saw for Korean-English. The only difference is that the scale of improvement is marginally less and our unified-cascade framework improves the BLEU scores in the range of 0.25 and 0.5 over the other baselines. The phrase-based model using Moses achieves 25.34 BLEU score, while the pialign achieves 24.90.

4.4 Future Extension

Thus far in this chapter, we presented a proof-of-concept for the unified-cascade framework by using an existing alignment model. Our experiments demonstrate the effectiveness of the unified-cascade framework and the BLEU scores are promising. Neubig et al.'s inference has been shown to be flawed due to the heuristic beam search approximation (Cohn and Haffari, 2013). While the pruning step allows their sampling to fast, the sampler fails to satisfy *positive recurrence* and *detailed balance* properties of a Markov chain. Additional their model was shown to perform poorly for various language pairs (Cohn and Haffari, 2013).

As a future extension, we would like to replace the Neubig et al. (2011) model with the following alignment model in order to improve the performance of our unified-cascade framework.

4.4.1 Phase-1: Inducing Alignments

In the first phase we generate the phrasal alignments of varying granularities for a given sentence. We use an ITG model for alignment having three types of links between source and target phrases:

terminal (TERM), regular (REG) and inverted (INV) ITG. The generative story is as follows:

1. Generate a phrase type z_p that can be one of TERM, REG and INV
2. If $z_p = $ REG or $z_p = $ INV:
    - Draw the left ($c_l = \langle f_l, e_l \rangle$) and right ($c_r = \langle f_r, e_r \rangle$) children from G_R - a distribution over *recursive* productions
    - Set $X \rightarrow \langle f_l, f_r, e_l e_r \rangle$ or $X \rightarrow \langle f_l, f_r, e_r e_l \rangle$ accordingly
3. If $z_p = $ TERM: decide the terminal phrase pair ($\langle f_t, e_t \rangle$) from an *emission* distribution G_E and set $X \rightarrow \langle f_t, e_t \rangle$

We assume a Pitman-Yor Process prior for G_R and a Dirichlet Process for G_E. Specifically:

$$G_R \sim \text{PYP}(d_R, s_R, P_R(c_l c_r)) \tag{4.1}$$

$$G_E \sim \text{DP}(\alpha_E, P_E) \tag{4.2}$$

The separation of priors for the recursive and terminal productions provides a better control over the diverse types. While the terminal phrase pairs are likely to be the minimal translation units (occurring frequently), the recursive productions yield larger units (and occur rarely) that are composed from smaller units in either monotone or swap orientations. We now explain the base distributions P_E and P_R to complete the generative story.

For the TERM type, we define the base distribution P_E using a formulation similar to that of (DeNero et al., 2008) in order to prefer short phrases in the leaf nodes.

$$P_E(\bar{f}, \bar{e}) = M_a(\bar{f}, \bar{e}) P_{Pois}(|\bar{f}|; \lambda_E) P_{Pois}(|\bar{e}|; \lambda_E)$$
$$M_a(\bar{f}, \bar{e}) = \left[P_f(\bar{f}) P_a(\bar{e}|\bar{f}) P_e(\bar{e}) P_a(\bar{f}|\bar{e}) \right]^{\frac{1}{2}}$$

where $P_{Pois}(.|\lambda_E)$ is a Poisson distribution over lengths of the phrases having a mean length of λ_E and M_a is the geometric mean of the joint distributions over phrase-pairs $(\bar{f}, \bar{e})$. P_a captures the probability of alignments according to some initial word alignments (most of the current works use IBM model-1 scores for this). P_f and P_e are the unigram models for source and target respectively.

Given a phrase type z_p rooted at some node and spanning the phrase-pair $(\bar{f}, \bar{e})$, the base distribution P_R for the recursive rules starts by deciding the source and target side lengths (uniformly over the respective phrase lengths) of the left child as:

$$|f_l| \sim P_u(|\bar{f}| - 1)$$
$$|e_l| \sim P_u(|\bar{e}| - 1)$$

The right child then gets the lengths: $|f_r| = |\bar{f}| - |f_l|$ and $e_r = |\bar{e}| - |e_l|$. We then generate the actual phrase-pairs according the geometric mean of the bi-directional alignment probabilities $M_a(f, e)$ as earlier. Thus the base distribution generating a pair of aligned phrase-pairs can be written as:

$$P_R(c_l, c_r|\bar{f}, \bar{e}) = P_u(|\bar{f}| - 1) * P_u(|\bar{e}| - 1) * M_a(f_l, e_l) * M_a(f_r, e_r) \tag{4.3}$$

The use of ITG productions intuitively explain the model in phase-1 as an ITG derivation over sentences. However, we prefer to use the alignment matrix representation as it allows us to explain it as an alignment process to capture the alignments between the source and target phrases. The yields of the subtrees in the ITG derivation then correspond to the aligned phrases of varying granularity.

Given this alternative representation, at each time step t the model either reuses an existing sub-phrase alignments (from the *cache*) or generate new alignments according to the base distribution.

We intend to use a Gibbs sampler in the first phase to infer a posterior distribution over the phrase pairs. We could perform the sampling efficiently by exploring the alignment matrix for each sentence in a top-down fashion similar to Levenberg et al. (2012) and drawing samples for each node in the ITG alignment tree.

4.4.2 Phase-2: Extracting Hiero Grammar

$\phi^z \sim$ Dirichlet$(\boldsymbol{\alpha_z})$	[draw derivation type parameters]
$G_H \sim$ Dirichlet$(\alpha_H P_H)$	[draw rule parameters]
$z_d \sim$ Multinomial(ϕ^z)	[decide the derivation type]
$r\vert\mathbf{r} \in d_x \sim G_H$	[generate rules deriving phr-pair x]

Figure 4.1: Model definition for Rule Extraction

We presented the model in the previous chapter (see section 3.3) and so we just present a summary of the generative story and the model. As earlier the model in phase-2 reasons over the space of phrase alignments (extracted in phase-1) to induce Hiero grammar.

1. Choose the derivation type z_d for generating the aligned phrase pair x. It can either be a terminal derivation or hierarchical derivation, i.e. $z_d = \{\text{TERM}, \text{HIER-A1}, \text{HIER-A2}\}$.

2. Draw the constituent rules $\mathbf{r}$ in the derivation to generate the phrase pair from a distribution over Hiero rules G_H.

The derivation-type z_d is sampled from a multinomial distribution parameterized by ϕ^z, where ϕ^z itself is drawn from a Dirichlet distribution having a hyper-parameter $\boldsymbol{\alpha_z}$.

We assume a Dirichlet Process prior for G_H, parameterized by a concentration parameter α_H and a base distribution P_H. While the model is same as the binary Hiero model (Sankaran et al., 2013a) presented in the previous chapter, we would like to replace our current parametric prior with a more generalized prior following (DeNero et al., 2008; Neubig et al., 2011). The base measure P_H is simple but yet informative, and is defined similar to $M_a(\bar{f}, \bar{e})$ in phase-1. This allows us to only explore the rules that would be consistent with the underlying word alignments. Thus our setting closely resembles that of the Hiero heuristic rule extraction.

Under this model the probability of a particular derivation $d \in \phi_x$ for a given phrase pair x can be expressed as:

$$p(d) \propto p(z_d) \prod_{r \in d} p(r|\mathcal{G}, \theta) \tag{4.4}$$

where r is a rule in grammar $\mathcal{G}$ and θ is the grammar parameter.

Our goal is thus to infer the joint posterior $p(\theta, \Phi|\alpha_H, P_H, \boldsymbol{\alpha_z}, \mathcal{X})$, where θ are the model parameters and Φ the latent derivations over all the phrase pairs. The base distribution P_H will use the alignment probabilities obtained from the first phase instead of the Model-1 alignments.

4.4.3 Unified-cascade: Iterative Inference

The two phases of the unified-cascade framework explained above are repeated iteratively for a fixed number of times, alternately yielding alignments and Hiero grammar.

1. Run the Gibbs sampler for n_1 iterations with some initial burn-in period and collect alignment samples at every k iterations after the burn-in period. (Phase-1)

2. Use the alignment samples from the Gibbs sampler run as initial phrase-pairs and perform Variational EM for extracting grammar rules. Variational EM is run for a fixed number n_2 (set to 10) of iterations. (Phase-2)

3. In the reverse direction, the sentences in the training corpus are parsed (by force decoding) with the extracted Hiero grammar and use the resulting alignments to initialize the aligner in the next iteration.

4. Repeat steps 1 through 3 for N times and at each iteration collect the samples independently.

The samples from each iteration of the outermost loop correspond to the different modes of the respective posterior distributions. The individual samples could then be aggregated (Neubig et al., 2011; Cohn and Haffari, 2013) potentially yielding improved performance.

4.5 Better Hiero Translation Model: Summary

We focussed on improving the Hiero translation model so far in this thesis. In Chapter 3, we presented a new Bayesian model for learning Hiero grammar as an alternative to the heuristic extraction. The Bayesian model yielded better parameter estimation for the extracted rules with peaked distribution, as opposed to the flat distribution of the heuristic method. Our Bayesian model also resulted in better translation performance in the small-data setting.

Approach	**BLEU**
Rule Extraction Step only	
Heuristic extraction	37.82
Binary model (VB)	37.76
Full training pipeline	
Heuristic baseline (Giza++ & Heuristic extraction)	25.13
Unified-cascade (3 iterations)	**25.45**

Table 4.4: Summary of improved Hiero translation models for two settings i) rule extraction step only and ii) entire training pipeline. For each setting BLEU scores for the baseline as well our proposed approach are listed. Notice that the training corpus for the full training pipeline use *only* a subset of the original training data and the BLEU scores are reflect this.

In this Chapter we presented the unified-cascade framework, where the key idea is to jointly learn the alignments and Hiero grammars. Our approach involves two distinct models for learning the alignments and Hiero grammars in a iterative setting. While joint models have been proposed earlier for phrase-based translation, to our knowledge this is the first time a joint approach is proposed for learning alignments and translation model for Hiero. The unified-cascade approach also leads to higher translation quality as we showed earlier.

Table 4.4 summarizes the results of the two chapters so far for Arabic-English dataset. Our binary model retains the BLEU score as the heuristic approach when it is applied on for the rule extraction step. Our unified-cascade approach shows marginal BLEU score gain over the heuristic baseline that uses separate steps for alignment and extraction.

Chapter 5

Compact Grammars for Hiero

In the previous two chapters, we examined the issue of improving the estimation of Hiero translation model in order to get better translation performance. In this chapter, we concern ourselves with the problem of excessively huge Hiero translation models and seek ways to induce compact models for Hiero. We address this issue from two different directions.

Firstly we consider the case of a simpler Hiero grammar, where the maximum rule arity is limited to 1. Such unary grammars have smaller footprints than binary Hiero grammar and we show them to be sufficient for close language pairs. We also propose different models for extracting grammars based on combinatorial optimization and Bayesian frameworks.

Secondly we demonstrate that the simple idea of pruning the extracted grammars can also lead to substantial reduction in the model sizes. While pruning could also be theoretically applied in the context of heuristically extracted Hiero grammars, the poor rule probability estimation of the heuristic approach negatively impacts the translation performance of the pruned grammars. Since these directions are orthogonal to each other, we combine them by pruning the unary grammars to gain further reduction.

5.1 Motivation

As we discussed, the primary advantage of Hiero-style systems lie in their unsupervised model of learning parallel tree structures. However, one of the major issue in Hiero systems is its large model size, which is three to five times larger than a phrase-based model trained from same training data. For example Figure 5.1 plots the model size against BLEU for phrase-based and Hiero models for five language pairs listed in Table 2.1. As earlier we use Moses for experiments with phrase-based

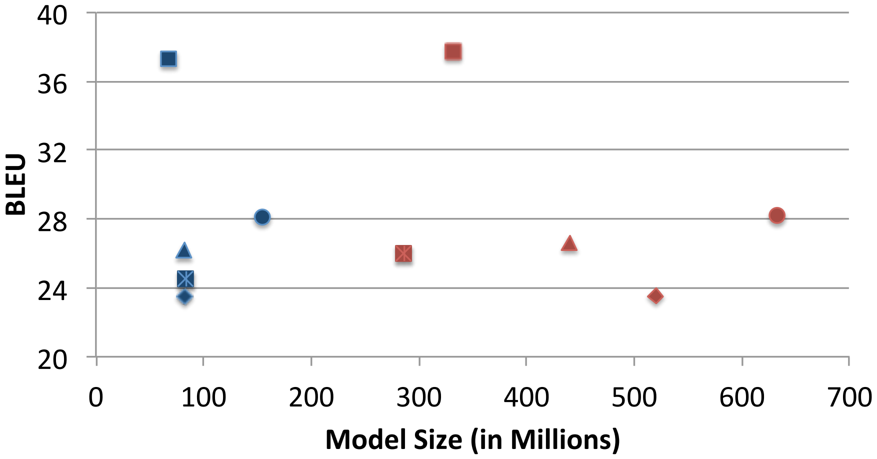


Figure 5.1: Model size vs. BLEU for Phrase-based (points in blue) and Hierarchical phrase-based (points in red) models for different language pairs mentioned in Table 2.1. We use Moses and Kriya systems for phrase-based and Hiero models respectively.

models and Kriya for Hiero models.

While the BLEU scores between the two systems are comparable for each language pair, the Hiero models are about 2.5 to 6 times larger than their phrase-based counterparts.

This leads to issues such as over-generation and slower decoding (Zollmann et al., 2008; de Gispert et al., 2010a). Additionally as we noted in Section 3.1 the heuristics used for learning the rule parameters results in a flat distribution as it does not have any means for discriminating poor translations from good ones locally within each phrase pair.

Secondly, the compact models could be extremely practical in the modern smartphone era, for running machine translation software/service in smartphones and other mobile devices that have limited memory and computing power. Finally compact translation models are also preferable from the perspectives of Occam's Razor and minimum description length (MDL) principle (Rissanen, 1983).

While the majority of the research in Hiero is focused on the improving the decoding, comparatively few papers have explored the inference of the probabilistic Hiero-style SCFG. Several methods have been proposed for pruning/ filtering the model *post-hoc* - after the heuristic step of extracting Hiero grammar. These approaches typically aim to reduce the model size by employing various pruning techniques such as removing redundancy arising from monotone composed rules (He et al., 2009), filtering rules based on certain patterns (Iglesias et al., 2009) and so on.

Another stream of research employs Bayesian models for learning the synchronous grammar (Blunsom et al., 2008a; Blunsom et al., 2009; de Gispert et al., 2010b; Levenberg et al., 2012). Our approach is similar to these in that, we propose a Bayesian approach for Hiero-style rule extraction. Our objective in this work is two-fold, i) to provide a principled rule extraction strategy using a Bayesian framework and ii) to extract a *minimal* SCFG grammar without reducing the translation performance measured through BLEU.

As mentioned earlier, we use a simpler grammar called *unary* Hiero grammar, where the maximum rule arity is restricted to 1. Consider the phrase-pair shown in Figure 3.4 from Chapter 3, which we reproduce here (Figure 5.2) for convenience. The unary grammar will have two types of derivations, viz. terminal (TERM) and arity-1 (HIER-A1), as shown in Figure 5.3 for this example phrase-pair.

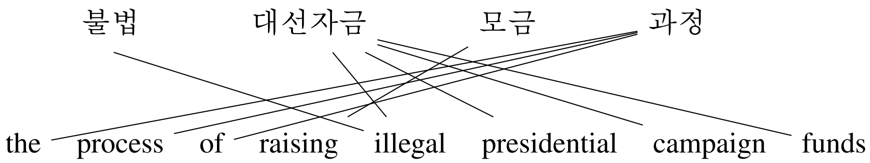


Figure 5.2: Korean-English phrase-pair shown with word alignments (reproduced from Figure 3.4)

#1 $X \rightarrow \langle$불법 대선자금 모금 과정, the process of raising illegal presidential campaign funds$\rangle$

#2 $X \rightarrow \langle$불법 대선자금 X_1, X_1 illegal presidential campaign funds$\rangle$
$X \rightarrow \langle$모금 과정, the process of raising$\rangle$

#3 $X \rightarrow \langle$불법 대선자금 X_1 과정, the process of X_1 illegal presidential campaign funds$\rangle$
$X \rightarrow \langle$모금, raising$\rangle$

Figure 5.3: Three possible derivations (among many others) of the phrase-pair in Fig 5.2. The unary grammars include two types of derivations: Terminal (TERM) as in derivation #1 and Arity-1 (HIER-A1) as in derivations #2 and #3.

Thus each rule in the unary grammar can have at most one non-terminal X for embedding a terminal sequence. While the resulting model is slightly weaker than the original Hiero grammar, it should be noted that the unary model *does* allow reordering and discontiguous alignments, which are the key characteristics of the Hiero formalism. For example unary model would include rules such as, $X \rightarrow \langle \alpha X_1 \beta, \alpha' \beta' X_1 \rangle$, which can capture phrases like (*not* X_1, *ne* X_1 *pas*) in the case of a English-French translation task. In terms of the long-distance reordering capability, unary model lies in between the regular Hiero and phrase-based models.

5.2 Unary model-1: Model

Given the word alignments and the heuristically extracted phrase pairs $\mathcal{X}$, our goal is to extract the minimal set of *hierarchical* rules $\mathcal{G}$ that would best explain $\mathcal{X}$. This is achieved by inferring a distribution over the derivations for each phrase pair, where the set of derivations collectively specify the grammar. In the following, we denote the sequence of derivations for the set of phrase pairs by Φ and denote the derivations of a specific phrase-pair x by ϕ_x, which is composed of grammar rules $\mathbf{r}$. We will essentially read off our learned grammar $\mathcal{G}$ from the set of sampled derivations.

Our Bayesian model reasons over the space of the (hierarchical and terminal) derivations and samples a derivation by employing a novel prior based on the alignment probability of the words in the phrase pairs. We hypothesize that the resulting grammar will be compact and also will explain the phrase pairs better (the SCFG rules will maximize the likelihood of producing the entire set of observed phrase pairs).

Using Bayes' rule, the posterior over the derivations $\mathcal{G}$ given the initial phrase-pairs $\mathcal{X}$ can be written as:

$$P(\mathcal{G}|\mathcal{X}) \propto P(\mathcal{X}|\mathcal{G})P(\mathcal{G}) \tag{5.1}$$

where $P(\mathcal{X}|\mathcal{G})$ is equal to one when some combination of rules in $\mathcal{G}$ is consistent with a particular phrase-pair x from $\mathcal{X}$, i.e. $\mathcal{G}$ can be partitioned into derivations to compose the phrase-pair x such that the derivations respect the given word alignments; otherwise $P(\mathcal{X}|\mathcal{G})$ is zero. The overall structure of the model is analogous to the Bayesian model for inducing Tree Substitution Grammars proposed by Cohn et al. (2009). Note that, our model extracts hierarchical rules for the word-aligned phrase pairs and not for the sentences.

As mentioned earlier, we use two types of rules: *terminal* and *hierarchical* rules. For each phrase-pair, our model either generates a terminal rule by *not* segmenting the phrase-pair, or decides to *segment* the phrase-pair and extract some rules.

Though it is possible to segment phrase-pairs by two (or more) non-overlapping spans, we propose to use the unary Hiero model in this chapter and consider the binary Hiero model in the next chapter. Under this model, our inference explores the phrase-pair and identifies a sub-span for introducing the non-terminal, so that the sub-span could be written as a separate *terminal rule* (note that the sub-span is *not* decomposed further). The derivations #2 and #3 in Figure 5.3 correspond to two possible derivations for the phrase-pair in Figure 5.2, where the non-terminals are introduced at different sub-spans.

We use a two step generative process for generating a phrase pair x from the grammar rules. In the first step, the model decides on the type of the rule $z_d \in \{\textsc{Term}, \textsc{Hier}\text{-A1}\}$ used to generate the phrase-pair based on a Bernoulli distribution, having a prior γ coming from a Beta distribution:

$$z_d \sim \text{Bernoulli}(\gamma) \tag{5.2}$$

$$\gamma \sim \text{Beta}(l_x, 0.5) \tag{5.3}$$

The lexical alignment probability l_x controls the tendency for extracting hierarchical rules from the phrase-pair x. For a given phrase-pair, l_x is computed by taking the arithmetic (or geometric) average of the reverse and forward alignment probabilities, which we explain later in this section. Integrating out γ gives us the conditional probabilities of choosing the rule type z_d as:

$$p(t_{term}|x) \propto n^x_{term} + l_x \tag{5.4}$$

$$p(t_{hier}|x) \propto n^x_{hier} + 0.5 \tag{5.5}$$

where n^x_{term} and n^x_{hier} denote the number of terminal or hierarchical rules, among the rules extracted so far from the phrase-pair x during the sampling.

In the second step, if the rule type $z_d = \textsc{Hier}\text{-A1}$, the model generates the phrase-pair by sampling from the hierarchical and terminal rules. We use a Dirichlet Process (DP) to model the generation of hierarchical rules r:

$$G \sim DP(\alpha_h, P_0(r))$$

$$r \sim G$$

Integrating out the grammar G, the predictive distribution of a hierarchical rule r_x for generating the current phrase-pair (conditioned on the rules from the rest of the phrase-pairs) is:

$$p(r_x|r^{-x}, \alpha_h, P_0) \propto n^{-x}_{r_x} + \alpha_h P_0(r_x) \tag{5.6}$$

where $n^{-x}_{r_x}$ is the count of the rule r_x in the rest of the phrase-pairs that is represented by r^{-x}, P_0 is the base measure, and α_h is the concentration parameter controlling the model's preference towards using an existing hierarchical rule from the cache or to create a new rule sanctioned by the base distribution. We use the lexical alignment probabilities of the component rules as our base measure:

$$P_0(r) \propto \Big[\Big(\prod_{(k,l)\in a} p(e_l|f_k)\Big)^{\frac{1}{|a|}} \Big(\prod_{(k,l)\in a} p(f_k|e_l)\Big)^{\frac{1}{|a|}}\Big]^{\frac{1}{2}} \tag{5.7}$$

where a is the set of alignments in the given sub-span; if the sub-span has multiple Viterbi alignments from different phrase-pairs, we consider the union of all such alignments. Note that this prior is different from DeNero et al. (2008) in that our prior is parametric and only supports phrase-pairs observed in the training data.[1] Because of this we normalize our base measure probabilities for each phrase-pair to get a distribution.

While Equation (5.7) uses the product of geometric means of the forward and reverse alignment scores, we also experimented with the arithmetic mean of the lexical alignment probabilities. The lexical prior l_x used in the first step of our generative story can be defined similarly. We found the particular combination of using arithmetic mean for the lexical prior l_x (in the first step) and geometric mean for the base distribution P_0 (in the second step) to work better, as we discuss later in Section 5.4.

This prior encodes our hypothesis that the lexical alignment probabilities of a phrase-pair would be a good indicator for whether or not it needs to decomposed into a derivation consisting of several rules. When the lexical alignment probability is high (close to 1), the Beta prior in Equation (5.3) would prefer a terminal derivation.

5.3 Unary model-1: Inference

We train our model by using a Gibbs sampler – a Markov Chain Monte Carlo (MCMC) method for sampling one variable in the model, conditional to the other variables. The sampling procedure is repeated for what is called a long Gibbs chain spanning several iterations, while the counts are collected at fixed *thin* intervals in the chain. As is common in the MCMC procedures, we ignore samples from a fixed number of initial *burn-in* iterations, allowing the model to move away from the initial bias. The rules in the final sampler state at the end of the Gibbs chain along with their counts averaged by the number of thin iterations become our translation model.

We initialize the sampler by using our lexical alignment prior and sampling from the distribution

[1] We intend to relax this for the unified-cascade framework (in Chapter 4) by using a non-parametric prior similar to DeNero et al. (2008).

of derivations as suggested by the priors. We found this to perform better in practice, than a naive sampler without an initializer.

At each iteration, the Gibbs sampler processes the phrase pairs in random order. For each phrase pair $\mathcal{X}$, it visits the nodes in the corresponding alignment tree and computes the posterior probability of the derivations and samples from this posterior distribution. To speedup the sampling, we store the pre-computed alignment tree for the phrase pairs and just recompute the derivation probabilities based on the sampler state at every iteration. While the sampler state is updated with the counts at each iteration, we accumulate the counts only at fixed intervals in the Gibbs chain. In applying the model for decoding, we use the grammar from the final sampler state.

Since our model includes only one hyperparameter α_h, we tune its value manually by empirically experimenting on a small set of initial phrase pairs. We keep for future work the task of automatically tuning for hyper-parameter values by sampling. We use the decomposition tree algorithm explained in Section 3.4.2 to efficiently encode the word aligned phrase-pair as a normalized decomposition tree (Zhang et al., 2008a). The possible derivations (that are consistent with the word alignments) could then be enumerated by simply traversing every node in the decomposition tree and replacing its span by a non-terminal X.

5.4 Unary model-1: Experiments

As we seek to evaluate the effectiveness of compact unary grammars, we want to choose a setting of closely related language pairs. The unary Hiero grammars have been shown to result in performance degradation for diverse language pairs such as Chinese-English and Urdu-English (Zollmann et al., 2008). We thus choose the setting of translating from English into Spanish. In the later part of this Chapter, we introduce more diversity by considering three language pairs.

We use the English-Spanish data from WMT-10 shared task for the experiments to evaluate the effectiveness of our Bayesian rule extraction approach. See Table 5.6 for the statistics of the dataset used in our experiments. We used the entire shared task training set except the UN data for training translation model and the language model was trained with the same set and an additional 2 million sentences from the UN data, using SRILM toolkit with Kneser-Ney discounting. We tuned the feature weights on the WMT-10 dev-set using MERT (Och, 2003) and evaluate on the test set by computing lower-cased BLEU score (Papineni et al., 2002) using the WMT-10 standard evaluation script.

We use *Kriya* with the following standard features (4 translation model features, rule penalty, word penalty and language model) as is typical in Hiero-style systems. For tuning the feature weights, we have adapted the MERT implementation in Moses[2] for use with Kriya.

We compare the performance of our model with two baselines i) binary Hiero and ii) unary Hiero, which were trained using the conventional heuristic extraction approach. For the unary grammar baseline, we modified the heuristic rule extraction algorithm appropriately[3].

Additionally we also wanted to compare our model to different rule filtering strategies. We first use the greedily trained pattern-based filtering (Iglesias et al., 2009), where the idea is to decide which patterns are important based on their effectiveness on a held out data. We use the patterns suggested by Iglesias et al. (2009) to filter the grammar[4]. For the other baseline filtering experiments, we retained only unary rules and then further limited it by retaining only non-monotone unary rules; in both cases the terminal rules were retained.

Model	**# of rules filtered for devset (in millions)**	**BLEU**
Binary Hiero		
Heuristic baseline	52.36	**27.45**
Unary Hiero		
Heuristic baseline	22.09	26.71
Pattern-based filtering†	18.78	24.61
Monotone & non-monotone	10.36	24.17
Non-monotone only	3.62	23.99

Table 5.1: English-Spanish Results for Heuristic baseline and filtered grammars. †: This is the initial rule set used in Iglesias et al. (2009) obtained by greedy filtering. Rows 4 and 5 represents the filtering that uses unary grammar with row 4 allowing monotone rules in addition to the non-monotone (reordering) rules.

Table 5.1 shows the results for baseline and the rule filtering experiments. Restricting rule extraction just to unary grammar does not affect the BLEU score significantly as we also saw in the previous chapter. Secondly, we find significant reduction in the BLEU for the pattern-based

[2] www.statmt.org/moses/

[3] Given an initial phrase pair, the algorithm would introduce a non-terminal for each sub-span consistent with the alignments and extract rules corresponding to each sub-span. The constraints relating to binary grammar case (such as, no adjacent non-terminals in source side) does not apply for the unary models.

[4] It should be noted that we didn't use the augmentations to the initial rule set (Iglesias et al., 2009) and our objective is to find the impact of the filtering approaches.

filtering strategy and this is because we only use the initial rule set obtained by greedy filtering without augmenting it with other specific patterns. The other two filtering methods reduced the BLEU further but not significantly. The second column in the table gives the number of SCFG rules filtered for the dev-set, which is typically much less than the full set of rules. We later use this to put in perspective the effective reduction in the model size achieved by our Bayesian model. We can ideally compare our Bayesian rule extraction using Gibbs sampling with the baselines and the filtering approaches. However, running our Gibbs sampler on the full set of phrase pairs requires sampling to be distributed, possibly with approximation (Newman et al., 2007; Asuncion et al., 2008).

Instead, we focus on evaluating our Gibbs sampler on reasonable sized set of phrase pairs with corresponding baselines. We filter the initial phrase pairs based on their frequency using three different thresholds, viz. 20, 10 and 3- resulting in smaller sets of initial phrase pairs because we throw out infrequent phrase pairs (the threshold-20 case is the smallest initial set of phrase pairs). This allows us to run our sampler as a stand-alone instance for the three sets, obviating the need for distributed sampling. While the earlier Bayesian approaches (Blunsom and Osborne, 2008; Blunsom et al., 2008a) used smaller training corpora ranging from 33-300K sentence pairs, we use larger training data (about 1.7M sentence pairs) but restrict the set of initial phrase-pairs based on three frequency thresholds as mentioned above.

Table 5.2 shows the number of unique phrase pairs in each set. While, the filtering reduces the number of phrase pairs to a small fraction of the total phrase pairs, it also increases the unknown words (OOV) in the test set by a factor between 1.8 and 3. In order to address this issue due to the OOV words, we additionally added *non-decomposable phrase pairs* having just one word at either source or target side, as coverage rules. The coverage rules (about 1.8 million) were added separately to the SCFG rules induced by both heuristic algorithm and Gibbs sampler. This is justified

Phrase-pairs set	**# of Unique phrase-pairs**	**Testset OOV**
All phrase-pairs	110782174	1136
Threshold-20	292336	3735
Threshold-10	606590	3056
Threshold-3	2689855	2067

Table 5.2: English-Spanish phrase-pair statistics for different frequency thresholds

Experiment	Threshold-20	Threshold-10	Threshold-3
Binary Hiero			
Heuristic Extraction	24.30	25.96	26.34
Unary Hiero			
Heuristic Extraction	**24.00**	**25.90**	**26.83**
Unary model-1	23.39	24.30	25.22

Table 5.3: English-Spanish BLEU scores: Heuristic vs Bayesian rule extraction

Experiment	Rules Extracted (in millions)		% Reduction
	Heuristic	Unary model-1	
Threshold-20	1.93 (0.117)	1.86 (0.07)	3.6 (38.34)
Threshold-10	2.91 (1.09)	2.10 (0.28)	27.7 (73.95)
Threshold-3	7.46 (5.64)	2.45 (0.71)	**67.2 (87.28)**

Table 5.4: Model compression for unary grammar: Heuristic vs Bayesian rule extraction. The numbers outside the parentheses correspond to the statistic of number of extracted rules (numbers inside the parentheses) together with the number of coverage rules.

because we only add the rules that can not be decomposed further by both rule extraction approaches. However, note that both approaches can independently induce rules that overlap with the coverage rules set and in such cases we simply add the original corpus count to the counts returned by the respective rule extraction method.

The Gibbs sampler considers the phrase pairs in random order at each iteration and induces SCFG rules by sampling a derivation for each phrase pair. Given a phrase pair x with raw corpus frequency f_x, we simply scale the count for its sampled derivation ϕ_x by its frequency f_x. Alternately, we also experimented with independently sampling for each instance of the phrase pair and found their performances to be comparable. Sampling phrase pairs once and then scaling the sampled derivation, help us to speed up the sampling process. In our experiments, we ran the Gibbs sampler for 2000 iterations with a burn-in period of 200, collecting counts every 50 iterations. We set the concentration parameter α_h to be 0.5 based on our experiments detailed later in this section.

The BLEU scores for the SCFG learned from the Gibbs sampler are shown in Table 5.3. We first note that, the threshold-20 set has lower baseline BLEU than threshold-10 and threshold-3 sets, as can be expected because threshold-20 set uses a much smaller subset of the full set of phrase pairs

Priors	α_h	**BLEU**
Arith + Arith means	0.5	22.46
Arith + Geom means	0.5	**23.39**
Geom + Arith means	0.5	22.96
Geom + Geom means	0.5	22.83
Arith + Geom means	0.1	22.88
Arith + Geom means	0.2	22.97
Arith + Geom means	0.3	22.98
Arith + Geom means	0.4	22.69
Arith + Geom means	0.5	**23.39**
Arith + Geom means	0.6	22.89
Arith + Geom means	0.7	22.82
Arith + Geom means	0.8	22.82
Arith + Geom means	0.9	22.67

Table 5.5: Effect of different priors and α_h on Threshold-20 set. The two priors correspond to the lexical prior l_x in the first step and the base distribution P_0 in the second step (in same order). Arithmetic means prior works better for deciding the derivation type z_d and geometric means is better for deciding where to segment the phrase-pair.

to extract hierarchical rules. The Bayesian approach results in a maximum BLEU score reduction of 1.6 for the sets using thresholds 10 and 3, compared to the unary grammar baseline. The binary grammar baseline is also provided to place our results in perspective.

Table 5.4 shows the model size, including the coverage rules for the two rule extraction approaches. The number of extracted rules, excluding the coverage rules are shown within the parenthesis. The last column shows the reduction in the model size for both with and without the coverage rules; yielding a maximum absolute reduction of 67.2% for the threshold-3 phrase pairs set. It can be seen that the number of rules are far fewer than the rules extracted using the baseline heuristic methods for filtering detailed in Table 5.1. Interestingly, we obtain a smaller model size, even as we decrease the threshold to include more initial phrase pairs used as input to the inference procedure, e.g. a 67.2% reduction over the rules extracted from the threshold-3 phrase pairs v.s. a 27.7% reduction for threshold-10.

These results show that our model is capable of extracting high-value Hiero-style SCFG rules, albeit with a reduction in the BLEU score. However, our current approach offers scope for improvement in several avenues, for example we can use annealing to perturb the initial sampling iterations to encourage the Gibbs sampler to explore several derivations for each phrase pair. Though this might result in slightly large models than the current ones, we still expect substantial reduction than the original Hiero rule extraction. In future, we also plan to sample the hyperparameter α_h, instead of using a fixed value.

Table 5.5 shows the effect of different values of the concentration parameter α_h and the priors used in the model. The order of priors in each setting correspond to the prior used in deciding the rule-type and identifying the non-terminal span for sampling a derivation. We found the geometric mean to work better in the latter while the derivation type z_d is best decided by the arithmetic mean prior. We further found that the concentration parameter α_h value 0.5 gives the best BLEU score.

5.5 Issues with Unary model-1

The proposed model was able to achieve over 65% reduction in the grammar size, as compared to the heuristic approach, albeit with a slight reduction in the BLEU scores. One problem for the less than desirable performance could be due to the strict requirement of sampling *one* derivation per phrase-pair. We can easily see that a given phrase pair could have more than one good derivation.

$X \rightarrow \langle X_1$ 모금 과정, the process of raising $X_1 \rangle$
$X \rightarrow \langle$불법 대선자금, illegal presidential campaign funds$\rangle$

$X \rightarrow \langle$불법 대선자금 X_1, X_1 illegal presidential campaign funds$\rangle$
$X \rightarrow \langle$모금 과정, the process of raising$\rangle$

$X \rightarrow \langle$불법 대선자금 X_1 과정, the process of X_1 illegal presidential campaign funds$\rangle$
$X \rightarrow \langle$모금, raising$\rangle$

Figure 5.4: Multiple-views of generating the phrase-pair in Fig 5.2. None of these derivations is distinguishable from others in terms of translation quality. They vary in which sub-phrase is being lexicalized; further the component rules appear to be reasonable translations and are likely to be used by the decoder wherever applicable.

To illustrate this consider the derivations of the aligned phrase-pair shown in Figure 5.2. There could be several derivations of this phrase-pair all of which encode reasonably good translations that are indistinguishable from each other, some of which are shown in Figure 5.4. And these derivations offer multiple views of generating the phrase-pair and the decoder might actually benefit by incorporating several views. The posterior distribution inferred by Gibbs sampler could theoretically factor in these different derivations, by sampling them at different time steps. This could however be influenced by the concentration parameter and the cache that embodies rich gets richer phenomenon. Additionally, the number of sampling iterations should be reasonably high in order for the inferred posterior to be close to the true posterior.

5.5.1 Where do we go from here?

We could potentially improve the performance with Gibbs sampling inference by addressing some of these concerns. We would also like to train our model on large-scale SMT datasets. This requires the Gibbs sampling to be distributed, which can only be an approximation of the single processor inference (Newman et al., 2007).

So in the rest of this chapter, we explore two other models with for inducing compact Hiero grammars but without compromising on the translation performance. Our first approach learns a minimal grammar by solving a combinatorial optimization problem over a tripartite graph consisting of three types of nodes: phrase pairs, derivations, and translation rules. This is reduced to a *minimum set cover* problem and we devise a greedy approach to extract a minimal set of translation rules to cover all the phrase pairs. Our second approach, which learns a compact but not necessarily a minimal grammar, is based on a Bayesian model for generating phrase pairs from the Hiero grammar. We call this unary model-2 and use Variational Bayes (VB) for its inference as opposed to the Gibbs sampling used for the unary model-1.

Our new Bayesian model induces a compact Hiero grammar that has comparable performance to the original Hiero grammar in terms of the translation quality, and even improves on the full Hiero grammar when faced with a small amount of bilingual training data. On several datasets, the VB method achieves a significant reduction in the grammar size. We analyze the different extracted grammars and explain why the Bayesian model works better.

5.6 Combinatorial Optimization Approach

In our first approach we pose the problem of learning a minimal Hiero grammar in the combinatorial optimization framework as follows. *To find the minimum subset of translation rules based on which, at least one derivation can be constructed for each initial phrase pair*. This problem is closely related to the *minimum set cover problem* (Vazirani, 2004), a well known NP-hard problem.[5]

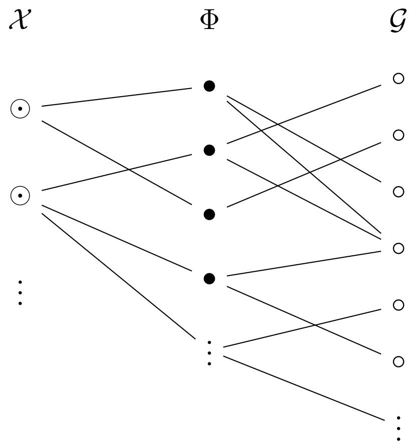


Figure 5.5: Combinatorial optimization approach: Tripartite graph representation of phrase-pairs ($\mathcal{X}$), derivations (Φ) and grammar rules ($\mathcal{G}$)

We represent the problem as a tripartite graph $\mathcal{T}_G$ consisting of three types of vertices as in Fig 5.5.

- v_x are vertices for phrase pairs for each phrase pair $x \in \mathcal{X}$
- $v_{d,x}$ are vertices for derivations for each phrase pair, where $d \in \phi_x$ is a derivation from the set of all derivations ϕ_x for an initial phrase pair x
- v_r are vertices for translation rules, for each $r \in \mathcal{G}$, where $\mathcal{G}$ is the set of all constituent rules observed in the derivations of the initial phrase pairs $\mathcal{X}$

In terms of $\mathcal{T}_G$, our aim is to select a minimal subset of rule vertices $\{v_r\}$ such that at least one derivation vertex $v_{d,x}$ is picked for each phrase-pair vertex v_x. We devise an efficient *greedy* algorithm to find an approximate solution for our optimization problem towards learning a compact Hiero grammar[6].

[5] Given a set of elements (called the universe) and some sets whose union comprises the universe, the *minimum set cover problem* is to identify the smallest number of sets whose union still contains all elements in the universe.

[6] In early experiments, we expressed these desiderata using an integer linear program (ILP) and its linear program (LP) relaxation. However, the size of resulting optimization problem was very large for the SMT datasets (which typically contain millions of sentence pairs). Hence, the solution of the ILP was beyond the capacity of available off-the-shelf solvers (CPLEX).

Algorithm 2 Greedy Algorithm for extracting Minimal Grammar

```
Input: Init phrases $\mathcal{X}$, derivations $\Phi$ and rules $\mathcal{G}$
$G_m \leftarrow \emptyset$ // minimal grammar
$C \leftarrow \mathcal{X}$ // initial phrases to be covered
while $C \neq \emptyset$ do
  $v_r \leftarrow \arg\max_{r' \in \mathcal{G}}$ degree$(v_{r'}, \mathcal{T}_G)$
  $G_m \leftarrow G_m \cup \{r\}$
  Remove $v_r$ from $\mathcal{T}_G$
  for $x \in C$ do
    if $\exists d \in \phi_x$ such that COV$(v_{d,x})$ is true then
      $C \leftarrow C - \{x\}$
      for $d \in \phi_x$ do
        Remove $v_{d,x}$ from $\mathcal{T}_G$
Output: Minimal grammar $G_m$
```

The greedy method, which is listed in Algorithm 2, extracts a minimal grammar G_m that explain the set of initial phrase pairs $\mathcal{X}$ by covering at least one derivation d for each phrase pair x. We iteratively repeat the following two steps until there are no initial phrase-pair vertices in the tripartite graph $\mathcal{T}_G$: (i) Select the rule vertex which is connected to the most number of derivations in the graph, and (ii) Remove this rule and all the derivations and phrase-pair vertices reachable from this rule provided the phrase-pair vertex is covered through at least one derivation vertex. The routine degree$(v_r, \mathcal{T}_G)$ returns the in-degree of a rule vertex v_r in $\mathcal{T}_G$. The function COV$(v_{d,x})$ is a Boolean function returning *true* if this vertex is not connected to any rule vertex, i.e. all the rules in derivation d are present in the extracted minimal grammar G_m.

The greedy approach chooses one derivation for each phrase, whose component rules are added to G_m. For each such extracted rule, we assign it the count of the bilingual phrase from which the rule was extracted and aggregate the count across all the phrase pairs. We then simply use the relative frequency estimation for computing the conditional probabilities of the rules.

5.7 Unary model-2: Model

As earlier, given a set of initial phrase pairs $\mathcal{X}$ as well as a prior over the grammars θ, our goal is to infer the posterior over grammars in the Bayesian framework. We describe our second unary model followed by the inference procedure using Variational Bayes.

As in unary model-1, we represent the generation of bilingual phrases from the grammar rules as a generative process, where the process first decides the type of derivation d to be either terminal ($z_d = \textsc{Term}$) or hierarchical ($z_d = \textsc{Hier-A1}$). It then identifies the constituent rules in the derivation to generate the phrase pair.

For a given phrase pair $x \in \mathcal{X}$, the probability of a derivation $d \in \phi_x$ can be expressed as:

$$P(d) \propto P(z_d) \prod_{r \in d} P(r|\mathcal{G}, \theta) \tag{5.8}$$

where r is a rule in grammar $\mathcal{G}$, and θ are grammar parameters (the vector of rule probabilities). We assume a Dirichlet prior over the parameters:

$$\theta \sim \text{Dirichlet}(\alpha_h p_0) \tag{5.9}$$

where α_h is the concentration hyperparameter, and p_0 is the base measure which we construct as follows. Let $x_r = \langle f, e \rangle$ denote the phrase-pair resulted from the lexical items in the right-hand-side of a translation rule r. This is depicted in the graphical representation in Figure 5.6.

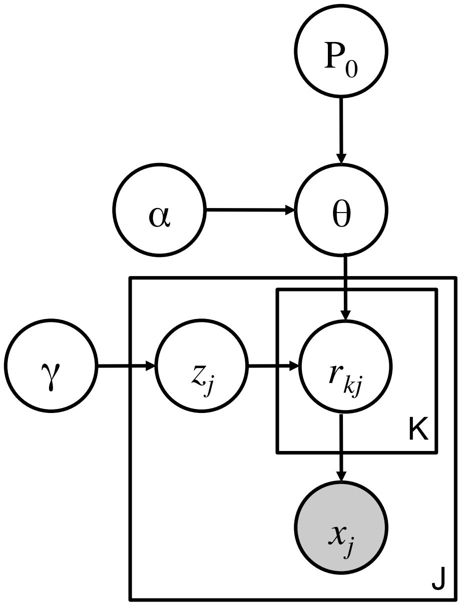


Figure 5.6: Graphical model representation for Unary model-2. The generative process first decides the derivation type z_j from a Bernoulli distribution parametrized by γ. It then generates the rules r_{kj} in the derivation by using a Dirichlet distribution θ with base measure P_0 and concentration parameter α. There are K rules in the derivation which yields the phrase-pair x_j.

There could be many different alignments a identified via learning the word alignments for different instances of x_r[7]. Define the forward *lf* alignment score to be (backward score *lb* is defined equivalently):

$$lf_{x_r} \propto \Big(\prod_{(m,n) \in a} p(e_n|f_m) \Big)^{\frac{1}{|a|}}$$

with a being the set of alignments for the lexical items x_r of rule r.

The base measure of a translation rule $p_0(r)$ is the arithmetic mean of the two alignment scores above. $p_0(r) \propto (lf_{x_r} + lb_{x_r})/2$.

Let l_x be the geometric mean[8] of the forward and backward alignment score over an initial phrase pair $x \in \mathcal{X}$, $l_x \propto \Big(lf_x lb_x\Big)^{\frac{1}{2}}$. We place a Beta$(l_x, 0.5)$ prior over the Bernoulli distribution that decides the derivation type z_d and this is normalized by the sum of lexical weights from all phrase pairs. The Beta prior prefers to consolidate a phrase pair fragment (within a larger phrase-pair) having a higher l_x as a single rule.

Notice that this is similar to the unary model-1 explained earlier, except for the following two differences, i) the unary model-2 uses a finite dimensional grammars (Dirichlet) as opposed to infinite dimensional grammars (DP) in unary model-1, and (ii) we employ Variational-Bayes inference for this model as opposed to Gibbs sampler inference in unary model-1. We now give a brief description of Variational inference and then explain our inference procedure in detail.

5.8 Unary model-2: Inference

Using Bayes' rule, we can express the posterior over the grammar $\mathcal{G}$ given the set of bilingual phrases $\mathcal{X}$ as: $P(\mathcal{G}|\mathcal{X}) \propto P(\mathcal{G})P(\mathcal{X}|\mathcal{G})$. Specifically, we are interested in the posterior over the grammar parameters θ and the latent derivations Φ given the data and the prior. Using Variational Bayes we assume the posterior to be factorized over θ and Φ resulting in the approximate posterior as:

$$p(\theta, \Phi|\alpha_h, p_0, \mathcal{X}) \approx q(\theta|\mathbf{u})q(\Phi|\pi)$$

where $\mathbf{u}$ and π are the parameters of the variational distributions.

The inference procedure is presented in Algorithm 3, where the parameters $\mathbf{u}^t$ and π^t are updated iteratively. Following our assumption of Dirichlet prior over grammar parameters, we initialize $\mathbf{u}^0 := \alpha_h p_0$, which is then updated using expected rule counts in subsequent iterations. The expected rule count can be written as:

[7] If there are multiple alignments for x_r (based on multiple initial phrase pairs), we take the union of these alignments as a.

[8] We use arithmetic mean for p_0 and geometric mean for l_x based on the unary model experiments presented earlier.

$$\mathbb{E}[r] = \sum_{d \in \phi_x} P(d|\pi^{t-1}, x) c_d(r) \tag{5.10}$$

where, $P(d|\pi^{t-1}, x)$ is the probability of the derivation d for the phrase pair x and $c_d(r)$ is the count of r in derivation d (the count is either 0 or 1).

Algorithm 3 Variational EM for learning Unary Hiero Grammar

```
Input: Init phrases $\mathcal{X}$ and base distribution $p_0$
Get prior distribution $\mathbf{u} = \{u_r = \alpha_h p_0(r) | r \in \mathcal{G}\}$
Set $\mathbf{u}^0 = \mathbf{u}$
for $t = 1, 2, \ldots$ do
  Estimate $\pi^{t-1}$:
    $\pi_r^{t-1} \leftarrow \exp\left(\psi(u_r^{t-1}) - \psi(\sum_r u_r^{t-1})\right)$
  for $x \in \mathcal{X}$ do
    for $d \in \phi_x$ do
      Compute $P(d|\pi^{t-1}, x)$ as in (5.11)
  for $r \in \mathcal{G}$ do
    Compute expected rule count $\mathbb{E}[r]$ using (5.10)
  Estimate posteriors $u_r^t$:
    $u_r^t \leftarrow u_r^0 + \sum_{x \in \mathcal{X}} \mathbb{E}[r]$
Output: Posterior distribution $\mathbf{u}^t$
```

The probability of a derivation in Equation 5.10 can be written in terms of l_x and π as:

$$P(d|\pi^{t-1}, x) \propto \begin{cases} \frac{l_x}{l_x + 0.5} \pi_r^{t-1} & \text{if } z_d = \textsc{Term} \\ \frac{0.5}{l_x + 0.5} \prod_{r' \in d} \pi_{r'}^{t-1} & \text{otherwise} \end{cases} \tag{5.11}$$

The probability of a derivation is normalized over all the derivations for a particular phrase pair. We fix α_h to be 0.5 in our experiments, which was manually set based on a small number of trials on development data. We run Variational Bayes for fixed number of iterations (10) and read off the grammar in the last iteration together with rule pseudo counts (the expected rule counts over e, f pairs). We then compute the probabilities $P(e|f)$ and $P(f|e)$ using relative frequency estimation over these pseudo counts similar to the estimation procedure in original Hiero.

5.8.1 VB Inference: Implementation Notes

Our model allows the unaligned source words to be attached at all possible positions in the derivation tree. This results in multiple interpretations of the unaligned words reflecting through large number of derivations, which include wider and richer rule contexts. This is analogous to the method used in (Galley et al., 2006) for context-rich syntactic translation models and we hope this to be useful in the Hiero models as well. In contrast the original Hiero grammar extraction restricts the unaligned words to be attached only to the top most position and so it can participate in just a single derivation.

To make VB inference practical, we need to efficiently enumerate all the derivations for a phrase pair such that they are consistent with the given word alignments. We use the decomposition tree algorithm proposed by (Zhang et al., 2008a) explained in Section 3.4.2.

5.9 Experiments

Language Pairs. We use three language pairs in our experiments: Arabic-English and English-Spanish (large bilingual data conditions), and Korean-English (small bilingual data condition). We retain the small data setting (Korean-English language pair) from the binary model experiments presented earlier in Chapter 3. This further helps us to verify if the compact models will have negative impact for the case where the training data is already limited. We use Arabic-English and English-Spanish language pairs for the large data setting and being similar language pairs they validate our hypothesis that unary models are sufficient for such languages. While we considered Chinese-English for our experiments in binary Hiero model, we drop this here for the reason that the unary models are not sufficient to capture the complex long-distance reordering requirements of these two diverse languages (Zollmann et al., 2008).

Corpora. Table 5.6 summarizes the statistics for the bilingual corpora used for these sets of experiments. For the language model, we use English Gigaword corpus (v4) for the Arabic-English and Korean-English translation tasks, and the WMT10 training data together with the UN data for the English-Spanish translation task and use 5-gram models for all language pairs.

We used the University of Rochester (Chung and Gildea, 2009) corpus for our Korean-English experiments without changing the tuning or test set splits, so our results are directly comparable to theirs. We also used the same rule-based morphological analyzer[9] as Chung and Gildea (2009) to segment the Korean side of the bitext.

SMT Models. We use Kriya with the standard features such as forward and reverse translation probabilities and lexical weights, phrase and word penalties, glue penalty and language model feature. For each experiment, we use MERT (Och, 2003) to optimize the feature weights on a tuning set, and evaluate using the corresponding optimal weights on the test set. To ensure robustness in

[9] http://nlp.kookmin.ac.kr/HAM/eng/main-e.html

Lang Pair	Dataset	Train/ Tune/ Test
Arabic-English	ISI web crawled corpus	1.1 M/ 1982/ 987
English-Spanish	WMT-10: no UN data	1.7 M/ 5061/ 2489
Korean-English	Univ of Rochester corpus	59218/ 1118/ 1118

Table 5.6: Corpus Statistics in # of sentences

Korean-English small data condition, we run MERT three times. The official NIST BLEU script[10] is used for computing the case-insensitive BLEU scores.

Evaluation. We compare our two translation grammar induction methods, based on variational Bayes (*VB*) and combinatorial optimization (*Greedy*), against the following grammars:

- *Original Hiero (binary).* The grammar as extracted by the heuristic rule extraction algorithm (Chiang, 2007) with two non-terminals
- *Original Hiero (unary).* A variant of the heuristic rule extraction algorithm restricted to unary grammar

We compare the model size and BLEU scores of the grammars induced by our approaches to the above two models for all the three language pairs. We also prune the VB grammar based on a count cutoff and decode using this compact grammar showing it to be competitive to the original Hiero models in terms of BLEU scores.

5.9.1 Experiments on Ar-En and En-Es

The VB inference is computationally prohibitive for Arabic-English and English-Spanish pairs due to the size of these datasets.[11] So, we filter the set of bilingual phrases (initial phrase pairs) for these corpora based on the frequency, and run our VB inference algorithm on the filtered set of initial phrase pairs[12]. We use threshold 3 for Arabic-English, and use two thresholds (10, 20) for English-Spanish.

For Arabic-English we compare our results with heuristic rule extraction method apart from three alternative approaches for pruning Hiero grammar. First we employ pruning based on fisher

[10] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13.pl

[11] We have subsequently implemented a distributed version of the inference using Map-Reduce framework and are able to experiment with them on full datasets. See Chapter 3 for details.

[12] As in the case of unary model-1, we add the *coverage* phrase pairs (those with non-decomposable source-target alignments) without the threshold limit to avoid OOVs (in training).

significance test (Yang and Zheng, 2009) to reduce the Hiero model. We also provide results for the pattern-based filtering (Iglesias et al., 2009) that filters the grammar extracted by the original rule extraction algorithm based on certain patterns that are found to be least useful in translation or in improving the quality. And finally, we apply a fixed count cut-off on the pseudo counts of the grammar rules and eliminate all rules having pseudo counts fewer than 1.0 (we call this parameter *mincount*). This is somewhat similar to the pruning of hierarchical rules (Zollmann et al., 2008) based on a threshold, except that here we prune both lexical and hierarchical rules. Table 5.7 shows the BLEU scores and grammar sizes for Arabic-English.

We first note that the unary grammar achieves comparable performance to that of the binary heuristic grammars and we observe the same for the other two language pairs as well. This shows that unary models does not reduce the expressive power or reordering ability and hence our Bayesian model is not handicapped by using unary. It also reduces the model size by 23% compared to the binary Hiero model.

Fisher significance pruning results in a slight drop of about 0.3 BLEU points. However it does not reduce the grammar size beyond the marginal 3.2% and this is because least frequent initial phrase-pairs are not considered in this thresholded setting. We also apply significance pruning for Korean-English- where we consider all phrase-pairs without any thresholding, and as we show later 5.9.2 it leads to substantial savings in the model sizes but with reduced BLEU scores. Pattern-based filtering reduces the model size by 8-26% compared to their respective baseline Hiero grammars; however the BLEU score drops by 0.7 suggesting that the blanket filtering, based purely on patterns, might actually be harmful. Using a count cutoff also significantly reduces the size of the grammars, but incurs a 1.5 point drop in the BLEU scores. The Greedy approach for combinatorial optimization worsens the BLEU score further but has a smaller model size compared to the filtering and count cutoff methods.

However, the trade-off relationship changes with our VB approach as we note that both the full VB grammar and the one pruned with mincount 1.0[13] perform to the same level as the original Hiero models (without pruning). The full VB grammar has slightly larger model size than its equivalent original Hiero (unary) model and this is due to the additional rules generated from the unaligned source words, which are attached to all possible positions in the derivation tree as we mentioned earlier. The pruned VB grammar substantially reduces the size of the grammars with the effective saving of 40.8% compared to the binary Hiero model. The model size can be reduced further by

[13] We explored a range of mincount values (1, 1.5 and 2) on tuning-set and present the test-set numbers that are interesting.

Grammar	**Bleu**	**Model Size**	**Speed** *(sent/min)*
Binary Hiero			
Heuristic Hiero	33.11	4.82	3.62
- Yang and Zheng (2009)	32.84	4.70	3.73
- Iglesias et al. (2009) filtered	32.52	3.59	4.99
- Pruned (mincount 1.0)	31.68	2.24	5.57
Unary Hiero			
Heuristic Hiero	33.08	3.71	4.43
- Yang and Zheng (2009)	32.80	3.59	4.87
- Iglesias et al. (2009) filtered	32.40	3.43	5.36
- Pruned (mincount 1.0)	31.64	2.28	5.70
Greedy Approach	31.20	1.88	6.53
Variational Bayes	33.13	3.75	4.62
- Pruned (mincount 1.0)	**33.05**	**2.90**	**4.87**
- Pruned (mincount 1.5)	32.44	1.84	5.33

Table 5.7: Arabic-English (Threshold-3): Results. Model sizes is in millions. **Boldface** indicate the best setting of high BLEU, model size and decoding speed.

pruning the VB grammar with a slightly larger mincount of 1.5 and this reduces the BLEU score modestly. This is due to the fact that VB inference produces a sharp approximation to the posterior distribution, so most of the expected counts (pseudo counts) fall below the threshold when it is slightly increased. VB provides a better trade-off between the translation quality and the model size compared to all the competing approaches. Finally we also note that the compact grammar results in 10-35% faster decoding (*speed* column in Table 5.7) for the pruned VB grammar compared to the original Hiero models.

We see a similar trend for the threshold-10 setting in English-Spanish experiment as seen in Table 5.8. Both full and pruned VB grammars achieve same translation performance as binary Hiero grammar but with 17.8% reduction in the grammar size. The threshold-20 setting for En-Es offers an interesting insight about the superiority of parameter estimation by VB. The pruned VB model (mincount 1) improves over the two full Hiero models by over 0.9 BLEU points, although it uses a marginally large grammar. The *italicized* BLEU scores indicate statistically significant improvement over both unary and binary Hiero grammars, computed using bootstrap resampling with $\alpha = 0.05$. We hypothesize this to be due to improved parameter estimation using VB (see Section 5.9.3). Finally, the pruned VB grammar results in faster decoding compared to the binary

Grammar	**Threshold-10**		**Threshold-20**	
	BLEU	**Model Size**	**BLEU**	**Model Size**
Binary Hiero				
Heuristic Hiero	26.72	3.14	24.95	1.95
Unary Hiero				
Heuristic Hiero	26.63	2.91	24.94	1.93
Greedy Approach	25.51	1.95	23.88	1.9
Variational Bayes	26.58	2.91	*25.65*	2.31
- Pruned (mincount 1.0)	**26.55**	**2.58**	***25.86***	**1.97**

Table 5.8: English-Spanish: Results. Model sizes is in millions. **Boldface** indicate the best setting of high BLEU and model size.

Hiero models by 30% and 8% in both cases.

5.9.2 Experiments on Korean-English

Table 5.9 shows the BLEU scores and the grammar sizes for the different rule extraction approaches and we report the testset BLEU for the MERT run achieving the best BLEU in the tuning set. We note that the BLEU score for the binary Hiero and VB models are higher than the 7.27 score obtained by Chung and Gildea (2009).

Interestingly, the greedy approach performs relatively better in this setting even though the BLEU scores of the other models are statistically significant than greedy approach. As we noted earlier significance pruning reduces the grammar by 70%, but it also hurts the translation performance as seen from the BLEU scores. This is also consistent with the more recent work by Zens et al. (2012) comparing different phrase-table pruning techniques applied to phrase-based models. Significance-based pruning was shown to perform poor compared to entropy-based pruning, even though they were better than probability-based pruning.

Unlike other languages, the Hiero unary and VB grammar gets a BLEU of 7.25 that is noticeably below the score of the heuristic binary grammar. However pruned VB grammars achieve higher BLEU scores possibly by reducing the over-generation and search error. The VB grammar pruned with threshold mincount[14] 0.25 is 57.6% slimmer than the heuristic binary grammar. While the

[14] For Ko-En, we experimented with different mincount values 1.0, 0.5, 0.25 and 0.1 on the tuning-set and chose the setting (0.25) that got the highest tuning-set BLEU.

Grammar	**BLEU**	**Model Size**	**Speed** *(sent/min)*
Binary Hiero			
Heuristic Hiero	7.53	2.64	3.82
- Yang and Zheng (2009)	6.85	0.75	5.89
Unary Hiero			
Heuristic Hiero	7.25	1.83	4.85
- Yang and Zheng (2009)	6.93	0.56	5.97
Greedy Approach	7.04	1.27	5.25
Variational Bayes	7.28	2.30	4.73
- Pruned (mincount 0.1)	7.40	2.11	4.83
- Pruned (mincount 0.25)	***7.51***	**1.12**	**5.41**

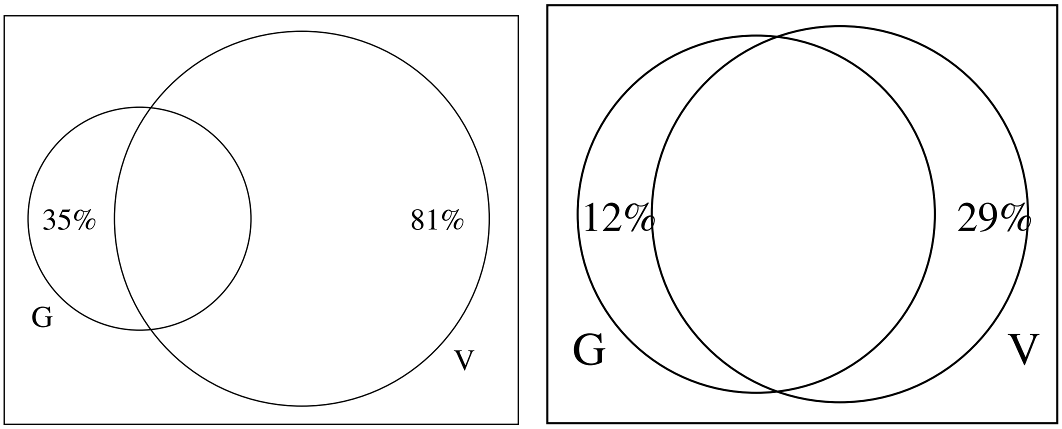
Table 5.9: Korean-English: Results. Model sizes is in millions. **Boldface** indicate the best setting of high BLEU, model size and decoding speed.

BLEU score of mincount 0.1 is closely behind the 0.25 setting (as also in the tuning-set); it only reduces the model size by 20.1%. Finally we also note that the BLEU score of mincount 0.25 is statistically significant ($\alpha = 0.1$) than the heuristically extracted baseline unary grammar.

5.9.3 Analysis

In this section, we investigate i) the reason for poor performance of the greedy approach and ii) why our VB inference performs to the same level as the Hiero rule extraction algorithm even after pruning. While this analysis was performed for the Ar-En, we find similar trend to hold for En-Es and Ko-En as well.

We first analyze the differences in the grammars in terms of the terminal and hierarchical rules and particularly look at the grammars generated by the VB-pruned (mincount 1.0) and that of greedy algorithm. The Venn diagrams in Figure 5.7 plots the overlap in the two rule types in either grammars. While 65% of hierarchical rules in greedy grammar (G) are also found in the pruned VB grammar (V), only 19% of the hierarchical rules in the VB grammar are included by the greedy approach. It suggests that the greedy grammar is missing crucial hierarchical rules compared to the VB grammar severely limiting its ability, for instance in reordering the phrases during decoding. Though the greedy grammar also misses 29% of terminal rules found in VB grammar, its impact is minimal and as we notice in the N-best list, it uses smaller terminal rules and composing them with



(a) Hierarchical

(b) Terminal

Figure 5.7: Venn diagrams of hierarchical and terminal rules in Greedy (G) and VB-pruned (V) grammars for Arabic-English (rows 4 and 9 in Table 5.7). The numbers indicate the % of unique rules.

glue rules (this is also because greedy approach typically prefers shorter terminal rules over longer ones). We found identical trend between greedy (G) and Hiero unary (H) grammars as well and this suggests the poor performance of the greedy approach to be mainly due to poor *model selection*.

We also analyze the percentage of shared terminal and hierarchical rules for the heuristic (H) and VB-pruned (V) grammars but they had high overlap (more than 90%). This clearly shows that the better performance of VB grammars is not only due to its ability in model selection, but also in better parameter estimation compared to the original Hiero rule extraction algorithm. Particularly, the VB is learning a sharper distribution by moving probability mass from poor translations towards rules capturing high quality translations. Further, the high overlap (not shown due to space limitation) of the VB-pruned grammar with the heuristic Hiero grammar, indicate that the additional rules resulting from the multiple interpretations of the unaligned source words are not particularly helpful for Hiero models, unlike in syntactic models (Galley et al., 2006).

As noted earlier the Hiero rule extraction uniformly distributes the weight to the rules extracted from an initial phrase pair. These locally distributed weights are aggregated globally for each rule as these rules can be extracted from other phrase pairs as well. Therefore, it does not allow subsequent re-weighting of the rules based on the global frequency of rules across the entire set of phrase pairs. In contrast, our VB inference naturally allows the rule pseudo counts to be updated at each iteration based on their global usage, thus pushing probability mass from low quality rules to high quality ones.

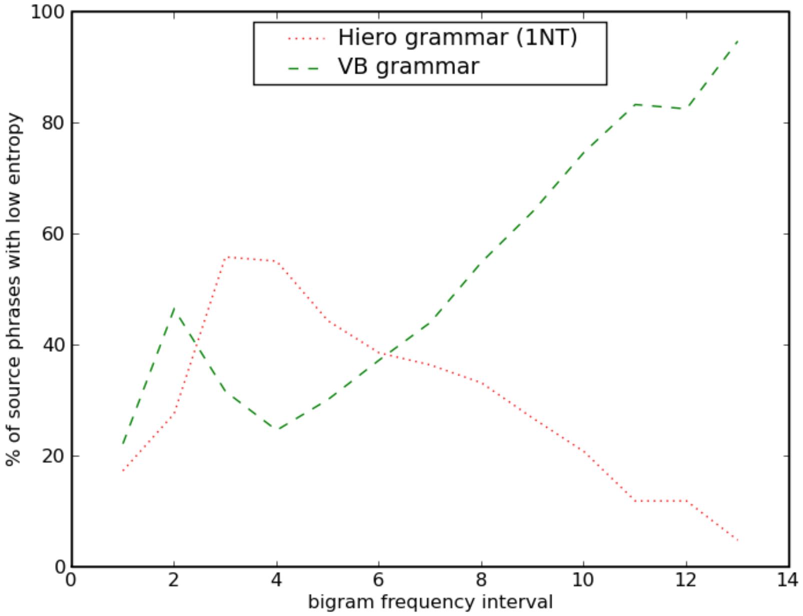


Figure 5.8: Entropy of the bigram source phrases of different frequencies in intervals 1-13. Intervals 1 and 2 correspond to $[2, 5)$ and $[5, 10)$ (see text for details).

In order to study this quantitatively, we analyze the entropy of the source phrases (in terminal rules) that are found in both heuristic and VB grammars. For better control in the experiment, we restrict ourselves to bigram source phrases and group them into bins based on their frequency in the initial phrase pairs. We consider frequencies in 13 different intervals spaced at frequencies 2, 5, 10, 25, 50, 125, 250, 500, 1K, 2.5K, 5K, 10K, 20K and *over*. For example the two initial intervals include source phrases having frequencies $[2, 5)$ and $[5, 10)$ respectively.

For each interval, we compute the entropy of rules for unique source phrases using the conditional probability $P(e|f)$. We compute the entropy for the source phrases that are found in both heuristic and VB grammars and aggregate this across all the source phrases within an interval. We compute the $\%$ of source phrases in VB grammar having lower entropy compared to the heuristic grammar and vice versa. Figure 5.8 plots the $\%$ of source phrases having lower entropy for heuristic and VB grammars at different intervals. For most of the intervals a large percent of source phrases in VB grammar has low entropy compared to the Hiero grammar, while for 3 intervals the percentage of source phrases in Hiero grammar exceed that of VB grammar. This clearly shows that VB inference produces a sharp distribution across different frequency ranges, for both frequent and rare phrases in the training data. We also observe similar trend for trigram source phrases.

Next, we particularly examine the ranking of the translation options for most frequent source phrases in both grammars. We consider 100 most frequent source side n-grams ($n = \{1, 2, 3\}$) in the training data and compare the ranking of the translation options preferred by the two grammars (We again use $P(e|f)$ as earlier). Comparing the highest ranking translation option for these source phrases, we find that both grammars agree on the same target translation for 88.5% of the n-grams. Additionally, in over 73% of the source phrases agreeing on the same target translation, rules of the VB grammar had higher probability than the corresponding heuristic grammar rules.

5.10 Compact Models with Binary Hiero Grammars

So far in this chapter, we explored compact models arising from i) unary grammars and ii) applying pruning to unary grammars and found that the model size could be substantially reduced without impacting the translation performance. We now seek to experiment if the pruning could also applied to our binary Hiero model presented earlier in Section 3.3 yielding substantial reduction in the traditional Hiero setting.

We applied threshold pruning to the traditional Hiero grammar extracted by our binary model as well as the original heuristic model for the language pairs experimented in Section 3.5.1. Table 5.10 shows the results for the pruned grammars, where we prune the rules having expected count below a mincount threshold. We present the results for specific mincount settings based on our experiments on the held-out tuning set for each language pair.

- **Retains score with smaller grammar**: The pruned grammars retain the performance of the full grammar, even while using just 18% of the complete model.
- **Higher reduction for large dataset**: Variational inference reduces the model size over 80% for the large corpora. While this is similar to the findings of Johnson et al. (2007) and that of the pruning strategies mentioned above; the question of whether an intelligent model selection strategy can yield higher BLEU scores is still open.
- **Faster decoding**: The compact grammars naturally result in faster decoding and we observed up to 20-30% speedup in the translation including the time spent for loading the model.

Our goal in this chapter is not about learning compact grammars, nevertheless, we pruned the learned grammar for some threshold mincount and compare its performance of the full VB grammar. The results summarized in the lower half of the Table 5.10 shows that the pruned model achieves

	Ko-En	**Ar-En**	**Zh-En**
Heuristic grammar - *Unpruned*			
Model size	2.7	331.8	471.9
BLEU	7.18	37.82	28.58
VB grammar - *Unpruned*			
Model size	2.67	331.6	471.7
BLEU	**7.68**	*37.76*	*28.40*
VB grammar - *Pruned*			
Pruning mincount	0.25	1.0	1.0
Model size	1.65	58.9	87.3
% Reduction	38.2%	82.2%	81.5%
BLEU	**7.64**	*37.58*	*28.45*

Table 5.10: Comparison of the full and pruned Variational-Bayes grammars. Model sizes are in millions of rules. Mincount implies the rule count threshold used for pruning the full VB grammar. BLEU scores that are statistically indistinguishable from the heuristic baselines are *italicized* and those that are statistically significant are in **boldface**.

substantial reduction in the model size. The model size reduction is higher for the binary grammar especially in the two large-data settings (over 80% reduction compared to the 57% reduction for unary grammars). The binary grammar achieves about 38% reduction for the Korean-English data.

This shows that the pruning could also effective for our binary Hiero model in order to achieve substantial reduction in the model size without sacrificing translation quality for different language pairs. In this context, we compare the effects of pruning the heuristic and VB grammars in Figure 5.9 for Chinese-English. For the same mincount threshold of 1 as the best performing VB setting, the BLEU score of the heuristic grammar drops by over 1 point. However this setting prunes over 99% of the arity-1 and arity-2 rules even while it retains all the terminal rules. This is primarily because of the way the heuristic method estimates rule counts by uniformly distributing the weight among all the rules. The terminal rules are sufficient for coverage but does not capture long distance movements; and the lack of arity-1 and arity-2 rules further restrict the reordering ability of the model. We have to substantially lower the mincount threshold to 0.05, in order to get performance comparable to the pruned VB grammar setting. Interestingly, this uses about 7M more rules (13M more arity-1 rules, but 6M fewer arity-2 rules) than VB-Pr (1.0), but its BLEU score is marginally lower than the latter. This could be ascribed to the missing arity-2 rules, which could be crucial for certain long-distance reordering.

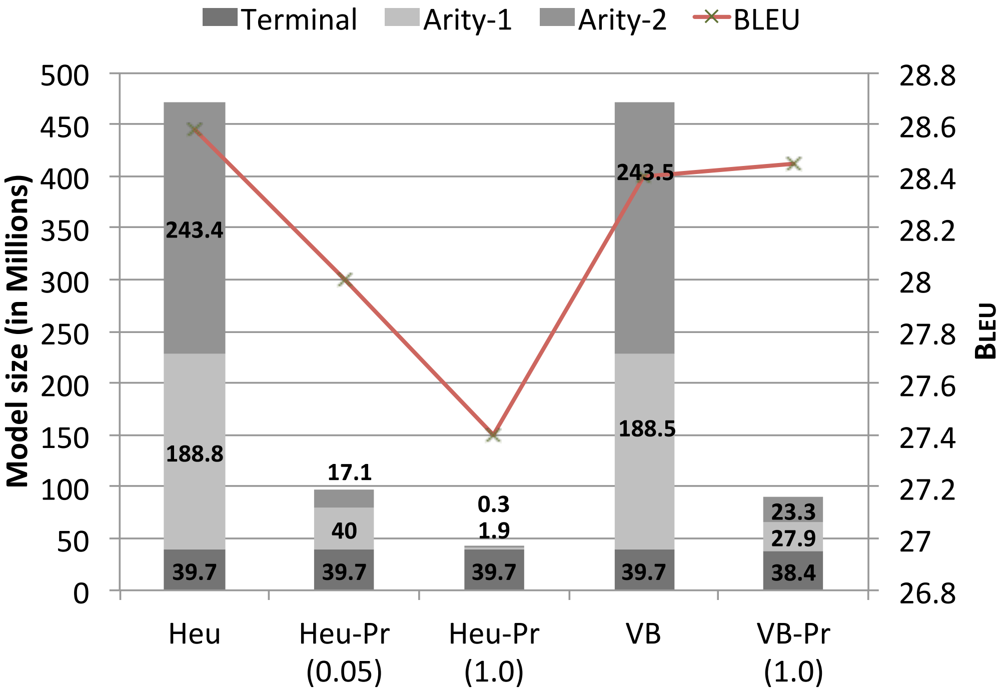


Figure 5.9: Cn-En: Model sizes and Bleu for different grammars. The pruned models are identified by the suffix 'Pr', whose mincount is shown in the brackets. The y-axis on the left marks the model sizes and that on the right denotes Bleu. The numbers in the stacked bars denote the # of rules (in millions) for the corresponding rule type.

5.11 Related Work

Some earlier works have focussed on reducing the Hiero grammar size by eliminating rule redundancies in some form such as by discarding rules that can be obtained by monotonically composing the smaller rules (He et al., 2009) or by filtering the grammar, based on certain patterns of hierarchical rules in which the useful patterns were identified in a greedy fashion (Iglesias et al., 2009). Yang and Zheng (2009) applied the Fisher's exact significance test for pruning the translation model, which has been earlier used for phrase-based models (Johnson et al., 2007). These approaches achieve up to 70% reduction in the phrase table by pruning the rules inferred by the Hiero grammar. However unlike our models, these pruning approaches negatively impacted the translation quality (Bleu), sometimes by a substantial margin that were statistically significant.

The idea of Shallow-n grammars (de Gispert et al., 2010a) takes an orthogonal direction for controlling the over-generation and search space issues in Hiero decoder by restricting the degree of nesting allowed for Hierarchical rules. Instead of allowing unlimited nesting of rules in Hiero, the shallow-n grammars introduce additional non-terminals and restrict the number of hierarchical nestings (equivalently height of the derivation tree) to n. In our earlier work, we generalized this by introducing shallow-n decoding (Sankaran and Sarkar, 2012) and also proposed a BITG style (Saers et al., 2009) reordering glue rule for improving the reordering in the shallow decoding framework. Our shallow-n decoding approach obviates the need for additional non-terminals and some of the manually tuned parameters that were used in shallow-n grammars. Both approaches showed that the shallow decoding could be effective for certain close language pairs (de Gispert et al., 2010a; Sankaran and Sarkar, 2012).

5.12 Compact Hiero Grammars: Summary

We focused on extracting compact grammars for Hiero that would perform to the same extent as full grammars and explored two distinct strategies towards this. First we considered a restricted form of Hiero grammar with unary SCFG as a means of reducing model size. We then applied the traditional pruning technique to further reduce the size of the extracted grammar. Finally we also show that the pruning could be applied for the binary model presented earlier in Chapter 3.

We presented two different approaches for extracting compact grammars for unary Hiero. Our approaches are similar in spirit to the heuristic extraction algorithm (Chiang, 2007) in that we extract rules from the word aligned initial phrase-pairs. We had two motivations in this work, i) to extract compact grammars for Hiero in order mitigate some issues associated with larger Hiero models (such as overgeneration and increased decoder complexity) and ii) to explore alternatives for the heuristic extraction algorithm used in Hiero.

Our combinatorial optimization approach formulates the problem as a variant of the *minimum set cover* problem over a novel tripartite graph representation. We proposed a greedy method for this, which extracted a minimal grammar even though it suffered BLEU losses. The problem is in the greedy search formulation, which only looks at the number of times each rule is used in derivations.

In the second approach we used Bayesian framework similar to the binary model proposed in Chapter 3 but adapted for the unary grammar. We described two approaches employing different inference techniques for extracting compact unary grammars. As in binary Hiero model, we employed a novel parametric prior that prefer good quality phrase-pairs to be memorized directly and opting to break up poor/ longer phrase-pairs for better generalization. The Gibbs sampler based unary model-1 suffered slight degradation in the performance even though it was able to extract reasonably good grammar with better parameter estimates.

We introduced a slightly different model called unary model-2 and employed Variational Bayes for inference as opposed to the Gibbs sampler for unary model-1. We show that the performance of unary model-2 is indistinguishable from binary Hiero models when the source and target languages do not have much diversity, as we showed for variety of language pairs. It also achieves statistically significant BLEU score improvement for resource poor and small data settings. Our experiment showed the unary model-2 to retain the same BLEU score as the heuristically extracted binary Hiero model. The results for compact grammar experiments are summarized in Figure 5.10 for Arabic-English (threshold-3) dataset.

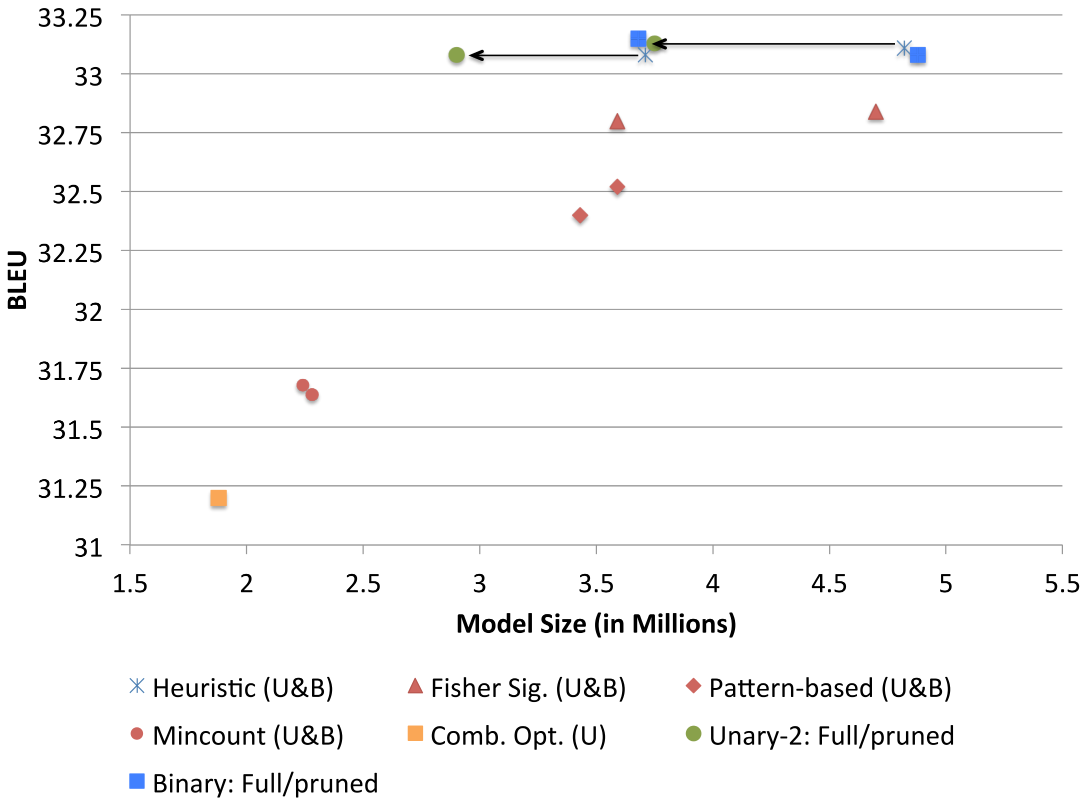


Figure 5.10: Compact Grammars Summary: Model size vs. BLEU for Arabic-English dataset (Threshold-3 setting). The heuristic baseline and different pruning approaches are shown for both unary (U) and binary (B) Hiero grammars. We fixed the mincount value to 1.0 for the ’mincount’ pruning and the pruning of our Bayesian models for fair comparison. The results for the combinatorial optimization approach is shown only for unary grammars. The arrows indicate the model size reduction compared to the Heuristic extraction baselines corresponding to unary and binary grammars.

We further reduced the grammar by applying traditional pruning to the unary and binary models. The pruned models are competitive to the heuristic rule extraction algorithm in terms of BLEU scores at the same time resulting in a 17.8% to 57.8% reduction in the model sizes across different language pairs. In contrast the pruning reduces the BLEU scores, when applied to the heuristically extracted grammars.

Finally the pruning of binary models for full training corpora (no initial phrase-pair thresholding) resulted in a model size reduction ranging between 38% and 82% (see Section 5.10) for different language pairs.

Chapter 6

Multi-Metric Optimization

So far in this thesis we focussed on the training part of the Hiero machine translation pipeline, specifically proposing new methods for learning alignments and Hiero translation grammars. We now shift our attention to the parameter optimization part of the translation pipeline. The goal of the parameter optimization step is find optimal weights for the different features in the MT system by tuning them to maximize the translation performance on a held-out set. This ensures that the MT system can generalize well for some unseen text in the same domain as the held-out set.

6.1 SMT Parameter Optimization

As mentioned above, tuning algorithms are used to find the weights for a statistical machine translation (MT) model by maximizing the translation performance (or equivalently minimizing the error) with respect to a single MT evaluation metric. The tuning process improves the performance of an SMT system as measured by this metric; with BLEU (Papineni et al., 2002) being the most popular choice. Minimum error-rate training (MERT) proposed by Och (2003) was the first approach in MT to directly optimize an evaluation metric. Several alternatives now exist: MIRA (Watanabe et al., 2007; Chiang et al., 2008), PRO (Hopkins and May, 2011), linear regression (Bazrafshan et al., 2012) and ORO (Watanabe, 2012) among others.

However these approaches optimize towards the best score as reported by a single evaluation metric. MT system developers typically use BLEU and ignore all the other metrics. This is done despite the fact that other metrics model wide-ranging aspects of translation: from measuring the translation edit rate (TER) in matching a translation output to a human reference (Snover et al., 2006), to capturing lexical choices in translation as in METEOR (Lavie and Denkowski, 2009) to modelling semantic similarity through textual entailment (Padó et al., 2009) to RIBES, an evaluation metric that pays attention to long-distance reordering (Isozaki et al., 2010). While some of these metrics such as TER, METEOR are gaining prominence, BLEU enjoys the status of being the *de facto* standard tuning metric as it is often claimed and sometimes observed that optimizing with BLEU produces better translations than other metrics (Callison-Burch et al., 2011).

The gains obtained by the MT system tuned on a particular metric do not improve performance as measured under other metrics (Cer et al., 2010), suggesting that over-fitting to a specific metric might happen without improvements in translation quality. In this chapter of the thesis we propose a new tuning framework for jointly optimizing multiple evaluation metrics.

Pareto-optimality is a natural way to think about multi-metric optimization (MMO) and this was recently explored in the Pareto-based Multi-objective Optimization (PMO) approach (Duh et al., 2012). PMO provides several equivalent solutions (parameter weights) having different trade-offs between the different MT metrics. In Duh et al. (2012) the choice of which option to use rests with the MT system developer and in that sense their approach is an *a posteriori* method to specify the preference (Marler and Arora, 2004).

In contrast to this, our tuning framework provides a principled way of using the Pareto optimal options using *ensemble* decoding (Razmara et al., 2012). We also introduce a novel method of *ensemble tuning* for jointly tuning multiple MT evaluation metrics and further combine this with the PMO approach (Duh et al., 2012). We also introduce three other approaches for multi-metric tuning and compare their performance to the ensemble tuning. Our experiments yield the highest metric scores across many different metrics (that are being optimized), something that has not been possible until now.

Our ensemble tuning method over multiple metrics produced superior translations than single metric tuning as measured by a post-editing task. HTER (Snover et al., 2006) scores in our human evaluation confirm that multi-metric optimization can lead to better MT output.

6.2 Related Work

In grammar induction and parsing Spitkovsky et al. (2011), Hall et al. (2011) and Auli and Lopez (2011) have proposed multi-objective methods based on round-robin iteration of single objective optimizations.

Research in SMT parameter tuning has seen a surge of interest recently, including online/batch learning (Watanabe, 2012; Cherry and Foster, 2012), large-scale training (Simianer et al., 2012; He and Deng, 2012), and new discriminative objectives (Gimpel and Smith, 2012; Zheng et al., 2012; Bazrafshan et al., 2012). However, few works have investigated the multi-metric tuning problem in depth. Linear combination of BLEU and TER is reported in Zaidan (2009), Dyer et al. (2009) and Servan and Schwenk (2011); an alternative is to optimize on BLEU with MERT while enforcing that TER does not degrade per iteration (He and Way, 2009). Studies on metric tunability (Liu et al., 2011; Callison-Burch et al., 2011; Chen et al., 2012) have found that the metric used for evaluation may not be the best metric used for tuning. For instance, Mauser et al. (2008) and Cer et al. (2010) report that tuning on linear combinations of BLEU-TER is more robust than a single metric like WER.

The approach in Devlin and Matsoukas (2012) modifies the optimization function to include traits such as output length so that the hypotheses produced by the decoder have maximal score according to one metric (BLEU) but are subject to an output length constraint, e.g. that the output is 5% shorter. This is done by rescoring an N-best list (forest) for the metric combined with each trait condition and then the different trait hypothesis are combined using a system combination step. The traits are independent of the reference (while tuning). In contrast, our method is able to combine multiple metrics (each of which compares to the reference) during the tuning step and we do not depend on N-best list (or forest) rescoring or system combination.

Duh et. al. Duh et al. (2012) proposed a Pareto-based approach to SMT multi-metric tuning, where the linear combination weights do not need to be known in advance. This is advantageous because the optimal weighting may not be known in advance. However, the notion of Pareto optimality implies that multiple "best" solutions may exist, so the MT system developer may be forced to make a choice after tuning.

These approaches require the MT system developer to make a choice either before tuning (e.g. in terms of linear combination weights) or afterwards (e.g. the Pareto approach). Our method here is different in that we do not require any choice. We use *ensemble decoding* (Razmara et al., 2012) (see sec 2.5) to combine the different solutions resulting from the multi-metric optimization, providing an elegant solution for deployment. We extend this idea further and introduce *ensemble tuning*, where the metrics have separate set of weights. The tuning process alternates between ensemble decoding and the update step where the weights for each metric are optimized separately followed by joint update of metric (meta) weights.

6.3 Multi-Metric Optimization

In statistical MT, the multi-metric optimization problem can be expressed as:

$$w^* = \arg\max_w g\Big([M_1(H), \ldots, M_k(H)] \Big) \tag{6.1}$$
$$\text{where } H = \mathcal{N}(\mathbf{f}; w)$$

where $\mathcal{N}(\mathbf{f}; w)$ is the decoding function generating a set of candidate hypotheses H (say N-best list) based on the model parameters w, for the source sentences $\mathbf{f}$. For each source sentence $f_i \in \mathbf{f}$ there is a set of candidate hypotheses $\{h_i\} \in H$.

$M_1, \ldots, M_k$ denote the target evaluation metrics being optimized and $M_i(H)$ is a vector of scores for H according to i^{th} metric. The goal of the optimization is to find the weights that leads to hypotheses that are better in the multi-metric space defined by $M_1, \ldots, M_k$. The function $g(.)$ represents different ways of combining the target metrics, for example Pareto-frontier.

It should be noted that the above formulation abstracts the idiosyncrasies of the underlying optimization algorithm (for example using PRO for optimization would involve a separate step for generating a candidate pool from the N-best hypotheses) and instead provides a optimization template, which could be modified according to our definition of $g(.)$. We introduce four methods based on the above formulation and each method uses a different instantiation of $g(\cdot)$ function for combining different metrics and we compare experimentally with existing methods.

6.3.1 Pareto Optimality

As we noted earlier, the notion of *Pareto optimality* (or *Pareto efficiency*) is a natural way of encoding the different possibilities in multi objective optimization scenarios and we briefly describe this here. Formally the Pareto efficiency is an equilibrium state in which no further improvements are possible for a player without penalizing some other player.

As an illustration consider the two-dimensional scatter plot of two metrics M_1 and M_2 shown in Figure 6.1. It shows a toy N-best list (H) consisting of 10 candidates plotted for their scores in two metrics. Here the hypothesis in point $(0.3, 0.75)$ *dominates* the one at $(0.2, 0.6)$ as the former point is better in both dimensions. We thus say that a hypothesis $h_1 \in H$ dominates some other hypothesis $h_2 \in H$, if $M(h_1) \geq M(h_2)$ for all metrics and that $M(h_1) > M(h_2)$ for at least one metric k. Following Duh et al. (2012) we express this as $M(h_1) \rhd M(h_2)$.

A candidate $h^* \in H$ is then said to be Pareto optimal if and only if there exist no other candidate $h \in H$ such that $M(h) \rhd M(h^*)$. A candidate $h^* \in H$ is said to be weakly Pareto optimal if there

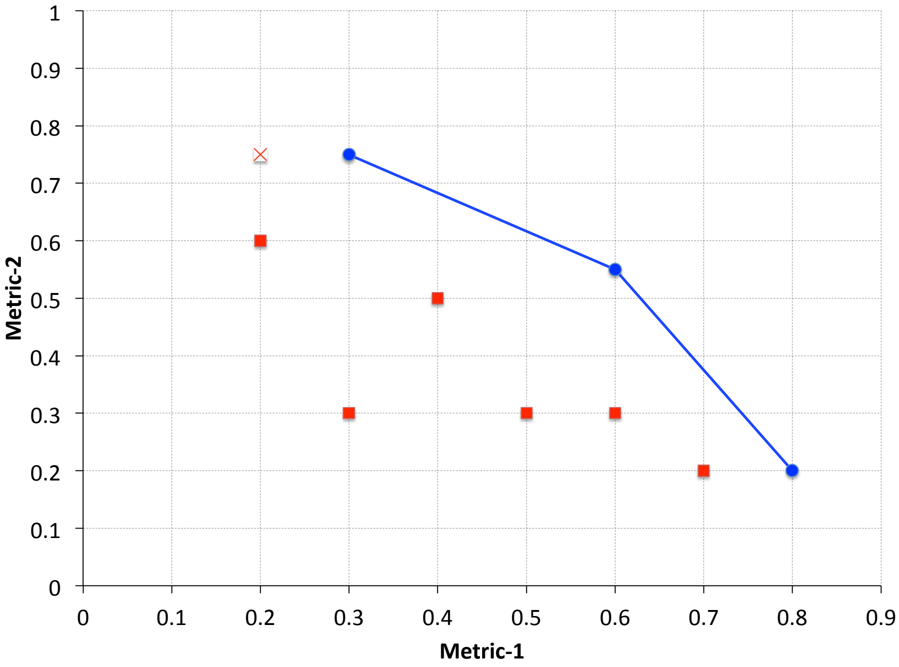


Figure 6.1: Illustration of Pareto optimality in MMO for two metrics for a toy list of 10-best candidates. Each point in the plot marks the score of the candidate in two metrics M_1 and M_2. The (blue) circles indicate Pareto optimal candidates; the (red) squares denote the non-Pareto optimal candidates and the candidate marked by $\times$ (in red) marks a weakly Pareto optimal candidate. The Pareto frontier is indicated by the lines connecting the Pareto optimal candidates.

is no other candidate $h \in H$ such that $M(h) > M(H^*)$. The point $(0.2, 0.75)$ is a weakly Pareto optimal point for this toy N-best list.

The Pareto frontier is defined to be the set of Pareto optimal candidates. Thus there can not be any other candidate in the N-best list that *dominates* any of the points on the Pareto frontier. In the machine translation setting, we typically deal with a set of top scoring candidates (N-best list in this work) and not with the exhaustive set of translations that are possible for the given model. For this reason, the Pareto optimal candidates in the N-best list are said to define an *approximate Pareto frontier* and not the true frontier. However, we abuse the terminology and use the two terms interchangeably in our work.

6.3.2 Linear Combination: Baseline MMO

As a baseline multi-metric optimization method, one could simply combine two or more metrics into a single metric by taking their weighted sum/ average. Such a function

$$g_{\text{wavg}}(\cdot) = \sum_k \lambda_k M_k(H)$$

would result in a linear combination of the metrics being considered and the tuning subsequently seeks to optimize this new joint metric.

In fact this has been explored earlier (Cer et al., 2010; Servan and Schwenk, 2011), where the objective was to minimize the joint metric: $(\text{TER} - \text{BLEU})/2$. However this approach does not provide a natural way of learning the meta-weights and they are usually fixed by the user *a priori*.

Alternately, one can tune the weights manually by trial and error method, based on its performance on a held-out devset for different settings of meta-weights as suggested by Duh et al. (2012). Considering the case of two-metric optimization as an example, the linear combination tuning can be repeated with five different weight settings, *viz.* $(0, 1)$, $(0.3, 0.7)$, $(0.5, 0.5)$, $(0.7, 0.3)$ and $(1, 0)$ and then choosing the optimal setting *post-hoc*. However, this approach would become impractical as the number of settings grow exponentially when tuning with three or more metrics. Further, the linear combination is limited in finding Pareto-optimal solutions and it fails to satisfy the necessary condition of finding *all* such solutions, irrespective of the values assigned to meta-weights λ_i (Duh et al., 2012).

6.3.3 PMO Ensemble

Pareto-based multi-objective (PMO-PRO) seeks to maximize the number of points in the Pareto frontier of the metrics considered (Duh et al., 2012). In this case, $g(\cdot)$ can be considered as a function f computing the Pareto frontier given by:

$$g_{\text{PMO}}(\cdot) = \underset{1,\ldots,k}{f}\Big(\lambda_k M_k(H)\Big)$$

The PMO-PRO is illustrated in Algorithms 4 and 5 for the case of tuning with two metrics M_1 and M_2. Given the held-out tuning set $(\mathbf{f}, \mathbf{e})$ along with initial feature weights and the meta-weights for the metrics, the main routine (Algorithm 4) runs PRO for a fixed number iterations (line 4). At each iteration j the held-out tuning set is decoded (line 5) with the current set of weights w^j tuning set followed by the optimization step (line 6) that finds new weights for the next iteration.

Algorithm 4 PMO-PRO Algorithm

```
Input: Tuning set $(\mathbf{f}, \mathbf{e})$;
          Metrics $M_1$ and $M_2$;
          Initial feature weights $w_{init}$ and
          Meta-weights $\lambda_1 = (0.0, 1.0), \lambda_2 = (0.3, 0.7), \ldots, \lambda_5 = (1.0, 0.0)$
for $\lambda_i$ such that $i = 1, \ldots |\lambda|$ do
    Initialize weights for the first iteration: $w^{(1)} \leftarrow w_{init}$
    for $j = 1, \ldots$ do
        Decode the tuning set f to get $N$-best hypotheses
          $H = \mathcal{N}(\mathbf{f}; w^{(j)})$
        Call optimizer routine to optimize feature weights $w$
          $w^{j+1} \leftarrow$ PMO-PRO$((\mathbf{f}, \mathbf{e}), H, w^j, \lambda_i)$        (Alg 5)
    Add the final optimal solution to set of Pareto-solutions
      $p_{s_i} \leftarrow w^j$
Output: Pareto-optimal solutions $\{p_s\}$
```

The optimization step then calls the inner routine listed in Algorithm 5 which in turn uses PRO for getting new weights.

The tuning process with PMO-PRO is independently repeated (line 2 in Algorithm 4) with different set of weights for metrics[1] yielding a set of equivalent solutions $\{p_{s_1}, \ldots, p_{s_n}\}$ which are points on the Pareto frontier. The user then chooses one solution by making a trade-off between the performance gains across different metrics. However, this *a posteriori* choice not only ignores other solutions that are indistinguishable from the chosen one but also becomes impractical as the number of metrics are increased as noted earlier.

We alleviate this by complementing PMO with ensemble decoding, which we call *PMO ensemble*, in which each point in the Pareto solution is a distinct component in the ensemble decoder. This idea can also be used in other MMO approaches such as linear combination of metrics $(g_{\text{wavg}}(.))$ mentioned above. In this view, PMO ensemble is a special case of *ensemble combination*, where the decoding is performed by an ensemble of optimal solutions.

The ensemble combination model introduces new hyperparameters $\boldsymbol{\beta}$ that are the weights of the ensemble components (meta weights). These ensemble weights could set to be uniform in a naïve implementation. Or the user can encode her beliefs or expectations about the individual solutions $\{p_{s_1}, \ldots, p_{s_n}\}$ to set the ensemble weights (based on the relative importance of the components).

[1] For example Duh et al. (2012) use five meta-weight settings $(0, 1)$, $(0.3, 0.7)$ and so on as mentioned earlier. They combine the metric weights λ_i with the sentence-level metric scores M_i as $\ell = \left(\sum_k \lambda_k M_k\right) / k$ where ℓ is the target value for negative examples (the *else* line in Alg 5) in the optimization step.

Algorithm 5 PMO-PRO(): Optimization (*inner*) routine

```
Input: Held-out set: $(\mathbf{f}, \mathbf{e})$;
       $N$-best hypotheses: H;
       Weights: $w$ and
       Meta-weights: $\lambda$
Initialize $\mathcal{T} = \{\}$
for each f in tuning set f do
    $\{h\} = H(f)$
    $\{M(\{h\})\}$ = ComputeMetricScore($\{h\}, \hat{e}$)
    $\{\mathcal{F}\}$ = FindParetoFrontier($\{M(\{h\})\}, \lambda$)
    for each h in $\{h\}$ do
        if $h \in \mathcal{F}$ then add $(1, h)$ to $\mathcal{T}$
        else add $(\ell, h)$ to $\mathcal{T}$        (see footnote 1)
$w^p \leftarrow$ PRO($\mathcal{T}$) (optimize using PRO)
Output: Pareto-optimal weights $w^p$
```

Finally, one could also include a meta-level tuning step to set the weights $\boldsymbol{\beta}$.

The PMO ensemble approach is graphically illustrated in Figure 6.2; we will also refer to this figure while discussing other methods.[2] The original PMO-PRO seeks to maximize the points on the Pareto frontier (blue curve in the figure) leading to Pareto-optimal solutions. On the other hand, the PMO ensemble combines the different Pareto-optimal solutions and potentially moving in the direction of dashed (green) arrows to some point that has higher score in either or both dimensions.

6.3.4 Lateen MMO

We explained in Section 2.7, the lateen-EM optimization introduced by Spitkovsky et al. (2011) for jointly optimizing multiple objectives in the context of dependency parsing. Broadly speaking, the lateen-EM method uses a secondary hard EM objective to move away, when the primary soft EM objective gets stuck in a local optima. The course correction could be performed under different conditions leading to different lateen strategies that are based on when and how often to shift from one objective function to another during optimization.

The lateen technique can be applied to the multi-metric optimization in SMT by treating the different metrics as different objective functions. While the different lateen strategies can also be applied for machine translation, our goal here is to improve performance across the different metrics

[2] The illustration is based on two metrics, metric-1 and metric-2, but could be applied to any number of metrics. Without loss of generality we assume accuracy metrics, i.e. higher metric score is better.

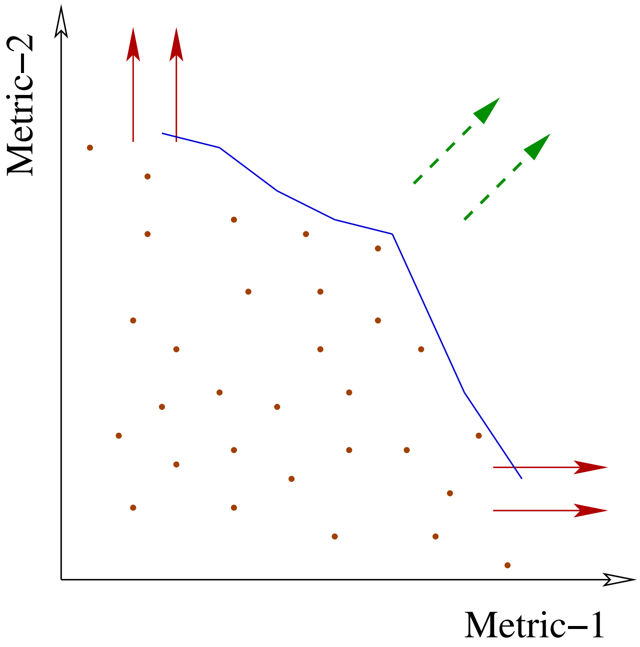


Figure 6.2: Illustration of different MMO approaches involving two metrics. Solid (red) arrows indicate optimizing two metrics independently and the dashed (green) arrow optimize them jointly. The Pareto frontier is indicated by the curve.

being optimized. Thus, we restrict ourselves to the simple lateen strategy where the search alternates between the metrics (in round-robin fashion) at each iteration. This can be expressed in terms of Equation 6.1 by introducing a k-dimensional indicator vector I, where only one element takes the value 1 in a given iteration j.

$$I = \begin{cases} 1 & \text{if } i \bmod j = 0 \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

The function g then represents the dot product of I and $M_i(H)$ so that our objective function searches for the $\arg\max$ in the space defined by g.

$$\begin{aligned} g(H) &= [I_1, \ldots, I_k].[M_1(H), \ldots, M_k(H)] \\ w^* &= \arg\max_{w}\ g(H) \end{aligned} \tag{6.3}$$

Since the notion of convergence is unclear in lateen setting, we stop after a fixed number of iterations optimizing the tuning set. In terms of Figure 6.2, lateen MMO corresponds to alternately maximizing the metrics along two dimensions as depicted by the solid arrows.

By the very nature of lateen-alternation, the weights obtained at each iteration are likely to be best for the metric that was optimized in that iteration. Thus, one could use weights from the last k iterations (for lateen-tuning with as many metrics) and then decode the test set with an ensemble of these weights as in PMO ensemble. However in practice we find the weights to converge and we simply use the weights from the final iteration to decode the test set in our lateen experiments.

6.3.5 Union of Metrics

While the lateen MMO optimizes multiple target metrics across different tuning iterations, it only considers one target metric for optimization at each iteration. An alternative would be to optimize the feature weights so as to jointly improve the performance across all the metrics at *every* iteration. The linear combination of metrics (Cer et al., 2010; Servan and Schwenk, 2011) approach discussed in Section 6.3.2 achieves this by fusing the different metrics into a joint metric (through linear combination), which is then used as the target metric.

However due to the scaling differences between the scores of different metrics, the linear combination might completely suppress the metric having scores in the lower-range. As an example, the RIBES scores that are typically in the high 0.7-0.8 range, dominate the BLEU scores that is typically around 0.3. While the weighted linear combination tries to address this imbalance, they introduce additional parameters that are manually fixed and not separately tuned.

Instead of fusing the different metric scores, we combine the positive (negative) candidate points for the different metrics to generate a single large set of positive (negative) examples, which then used to train the binary classifier in PRO tuning. Thus the key difference between the linear combination PRO and our union MMO is that the former method fuses the scores of the different metrics to create the target labels (for PRO), while the latter approach takes the union of positive (negative) examples from different metrics to create a single training set for the binary classifier. The candidates in the training set of the union method are labelled as 1 and 0 for positive and negative candidates respectively.

We represent this mathematically by using a unit vector for I (in Equation 6.2) so that the dot product results in the *union* of all $M_i(H)$ as:

$$g(H) = M_1(H) \cup \ldots \cup M_k(H) \tag{6.4}$$

Most of the optimization approaches involve two phases: i) select positive and negative examples and ii) optimize parameters to favour positive examples while penalizing negative ones. In the *union* approach, we independently generate positive and negative sets of examples for all the metrics and take their union. The optimizer now seeks to move towards positive examples from all metrics, while penalizing others.

This is similar to the PMO-PRO approach except that here the optimizer tries to simultaneously maximize the number of high scoring points across all metrics. Thus, instead of the entire Pareto frontier curve in Figure 6.2, the union approach optimizes the two dimensions simultaneously in each iteration.

6.4 Ensemble Tuning

These methods, even though novel, under utilize the power of ensembles as they combine the solution only at the end of the tuning process. We would prefer to tightly integrate the idea of ensembles into the tuning. We thus extend the ensemble decoding to *ensemble tuning*. The feature weights are replicated separately for each evaluation metric, which are treated as components in the ensemble decoding and tuned independently in the optimization step. Initially the ensemble decoder decodes a devset using a weighted ensemble to produce a single N-best list. For the optimization, we employ a two-step approach of optimizing the feature weights (of each ensemble component) followed by a step for tuning the meta (component) weights. The optimized weights are then used for decoding the devset in the next iteration and the process is repeated for a fixed number of iterations.

Modifying the MMO representation in Equation 6.1, we formulate *ensemble tuning* as:

$$H_{ens} = \mathcal{N}_{ens}\Big(\mathbf{f}; \{w_M\}; \otimes; \boldsymbol{\lambda}\Big) \tag{6.5}$$

$$\mathbf{w}^* = \Big\{ \arg\max_{w_{M_i}} H_{ens} \mid {\scriptstyle 1 \le i \le k} \Big\} \tag{6.6}$$

$$\boldsymbol{\lambda} = \arg\max_{\lambda} g\left(\{M_i(H_{ens}) | {\scriptstyle 1 \le i \le k}\} ; \mathbf{w}^* \right) \tag{6.7}$$

Here the ensemble decoder function $\mathcal{N}_{ens}(.)$ is parameterized by an ensemble of weights w_{M_1}, $\ldots, w_{M_k}$ (denoted as $\{w_M\}$ in Eq 6.5) for each metric and a mixture operation ($\otimes$). $\boldsymbol{\lambda}$ represents the weights of the ensemble components.

Pseudo-code for ensemble tuning is shown in Algorithm 6. In the beginning of each iteration (line 2), the tuning process ensemble decodes (line 4) the tuning set using the weights obtained from the previous iteration. Equation 6.5 gives the detailed expression for the ensemble decoding, where H_{ens} denotes the N-best list generated by the ensemble decoder.

The method now uses a dual tuning strategy involving two phases to optimize the weights. In the first step it optimizes each of the k metrics independently (lines 6-7) along its respective dimension

Algorithm 6 Ensemble Tuning Algorithm

```
1:  Input: Tuning set $\mathbf{f}$,
        Metrics $M_1, \ldots, M_k$ (ensemble components)
        Initial weights $\{w_M\} \leftarrow w_{M_1}, \ldots w_{M_k}$ and
        Component (meta) weights $\boldsymbol{\lambda}$
2:  for $j = 1, \ldots$ do
3:      $\{w_M^{(j)}\} \leftarrow \{w_M\}$
4:      Ensemble decode the tuning set
            $H_{ens} = \mathcal{N}_{ens}(\mathbf{f}; \{w_M^{(j)}\}; \otimes; \boldsymbol{\lambda})$
5:      $\{w_M\} = \{\}$
6:      for each metric $M_i \in \{M\}$ do
7:          $w_{M_i}^* \leftarrow$ PRO$((\mathbf{f}, \mathbf{e}), H_{ens}, w_{M_i})$        (use PRO)
8:          Add $w_{M_i}^*$ to $\{w_M\}$
9:      $\boldsymbol{\lambda} \leftarrow$ PMO-PRO$((\mathbf{f}, \mathbf{e}), H_{ens}, \{w_M\}, \boldsymbol{\lambda})$        (Alg 5)
10: Output: Optimal weights $\{w^M\}$ and $\boldsymbol{\lambda}$
```

in the multi-metric space (as shown by the solid arrows along the two axes in Figure 6.2). This yields a new set of weights $\mathbf{w}^*$ for the features in each metric.

The second tuning step (line 9) then optimizes the meta weights ($\boldsymbol{\lambda}$) so as to maximize the multi-metric objective along the joint k-dimensional space as shown in Equation 6.7. This is illustrated by the dashed arrows in the Figure 6.2. While $g(.)$ could be any function that combines multiple metrics, we use the PMO-PRO algorithm (Alg. 5) for this step.

The main difference between *ensemble tuning* and *PMO ensemble* is that the former is an ensemble model over metrics and the latter is an ensemble model over Pareto solutions. Additionally, PMO ensemble uses the notion of ensembles only for the final decoding after tuning has completed.

6.4.1 Implementation Notes

All the proposed methods fit naturally within the usual SMT tuning framework. However, some changes are required in the decoder to support ensemble decoding and in the tuning scripts for optimizing with multiple metrics. For ensemble decoding, the decoder should be able to use multiple weight vectors and dynamically combine them according to some desired mixture operation. Note that, unlike Razmara et al. (2012), our approach uses just one model but has different weight vectors for each metric and the required decoder modifications are simpler than full ensemble decoding.

While any of the mixture operations proposed by Razmara et al. (2012) could be used, here we use *log-wsum* – the linear combination of the ensemble components and *log-wmax* – the combination that prefers the locally best component. These are simpler to implement and also performed competitively in their domain adaptation experiments. Unless explicitly noted otherwise, the results presented in Section 6.5 are based on linear mixture operation log-wsum, which empirically performed better than the log-wmax for ensemble tuning.

6.5 Experiments

We evaluate the different methods on Arabic-English translation in single as well as multiple references scenario. Corpus statistics are shown in Table 6.1. For all the experiments in this chapter, we use our in-house implementation of Hierarchical phrase-based system - Kriya (Sankaran et al., 2012b), with additional support for ensemble decoding.

We use PRO (Hopkins and May, 2011) for optimizing the feature weights and PMO-PRO (Duh et al., 2012) for optimizing meta weights, wherever applicable. In both cases, we use SVM-Rank (Joachims, 2006) as the optimizer.

We used the default parameter settings for different MT tuning metrics. For METEOR, we tried both METEOR-tune and METEOR-hter settings and found the latter to perform better in BLEU and TER scores, even though the former was marginally better in METEOR[3] and RIBES scores. We observed the margin of loss in BLEU and TER to outweigh the gains in METEOR and RIBES and we chose METEOR-hter setting for both optimization and evaluation of all our experiments.

6.5.1 Evaluation on tuning set: Arabic-English

Language	Corpus	Training size	Tune/ test set
Arabic-English	ISI web-crawled parallel corpus	1.1 M	1664/ 1313 (MTA)
			1982/ 987 (ISI)
Chinese-English	HK Parallel text + GALE phase-1	2.3M	1928/ 919 (MTC)

Table 6.1: Corpus Statistics (# of sentences). We experiment our proposed multi-metric tuning approaches on two language pairs. We also examine the robustness of MMO strategies by experimenting in both single-reference (ISI) and multiple-references (MTA) settings (Arabic-English).

Unlike conventional tuning methods, PMO (Duh et al., 2012) was originally evaluated on the tuning set to avoid confounding due to potential mismatch with the test set. In order to ensure

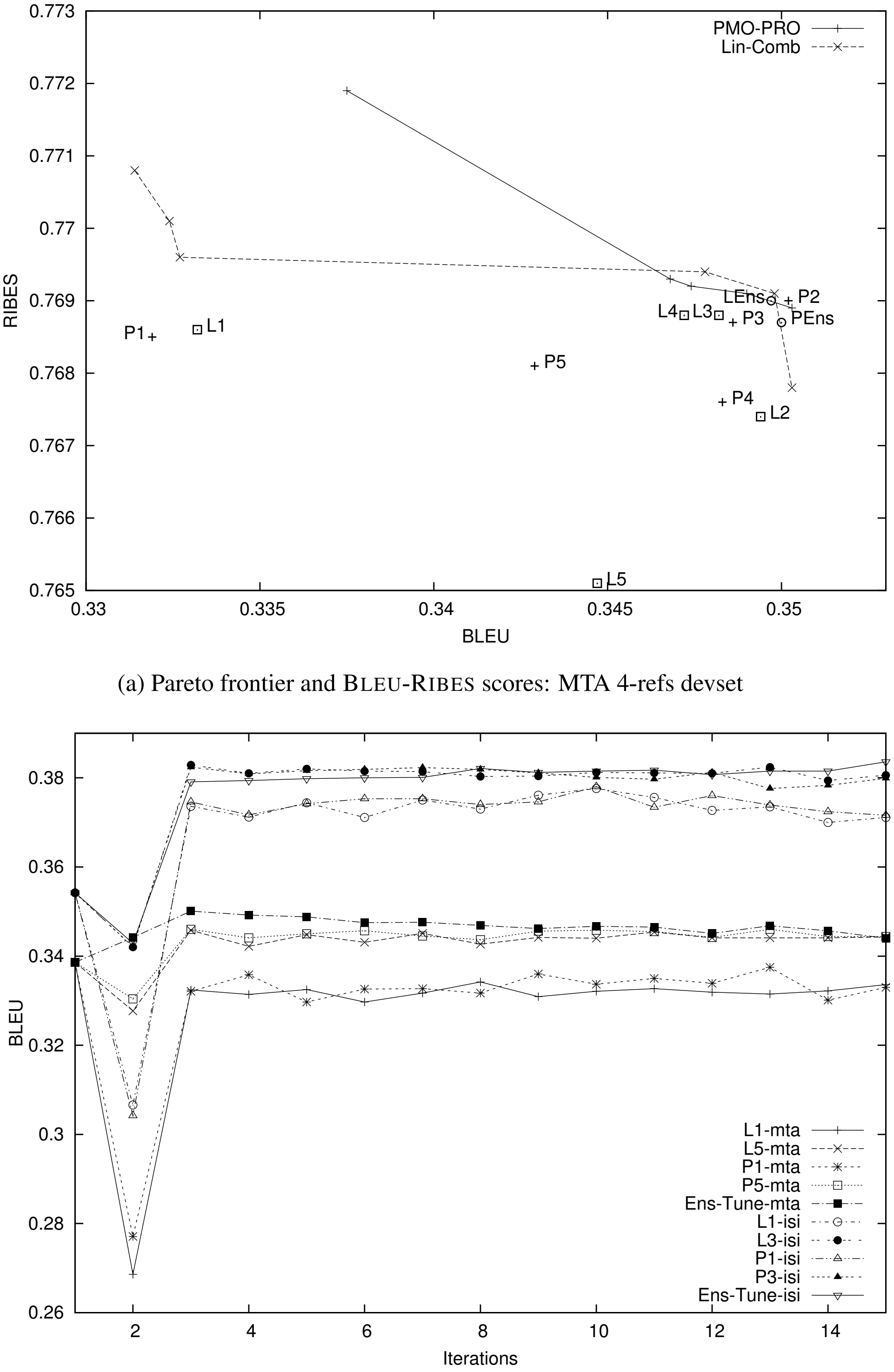
[3] This behaviour was also noted by Denkowski and Lavie (2011) in their analysis of Urdu-English system for *tunable metrics* task in WMT11.

robustness of evaluation, they *re-decode* the devset using the optimal weights from the last tuning iteration and report the scores on 1-best candidates. However this is inadequate in practice as this does not tell whether the optimal solution generalizes for an unseen set. We also find the re-decoding strategy to be misleading and we see the effects of over-training in the resulting scores. For example, we find the re-decoded scores to be comparable for conventional single-metric tuning as well as for our different multi-metric tuning approaches, even though weights tuned by our multi-metric tuning approaches generalize better for unseen test, with statistically significant improvements.

Nevertheless, we provide the results for re-decoding the devset in order to situate our work in the context of PMO-PRO. In the following we compare our PMO-ensemble approach with PMO-PRO (denoted P) and a linear combination (denoted L) baseline. Similar to Duh et al. (2012), we use five different BLEU:RIBES weight settings, viz. $(0.0, 1.0)$, $(0.3, 0.7)$, $(0.5, 0.5)$, $(0.7, 0.3)$ and $(1.0, 0.0)$, marked L1 through L5 or P1 through P5. The Pareto frontier is then computed from 80 points (5 runs and 15 iterations per run) on the devset.

Figure 6.3a shows the Pareto frontier of L and P baselines using BLEU and RIBES as two metrics. The frontier of the P dominates that of L for most part showing that the PMO approach benefits from picking Pareto points during the optimization.

We use the PMO-ensemble approach to combine the optimized weights from the 5 tuning runs and re-decode the devset employing ensemble decoding. This yields the points *LEns* and *PEns* in the plot, which obtain better scores than most of the individual runs of L and P. This ensemble approach of combining the final weights also generalizes to the unseen test set as we show later.



(a) Pareto frontier and BLEU-RIBES scores: MTA 4-refs devset

(b) Tuning BLEU scores: MTA 4-refs and ISI 1-ref devsets

Figure 6.3: Devset (redecode): Comparison of Lin-comb (L) and PMO-PRO (P) with Ensemble decoding (Lens and PEns) and Ensemble tuning (Ens-Tune)

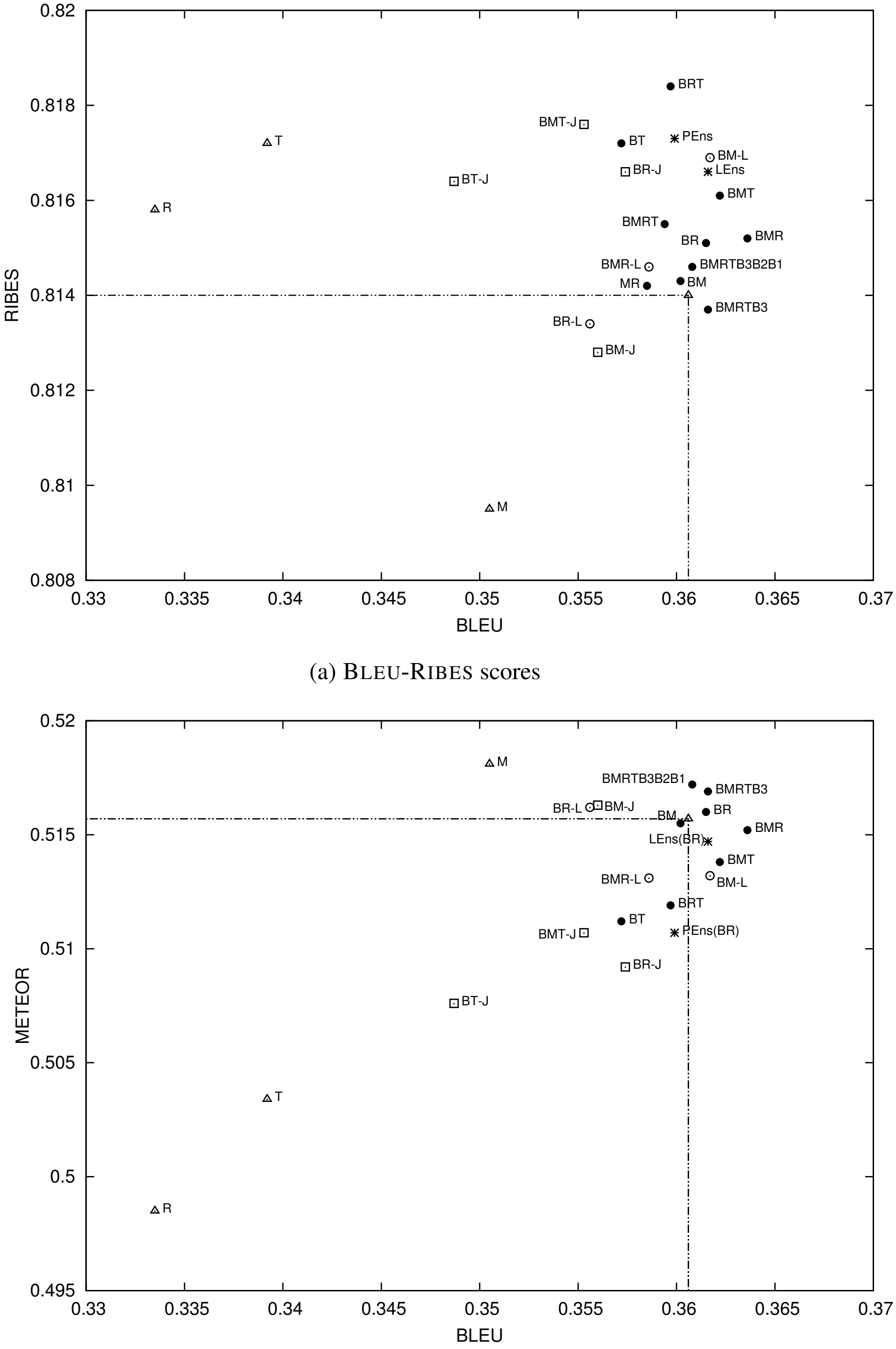
Figure 6.3b plots the change in BLEU during tuning in the multiple references and the single reference scenarios. We show for each baseline method L and P, plots for two different weight settings that obtain high BLEU and RIBES scores. In both datasets, our ensemble tuning approach dominates the curves of the (L and P) baselines. In summary, these results confirm that the ensemble approach achieves results that are competitive with previous MMO methods on the devset Pareto curve. We now provide a more comprehensive evaluation on the test set.

6.5.2 Evaluation on test set: Arabic-English

This section contains multi-metric optimization results on the unseen test sets, one test set has multiple references and the other has a single-reference. We plot BLEU scores against other metrics (RIBES, METEOR and TER) and this allows us to compare the performance of each metric relative to the de-facto standard BLEU metric.

Baseline points (marked as a triangle with a dot in the middle) are identified by single letters B for BLEU, T for TER, etc. and the scores of the BLEU-only baseline are indicated by the dashed lines on the two axes. MMO points use a series of single letters referring to the metrics used, e.g. BT for BLEU-TER. The union of metrics method is identified with the suffix J (appear as a square with a single dot in the middle) and lateen method with suffix L (appear as a circle with a single dot in the middle). The ensemble tuned points appear as dark circles and are marked only with the first letters of the metrics used (without any suffix). Finally the ensemble combination method are starred and labelled with suffix Ens. For example 'BT-L' refers to a lateen system tuned with BLEU and TER and a point labelled 'BM' refers to an ensemble tuned system using BLEU and METEOR.

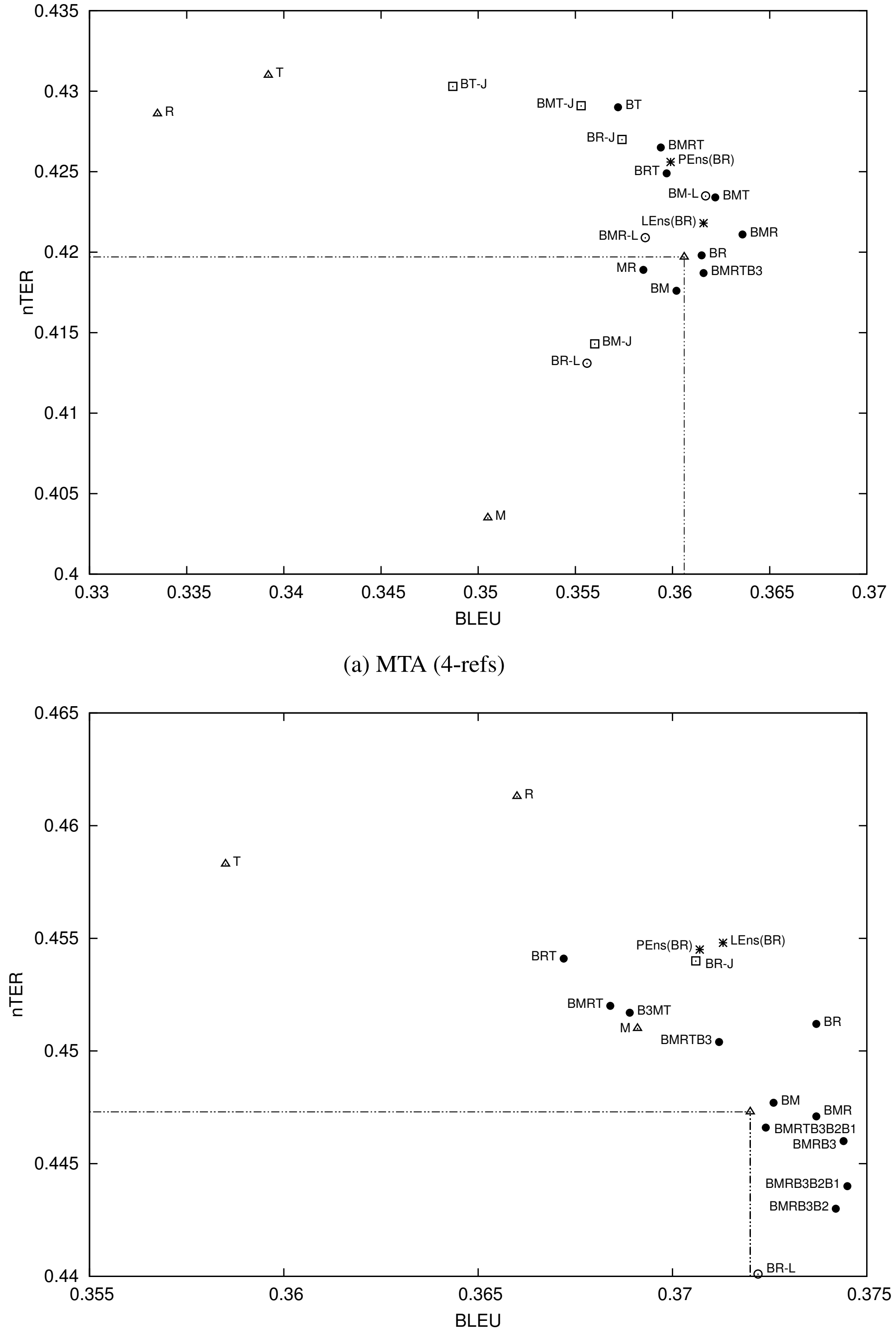
Figures 6.4 and 6.5a plot the scores for the MTA test set with 4-references. We see several noticeable and some statistically significant improvements in BLEU and RIBES (see Table 6.2 for BLEU improvements). All our MMO approaches, except for the union method, show gains on both BLEU and RIBES axes.



(a) BLEU-RIBES scores

(b) BLEU-METEOR scores

Figure 6.4: MTA 4-refs testset: Comparison of different MMO approaches. The dashed lines correspond to the classic BLEU optimization. The lateen method is marked with suffix L, union with the suffix J and ensemble combination with suffix Ens. The ensemble tuning points are labelled with the first letters of metrics used.



(a) MTA (4-refs)

(b) ISI (1-ref)

Figure 6.5: BLEU-nTER scores: Comparison of different MMO approaches. We plot nTER (1-TER) scores for easy reading of the plots. The dashed lines correspond to the classic BLEU optimization. The lateen method is marked with suffix L, union with the suffix J and ensemble combination with suffix Ens. The ensemble tuning points are labelled with the first letters of metrics used.

Approach and Tuning Metric(s)	**Bleu**	
	MTA	**ISI**
Single Objective Baselines		
Bleu	36.06	37.20
Meteor	35.05	36.91
Ribes	33.35	36.60
Ter	33.92	35.85
Ensemble Tuning: 2 Metrics		
B-M	36.02	***37.26***
B-R	**36.15**	***37.37***
B-T	35.72	36.31
Ensemble Tuning: 3 Metrics		
B-M-R	***36.36***	***37.37***
B-M-T	**36.22**	36.89
B-R-T	35.97	36.72
Ensemble Tuning: $>$ 3 Metrics		
B-M-R-T	35.94	36.84
B-M-R-T-B3	***36.16***	**37.12**
B-M-R-T-B3-B2-B1	***36.08***	***37.24***

Table 6.2: Bleu Scores on MTA (4 refs) and ISI (1 ref) test sets using the standard *mteval* script. Boldface scores indicate scores that are comparable to or better than the baseline Bleu-only tuning. *Italicized* scores indicate statistically significant differences at p-value 0.05 computed with bootstrap significance test.

Figures 6.4b and 6.5a show that none of the proposed methods managed to improve the baseline scores for Meteor and Ter. However, several of our ensemble tuning combinations work well for both Meteor (BR, BMRTB3, etc.) and Ter (BMRT and BRT) in that they improved or were close to the baseline scores in either dimension. We again see in these figures that the MMO approaches can improve the Bleu-only tuning by 0.3 Bleu points, without much drop in other metrics. This is in tune with the finding that Bleu could be tuned easily (Callison-Burch et al., 2011) and also explains why it remains a popular choice for optimizing SMT systems.

Among the different MMO methods the ensemble tuning performs better than lateen or union approaches. In terms of the number of metrics being optimized jointly, we see substantial gains when using a small number (typically 2 or 3) of metrics. Results seem to suffer beyond this number; probably because there might not be a space that contain solution(s) optimal for *all* the metrics that are jointly optimized.

The lateen and union approaches appear to be very sensitive to the number of metrics and they generally perform well for two metrics case and show degradation for more metrics. Unlike other approaches, the union approach failed to improve over the baseline BLEU and this could be attributed to the conflict of interest among the metrics, while choosing example points for the optimization step. The positive example preferred by a particular metric could be a negative example for the other metric. This would only confuse the optimizer resulting in poor solutions. Our future line of work would be to study the effect of avoiding such of conflicting examples in the union approach.

For the single-reference (ISI) dataset, we only plot the BLEU-TER case in Figure 6.5b. The results are similar to the multiple references set indicating that MMO approaches are equally effective for single references[4]. Table 6.2 shows the BLEU scores for our ensemble tuning method (for various combinations) and we again see improvements over the baseline BLEU-only tuning.

6.5.3 Evaluation on test set: Chinese-English

We present the results for the unseen test set for Chinese-English in a two-dimensional BLEU-METEOR plot. As earlier the ensemble tuning points are labelled with the dark circles and we see that several of the ensemble tuning points to do significantly better in both dimensions than the BLEU-only tuning. In effect we see an absolute BLEU gain of 0.5 points over the single metric baseline, which is statistically significant at p-value 0.05.

[4] One could argue that MMO methods require multiple references since each metric might be picking out a different reference sentence. Our experiment shows that MMO methods can perform well even with just one reference.

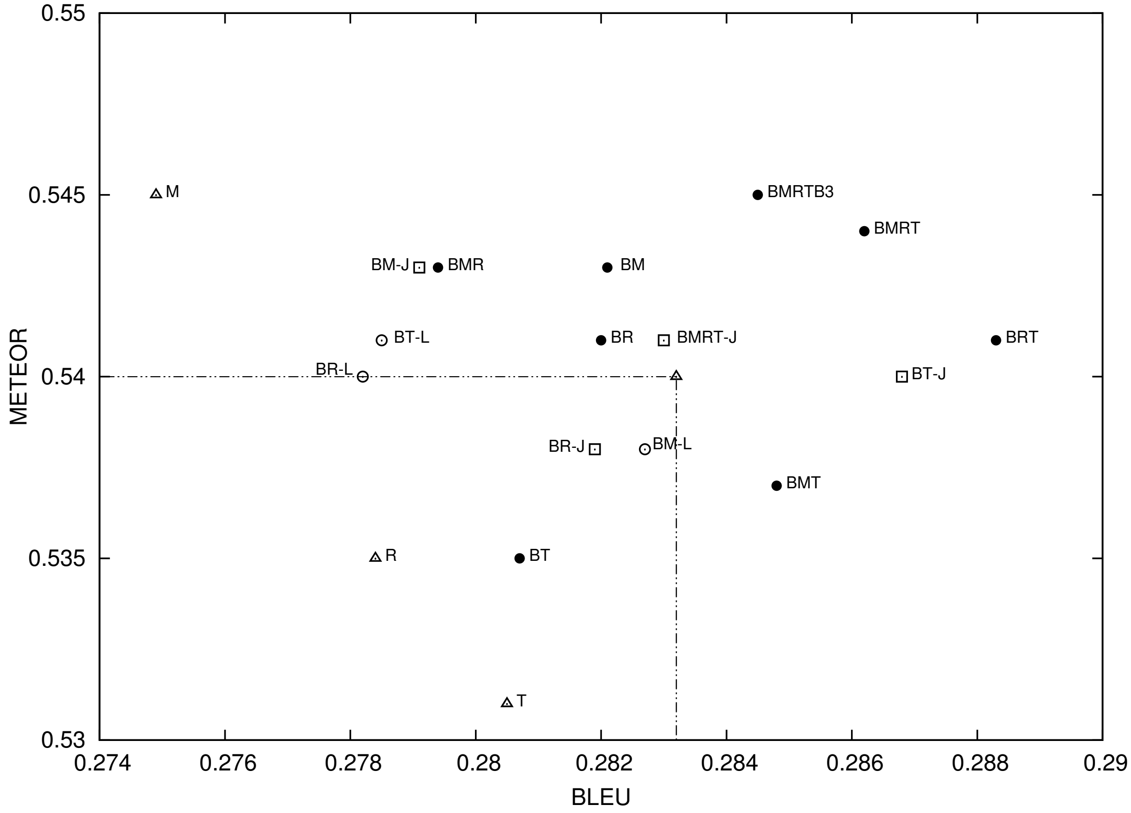


Figure 6.6: MTC 4-refs testset: Comparison of different MMO approaches. The dashed lines correspond to the classic BLEU optimization. The lateen method is marked with suffix L and union with the suffix J. The ensemble tuning points are labelled with the first letters of metrics used.

6.5.4 Metric Sensitivity

We would like to analyze how sensitive the metrics are to the smaller changes in the translation outputs as the translation quality improves. Given a reasonable sized test set, this could to done by having translations of varying levels of quality and then computing the scores for these different translations. Since there is no natural way of experimenting this without involving substantial human effort to generate varying quality of translations, we use a synthetic experimental setup to generate these translations automatically.

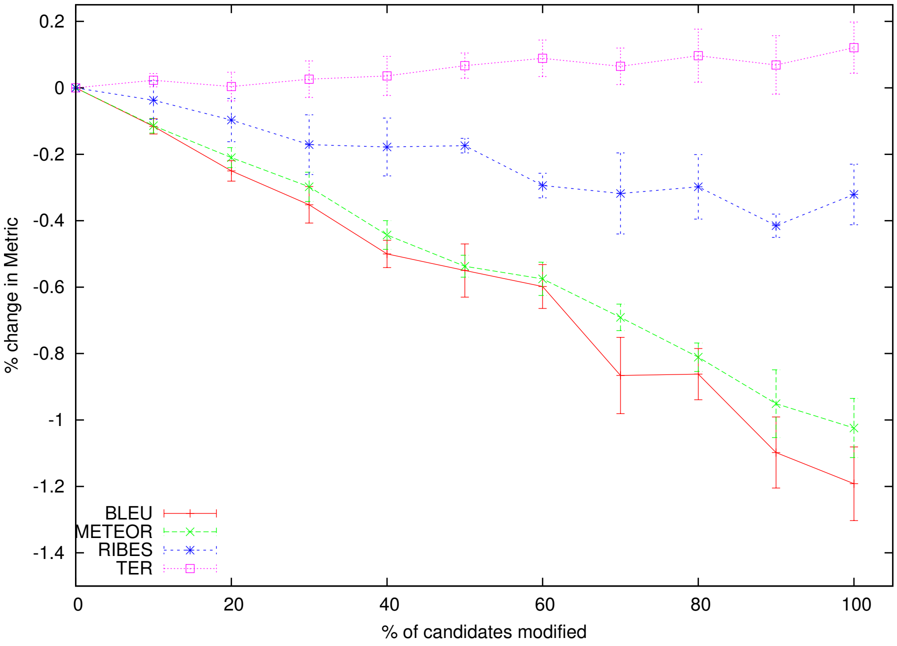
Given a set of translations produced by a system, we choose a fixed % of sentences (varied between 0% and 100% in intervals of 10) for modification. These translations are then modified automatically by applying one of the four edit operations.

- *delete* operation models the situation where the decoder misses a word. We approximate this by removing a random word from a sentence. Note that since we choose a random word for removal, we might actually be removing a OOV word appearing in the final output.
- *insert* simulates the effect of adding a new word in a randomly chosen position in the sentence. This corresponds to the case where the decoder inserts an additional word while decoding.
- *replace* operation substitutes a random word in the sentence with a new word. This would happen, when the decoder uses an alternative word choice (note that the operation doesn't necessarily replace it with an alternate translation). The new word for *insert* and *replace* operations is chosen at random from a list of top-50 high frequency words from the phrase-table.
- *swap* choses two positions at random (within each sentence) and interchange the words in those positions. This approximates the case where the decoder makes a reordering mistake and interchanges two target words.

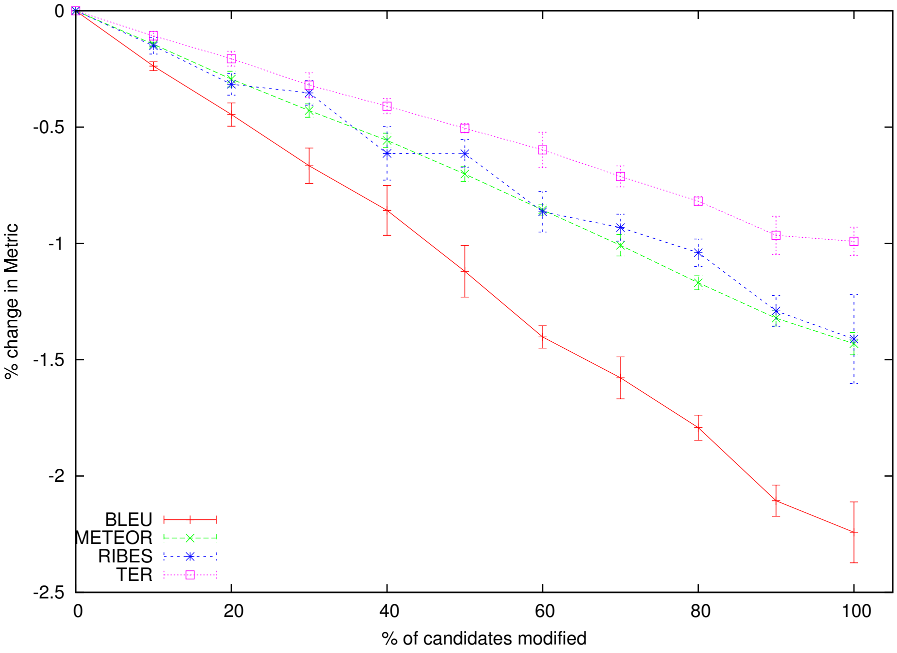
We compute the scores for the four evaluation metrics for every 10% interval of sentences modified. We then calculate the $\%$ change in the score of individual evaluation metrics with respect to the original unmodified translation. The percentage change in the metric scores allow us to compare the metrics at equal footing without being concerned about their different scales. We repeat the entire experiment for five times and average the $\%$ change in the metric scores across the runs. Finally we also find the standard deviation (σ) of the $\%$ change for each interval.

We perform this synthetic experiment on multiple-references (Chinese-English) and single-reference (Arabic-English) settings. For Chinese-English, we use the translations of the MTC test set (919 sentences) produced by a system optimized for BLEU, RIBES and TER using ensemble tuning MMO. Similarly for Arabic-English, we translate the single-reference test set (987 sentences) using the optimal weights from the ensemble tuning method that optimized BLEU and RIBES.

Figure 6.7a plots the results for the multiple-references for the *delete* operation; interestingly the TER scores actually improve as we remove the words in the translation (although the percentage improvement is very small). BLEU and METEOR appear to be equally sensitive and in the case of BLEU this could largely be due to a higher brevity penalty. For the plot in Figure 6.7b, we randomly decide the edit operation to be applied for modifying a sentence (marked as *any* in the plots). But for each sentence, only one modification is applied. Here BLEU varies almost twice as fast as METEOR and RIBES, with TER demonstrating smallest % change.

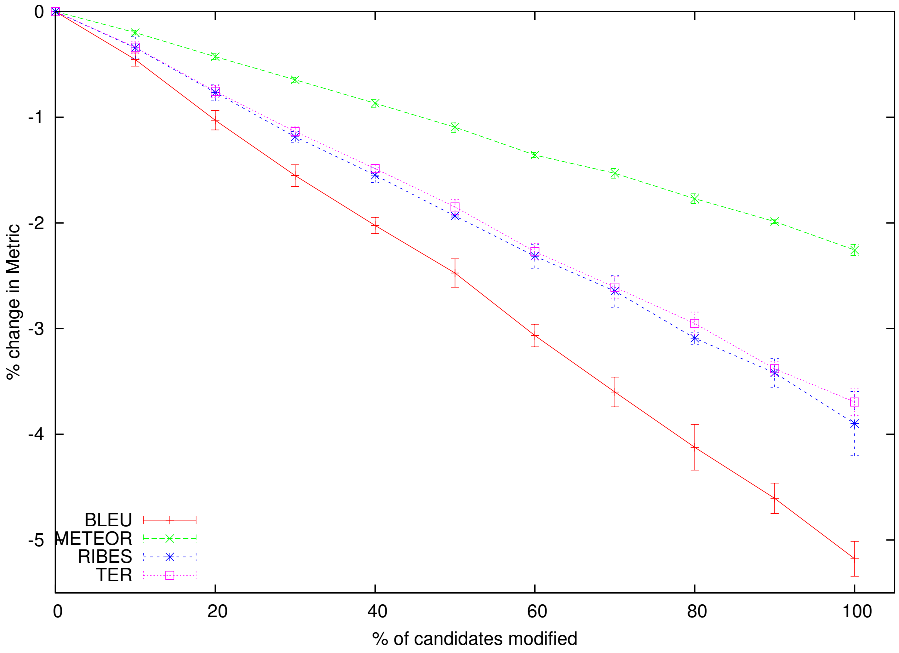


(a) Modification operation: *delete*. TER scores actually improve as the words are deleted from the translation output.

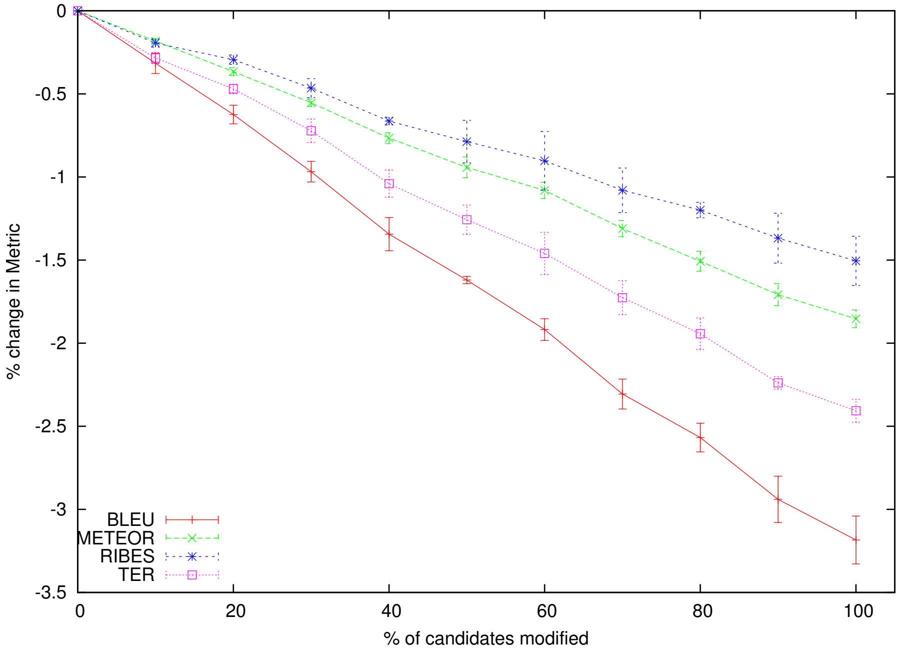


(b) Modification operation: *any*. The operation is chosen at random for each sentence (with only one operation applied for a single sentence).

Figure 6.7: **Zh-En multiple refs**: Synthetic experiment to measure to sensitivity of MT evaluation metrics on translation output changes. We modify a fixed percentage of sentences $(0\%, 10\%, \ldots, 100\%)$ with one of four different operations: *delete*, *insert* , *replace* and *swap*. The experiment is repeated 5 times and the $\%$ changes in the metric scores are averaged.



(a) Modification operation: *swap*.



(b) Modification operation: *any*. The operation is chosen at random for each sentence (with only one operation applied for a single sentence).

Figure 6.8: **Ar-En single ref**: Synthetic experiment to measure to sensitivity of MT evaluation metrics on translation output changes. We modify a fixed percentage of sentences $(0\%, 10\%, \ldots, 100\%)$ with one of four different operations: *delete*, *insert* , *replace* and *swap*. The experiment is repeated 5 times and the $\%$ changes in the metric scores are averaged.

Additionally as one can expect, the magnitude of percentage change in the metrics scores for the multiple-references setting is less than the single-reference (Figure 6.8) case. The single reference setting is very sensitive to the incorrect word ordering changes as seen in Figure 6.8a depicting metrics' sensitivity to the *swap* operation. METEOR doesn't seem to capture this as much as the other three metrics. The 5% reduction in BLEU scores clearly show the impact of higher order n-grams; because the *swap* operation doesn't affect the unigram precision as was shown earlier by (Koehn, 2004). The other operations have lesser impact on the metrics and the magnitude of the change is reduced as we randomly decide the operation instead of *swap* (Figure 6.8b).

While these plots show the percentage reduction in the scores as we corrupt the translation, we can use these to extrapolate the gains that we'll likely get as the quality of translations improve. For example to get a BLEU score gain of 0.5%, we need about $\tilde{2}2\%$ of the translations in the multiple-references setting to have at least one positive change. However for the same 0.5% gain in METEOR, RIBES and TER the translations have to improve in between 35% and 50% of the sentences. We see similar behaviour in the single-reference setting, even though the difference between BLEU and other metrics is lesser.

We present more plots in Appendix A. that shows the effect of individual edit operations. In addition we also explore whether the choice of optimization metric could influence the sensitivity of the different metrics, by comparing the metrics' sensitivity for two different translations produced by TER-only and BLEU-RIBES ensemble optimized systems.

In general for different edit operations, we found BLEU to be *most* sensitive and RIBES to be *least* sensitive with METEOR and TER falling in the middle. This also explains why BLEU remains a widely preferred choice to optimize for. However, our experiments show that we could gain higher BLEU scores by optimizing for multiple metrics instead of tuning just BLEU, in addition to doing comparably on other metrics.

6.5.5 Metric Dichotomy

We are interested in finding which metrics are useful for getting better BLEU scores in the MMO setting where they are jointly optimized. Figure 6.9 plots the (unseen test set) BLEU scores for the various ensemble tuning settings in multiple-references and single-reference Arabic-English datasets. We observe that METEOR and RIBES to generally help BLEU, and we see moderate to statistically significant gains in BLEU score when we use any of these metrics in the MMO. On the other hand, the BLEU scores are adversely affected for most of the ensemble tuning settings involving TER and frequently the reduction in BLEU is substantial. We hypothesize that certain groups of metrics are naturally amenable for being tuned together in the multi-metric setting (at least from the limited perspective of better BLEU scores), whereas some other metrics (such as TER) might not be amenable. We call this *metric dichotomy* and it is not clear if such dichotomous behaviour could ever be avoided while tuning with multiple metrics.

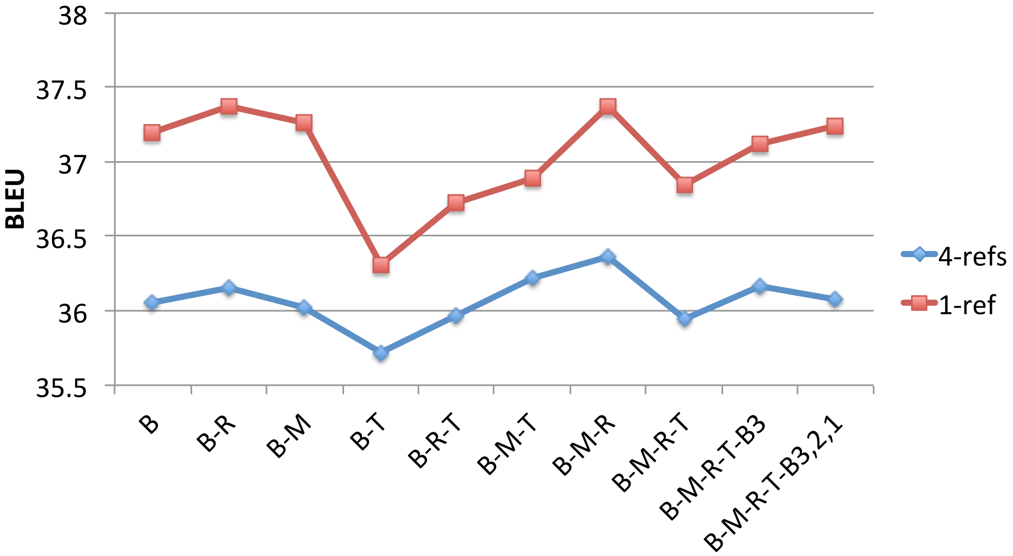


Figure 6.9: Metric dichotomy: BLEU scores for different ensemble tuning settings for the two Arabic-English test sets. The left-most point corresponds to BLEU-only baseline.

Interestingly the dichotomous behaviour of TER disappears in the case of Chinese-English as we see from the plot in Figure 6.10. Here contrary to the Arabic-English dataset, we see substantial BLEU gains when using TER as a metric in the ensemble tuning (except for BT). We also see the BLEU scores to peak for the ensemble tuning setting involving three metrics, which then tapers off as we add more metrics. Finally we believe that the metric dichotomy behaviour requires further study as it remains key to gain better understanding of the interplay between different metrics in the multi-metric tuning.

6.5.6 Human Evaluation

So far we have shown that multi-metric optimization can improve over single-metric tuning on a single metric like BLEU and we have shown that our methods find a tuned model that performs well with respect to multiple metrics. Is the output that scores higher on multiple metrics actually a better translation? To verify this, we conducted a post-editing human evaluation experiment.

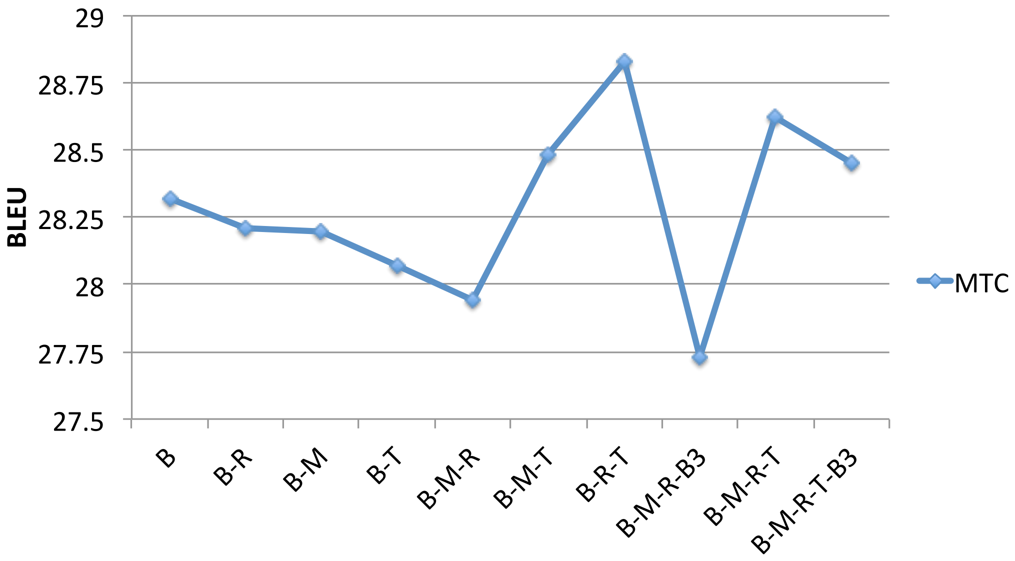


Figure 6.10: Metric dichotomy: BLEU scores for different ensemble tuning settings for the Chinese-English test set. The left-most point corresponds to BLEU-only baseline.

We compared our ensemble tuning approach involving BLEU, METEOR and RIBES (B-M-R) with systems optimized for BLEU (B-only) and METEOR (M-only).

We selected 100 random sentences (that are at least 15 words long) from the Arabic-English MTA (4 references) test set and translated them using the three systems (two single metric systems and BMR ensemble tuning). We shuffled the resulting translations and split them into 3 sets such that each set has equal number of the translations from three systems. The translations were edited by three human annotators in a post-editing setup, where the goal was to edit the translations to make them as close to the references as possible, using the Post-Editing Tool: PET (Aziz et al., 2012). The annotators were not Arabic-literate and relied only on the reference translations during post-editing. The identifiers that link each translation to the system that generated it are removed to avoid annotator bias.

In the end we collated post-edited translations for each system and then computed the system-level human-targeted (HBLEU, HMETEOR, HTER) scores, by using respective post-edited translations as the reference. First comparing the HTER (Snover et al., 2006) scores shown in Table 6.3, we see that the single-metric system optimized for METEOR performs slightly worse than the one optimized for BLEU, despite using METEOR-hter version (Denkowski and Lavie, 2011). Ensemble tuning-based system optimized for three metrics (B-M-R) improves HTER by 4% and 6.3% over

Metric	**Single-metric Tuning**		**Ensemble Tuning**
	B-only	M-only	B-M-R
BLEU	37.89	37.18	***39.01***
HBLEU	51.93	**53.59**	53.14
METEOR	61.31	61.56	**61.68**
HMETEOR	72.35	72.39	**72.74**
TER	0.520	0.532	***0.516***
HTER	0.361	0.370	**0.346**

Table 6.3: Post-editing Human Evaluation: Regular (untargeted) and human-targeted scores. Human targeted scores are computed against the post-edited reference and regular scores are computed with the original references. **Best** scores are in bold-face and *statistically significant* ones (at p = 0.05) are italicized.

BLEU and METEOR optimized systems respectively.

The single-metric system tuned with M-only setting scores high on HBLEU, closely followed by the ensemble system. We believe this to be caused by chance rather than any systematic gains by the M-only tuning; the ensemble system scores high on HMETEOR compared to the M-only system. While HTER captures the edit distance to the targeted reference, HMETEOR and HBLEU metrics capture missing content words or synonyms by exploiting n-grams and paraphrase matching.

We also computed the regular variants (BLEU, METEOR and TER), which are scored against original references. The ensemble system outperformed the single-metric systems in all the three metrics. The improvements were also statistically significant at p-value of 0.05 for BLEU and TER.

6.6 Summary and Future work

We propose and present a comprehensive study of several multi-metric optimization (MMO) methods in SMT. First, by exploiting the idea of ensemble decoding (Razmara et al., 2012), we propose an effective way to combine multiple Pareto-optimal model weights from previous MMO methods (e.g. Duh et al. (2012)), obviating the need for manually trading off among metrics. We also proposed two new variants: lateen-style MMO and union of metrics. We also presented an extensive analysis of the sensitivity of different metrics in order to understand the amount of improvements we require for meaningful gains in them. We formalized the metric dichotomy from the perspective of improving BLEU scores so as to analyze the interplay of different metrics.

We also extended ensemble decoding to a new tuning algorithm called *ensemble tuning*. This method demonstrates statistically significant gains for BLEU and RIBES with modest reduction in METEOR and TER. Further, in our human evaluation, ensemble tuning obtains the best HTER among competing baselines, confirming that optimizing on multiple metrics produces human-preferred translations compared to the conventional optimization approach involving a single metric.

6.6.1 Game Theoretic Perspective of MMO

We plan to extend MMO by using Game theoretic perspective, under which the metrics of interest are treated as individual players and the actual optimization corresponds to each player deciding its strategy with respect to others, so as to maximize their pay-off. The pay-off can correspond to the gain for an individual metric or a collective gain for a set of metrics.

We could view some of the methods discussed earlier as non-cooperative games, where the players make their decision in isolation solely to maximize the respective pay-offs. Our formulation of the lateen MMO approach that strictly alternates between the objective functions falls under this type.[5] Both PMO-PRO and Ensemble tuning[6] use the notion of Pareto efficiency, which inherently corresponds to the strong Nash equilibrium (Bernheim et al., 1987). To elaborate this point further, consider ensemble tuning MMO with two metrics. The points on the two extremes of a frontier curve (as in Figure 6.2) maximize the pay-off for one metric, while heavily penalizing the other. This non-cooperative playing strategy might not lead to increased pay-off for all the metrics. Additionally, each metric has different range of scores and this means that an improvement of a certain magnitude has varying significance for each metric. Thus the non-cooperative game setting doesn't model this directly during optimization (while the meta-weights tuning can capture this to certain extent, it only looks at the Pareto frontier which inherently corresponds to maximizing the respective gains).

In contrast, a cooperative game setting would allow us to define a *collective pay-off* (Marler and Arora, 2004) for all the metrics, which can then be enforced by the optimization step. We could further model the varying significance levels of the metrics by appropriately defining the collective pay-off. Thus in this scheme, we form a grand coalition of metrics and seek to maximize the pay-offs for all the coalition members. We could use the following two broad notions in defining the collective pay-off for our MMO context in machine translation, i) a player should be willing to

[5] Though the lateen MMO does not optimize multiple metrics simultaneously (in the *same* iteration), the point is that there is no *contract* between the different metrics across iterations.

[6] The first step of ensemble tuning only optimizes the respective metrics (independent from each other); it is the second tuning step that optimizes the meta-weights by jointly optimizing for multiple metrics.

sacrifice its gain, if this would result in *disproportionate* loss to one or more of the other players and ii) improve the pay-offs for maximum number of players. We would also like to exploit the varying significance levels of the metrics in quantifying proportionate gain/ loss that would be acceptable for each metric.

Note that the linear combination baseline (discussed in Sec 6.3.2) could be considered as a simple case of the cooperative MMO, where the collective pay-off is defined to be the weighted sum. We plan to explore the collective bargaining strategy, particularly using methods such as simple product or Nash product (Venugopal and Narendran, 1990; Marler and Arora, 2004) of metrics.

Chapter 7

Conclusion and Future Directions

We presented improvements to the well-known Hierarchical phrase-based (Hiero) models (Chiang, 2007) for statistical machine translation. Our improvements broadly apply for the two different stages in Hiero's multi-step pipeline, i) training Hiero grammars for translation and ii) training the weights of the log-linear model feature functions by optimizing some MT evaluation metric.

In the first part of the thesis research, we focused on the heuristic rule extraction algorithm in Hiero that produces a large grammar with rule weights having a flat distribution. The grammars generated by this heuristic algorithm affects the decoding speed due to the huge model size (as compared to an equivalent phrase-based model) and also leads to *overgeneration* and *spurious* ambiguities due to its model size and the flat distribution of the estimated rule weights.

We addressed these issues by proposing alternative Bayesian models employing a novel prior based on the lexical alignment probabilities of the rules. Our Bayesian models are applicable for unary Hiero (with maximum rule arity of 1) as well as binary Hiero (having maximum rule arity of 2) settings. We employed two different inference methods, Gibbs sampling (Sankaran et al., 2011) and Variational (Sankaran et al., 2012a) approximation; with the Variational inference achieving the competitive performance for a wide range of language pairs, using just a fraction of the original model size. We achieved highest reduction of over 57% with our unary model-2 and over 80% with binary model (Sankaran et al., 2013a). Our models also produced a peaked distribution of the grammars making it easier to prune the model, without impacting the translation quality. We further presented a distributed version of our Variational inference that can scale well with the large amounts of training corpora that are typically used in statistical MT.

In the second part, we presented a *unified-cascade framework* for jointly training the alignments and Hiero grammars. The key motivation for the unified framework is to address the traditional disconnect between the different steps of the Hiero translation pipeline with some of them relying on heuristic methods (such as for symmetrizing bidirectional alignments and for getting many-to-many alignments). In contrast to the traditional multi-step pipeline, our unified-cascade model learns the alignments and grammars in two steps in an iterative fashion. At the same time, our novel framework retains the separation between the structurally dissimilar alignments and Hiero rules, allowing us to efficiently employ distinct models for the two steps. Our validation experiments using an existing alignment model (Neubig et al., 2011) and our binary Hiero model (Sankaran et al., 2013a) show promising results compared to the heuristic baseline.

For future exploration, we intend to replace the existing alignment model in the unified-cascade framework with a new model. We seek to improve the alignments under this new model as the current aligner (Neubig et al., 2011) uses a beam search approximation in the search process, thus violating the *detailed balance* and *positive recurrence* properties of the Markov chain (Cohn and Haffari, 2013). We further plan to use the extracted translation rules in initializing the next iteration of the aligner. This could be done by parsing the source sentences of the training corpus with the extracted rules to force decode towards the target side sentences. The alignments induced by the synchronous derivation could then used to initialize the aligner in the next iteration.

In the final part of this thesis, we shift our focus to the parameter optimization step of the Hiero pipeline. Traditional approaches to MT parameter tuning, optimize the feature weights to improve the translation performance on a single metric (typically BLEU). In contrast, we propose four novel approaches to jointly optimize for multiple evaluation metrics. Our multi-metric optimization (MMO) approaches draw ideas from different areas such as Economics and Engineering. The ensemble tuning method combines the advantages for an ensemble model of decoding with the Pareto-based optimization to yield gains across a range of evaluation metrics, over the classic BLEU-only optimization. Our methods also achieve statistically significant BLEU score improvements of up to 0.5 points in two language pairs.

We finally presented some synthetic experiments for analyzing the sensitivity of the individual evaluation metrics to incremental degradations in the translation quality. We find BLEU to be easily tuneable, i.e it is able to detect small changes in the translations and adjust its score to better reflect the changes. We also analyzed the interplay of different metrics (*metric dichotomy*) with BLEU to identify which metrics are amenable to be tuned jointly with it (as a target metric). We find the interplay of a metric with BLEU to be language specific. For example, TER does not work well with BLEU in the Arabic-English setting, whereas it works exceedingly well in the case of Chinese-English.

As a future direction, we plan to further the analysis about the behaviour of the different MT evaluation metrics. We hope that this would lead to better understanding of the metrics, eventually for improving the translation quality. As a second direction, we also intend to pursue some novel MMO approaches, formulated in the Game theoretic perspective. As we have mentioned in Section 6.6 we plan to explore *collective bargaining* based approaches including a Nash product (Venugopal and Narendran, 1990; Marler and Arora, 2004) method.

Bibliography

[Asuncion et al.2008] Arthur Asuncion, Padhraic Smyth, and Max Welling. 2008. Asynchronous distributed learning of topic models. In *Proceedings of Neural Information Processing Systems-08*, pages 81–88.

[Attias2000] Hagai Attias. 2000. A variational bayesian framework for graphical models. In *Advances in Neural Information Processing Systems 12*, pages 209–215. MIT Press.

[Auli and Lopez2011] Michael Auli and Adam Lopez. 2011. Training a log-linear parser with loss functions via softmax-margin. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 333–343, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

[Aziz et al.2012] Wilker Aziz, Sheila Castilho Monteiro de Sousa, and Lucia Specia. 2012. PET: a tool for post-editing and assessing machine translation. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

[Bazrafshan et al.2012] Marzieh Bazrafshan, Tagyoung Chung, and Daniel Gildea. 2012. Tuning as linear regression. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 543–547, Montréal, Canada. ACL.

[Bernheim et al.1987] B.Douglas Bernheim, Bezalel Peleg, and Michael D Whinston. 1987. Coalition-proof nash equilibria i. concepts. *Journal of Economic Theory*, 42(1):1 – 12.

[Blunsom and Osborne2008] Phil Blunsom and Miles Osborne. 2008. Probabilistic inference for machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 215–223.

[Blunsom et al.2008a] Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008a. Bayesian synchronous grammar induction. In *Proceedings of Neural Information Processing Systems-08*.

[Blunsom et al.2008b] Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008b. A discriminative latent variable model for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 200–208, Columbus, Ohio.

[Blunsom et al.2009] Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of Association of Computational Linguistics-09*, pages 782–790. Association for Computational Linguistics.

[Bod1998] Rens Bod. 1998. *Beyond Grammar: An Experience-Based Theory of Language*. CSLI Publications, Stanford.

[Brown et al.1993] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19(2):263–311, June.

[Burkett and Klein2008] David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic parsing). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 877–886, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Burkett et al.2010] David Burkett, John Blitzer, and Dan Klein. 2010. Joint parsing and alignment with weakly synchronized grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–135. Association for Computational Linguistics.

[Callison-Burch et al.2011] Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. ACL.

[Callison-Burch et al.2012] Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. ACL.

[Cer et al.2010] Daniel Cer, Christopher D. Manning, and Daniel Jurafsky. 2010. The best lexical metric for phrase-based statistical mt system optimization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 555–563. ACL.

[Chen et al.2012] Boxing Chen, Roland Kuhn, and Samuel Larkin. 2012. Port: a precision-order-recall mt evaluation metric for tuning. In *Proceedings of the 50th Annual Meeting of the ACL (Volume 1: Long Papers)*, pages 930–939, Jeju Island, Korea. ACL.

[Cherry and Foster2012] Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 427–436, Montréal, Canada. ACL.

[Cherry2013] Colin Cherry. 2013. Improved reordering for phrase-based translation using sparse features. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 22–31, Atlanta, Georgia, June. Association for Computational Linguistics.

[Chiang et al.2008] David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 224–233. ACL.

[Chiang et al.2009] David Chiang, Kevin Knight, and Wei Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 218–226, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Chiang2005] David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics.

[Chiang2007] David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33.

[Chung and Gildea2009] Tagyoung Chung and Daniel Gildea. 2009. Unsupervised tokenization for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 718–726. Association for Computational Linguistics.

[Cocke1969] John Cocke. 1969. *Programming languages and their compilers: Preliminary notes*. Courant Institute of Mathematical Sciences, New York University.

[Cohn and Haffari2013] Trevor Cohn and Gholamreza Haffari. 2013. An infinite hierarchical bayesian model of phrasal translation. In *Proceedings of the Annual Meeting of Association for Computational Linguistics*.

[Cohn et al.2009] Trevor Cohn, Sharon Goldwater, and Phil Blunsom. 2009. Inducing compact but accurate tree-substitution grammars. In *Proceedings of Human Language Technologies: North American Chapter of the Association for Computational Linguistics-09*, pages 548–556. Association for Computational Linguistics.

[Daumé2004] Hal Daumé. 2004. Notes on CG and LM-BFGS Optimization of Logistic Regression. Software available for download from http://www.umiacs.umd.edu/~hal/megam/.

[de Gispert et al.2010a] Adrià de Gispert, Gonzalo Iglesias, Graeme Blackwood, Eduardo R. Banga, and William Byrne. 2010a. Hierarchical phrase-based translation with weighted finite-state transducers and shallow-n grammars. *Computational Linguistics*, 36.

[de Gispert et al.2010b] Adrià de Gispert, Juan Pino, and William Byrne. 2010b. Hierarchical phrase-based translation grammars extracted from alignment posterior probabilities. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 545–554. Association for Computational Linguistics.

[DeNero and Klein2007] John DeNero and Dan Klein. 2007. Tailoring word alignments to syntactic machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 17–24, Prague, Czech Republic, June. Association for Computational Linguistics.

[DeNero and Klein2010] John DeNero and Dan Klein. 2010. Discriminative modeling of extraction sets for machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1453–1463. Association for Computational Linguistics.

[DeNero et al.2008] John DeNero, Alexandre Bouchard-Cote, and Klein Dan. 2008. Sampling alignment structure under a bayesian translation model. In *Proceedings of Empirical Methods in Natural Language Processing-08*, pages 314–323. Association for Computational Linguistics.

[Denkowski and Lavie2011] Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, July. ACL.

[Devlin and Matsoukas2012] Jacob Devlin and Spyros Matsoukas. 2012. Trait-based hypothesis selection for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 528–532. ACL.

[Duh et al.2012] Kevin Duh, Katsuhito Sudoh, Xianchao Wu, Hajime Tsukada, and Masaaki Nagata. 2012. Learning to translate with multiple objectives. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju Island, Korea. ACL.

[Duh et al.2013] Kevin Duh, Baskaran Sankaran, and Anoop Sarkar. 2013. Multi-objective optimization problems in statistical machine translation. In *22nd International Conference on Multiple Criteria Decision Making (MCDM)*, Málaga, Spain.

[Dyer et al.2009] Chris Dyer, Hendra Setiawan, Yuval Marton, and Philip Resnik. 2009. The university of maryland statistical machine translation system for the fourth workshop on machine translation. In *Proc. of the Fourth Workshop on Machine Translation*.

[Fossum et al.2008] Victoria Fossum, Kevin Knight, and Steven Abney. 2008. Using syntax to improve word alignment precision for syntax-based machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 44–52. Association for Computational Linguistics.

[Foster and Kuhn2007] George Foster and Roland Kuhn. 2007. Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135. ACL.

[Foster et al.2010] George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459. ACL.

[Galley et al.2006] Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics*, pages 961–968. Association for Computational Linguistics.

[Ghahramani and Beal2000] Zoubin Ghahramani and Matthew J. Beal. 2000. Variational inference for bayesian mixtures of factor analysers. In *Advances in Neural Information Processing Systems 12*, pages 449–455. MIT Press.

[Gimpel and Smith2012] Kevin Gimpel and Noah A. Smith. 2012. Structured ramp loss minimization for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 221–231, Montréal, Canada. ACL.

[Hall et al.2011] Keith Hall, Ryan T. McDonald, Jason Katz-Brown, and Michael Ringgaard. 2011. Training dependency parsers by jointly optimizing multiple objectives. In *Proceedings of the Empirical Methods in Natural Language Processing*, pages 1489–1499.

[He and Deng2012] Xiaodong He and Li Deng. 2012. Maximum expected bleu training of phrase and lexicon translation models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 292–301, Jeju Island, Korea. ACL.

[He and Way2009] Yifan He and Andy Way. 2009. Improving the objective function in minimum error rate training. In *MT Summit*.

[He et al.2009] Zhongjun He, Yao Meng, and Hao Yu. 2009. Discarding monotone composed rule for hierarchical phrase-based statistical machine translation. In *Proceedings of the 3rd International Universal Communication Symposium*, pages 25–29. ACM.

[Hopkins and May2011] Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland. ACL.

[Iglesias et al.2009] Gonzalo Iglesias, Adrià de Gispert, Eduardo R. Banga, and William Byrne. 2009. Rule filtering by pattern for efficient hierarchical translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 380–388. Association for Computational Linguistics.

[Isozaki et al.2010] Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. ACL.

[Joachims2006] Thorsten Joachims. 2006. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. Software available for download from http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html.

[Johnson et al.2007] Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the EMNLP-CoNLL*.

[Johnson2002] Mark Johnson. 2002. The DOP estimation method is biased and inconsistent. *Computational Linguistics*, 28(1):71–76, March.

[Kasami1965] Tadao Kasami. 1965. An efficient recognition and syntax-analysis algorithm for context-free languages. Technical report, Air Force Cambridge Research Lab, Bedford, MA.

[Koehn et al.2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL ’03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Koehn et al.2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. ACL.

[Koehn2004] Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

[Kurihara and Sato2006] Kenichi Kurihara and Taisuke Sato. 2006. Variational bayesian grammar induction for natural language. In *International Colloquium on Grammatical Inference (ICGI)*, pages 84–96.

[Lavie and Denkowski2009] Alon Lavie and Michael J. Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23(2-3):105–115.

[Levenberg et al.2012] Abby Levenberg, Chris Dyer, and Phil Blunsom. 2012. A bayesian model for learning scfgs with discontiguous rules. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 223–232, Jeju Island, Korea. Association for Computational Linguistics.

[Li et al.2009] Zhifei Li, Jason Eisner, and Sanjeev Khudanpur. 2009. Variational decoding for statistical machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 593–601, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Liu et al.2011] Chang Liu, Daniel Dahlmeier, and Hwee Tou Ng. 2011. Better evaluation metrics lead to better machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

[Marcu and Wong2002] Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of Empirical Methods in Natural Language Processing-02*, pages 133–139. ACL.

[Marler and Arora2004] R. T. Marler and J. S. Arora. 2004. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, April.

[Mauser et al.2008] Arne Mauser, Saša Hasan, and Hermann Ney. 2008. Automatic evaluation measures for statistical machine translation system optimization. In *International Conference on Language Resources and Evaluation*, Marrakech, Morocco.

[May and Knight2007] Jonathan May and Kevin Knight. 2007. Syntactic re-alignment models for machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 360–368, Prague, Czech Republic, June. Association for Computational Linguistics.

[Neubig et al.2011] Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 632–641. Association for Computational Linguistics.

[Newman et al.2007] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2007. Distributed inference for latent dirichlet allocation. In *Proceedings of Neural Information Processing Systems-07*.

[Och and Ney2000] Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 440–447. Association for Computational Linguistics.

[Och and Ney2002] Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics.

[Och and Ney2003] Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

[Och and Ney2004] Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30:417–449.

[Och et al.1999] F. J. Och, C. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP99)*, pages 20–28, University of Maryland, College Park, MD, USA.

[Och2003] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the ACL*, pages 160–167. ACL.

[Padó et al.2009] Sebastian Padó, Daniel Cer, Michel Galley, Dan Jurafsky, and Christopher D. Manning. 2009. Measuring machine translation quality as semantic equivalence: A metric based on entailment features. *Machine Translation*, 23(2-3):181–193.

[Papineni et al.2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wie-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of Association of Computational Linguistics*, pages 311–318. ACL.

[Prescher et al.2004] Detlef Prescher, Remko Scha, Khalil Sima'an, and Andreas Zollmann. 2004. On the statistical consistency of dop estimators. In *Proceedings of the 14th Meeting of Computational Linguistics in Netherlands*, volume 111 of *CLIN*. University of Antwerp.

[Quirk et al.2005] Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: syntactically informed phrasal SMT. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 271–279. Association for Computational Linguistics.

[Razmara et al.2012] Majid Razmara, George Foster, Baskaran Sankaran, and Anoop Sarkar. 2012. Mixing multiple translation models in statistical machine translation. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju, Republic of Korea. ACL.

[Rissanen1983] Jorma Rissanen. 1983. A universal prior for integers and estimation by minimum description length. *The Annals of statistics*, pages 416–431.

[Saers et al.2009] Markus Saers, Joakim Nivre, and Dekai Wu. 2009. Learning stochastic bracketing inversion transduction grammars with a cubic time biparsing algorithm. In *Proceedings of the 11th International Conference on Parsing Technologies*, pages 29–32. Association for Computational Linguistics.

[Saers et al.2013a] Markus Saers, Karteek Addanki, and Dekai Wu. 2013a. Iterative rule segmentation under minimum description length for unsupervised transduction grammar induction. In Adrian-Horia Dediu, Carlos Martín-Vide, Ruslan Mitkov, and Bianca Truthe, editors, *Statistical Language and Speech Processing*, volume 7978 of *Lecture Notes in Computer Science*, pages 224–235. Springer Berlin Heidelberg.

[Saers et al.2013b] Markus Saers, Karteek Addanki, and Dekai Wu. 2013b. Combining top-down and bottom-up search for unsupervised induction of transduction grammars. In *Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 48–57, Atlanta, Georgia, June. Association for Computational Linguistics.

[Saers et al.2013c] Markus Saers, Karteek Addanki, and Dekai Wu. 2013c. Unsupervised transduction grammar induction via minimum description length. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation*, pages 67–73, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Sankaran and Sarkar2012] Baskaran Sankaran and Anoop Sarkar. 2012. Improved reordering for shallow-n grammar based hierarchical phrase-based translation. In *Proceedings of the North-American Chapter of Association of Computational Linguistics (NAACL)*, pages 533–537, Montréal, Canada.

[Sankaran et al.2011] Baskaran Sankaran, Gholamreza Haffari, and Anoop Sarkar. 2011. Bayesian extraction of minimal scfg rules for hierarchical phrase-based translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 533–541, Edinburgh, Scotland, jul. Association for Computational Linguistics.

[Sankaran et al.2012a] Baskaran Sankaran, Gholamreza Haffari, and Anoop Sarkar. 2012a. Compact rule extraction for hierarchical phrase-based translation. In *The 10th biennial conference of the Association for Machine Translation in the Americas (AMTA)*, San Diego, CA, oct. Association for Computational Linguistics.

[Sankaran et al.2012b] Baskaran Sankaran, Majid Razmara, and Anoop Sarkar. 2012b. *Kriya* – an end-to-end hierarchical phrase-based mt system. *The Prague Bulletin of Mathematical Linguistics (PBML)*, 97(97):83–98.

[Sankaran et al.2013a] Baskaran Sankaran, Gholamreza Haffari, and Anoop Sarkar. 2013a. Scalable variational inference for extracting hierarchical phrase-based translation rules. In *Submitted to the 6th International Joint Conference on Natural Language Processing*, Nagoya, Japan, oct. Association for Computational Linguistics.

[Sankaran et al.2013b] Baskaran Sankaran, Anoop Sarkar, and Kevin Duh. 2013b. Multi-metric optimization using ensemble tuning. In *Proceedings of the North-American Chapter of Association of Computational Linguistics (NAACL)*, Atlanta, GA. Association for Computational Linguistics.

[Servan and Schwenk2011] Christophe Servan and Holger Schwenk. 2011. Optimising multiple metrics with mert. *Prague Bull. Math. Linguistics*, 96:109–118.

[Shen2011] Libin Shen. 2011. Understanding exhaustive pattern learning. *CoRR*, abs/1104.3929.

[Simianer et al.2012] Patrick Simianer, Stefan Riezler, and Chris Dyer. 2012. Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–21, Jeju Island, Korea. ACL.

[Snover et al.2006] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

[Snyder et al.2009] Benjamin Snyder, Tahira Naseem, and Regina Barzilay. 2009. Unsupervised multilingual grammar induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 73–81, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Spitkovsky et al.2011] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2011. Lateen EM: Unsupervised training with multiple objectives, applied to dependency grammar induction. In *Proceedings of the Empirical Methods in Natural Language Processing*, pages 1269–1280. Association of Computational Linguistics.

[Vazirani2004] Vijay Vazirani. 2004. *Approximation Algorithms*. Springer.

[Venugopal and Narendran1990] V. Venugopal and T.T. Narendran. 1990. An interactive procedure for multiobjective optimization using nash bargaining principle. *Decision Support Systems*, 6(3):261 – 268.

[Watanabe et al.2007] Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773. ACL.

[Watanabe2012] Taro Watanabe. 2012. Optimized online rank learning for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 253–262, Montréal, Canada, June. ACL.

[Wolfe et al.2008] Jason Wolfe, Aria Haghighi, and Dan Klein. 2008. Fully distributed em for very large datasets. In *Proceedings of the 25th International Conference on Machine learning*, ICML '08, pages 1184–1191, New York, NY, USA. ACM.

[Yamada and Knight2001] Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530. Association for Computational Linguistics.

[Yang and Zheng2009] Mei Yang and Jing Zheng. 2009. Toward smaller, faster, and better hierarchical phrase-based smt. In *Proceedings of the ACL-IJCNLP*.

[Younger1967] Daniel H. Younger. 1967. Recognition and parsing of context-free languages in time n3. *Information and Control*, 10(2):189 – 208.

[Zaidan2009] Omar F. Zaidan. 2009. Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.

[Zens et al.2012] Richard Zens, Daisy Stanton, and Peng Xu. 2012. A systematic comparison of phrase table pruning techniques. In *Proceedings of the Empirical Methods in Natural Language Processing*.

[Zhang et al.2008a] Hao Zhang, Daniel Gildea, and David Chiang. 2008a. Extracting synchronous grammar rules from word-level alignments in linear time. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING) - Volume 1*, pages 1081–1088. Association for Computational Linguistics.

[Zhang et al.2008b] Hao Zhang, Chris Quirk, Robert C. Moore, and Daniel Gildea. 2008b. Bayesian learning of non-compositional phrases with synchronous parsing. In *Proceedings of ACL-08: HLT*, pages 97–105, Columbus, Ohio, June. Association for Computational Linguistics.

[Zheng et al.2012] Daqi Zheng, Yifan He, Yang Liu, and Qun Liu. 2012. Maximum rank correlation training for statistical machine translation. In *MT Summit XIII*.

[Zollmann et al.2008] Andreas Zollmann, Ashish Venugopal, Franz Och, and Jay Ponte. 2008. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical mt. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING) - Volume 1*, pages 1145–1152. Association for Computational Linguistics.

Appendix A.

Metric Sensitivity analysis

In the following we show some additional plots analyzing the sensitivity of different evaluation metrics. These plots show that BLEU is the easiest metric to tune for. In other words BLEU detects smaller changes in the translation outputs much more easily than some other metric, such as RIBES. The sensitivity of a particular metric M_i does not show much variance when the underlying system was optimized for the different combinations of metric.

In all the plots listed in this Appendix, the x-axis represents the percentage $(0\%, \ldots, 100\%)$ of sentences that were modified by one of four different operations: *delete*, *insert* , *replace* and *swap*. The experiment is repeated 5 times and the percentage changes in the metric scores are averaged. See Section 6.5.4 for additional details about the analysis.

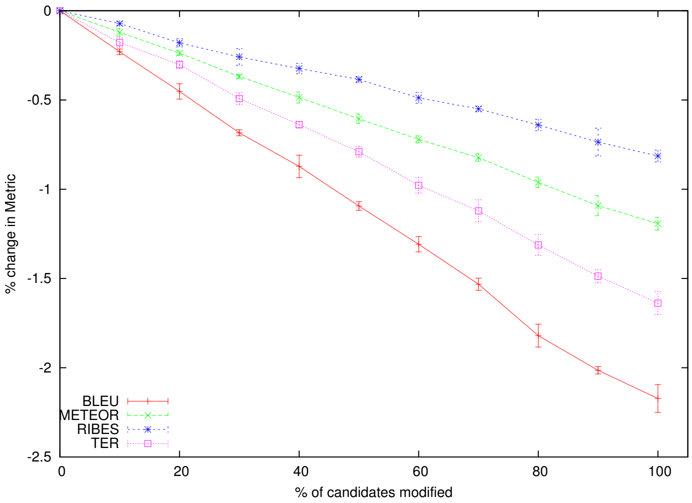
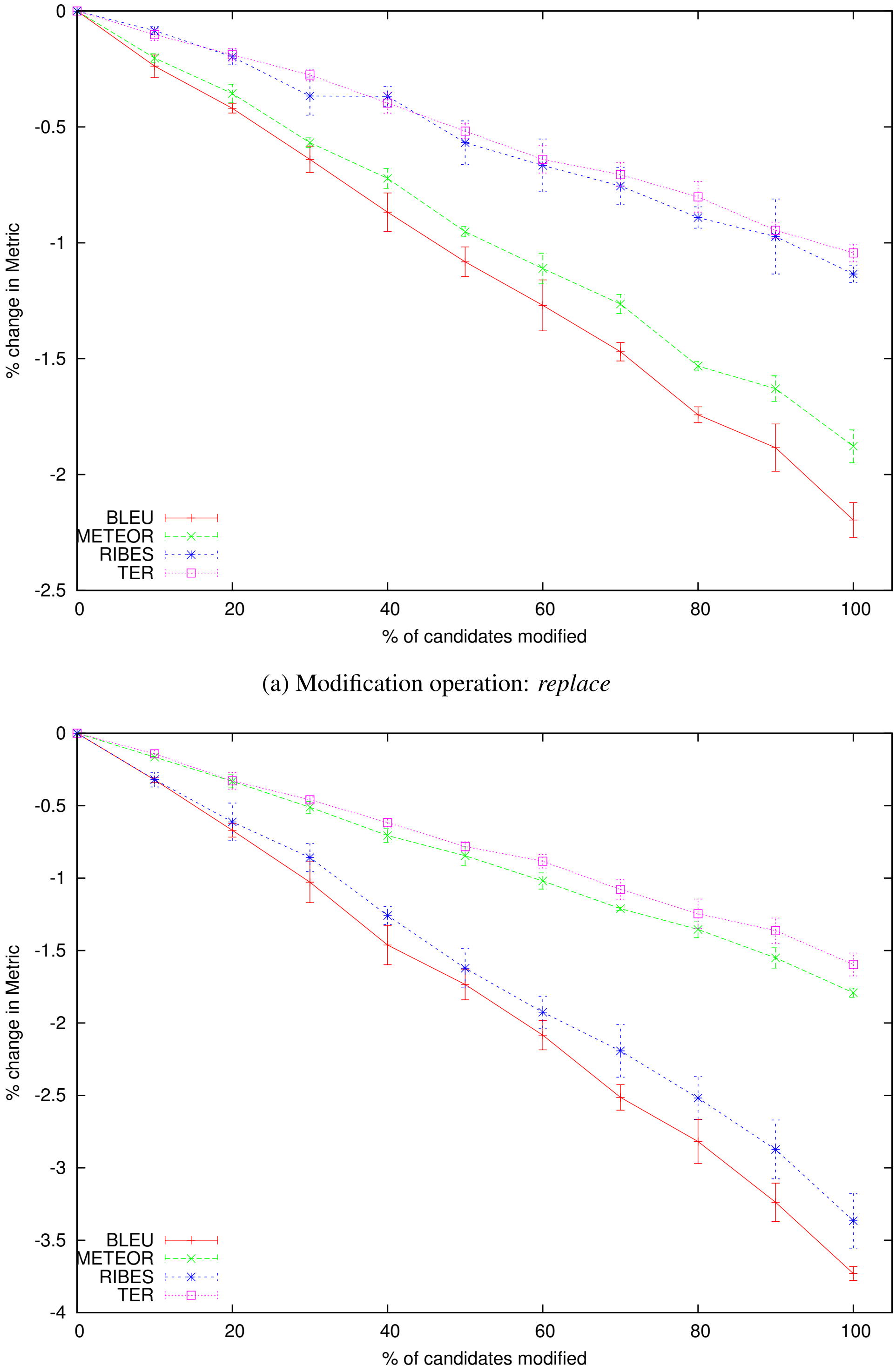


Figure A.1: **Zh-En multiple refs**: Modification operation: *insert*



(a) Modification operation: *replace*

(b) Modification operation: *swap*

Figure A.2: **Zh-En multiple refs**

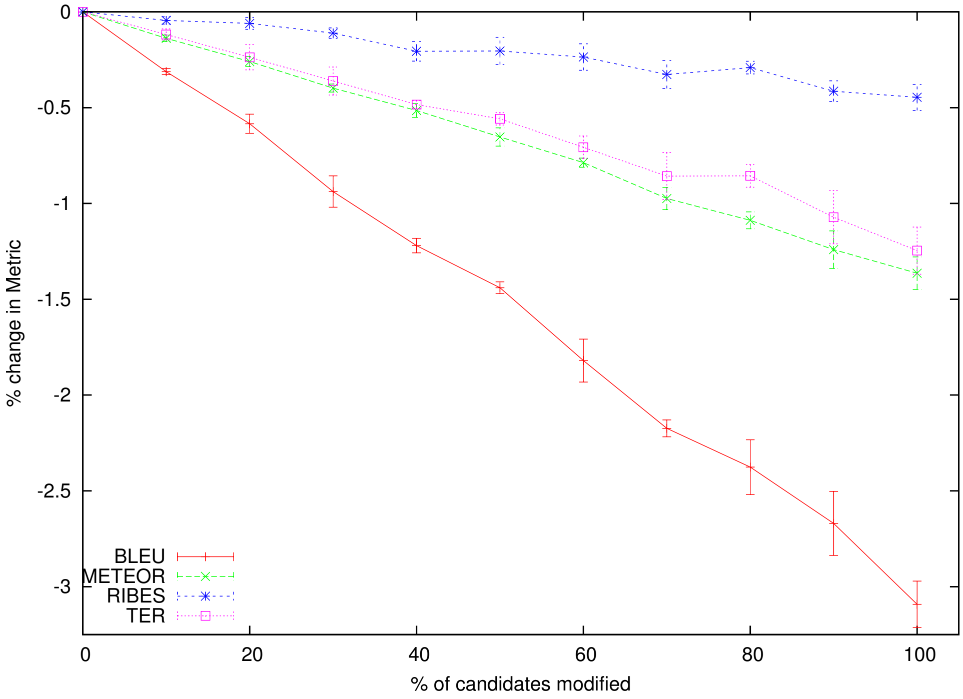
A.1 Effect of Target-metric on Metrics' Sensitivity

As we noted above, the synthetic experiments are performed on the test set outputs by using the weights optimized by the ensemble tuning methods for the metrics B-R-T (meta-weights: 0.455, 0.358 and 0.187 for BLEU, RIBES and TER respectively) for Chinese-English and B-R (meta-weights: 0.704 and 0.296 for B and R respectively) for Arabic-English.

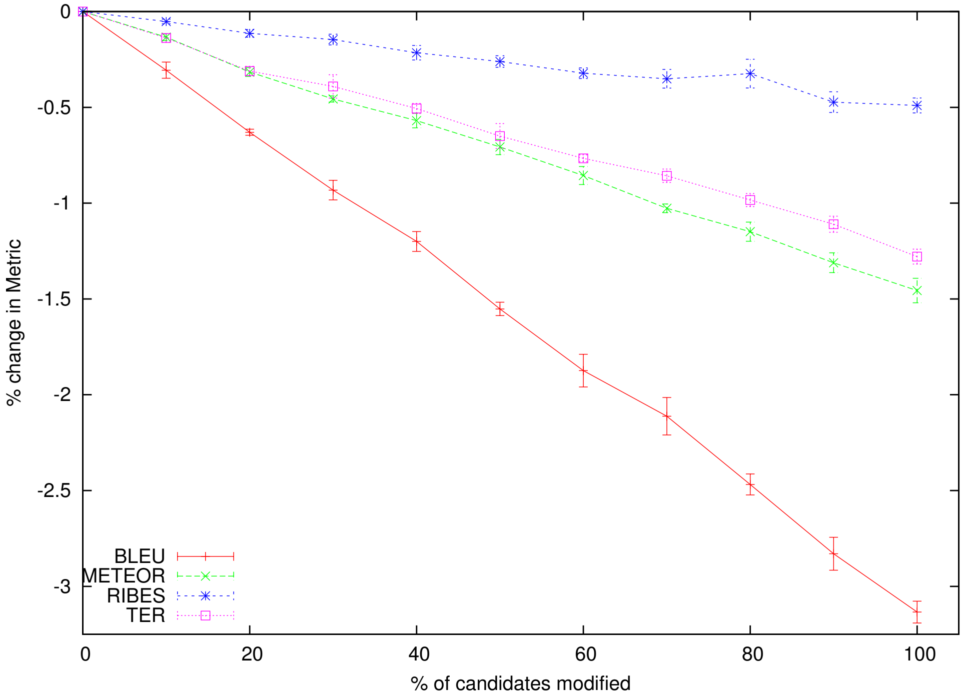
Even though RIBES gets reasonably high meta weights (around 0.3 in both languages), it does not seem to be sensitive to the changes. On the other hand none of these settings use METEOR as a target metric during optimization and similarly the Ar-En experiment does not use TER as target metric.

A natural question then would be "whether the choice of the target metric could influence the sensitivity of the different metrics". Since it is prohibitive to experiment with different combinations of the target metrics, we compare the sensitivity of different metrics in two scenarios, i) TER is used as a target metric and ii) TER is not used as a target metric. For the latter, we use the ensemble tuned (B-R) setting and for the former we use the system optimized only for TER. The plots compare the two scenarios for different edit operations for Arabic-English language pair.

Generally we see very small variations for the first half of the x-axis (where less then 50% of the sentences are corrupted). The amount of variation marginally increases as more sentences are corrupted by the edit operations. The standard deviations (error bars) of the BLEU are noticeably high for the ensemble tuned setting when compared with the TER- optimized setting. This is expected because the TER- optimized setting already gets a very low BLEU score (35.85) compared to the B-R ensemble tuned score (37.37). Since the BLEU is already degraded, randomly corrupting the sentence does not significant deteriorate the score and the degradation is gradual. Note that we can not directly compare the amount of percentage change between the two settings, because the absolute scores will be different.

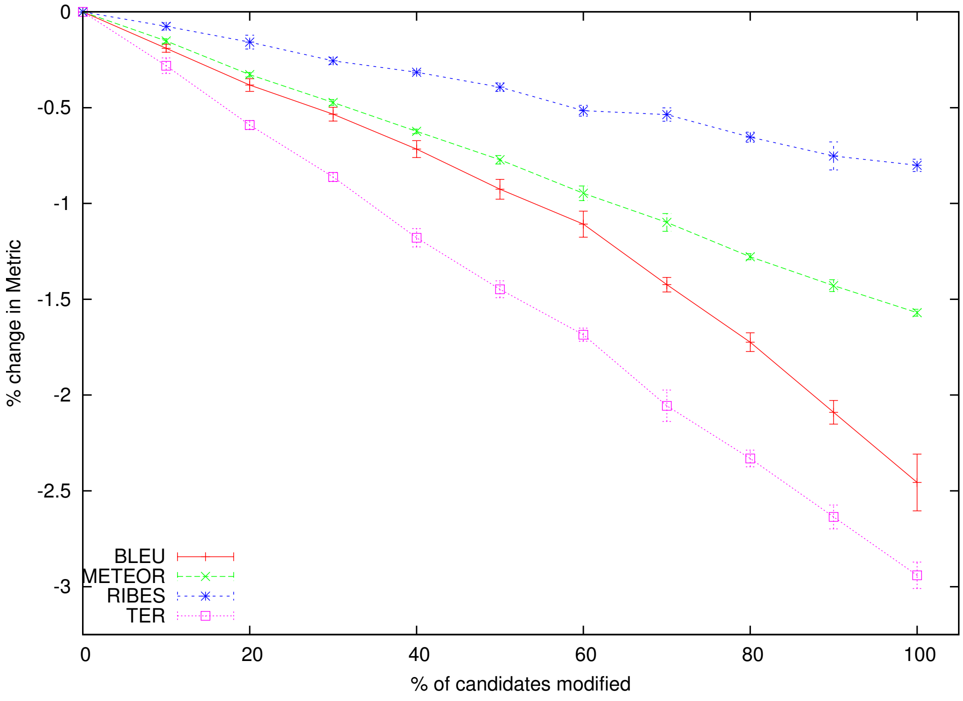


(a) Optimization setting: BLEU-RIBES Ensemble tune

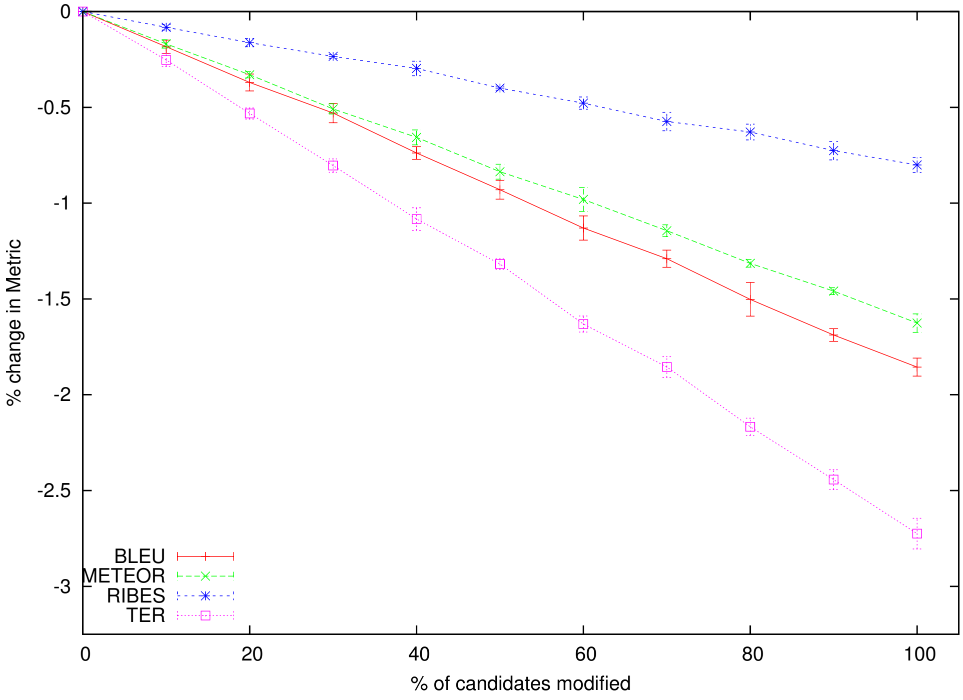


(b) Optimization setting: TER-only

Figure A.3: **Ar-En single ref** - Modification operation: *delete*

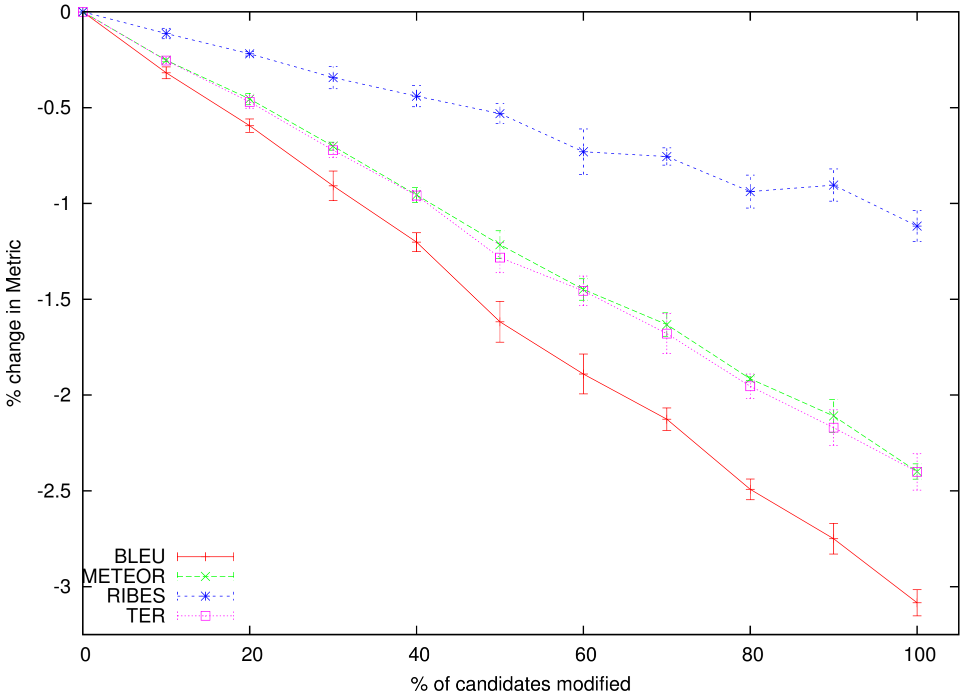


(a) Optimization setting: BLEU-RIBES Ensemble tune

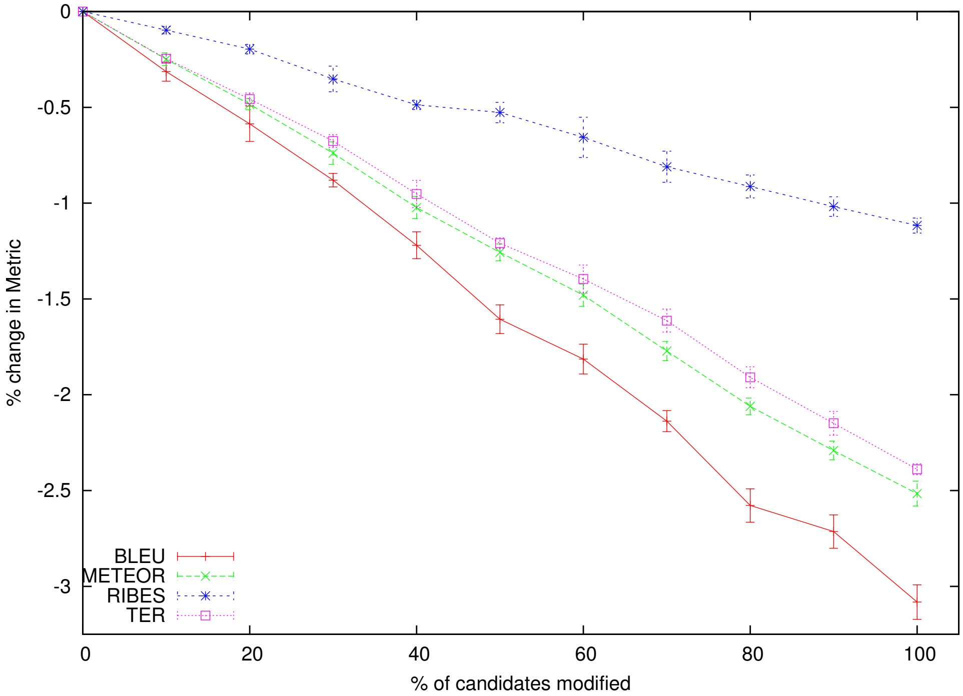


(b) Optimization setting: TER-only

Figure A.4: **Ar-En single ref** - Modification operation: *insert*

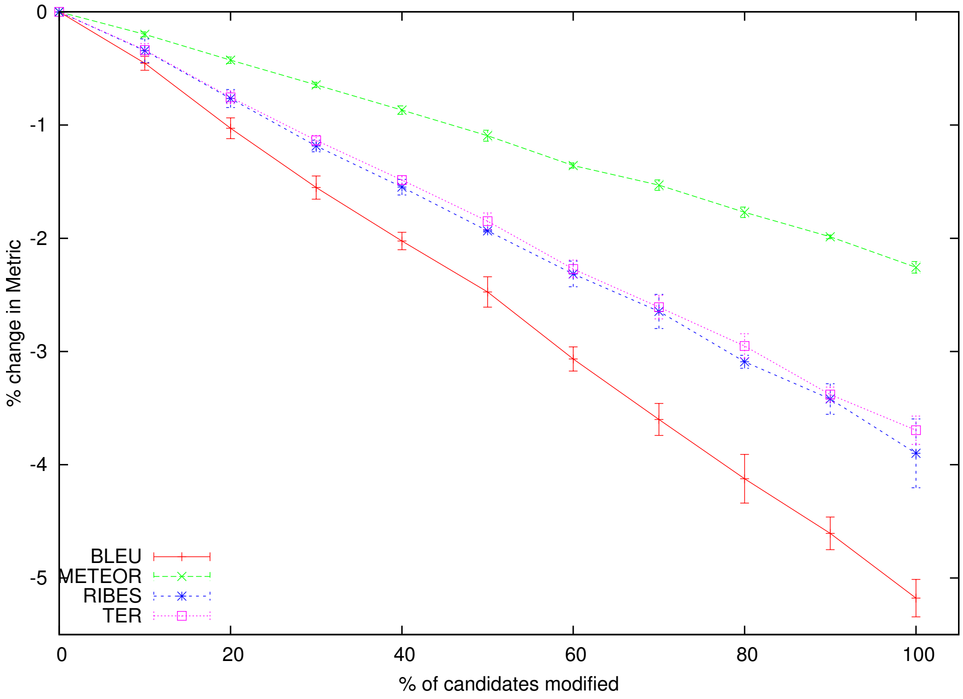


(a) Optimization setting: BLEU-RIBES Ensemble tune

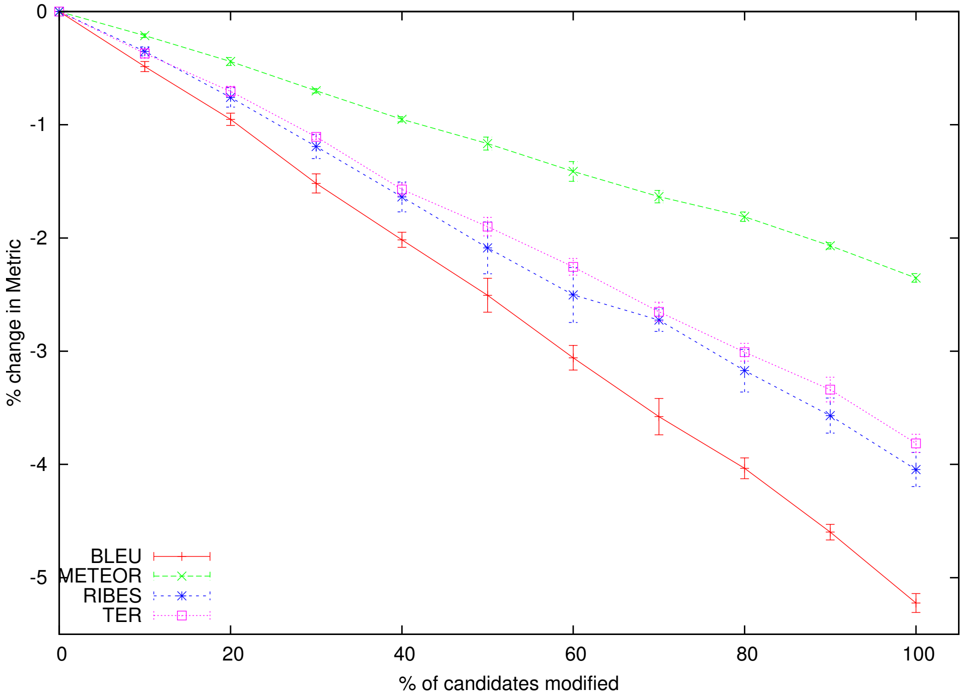


(b) Optimization setting: TER-only

Figure A.5: **Ar-En single ref** - Modification operation: *replace*

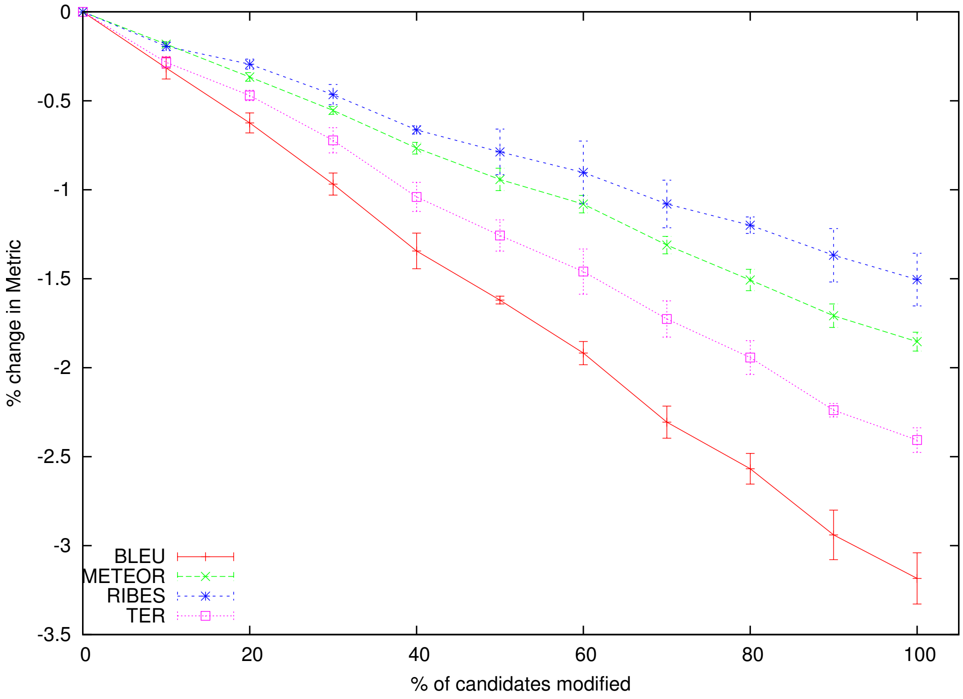


(a) Optimization setting: BLEU-RIBES Ensemble tune

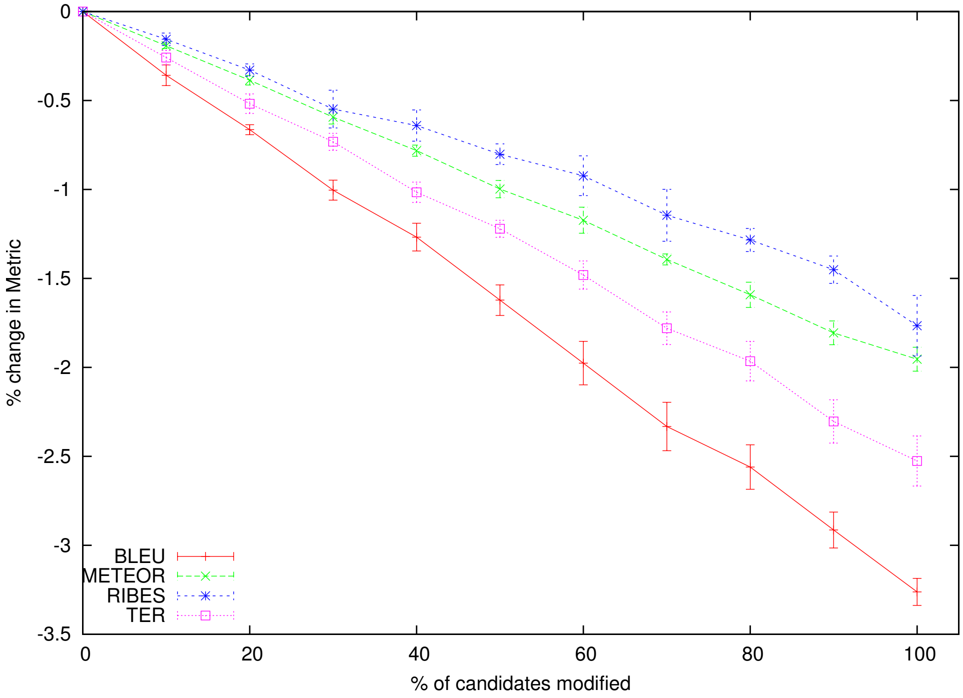


(b) Optimization setting: TER-only

Figure A.6: **Ar-En single ref** - Modification operation: *swap*



(a) Optimization setting: BLEU-RIBES Ensemble tune



(b) Optimization setting: TER-only

Figure A.7: **Ar-En single ref** - Modification operation: *any*