Co-training for Statistical Machine Translation

Chris Callison-Burch

Master of Science
Division of Informatics
University of Edinburgh
2002

Abstract

I propose a novel co-training method for statistical machine translation. As co-training requires multiple learners trained on views of the data which are disjoint and sufficient for the labeling task, I use multiple source documents as views on translation. Co-training for statistical machine translation is therefore a type of multi-source translation. Unlike previous mutli-source methods, it improves the overall quality of translations produced by a model, rather than single translations. This is achieved by augmenting the parallel corpora on which the statistical translation models are trained. Experiments suggest that co-training is especially effective for languages with highly impoverished parallel corpora.

Acknowledgements

I would like to thank Mark Steedman and Miles Osborne for allowing me to visit their summer workshop Johns Hopkins University. The discussions about applying co-training to parsing were useful when my own project was taking shape.

I am indebted to Franz Joseph Och, who gave me his EU corpus and saved me the trouble of having to collect the material myself.

I grateful to my fellow MSc students for their patience with my hogging the computational resources in our shared lab. All told, I used more CPU-time than there was in the entire project period.

A word of thanks must go to Marco Kuhlmann, for generally knowing everything.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Chris Callison-Burch*)

Table of Contents

Chapter 1

Introduction

Statistical machine translation (Brown et al. (1993)) is a technique that uses parallel corpora (documents in one language paired with their translations into another language) to automatically induce bilingual dictionaries and translation rules. By analyzing the co-occurrence and relative orderings of words in large amounts of such texts a statistical model of the translation process can be approximated. In order for statistical machine translation systems to achieve an acceptable level of translation quality, they must be trained on very large corpora. For example, the Candide system (Berger et al. (1994)), which translates between French and English, was trained on the Hansard corpus, a decade's worth of Canadian Parliament proceedings consisting of nearly three million parallel sentences. However, bilingual corpora as large as the Hansard corpus are extremely rare. In order for statistical machine translation to be possible between languages for which large parallel corpora are not available, one of two things need to be done:

1. Additional parallel corpora need to be assembled, or
2. Current statistical machine translation techniques need to be adapted to work with scarce linguistic resources.

This thesis combines the two: small amounts of parallel text are bootstrapped to create much larger parallel corpora.

Most machine learning techniques are *supervised*, that is, they rely crucially on labeled training data. Statistical machine translation falls into the category of super-

vised learning – it requires sentences "labeled" with their translations. Since labeled data must be created from unlabeled data at some cost, the amount of unlabeled data available is frequently much greater than the amount of labeled data. Because of this interest has developed in the area of *weakly supervised leaning* in which unlabeled data is utilized in addition to labeled data. Weakly supervised learning tries to reduce the cost associated with annotating data by having learners do it automatically. For example, in *self-training* a learner is trained from an initially small pool of labeled data, and then used to label additional data. The machine-labeled data is then added to the original pool of examples, and the learner is retrained. Charniak (1996) applied self-training to statistical parsing. Charniak trained a parser on one million words of parsed data, ran the parser over an additional 30 million words, and used the resulting parses to retrain the parser. Doing so gave a small improvement over just using the manually parsed data.

Co-training (Blum and Mitchell (1998)) is similar to self-training in that it increases the amount of labeled data by automatically annotating unlabeled data. Co-training differs from self-training because it uses multiple learners to do the annotation. The diversity of perspectives afforded by multiple learners produces more useful information for each learner than it would be able to produce for itself. Co-training is effective for a particular type of problem wherein the features used to label new items can be divided into distinct groups, where each group contains sufficient information to perform the annotation. For example, Blum and Mitchell (1998) apply co-training to web page classification. They train two naive Bayes classifiers on independent "views" of web pages: one uses the text of the web page itself, and one on the text of the hyperlinks pointing to the page. New pages can be classified using either of the learners, and then added to the pool of labeled examples from which both are trained. Blum and Mitchell show that iteratively retraining on the augmented data set provided significant increases in the performance of both learners.

While arbitrary features splits can be used to perform co-training, it is most effective when there is a natural division into distinct views (Nigam and Ghani (2000)). Statistical machine translation has a natural division of views. In machine translation "labels" are the target translations for source texts. The source text can therefore be

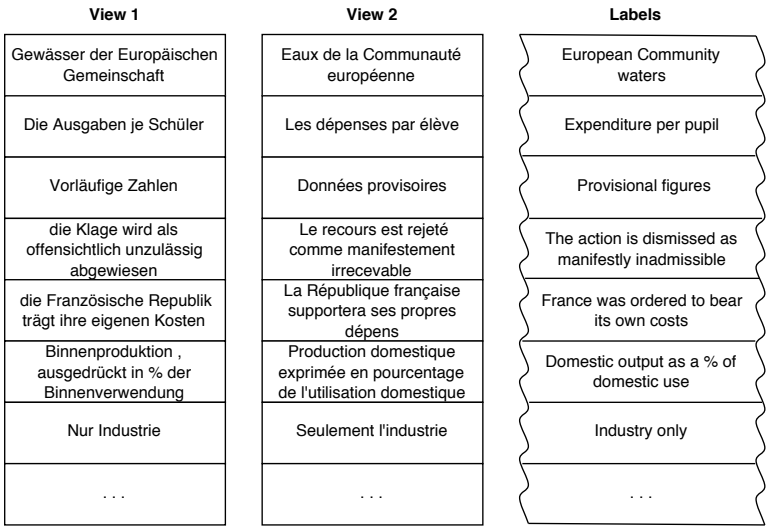


Figure 1.1: German and French as distinct views on English labels

considered a "view" on the translation. Other views that are sufficient for producing a translation would be existing translations of the source text into other languages. For example, a French text and its translation into German can be used as two distinct views, either of which could be used to produce a target translation into English (see Figure 1.1). Co-training can therefore be applied to statistical machine translation by using multiple sources. The use of multiple source documents to augment the quality of translation puts the method proposed in this thesis in the category of *multi-source translation* (Kay (2000)).

Kay observes that if a document is translated into one language, then there is a very strong chance that it will need to be translated into many languages. This is because international organizations publish legal documents in the languages of all of their member states, multi-national corporations produce product descriptions for many countries, and so forth. Kay (2000) proposes using multiple source documents as a way of informing subsequent machine translations, suggesting that much of what makes it hard to translate a text into another language may be resolved if a translation into some third language is available as a reference. Kay does not propose a method for how to go about performing this improvement, but instead challenges others to find general techniques that will exploit the information in multiple source to improve the

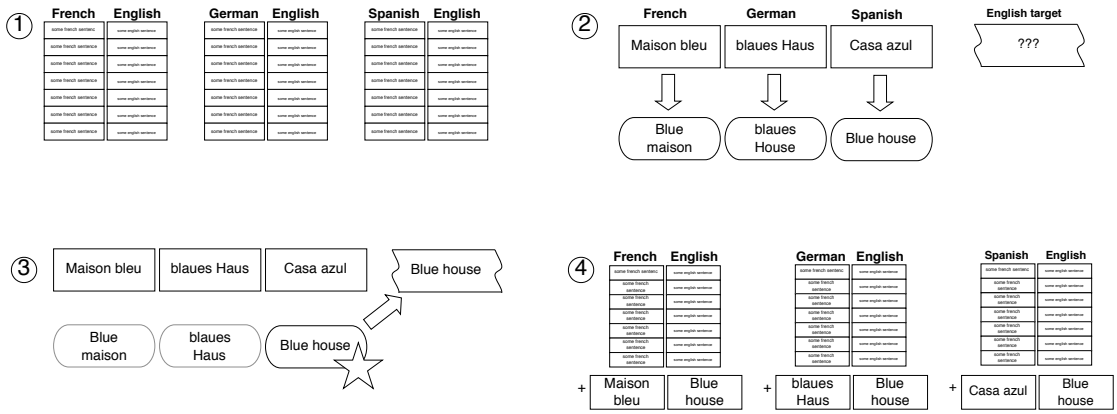


Figure 1.2: Co-training for Statistical Machine Translation

quality of machine translation.

This thesis describes a method that rises to that challenge. *Co-training for statistical machine translation* uses multiple source documents to augment the amount of data available for training machine translation systems. Figure 1.2 shows the process:

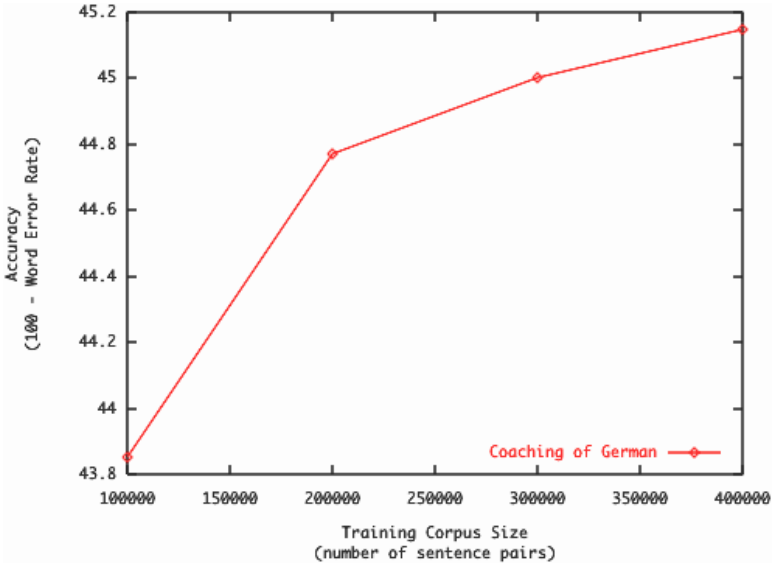
1. A number of statistical translation models are trained with an initially small set of bilingual parallel corpora
2. Those translation models are then used to translate a text with multiple sources
3. The best translation is picked from candidate translations, and aligned with the source texts
4. The alignment is added to each of the original bilingual corpora
5. The process is repeated, adding further machine-translated items to the bilingual corpora, and retraining the translation models on the augmented corpora.

Co-training for statistical machine translation thereby applies weakly supervised learning methods to statistical machine translation, and augments limited bilingual corpora with additional data.

This thesis empirically evaluates the efficacy of applying co-training to statistical machine translation. It uses a multi-source corpus assembled from European Union

web pages. The multi-source corpus contains text in German, French, Spanish, Italian, and Portuguese, and is used to create parallel English translations. The results of the experiment suggest that bootstrapping from additional machine-translated English data has a positive effect on translation quality between German and English, French and English, Spanish and English, and so on.

Additional experiments simulate the problem of trying translate from a language that lacks the linguistic resources necessary for statistical translation. Using a special case of co-training, which I call "coaching", I was able to achieve 45% accuracy for German to English translation (equivalent to training on about 20,000 human translate sentences) using *no human-translated data at all*. This was achieved by training from 400,000 machine-translated sentences produced by other translation models. The accuracy achieved by bootstrapping from the machine-translate data is shown in the following graph:



This suggests that co-training for machine translation would be useful for adapting existing translation resources for use with new languages.

The rest of my thesis is structured thusly:

- Chapter 2 gives an overview of statistical machine translation. It discusses how statistical translation models are learned from data, and gives details why so

much data is needed in order to achieve high quality translations.

- Chapter 3 introduces a method for bootstrapping the amount of available training data. It discusses Blum and Mitchell (1998) co-training scheme, and gives a subsequent formalization of the work as presented in Abney (2002). The chapter also describes a number of pervious applications of co-training to natural language processing tasks.

- Chapter 4 describes the novel method used to apply co-training to statistical machine translation. It details the use of multiple source documents as the independent views used to train classifiers. It further discusses the corpus that was assembled to test the method, and the evaluation metrics used.

- Chapter 5 presents experimental results. It begins with preliminary experiments that motivate increasing the size of training corpora. It describes various selection techniques that were used to add machine translations to the pool of training examples, and describes their effectiveness. It concludes with an illustration of the "coaching" variant.

- Chapter 6 discusses the implication of my work, and suggests future research directions.

Chapter 2

Statistical Machine Translation

In the early 1990s the increasing availability of machine-readable bilingual corpora, such as the proceedings of the Canadian Parliament which are written in both French and English, lead to investigation of ways of extracting useful linguistic information from them. Building on research into automatically aligning sentences with their translations across the languages in a bilingual corpus (such as Gale and Church (1993)), IBM researchers developed a statistical technique for automatically learning translation models using parallel corpora as training data (Brown et al. (1993)).

Extending the work on aligning sentences, Brown et al. addressed the problem of matching up words within the aligned sentences. The IBM statistical translation models use an algorithm developed for estimating the probability that a French word will be translated into any particular English word, incorporating it into a "statistical model of the translation process" to align the words in a French sentence with the words in its English translation. Brown et al. describe the translation process in statistical terms as follows:

> A string of French words, f, can be translated into a string of English words in many different ways. Often, knowing the broader context in which f occurs may serve to winnow the field of acceptable English translations, but even so, many acceptable translations will remain; the choice among them is largely a matter of taste. In statistical translation, we take the view that every English string, e, as a possible translation of f. We assign to every pair of strings $< f, e >$ a number $P(e|f)$, which we interpret as the probability that a translator, when presented with f will produce e as its translation.

Given a French string f, the job of the machine translation system is to find that English string $\hat{e}$, from all possible English strings for which $P(e|f)$ is greatest.

$$P(e|f) = \frac{P(e)P(f|e)}{P(f)}$$

$$P(e|f) \propto P(e)P(f|e)$$

$$\hat{e} = \text{argmax}_e \, P(e)P(f|e)$$

Brown et al. call this the *Fundamental Equation of Statistical Machine Translation*. Though these equations discuss translations between French and English, any languages maybe used. The target language is signified by e, and the source by f.

The terms $P(e)$ and $P(f|e)$ in the Fundamental Equation of Statistical Machine Translation can be though of as the *language model probability* and the *translation model probability*. The language model probability is essentially the same as that for speech recognition or other such recognition tasks; it represents a prior belief about how likely the translation is a sentence in the target language. Another way of thinking about the language model probability is as an estimate of the grammaticality of the translation. This notion of grammaticality could either be achieved directly by using a probabilistic grammar, or approximated using a standard n-gram language model. Separating out grammaticality into another term frees the translation model probability from having to encode it. This simplifies the estimation of $P(f|e)$ and motivates reducing the problem using Bayes' rule, rather than trying to estimate $P(e|f)$ directly.

The purpose of the Fundamental Equation of Statistical Machine Translation is to assign a high probability to well-formed English sentences that account well for an French source sentence. The language model assigns high probability to well-formed English strings regardless of their connection to the French. The translation model assigns high probability to English sentences (regardless of grammaticality) that have the necessary words in them in roughly the right places to account for the French. The product of the two is weighted towards the best translation. Together they assign a high probability to translations that are reasonable. Figure 2.1 shows how the two models interact when translating the French sentence "John est passè a la tele." The tick marks indicate which sentences pass each model by being assigned a high probability.

	$P(e)$	$P(f\|e)$
John appeared in TV.		✓
In appeared TV John.		
John is happy today.	✓	
John appeared on TV.	✓	✓
TV appeared on John.	✓	

Figure 2.1: Interaction of language and translation model probabilities (taken from Al-Onaizan et al. (1999))

Brown et al. focus on the translation modeling component of the translation problem.

2.1 Translation Modeling

There are a number of techniques that could be used to translate between English and French strings. One technique is an interlingua representation wherein the source language string gets converted into predicate logic, or some other sort of logical representation (like Copestake et al. (1995)), which is used to generate the target language string. For example the source sentence "John must not go" would get converted into OBLIGATORY(NOT(GO(JOHN))) whereas "John may not go" would get converted into NOT(PERMITTED(GO(JOHN))). The logical representation captures the correct interaction of "not" with the modal verbs despite the syntactic similarity of the sentences. The predicate logic could then be used to generate French sentences. Due to the lack of corpora with semantic annotation, there are no statistical approaches to interlingual machine translation.

Another technique is syntactic transfer wherein a parse tree is created for the source sentence, and assigns syntactic relationships between heads and modifiers, such as subject/verb, adjective/noun, prepositional phrase/verb phrase, etc. This parse tree is then transformed into a tree in the target language by reordering phrases, replacing English words with French translations, etc., while obeying linear precedence constraints on the ordering of heads and modifiers in the target language. Recent work has formu-

1. Mary did not slap the green witch
   Begin with an English sentence that will be rewritten into Spanish.

2. Mary not slap slap slap the green witch
   Choose fertilities for each of the English words. Here "did" has a fertility of zero and "slap" has a fertility of three.

3. Mary not slap slap slap NULL the green witch
   Choose the number of spurious words to be inserted, and insert a NULL word for each one. Here one spurious word is added.

4. Mary no daba una botefada a la verde bruja
   Choose Spanish translations for each of the English words, including NULL.

5. Mary no daba una botefada a la bruja verda
   Finally, choose target positions for each of the words, and reorder them.

Figure 2.2: Translation as string rewriting

lated a statistical transfer-based translation model (Yamada and Knight (2001)), but requires parsing machinery for the language being translated between.

Another much simpler technique is to treat translation as string rewriting, wherein the words in an source sentence are replaced by words in the target language, which are then reordered in some fashion. For example, every word in an English sentence could be replaced with zero or more Spanish words. Additionally, a number of "spurious" Spanish words might be inserted with no direct connection to the original English words. The Spanish words could then be rearranged into a better order. Figure 2.2 illustrates this process. Brown et al. (1993) use string rewriting as the core of their statistical translation models.

String rewriting fails to capture the often subtle mapping between meaning and surface form, which the interlingua approach does successfully. It further fails to capture syntactic relationships between words within a sentence in the source language, and their correspondences to the syntactic relationships in the target language, which the transfer based approach does successfully. However, it does have one main advantage

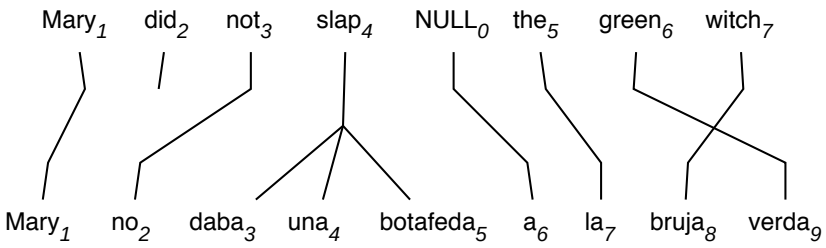


Figure 2.3: An alignment between a source string and its translation

over the other two approaches: it is feasible to learn string rewriting rules from available data. It is therefore a good technique for creating statistical translation models.

Brown et al. (1993) describes a method for using a bilingual corpus to produce a translation system. They first parameterize string rewriting into four components: word fertilities, word-for-word translations, target positions for word reordering, and spurious word insertion. They use these parameters to define *alignments* between sentences and their translations. Figure 2.3 gives a graphical representation of the alignment for the translation from Figure 2.2. The alignment can also be represented as a vector: (1 3 4 4 4 0 5 7 6). The vector shows which of the source words gave rise to which words in the translation. The first word in the target sentence "Mary" comes from the first word in the source sentence, the second word "no" comes from the third word in the source sentence, and the "daba", "una", and "botefada" all come from the forth word, and so on.

The probability of an alignment vector a given an English string e and a French string f is written as $P(a|e,f)$, and can be equated with the translation model probability term $P(f|e)$ of the Fundamental Equation of Statistical Machine Translation:

$$P(a|e,f) = \frac{P(a,e,f)}{P(e,f)}$$

$$= \frac{P(a,f|e) * P(e)}{P(f|e) * P(e)}$$

$$= \frac{P(a,f|e)}{P(f|e)}$$

$$P(f|e) = \sum_{a} P(a, f|e)$$

The probability of a translation f given a source string e is equivalent to the sum of the probabilities of all possible alignments between f and e. The probability of an alignment for a translation given a source string is calculated using the probability of each of the components involved in string rewriting:

translation probabilities	$t(f_j\|e_i)$	The probability that a French word f_j is the translation of an English word e_i.
fertility probabilities	$n(\phi_i\|e_i)$	The probability that a word e_i will expand into ϕ_i words in the target language.
spurious word probability	p	The probability that a spurious word will be inserted at any point in a sentence.
distortion probabilities	$d(p_i\|i,l,m)$	The probability that a target position p_i will be chosen for a word given the index of the English word that this was translated from i, and the lengths of the English source string l and French target string m.

The probability of an alignment $P(a, f|e)$ is calculated as the product of the fertility probabilities of each of the words in source sentence, times the product of the translations between each pair of connected words, times the target positions selected for each of the French words:[1]

$$P(a, f|e) = \prod_{i=1}^{l} n(\phi_i|e_i) * \prod_{j=1}^{m} t(f_j|e_i) * \prod_{j=1}^{m} d(j|a_j, l, m)$$

2.2 Estimating Parameters

The t, n, and d probability tables would be simple to generate if given a bilingual corpus in which translations were aligned on the word-level as in Figure 2.3. The number

[1] The true equation also includes the probabilities of spurious words arising from the "NULL" word at position zero of the English source string, but it is simplified here for clarity.

of times the French word "mason" occurred as the translation of "house" divided by the total number of times "house" occurred would give $t(\textit{mason} \mid \textit{house})$. The number of times the first word in English sentences of length 10 was connected to the second word in French sentences of length 12, divided by the total number of times English sentences with 10 words were translated with 12 French words would give $d(2 \mid 1, 10, 12)$. And so on.

Unfortunately, since bilingual corpora aligned on the word-level do not exist, direct estimation is not possible. Instead one must use bilingual corpora aligned on the sentence-level to estimate the word-level alignments. This is a difficult task for a number of reasons. Firstly, sentences have a huge number of possible alignments because of the interaction of the translation and distortion probabilities. Without first knowing which words translate to which other words, it is extremely difficult to estimate the distortion parameters, since any word could align with any other word and therefore take any target position in the translated sentence. Secondly, translation probabilities are difficult to estimate without first knowing how many words in the target language each word translates to. Given this simple model, a word could translate to anywhere between zero words to an entire sentence.

Brown et al. (1993) uses expectation maximization (EM) to address the problem of recovering word-level alignments from sentence-aligned corpora. A variety of EM algorithms are commonly used for estimating hidden variables, such as word-level alignments. EM searches for the maximally likely parameter assignments by trying to minimize the perplexity of a model. Perplexity is a measure of the "goodness" of a model. The assumption is that a good translation model will assign a high $P(f|e)$ to all sentence pairs in the some bilingual test data. We can measure the cumulative probability assigned by any given model by taking the product of the probability that it assigns to all of the sentence pairs in some testing data. Then comparing models is simply a matter of comparing the resulting product. A model which assigns a higher probability to all of the test data will be better than a model which assigns a lower probability.

However despite the fact that EM is guaranteed to improve a model on each iteration, the algorithm is not guaranteed to find a globally optimal solution. Since there are

so many factors contributing towards $P(a, f|e)$, and because those factors are equally weighted, it would be easy for EM to work towards a suboptimal local maximum. For example, EM could work towards optimizing an imagined correspondence of d positions in source and target sentences rather than actually optimizing the translation probabilities for words in the t table. The search path that optimized some aspect of d while neglecting t would reach a local optima, but would not reach a suitable parameter set for translation. In situations where there is a large search space, such as this one, the EM algorithm is greatly affected by initial starting parameters.

To address this search problem Brown et al. first train a simpler model to find sensible estimates for the t table, and then use those values to prime the parameters for incrementally more complex model which estimate d and n.

2.3 Training

Knight (1999) describes the incremental models used by Brown et al. to estimate the parameters of the translation model. The first model, called IBM Model 1, ignores distortion probabilities and spurious word introduction, and requires that each word have a fertility of one. Given the simplifications, the only thing that bears on the alignment probabilities are the word translation t parameter values. So the initial $P(a, f|e)$ formula can be expressed as

$$P(a, f|e) = \prod_{j=i}^{m} t(f_j|e_i)$$

Figure 2.4 shows how IBM Model 1 can be used to estimate the translation parameters for four words contained in the two sentence pairs $<$ *maison bleue*, *blue house* $>$ and $<$ *maison*, *house* $>$. The translation probabilities are initially uniformly set. Then through a process of iterative re-estimation, which involves changing the weight of the two possible alignments for $<$ *maison bleue*, *blue house* $>$ given the one alignment for $<$ *maison*, *house* $>$, the translation probabilities for the four words are made more linguistically plausible. If we had been given word-aligned sentence pairs we could calculate the translation probabilities directly. This method lets us bootstrap better translation probabilities by examining the likelihood of all possible alignments.

Step 1. Set parameter values uniformly.

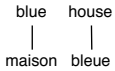
$t(bleue|house) = \frac{1}{2}$

$t(maison|house) = \frac{1}{2}$

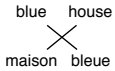
$t(bleue|blue) = \frac{1}{2}$

$t(maison|blue) = \frac{1}{2}$

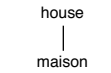
Step 2. Compute $P(a, f|e)$ for all alignments.



$P(a, f|e) = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$

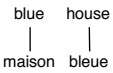


$P(a, f|e) = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$

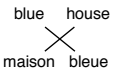


$P(a, f|e) = \frac{1}{2}$

Step 3. Normalize $P(a, f|e)$ values to yield $P(a|e, f)$ values.



$P(a|e, f) = \frac{1}{4}/\frac{2}{4} = \frac{1}{2}$



$P(a|e, f) = \frac{1}{4}/\frac{2}{4} = \frac{1}{2}$

$P(a|e, f) = \frac{1}{2}/\frac{1}{2} = 1$ There's only one alignment, so $P(a|e, f)$ will be 1 always.

Step 4. Collect fractional counts.

$tc(bleue|house) = \frac{1}{2}$

$tc(maison|house) = \frac{1}{2} + 1 = \frac{3}{2}$

$tc(bleue|blue) = \frac{1}{2}$

$tc(maison|blue) = \frac{1}{2}$

Step 5. Normalize fractional counts to get revised parameter values.

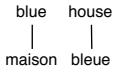
$t(bleue|house) = \frac{1}{2}/\frac{4}{2} = \frac{1}{4}$

$t(maison|house) = \frac{3}{2}/\frac{4}{2} = \frac{3}{4}$

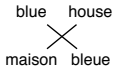
$t(bleue|blue) = \frac{1}{2}/1 = \frac{1}{2}$

$t(maison|blue) = \frac{1}{2}/1 = \frac{1}{2}$

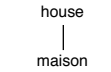
Repeat Step 2. Compute $P(a, f|e)$ for all alignments.



$P(a, f|e) = \frac{1}{4} * \frac{1}{2} = \frac{1}{8}$

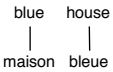


$P(a, f|e) = \frac{3}{4} * \frac{1}{2} = \frac{3}{8}$

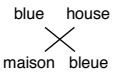


$P(a, f|e) = \frac{3}{4}$

Repeat Step 3. Normalize $P(a, f|e)$ values to yield $P(a|e, f)$ values.



$P(a|e, f) = \frac{1}{4}$



$P(a|e, f) = \frac{1}{4}/\frac{2}{4} = \frac{3}{4}$

$P(a|e, f) = 1$

Repeat Step 4. Collect fractional counts.

$tc(bleue|house) = \frac{1}{4}$

$tc(maison|house) = \frac{3}{4} + 1 = \frac{7}{4}$

$tc(bleue|blue) = \frac{3}{4}$

$tc(maison|blue) = \frac{1}{4}$

Repeat Step 5. Normalize fractional counts to get revised parameter values.

$t(bleue|house) = \frac{1}{8}$

$t(maison|house) = \frac{7}{8}$

$t(bleue|blue) = \frac{3}{4}$

$t(maison|blue) = \frac{1}{4}$

Repeating steps 2-5 many times yields:

$t(bleue|house) = 0.0001$

$t(maison|house) = 0.9999$

$t(bleue|blue) = 0.9999$

$t(maison|blue) = 0.0001$

Figure 2.4: Using EM to estimate translation values from two sentences.

The translation values t estimated from IBM Model 1 are then given as the starting conditions to IBM Model two which adds further complexity of the distortion probabilities d to the calculation of the probability of alignments. The t and d values produced through using EM on Model 2 are transfered to Model 3, the equation for which is given at the end of Section 2.1. In incrementally increasing the complexity of the way that the probability of an alignment is calculated, Brown et al. lessen the effects that starting conditions can have on the outcome of the EM algorithm. Transferring parameters reduces the chance that EM will find a suboptimal local maxima for the final values of the translation model parameters.

However, estimating parameters that will be useful in producing reasonable-quality translations is not only constrained by the features of the EM algorithm; it is also constrained by the amount of available training data.

2.4 Parallel Corpora

An additional difficulty in estimating values for the the parameters of statistical machine translation is the sheer number of parameters. Of the three main components used to estimate the probability of an alignment between two sentences, the translation table t has the largest number of parameters. Because t gives the probability of any particular word being translated to any other word, the size of the t table is the size of vocabulary squared. The distortion probability $d(p_i|i,l,m)$ gives the probability that the translation of a word at position i will take target position p_i given a source string of length l and target string of length m. The number of parameters for d is essentially the length of the longest sentence to the power of four. The fertility table f is the size of the vocabulary times the size of the longest sentence. To get reasonable estimate for all those parameters requires a lot of training data.

The translation system Candide (Berger et al. (1994)), which was developed to demonstrate the mathematical theory detailed in Brown et al. (1993) was trained using the Hansard Corpus. The Hansard Corpus is drawn from over a decade's worth of Canadian Parliament proceedings, and contains nearly 2.87 million parallel sentence pairs. Such large collections of machine-readable parallel corpora are extremely rare,

however. Current corpus-based machine translation techniques do not work well when given scarce linguistic resources. The problem of limited amounts of parallel text needs to be addressed in order to create statistical machine translation systems for languages which do not have extensive parallel corpora available. Furthermore, evidence suggests that machine translation can be improved by increasing the amount of training data even for language pairs like French-English for which large parallel corpora do already exist (as shown in Figure 5.1).

Various approaches have been taken to address the problem of scarce training data for machine translation:

- Resnik (1998) describes a method for mining the web for bilingual texts. The STRAND method automatically gathers web pages that are potential translations of each other by looking for documents in one language which have links whose text contains the name of another language. For example if an English web page had a link with the text "Español" or "en Español" then the page linked to is treated as a candidate translation of the English page. Further checks verify the plausibility of its being a translation (Smith (2002)).

- Al-Onaizan et al. (2000) investigates how human translators cope with scarce linguistic resources by designing an experiment in which human beings were asked to translate an unknown language (Tetun, which is spoken on East Timor) into English on the sole basis of a very small bilingual text. Participants performed quite well and debriefings revealed a number of interesting strategies that one could potentially incorporate into a machine translation system.

- Koehn and Knight (2000) describes a method for improving word translation probabilities using unrelated monolingual corpora. The method treats the problem of choosing the right word for translation among several possible translations as an instance of word sense disambiguation. The context provided by monolingual texts helps with word choice in translation.

This thesis investigates yet another way of dealing with the limited availability of bilingual corpora. Rather than trying to increase the size of bilingual data or trying to add extra mechanisms to the translation model, I propose a simple, inexpensive approach.

I employ *weakly supervised learning* that leverages limited bilingual corpora, and uses them to create larger corpora. *Co-training* for statistical machine translation uses a small amount of human-translated data to create incrementally larger corpora that incorporate machine translated data.

Co-training is explained in depth in the next chapter, and its application to statistical machine translation is described in Chapter 4 .

Chapter 3

Co-training

Most machine learning techniques crucially rely on labeled training data. These *supervised* learning techniques generally use data that has been hand-labeled, or assembled at considerable cost. Because labeled data must be created from unlabeled data with some associated cost, there is frequently much more unlabeled data available than labeled data. There has recently been interest in the area of *weakly supervised leaning*, in which unlabeled data is utilized in addition to labeled data. Blum and Mitchell (1998) and Mitchell (1999) develop a method called *co-training*, which which uses unlabeled data to boost performance of a learning algorithm when only a small set of labeled examples is available.

The co-training method uses unlabeled data to improve learning accuracy for a certain type of problem wherein new items can be labeled using different "views". That is, the features that are used by some learner to label an item must be divisible into independent groups, or views, and that each view must be sufficient in and of itself for labeling items. The example problem that Blum and Mitchell use to illustrate this is web page classification, wherein web pages are classified into some category like "computer science course page" or "faculty home page". Blum and Mitchell observe that the web contains an interesting kind of redundant information about each web page. Figure 3.1 shows a training example of a "faculty home page". The task of classifying the page could be achieved either by considering just the words in the web page, or by considering just the words in the text of the links that point the the web page. When examples can be described by two different sources of information, each

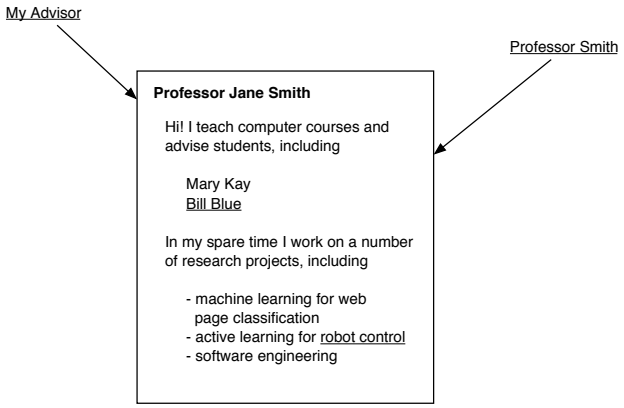


Figure 3.1: Hyperlinks and page text as views on web page classification

capable of training a learner for the classification task, unlabeled data can be used to boost learning accuracy.

Sufficiently redundant views can be used to train two independent classifiers rather than a single classifier. The unlabeled data is then used as follows: each classifier is allowed to examine the unlabeled data and to pick its most confidently predicted positive and negative examples, and add these to the set of labeled examples. Each classifier is thereby augmenting the pool of labeled examples. Both classifiers are retrained on the augmented set of labeled examples, and the process is repeated. Figure 3.2 summarizes this co-training algorithm.[1]ote that this is only one of many possible co-training algorithms.

The intuition on why retraining ought to lead to more accurate classifiers is that if the hyperlink classifier finds an "easily classified" hyperlink in the unlabeled data (that is, one that is quite similar to one of the labeled examples on which it was trained), the web page that it points to be will added to the labeled pool of examples as well. Of course, just because the hyperlink happened to be easy to classify does not mean the web page will be easily classified by the other classifier. If not then the hyperlink classifier has added useful training information to improve the other classifier. Similarly, the web page classifier can add examples that provide useful information to improve the accuracy of the hyperlink classifier.

[1]N

Given:

- a set L of labeled training examples
- a set U of unlabeled examples

Create a pool U' of examples by choosing u examples at random from U

Loop for k iterations:

Use L to train a classifier c_1 that considers only the v_1 portion of each item

Use L to train a classifier c_2 that considers only the v_2 portion of each item

Allow c_1 to label p positive and n negative examples from U'

Allow c_2 to label p positive and n negative examples from U'

Add these self-labeled examples to L

Randomly choose $2p+2n$ examples from U to replenish U'

Figure 3.2: The co-training algorithm (as presented in Blum and Mitchell (1998))

This intuitive explanation that co-training will improve results is concretized in Blum and Mitchell (1998) with an empirical investigation that shows that the error rates of both the page-based and hyperlink-based classifiers can be reduced. Training naive Bayes classifiers on bags of words extracted from the web page and from the text of the hyperlinks that point to them, Blum and Mitchell bootstrap from an initial training pool of a dozen labeled examples to a training pool containing ten times that many examples. The error rate decreases for both classifiers, and was halved in the cause of the page-based classified. Blum and Mitchell go on two prove that if two views are *conditionally independent* then any weak classifier can be boosted to arbitrarily high accuracy using unlabeled examples only through co-training.

However, as Abney (2002) points out, and as in demonstrated in the experiments described below, conditional independence is an unreasonably strict assumption. Abney reformulates the proof of efficacy for co-training by crucially integrating the agreement rate between classifiers. Blum and Mitchell's intuitive explanation on why co-training works is essentially described in terms of maximizing the agreement between classifiers on unlabeled data, though their proof neglects this point. Abney proves that the rate of disagreement between classifiers provides an upper-bounds on their error

rate. Specifically, he shows that this holds for binary, non-trivial (that is, fairly accurate), conditionally independent classifiers. As a corollary Abney proves that if the precision of one classifier is known then the precision of any other independent classifier can be precisely calculated using their agreement rates. Abney shows empirically that this corollary *does not hold* for the named entity data presented in Collins and Singer (1999). Knowing the precision of one classifier and its agreement rate with another does not allow the precision of the other classifiers to be precisely calculated, because the views that they are trained on are *not* conditionally independent.

Abney shows that the bounding of error rate by the level of disagreement between classifiers holds under a weaker assumption, however. Rather than having to posit view independence, which *is not* satisfied by the data, one can loosen the assumption and prove that the bounding of error rate still holds for classifiers with weak dependence, which is satisfied by the data. The proof for conditionally independent classifiers holds for negatively correlated classifiers, and an additional proof shows that so long as the positive correlation of one classifier within a small proportion of the other, that the bounding still holds. Abney expands this proof beyond binary non-abstaining classifiers to more complex classifiers, and further uses it to define The Greedy Agreement Algorithm for selecting which rules that can be derived from the unlabeled data set ought to be be added to the classifier. The *Greedy Agreement Algorithm* performs as well as the co-training algorithms presented in Collins and Singer (1999) and Yarowsky (1995) and has an appropriate theoretical framework to explain why it works.

3.1 Applied to NLP

In this section I describe a number of attempts to apply co-training to natural language processing, which were inspired by the Blum and Mitchell (1998) paper and whose success despite the lack of conditional independence led to the Abney (2002) paper. I summarize the findings of Pierce and Cardie (2001), Collins and Singer (1999), and Sankar (2001).

Collins and Singer (1999) develops a co-training method for named entity classification. The task is to learn a function from an input string (proper name) to its type,

which is either *person*, *organization*, or *location*. The two views adopted are "contextual" and "spelling" rules. Contextual rules use the words surrounding the string (for example, a name modified by an appositive with the word *president* is likely to be a person). Spelling rule examine the words being classified themselves (for example, a string containing *Mr.* is a person).

Collins and Singer start with only seven supervised seed rules: that *New York*, *California*, and *U.S.* are locations; that any name containing *Mr.* is a person; that any name containing *Incorporated* is an organization; and that *IBM* and *Microsoft* are organizations. Collins and Singer are able to harness the agreement between these seven rules to bootstrap from 90,000 unlabeled examples, and achieve a 91.3% accuracy (compared with 81.3% using EM) .

Pierce and Cardie (2001) uses co-training for base noun phrase identification. Base noun phrase identification is the task of identifying non-recursive noun phrases using the words in the sentence and their part-of-speech tags. Pierce and Cardie train naive Bayes classifiers on two contrived views – one classifier looks at the current tag and tags to the left of the word being classified, and the other classifier looks at the current tag and tags to the right of the word being classified – and include examples at each new round based on the examples labeled with the highest probabilities, rather than agreement between the two classifiers. Pierce and Cardie note that their views violate Blum and Mitchell's desideratum of conditional independence between views, since they both include the current tag. They point to the experiments in Nigam and Ghani (2000), which suggest that co-training may still prove effective even when an explicit feature split is unknown, provided that there is enough implicit redundancy to allow an arbitrary division of features.

Pierce and Cardie determine the best settings for the parameters of the co-training algorithm by testing multiple values for the initial amount of labeled data, the pool size of unlabeled data, and the number of examples added at each round of retraining. They report results given the best settings in terms of improving the accuracy of the classifier, after trying multiple values. Pierce and Cardie found initial gains above the accuracy of the initial seed corpus but only to a point. After that point, the test set accuracy begins to decline as additional training instances are added.

Corduneanu and Jaakkola (2001) attempts to address the problem that as amount of unlabeled data being added increases beyond a certain point, it overwhelms the labeled data and performance may drop. Corduneanu and Jaakkola seek a way of balancing labeled and unlabeled data in order to maximize the accuracy of the resulting model. When two classifiers begin to agree, both may converge to points that are significantly different than the privileged one that remains closest to the labeled data maximum likelihood solution. Grounding the solution to the labeled data is desirable since otherwise there is little reason to expect that the resulting model would serve well as a classifier. Corduneanu and Jaakkola suggest that it is best to avoid solutions that are unsupported by the labeled data. This could perhaps be employed by only adding those examples which increase performance on a held out set.

Sankar (2001) applies co-training to the task of statistical parsing. Statistical parsing is an especially interesting case study, since pervious applications of co-training to NLP tasks has generally been to classification tasks with relatively small set of possible labels. Statistical parsing labels sentences with parse trees, and those trees are decomposed into lexical trees attaching at the word level and the attachments between them. Sankar divides the parsing task into two components suitable for co-training: one selects trees based on the local context (a tagging probability-based model), and one which chooses the best parse based on attachment between trees (a parsing probability-based model). The tagging probability-based model uses a tri-gram model based on a SuperTagging parser. It creates an *n*-best lattice of tree assignments for an input sentence composed of elementary trees for each word in the sentence. The parsing probability-based model creates a consistent bracketing of a sentence by attaching the elementary trees together.

Co-training starts with a set of 9,600 sentences labeled with parse trees (four sections of the Penn Treebank) and an unlabeled set of 30,0000 sentences (fourteen sections, stripped of all annotation). On each round of co-training a subset of the unlabeled sentences are parsed. The *n* most probable sentences from the tagging probability-based model (run through the parsing probability-based model) and the *n* most probable sentences form the parsing probability-based model are included in the next round of training. Sankar obtained a labeled bracketing precision of 80% and recall of

79.64%. The co-trained model significantly outperformed the baseline that was trained on only the 9,600 labeled sentences, which scored 72.23% and 69.12%.

The next chapter details how I went about applying co-training to machine translation. It explains what the separate views are within translation, and tailors the co-training algorithm to machine translation. Chapter 5 gives the experimental results for the application of co-training to the task.

Chapter 4

Applying Co-training to Machine Translation

Co-training relies on having distinct "views" of the items being classified. Each view needs to have sufficient information to label the material, and needs to be (relatively) independent of each other view. Many problems in natural language processing do not naturally divide into different views and have to be artificially constructed. Translation, on the other hand, has a very natural division of views onto the labels. In machine translation "labels" are the target translations for source texts. The source text can therefore be considered a "view" on the translation. Other views that are sufficient for producing a translation would be existing translations of the source text into other languages. For example, a French text and its translation into German can be used as two distinct views, either of which could be used to produce a target translation into English (see Figure 4.1). The use of multiple source documents to augment the quality of translation puts the method proposed in this thesis in the category of *multi-source translation* (Kay (2000)).

4.1 Multi-source Translation

Martin Kay observes that if a document is translated into one language, then there is a very strong chance that it will need to be translated into many languages. This is

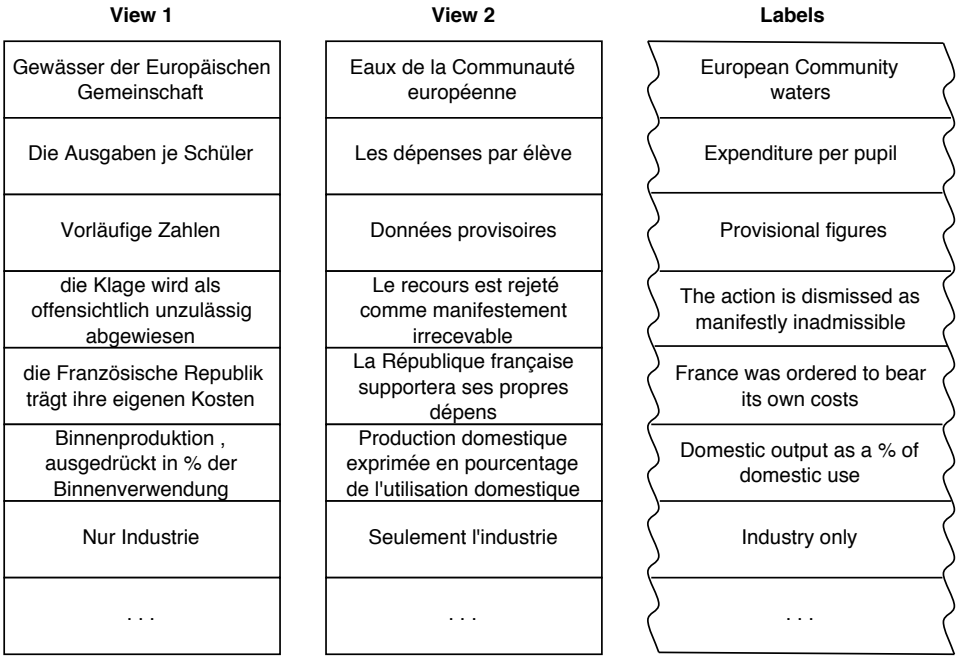


Figure 4.1: German and French as distinct views on English labels

because international organizations like the European Union must publish legal documents in the languages of all of their member states; multi-national corporations like Sony need to produce product descriptions and manuals in the languages of each country that they do business in; and so forth. Kay (2000) proposes using multiple source documents as a way of informing subsequent machine translations, suggesting that much of the ambiguity of a text that makes it hard to translate into another language may be resolved if a translation into some third language is available. He calls the use of existing translations to resolve underspecification in a source text "triangulation in translation", but does not propose a method for how to go about performing this triangulation. The challenge is to find general techniques that will exploit the information in multiple source to improve the quality of machine translation.

One approach that has been proposed is a straightforward adaptation of the Brown et al. (1993) technique. Och and Ney (2001) redefines the Fundamental Equation of Statistical Machine Translation so that it is suitable for statistical multi-source translation

$$\hat{e} = \text{argmax}_e\{P(e|f_1^N)\}$$

$$= \text{argmax}_e\{P(e) * P(f_1^N|e)\}$$

for source strings $f_1^N = f_1, ..., f_N$, in N source languages, which are to be translated in the target string e. Och and Ney give two ways of calculating $P(f_1^N|e)$. The first is simply to have each language's translation model produce separate translations and then take the translation with the highest probability assigned by its own translation model

$$\hat{e} = \text{argmax}_e\{P(e) * \text{max}_{f_n} P(f_n|e)\}$$

$$= \text{argmax}_{e,f_n}\{P(e) * P(f_n|e)\}$$

The other method is to take the translation which maximizes the product of the probabilities assigned by all translation models

$$\hat{e} = \text{argmax}_e\{P(e) * \prod_{n=1}^{N} P(f_n|e)\}$$

Och and Ney find improvement using multiple source sentences over using a single source sentence, and attribute the better quality to:

- Better word sense disambiguation: often ambiguities that need to be resolved between two languages do not exist between other languages.
- Better word reordering: a significant portion of errors in statistical machine translation are due to word order problems. The word order between related languages is often very similar while the word order between distant languages might differ significantly. Using more source languages increases the chances of having one with similar word order to the target language.
- Reduction of the need for explicit anaphora resolution: by having various translations of a pronoun in different languages the probability increases that it can be translated correctly without performing a full anaphora resolution.

Instead of taking the idea of multi-source translation as one that applies at the time of translation as Och and Ney do, I adapt the idea so that it may be used to build better translation models. Och and Ney use multiple source strings to improve quality of one translation only. My co-training method attempts to improve the quality of translation overall by bootstrapping more training data from multiple source documents. Increasing the amount of source data should lead to better estimation of translation model parameters, thus improving the overall quality of translations produced by a model.

4.2 Multiple Sources for Co-training

I adapt co-training to improve translation quality using multiple source texts to increase the amount of available training data. Just as Blum and Mitchell use the hyperlink and web page text to iteratively increase the number of classified web pages used to train the classifiers, I use multiple source documents to iteratively increase the size of parallel corpora used to train translation models. Figure 4.2 gives my co-training algorithm for machine translation. Here is how it would work using a German-French parallel corpus to bootstrap the quality of German to English and French to English translations:

1. Use a German-English parallel corpus to train a DE⇒EN translation model. Use a French-English parallel corpus to train a FR⇒EN translation model.

2. For each sentence pair in a French-German parallel corpus, use the FR⇒EN translation model to translate the French sentences into English, and use the DE⇒EN translation model to translate the German sentences into English.

3. Choose the best translation from the two translations created by the FR⇒EN and DE⇒EN translation models, and align it with the French-German sentence pair. This creates a French-German-English parallel corpus, which can be divided into a French-English corpus and a German-English corpus.

4. Take the top N sentences from the newly created French-English corpus, and add them to the original French-English corpus. Do the same for the German. Remove those items from the French-German corpus.

Given:

- parallel corpora $L_1||EN$, $L_2||EN$, ... $L_n||EN$ of sentences in languages $L_1, L_2 ... L_n$ aligned with their translations into a target language, here we choose English
- and a parallel corpus $L_1||...||L_n$ aligning sentences across languages $L_1, ... L_n$

Co-train translation models:

1. Create translation models $L_1 \Rightarrow EN$ ... $L_n \Rightarrow EN$ from each of the parallel corpora $L_1||EN$, ... $L_n||EN$
2. For each sentence alignment $l_1||...||l_n$ in the $L_1||...||L_n$ corpus create a candidate pool of translations $en_1 ... en_n$ by translating l_1 into English using $L_1 \Rightarrow EN$, ..., and translating l_n into English using $L_n \Rightarrow EN$
3. Build up machine-translated parallel corpora $L_1||EN' ... L_n||EN'$. Choose a translation en' from $en_1 ... en_n$ and creating an alignment $l_n||en'$ for each language $l_1 ... l_n$. Add these alignments to the appropriate machine-translated parallel corpora
4. Select some subset of each of the parallel corpora $L_1||EN' ... L_n||EN'$ and add them to $L_1||EN ... L_n||EN$, respectively. Remove the subset from consideration on subsequent rounds of co-training
5. Repeat steps 1 – 4 until $L_1||...||L_n$ is exhausted, or until some other stopping criteria is met.

Figure 4.2: The co-training algorithm for machine translation

5. Repeat steps 1-4, retraining the DE⇒EN and FR⇒EN translation models on the incrementally larger corpora until the French-German corpus is exhausted, or until some other stopping criteria is met.

This is also illustrated in Figure 4.3.

As Figure 4.2 indicates, the process need not be limited to two languages. The technique can be used with as many languages as are available in a multiply-aligned parallel corpus. Indeed, one would expect increasing the number of languages to improve the overall effectiveness of co-training. The algorithm also omits two important points. Firstly, it does not describe the selection criteria for which (and how many) sentences to include in the next round of training. This point is addressed in Section 5.2. Secondly, it leaves open the technique for choosing the best translation for each sentence alignment. This could be addressed by using Och and Ney's multi-source translation equations.

In order to test the feasibility of applying co-training to machine translation, I first needed a large, multiply parallel corpus. I constructed such a corpus from a number of bilingual parallel corpora that were gathered from European Union web site for experiments in multi-source translation.

4.3 Corpus

The multilingual corpus used in this project was created for the Och and Ney (2001) work on statistical multi-source translation. The corpus was assembled from the *Bulletin of the European Union* which is published on the Internet in the eleven official languages of the European Union. Och and Ney performed the following steps to create the multilingual corpus:

1. Web pages in the eleven languages of the European Union were downloaded in HTML format.

2. A document-level alignment was performed by aligning the URLs of the files downloaded.

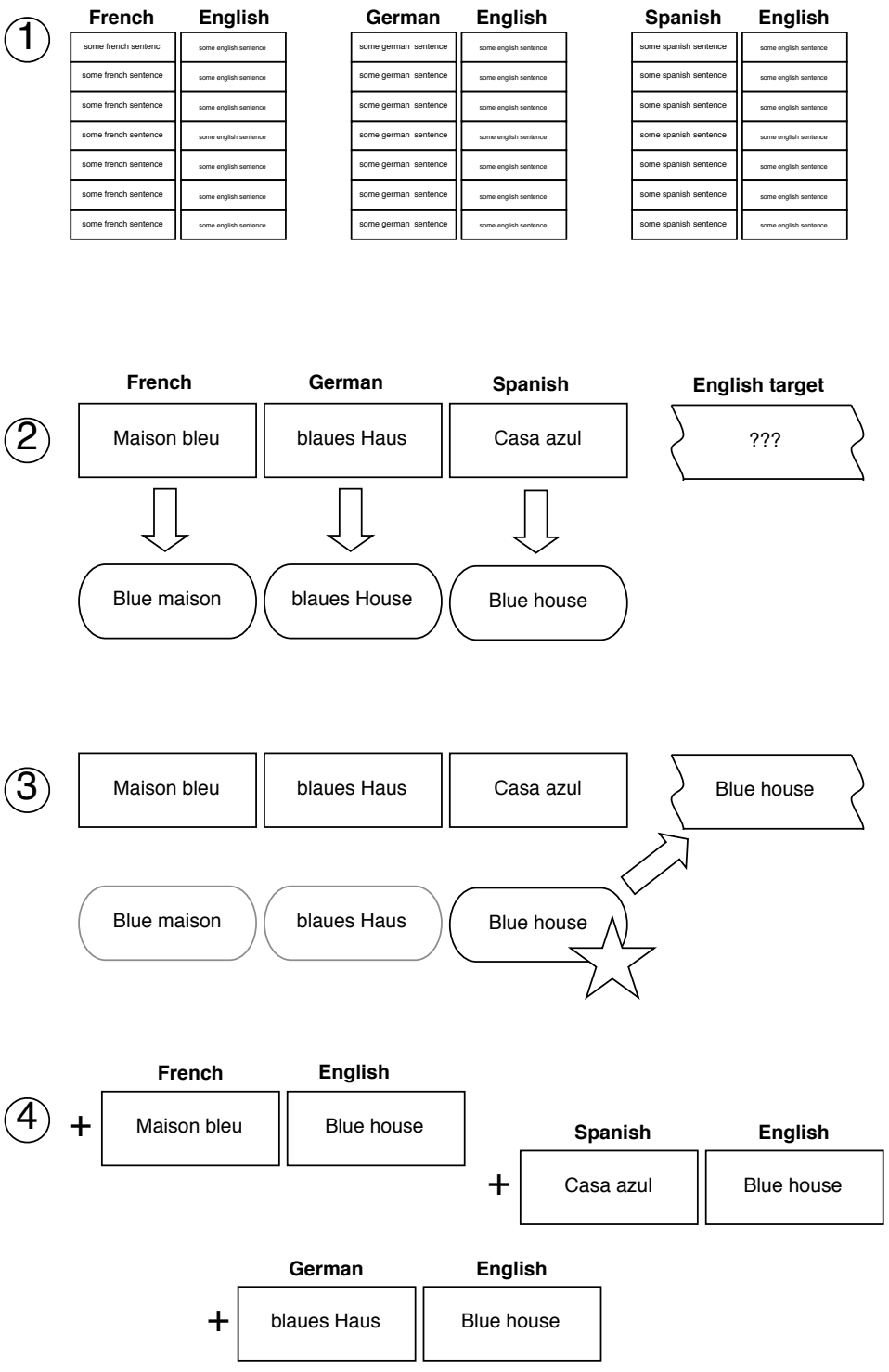


Figure 4.3: An illustration of the co-training algorithm

3. The raw text was extracted from all text segments within HTML tags. This created a sequence of paragraph-sized text segments for every document in every language.

4. A segment-level alignment was performed using a dynamic program algorithm which optimizes a length-based heuristic (as described in Gale and Church (1993)). This segmentation was performed between English and each of the other ten languages, thereby creating ten bilingual corpora aligned on the paragraph level.

5. A sentence-level alignment was performed using similar heuristics, thereby creating ten bilingual corpora aligned on the sentence level.

6. The resulting bilingual corpora were made more accurate by eliminating all sentences which had obviously wrong alignments. This was done by filtering out very long sentences that had been paired with very short sentences or pairs which had a very low probability according to the alignment model.

The following table gives size statistics for the Och and Ney (2001) corpus:

Lang	Sentences	Words	Vocab
French	117K	2.32M	50462
Spanish	120K	2.32M	50949
Portuguese	120K	2.30M	50216
Italian	120K	2.21M	54986
Swedish	125K	2.02M	72517
Danish	131K	2.21M	70713
Dutch	121K	2.30M	58550
German	139K	2.23M	73506
Greek	131K	2.28M	68811
Finnish	120K	1.61M	106159
English		2.1M	45K

In order to create a multiply parallel corpus, I used the English sentences as ‘keys’ to align the sentences across all languages. Each time an English sentence was con-

tained in an alignment in more than one of the parallel corpora, I aligned the non-English sentence with each other as well as with the English sentence. Note that this has the potential of introducing alignment errors, especially in cases where multiple translations exist for an English sentence in a single language.

4.4 Evaluation

One crucial aspect of testing the efficacy of any machine learning task is having an appropriate evaluation mechanism. In order to judge whether progress is being made, there needs to be a way to tell how well a system is doing. The evaluation of the quality of machine translation (and of translation in general) is a difficult task. There are two methods that can be used to evaluate machine translation systems:

1. manual evaluation, which uses subjective human judgments about the quality of translations

2. automatic evaluation, which uses some similarity metric to compare a system's translations to reference translations

Because translations can vary based on context, and because in a given context various translations may be equally acceptable, there is no single "correct" translation of a sentence. The most straightforward way to address this is to manually evaluate the correctness or quality of translations. In manual evaluation translations are scored by bilingual judges, and classified into a small number of categories ranging from "perfect" to "incomprehensible" (Callison-Burch and Flournoy (2001), NieBen et al. (2000)). Translation systems can be compared by translating a set of sentences, and comparing human judgments about the translations produced by each system.

Despite the fact that using subjective judgments addresses the inherent variability in translation, the approach is generally impractical for continuing MT research. The number of times that a prototype system needs to be evaluated (for example, every time a parameter like corpus training size is adjusted, and every time a slight change in system design is made) makes qualitative human evaluation infeasible because of the amount of time manual evaluation requires. The requirement that human judges

be bilingual is also a significant impairment for research that examines translation into many languages.

Automatic evaluation uses similarity metrics to compare translations produced by a machine translation systems to reference translations. Evaluation takes place by dividing a bilingual corpus into separate training and testing sets. The source sentences in the testing set are translated by the machine translation system and compared to the existing human translations. The most commonly used evaluation metrics are

- Sentence Error Rate: The number of times that the generated sentence corresponds exactly to its reference translation.
- Word Error Rate (WER): The minimum number of substitution, insertion, and deletion operations that have to be performed to convert the generated sentence to reference translation.
- Position-independent WER: A shortcoming of WER is the fact that it requires a perfect match with the reference translation's word order. Because different constituent orderings in a translation are often acceptable, WER alone can be misleading. Position-independent WER compares the words in the two sentences ignoring the word order.

The advantage of these methods for evaluating translation quality is that they can be conducted fully-automatically, and do not require expertise in the languages being translated between. However, the use of a single reference translation is an inexact measure of translation quality, as there are often multiple ways of translating of a sentence. Comparing against a reference translation supposes that the given translation only correct translation.

Various techniques attempt to combine manual and automatic evaluation in order to reap the benefits of subjective human evaluation while decreasing the time that it takes to evaluate a system. NieBen et al. (2000) describes a technique for increasing the speed of human evaluation by caching scores for previously seen translations. Keeping a database of previously scored translations recognizes the fact that while evaluation takes place many times while prototype systems are being optimized, often translations differ between adjustments by only a few words. NieBen et al.'s evaluation tool speeds

the manual evaluation process by automatically returning scores of translations that have already occurred, facilitating the evaluation of new translations that differ only slightly from previous ones by highlighting the differences, and extrapolating scores for new translations by comparison with similar sentences in the database. Having a record of scores for multiple translations of a single source sentences allows NieBen et al. to introduce a new automatic evaluation metric, Multi-reference Word Error Rate. Multi-reference WER treats all translations rated as "perfect" as reference translations, and gives a new translation the best WER score that it receives when compared to each of the reference translations.[1]

Two other techniques (Callison-Burch and Flournoy (2001) and Papineni et al. (2001)) attempt the approximate human evaluations automatically. Callison-Burch and Flournoy (2001) demonstrates that n-gram language models of the target language can be used to choose the highest human-ranked translation from a set of translations produced by multiple machine translation engines. Papineni et al. (2001) also uses n-gram language models to try to predict the rank assigned by a human evaluator, but further includes comparison to reference translations, and a number of heuristics such as a "brevity penalty". Both techniques have the advantage of being fully automatic evaluation that strongly correlate with human judgments (manual evaluation). However, their heavy reliance on language models of the target language make them an inappropriate measure of statistical machine translation of the IBM variety, since they disproportionately weight the language model component over the translation model component.

There is no single agreed upon metric for the evaluation of machine translation systems. Consequently most papers present results using a variety of metrics, generally using all those that are available to the authors. For example, Och and Ney (2002) evaluate their system using Sentence Error Rate, Word Error Rate, Position-independent WER, Multi-reference WER, BLEU Score, Subjective Sentence Error Rate (human judgments about sentence quality), and Information Item Error Rate (human judgments about the quality of sub-sentential elements).

[1]Note that the use of multiple references could also be applied to the other automatic evaluation metrics, Sentence Error Rate, and Position-independent WER.

Since the research conducted in this thesis is across many languages, and because of the large number of evaluations that needed to be conducted, automatic evaluation techniques were adopted in place of manual evaluation. Specifically, the results are presented using word error rate as the evaluation metric. This is an appropriate metric because it is fine-grained enough to show that improvement to translation quality is being made, and generic enough to be used across all languages and applied automatically.

Chapter 5

Experimental Results

This chapter describes the various experiments that I conducted up to and including empirically examining the efficacy of co-training for statistical machine translation. All experiments are of a similar nature: they compare the quality of translation models (using word error rate as the evaluation criterion, as discussed in Section 4.4) built from varied bilingual corpora. In the preliminary experiments, bilingual corpora are varied to see how training corpus size affects performance. In the co-training experiments, the bilingual corpora varied in how they are created; rather than using only human translations the bilingual corpora also contain machine translations.

The co-training experiments were performed using a German-French-Spanish-Italian-Portuguese-English parallel corpus, which contained 63,000 sentences in each language. The creation of the corpus is described in Section 4.3. The English section of the corpus was reserved, and used for evaluation of the machine translations. Five small translation models were trained for German, French, Spanish, Italian, and Portuguese into English. These initial "labeled" sets ranged from about 16,000 to 20,000 sentence pairs. The translation models were created from sentences outside the multilingual corpus – they were the sentences which did not have an English key which had a translation into all five languages. The translation models were used to translate sentences from their respective languages in the multilingual corpus. A subset of these machine translations were selected to retrain translation models, using a process described in Section 5.2.

Translation models were compiled using GIZA++, an open-source software pack-

age. Language models were compiled using the CMU-Cambridge Language model toolkit. Translations were produced with the translation and language models using the ISI ReWrite Decoder. The software is described in Appendix A.

5.1 Preliminary Experiments

As a preliminary exercise, I evaluated the translation quality of translation models built from incrementally larger training corpora. This was done by testing the performance of ten different French to English translation models. The French-English component of the EU parallel corpus was divided into chunks each containing 10,000 sentence pairs. These were combined at each round to create increasingly large training corpora. A held-out testing set of 1,000 sentence pairs was used to evaluate the quality of each of the translation models. Each training corpus was compiled into a translation model using GIZA++. The French sentences in the testing set were then translated into English with the decoder software using the translation models and a single language model (consisting of all 100,000 French sentences from the training sets). The English translations were then compared with the human translations in the test set. A word error rate was calculated for each sentence, and the average WER was determined for each model. The accuracy of each model (which is defined throughout this section as 100 minus the word error rate) was plotted against the size of the training corpus that is was built with. This is given in Figure 5.1.

If performance had plateaued as the number of human translated sentences in the training corpus was increased, then adding machine translations to the training corpus would be of dubious value. Despite the fact that the rate of quality improvement decreases after the first 20,000 training examples, the performance does not plateau. Therefore, using co-training to produce more data may be a fruitful endeavor. I performed similar experiments for German to English and Spanish to English translation models to verify that the same behavior held for other languages. The same trend of quality increasing with training corpus size was observed. This can be seen in Figure 5.2, which gives performance of all of the models.

Another preliminary experiment that I performed was to compare translation qual-

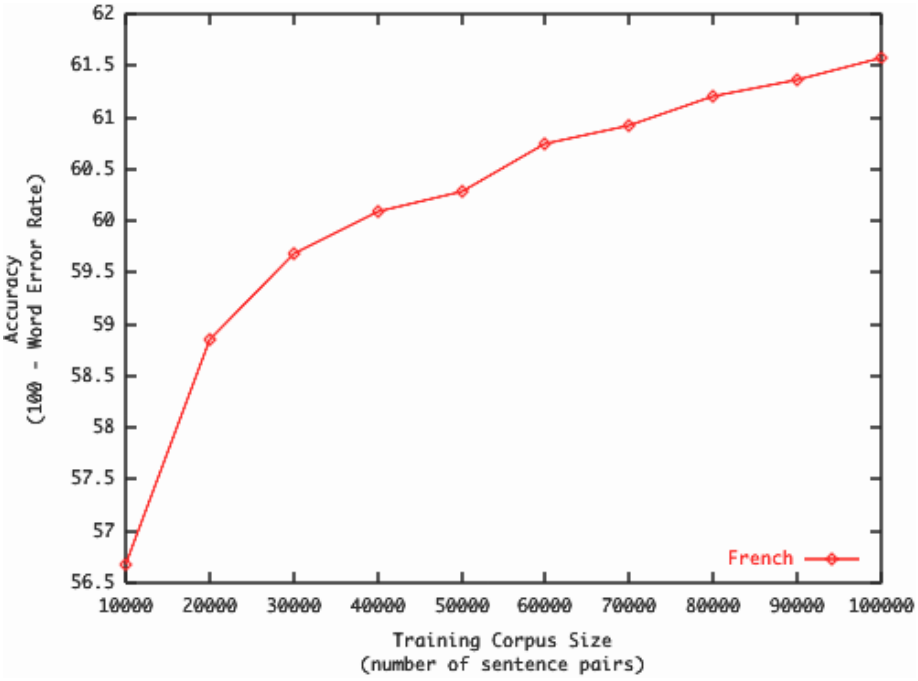


Figure 5.1: Accuracy vs training corpus size for French to English translation models

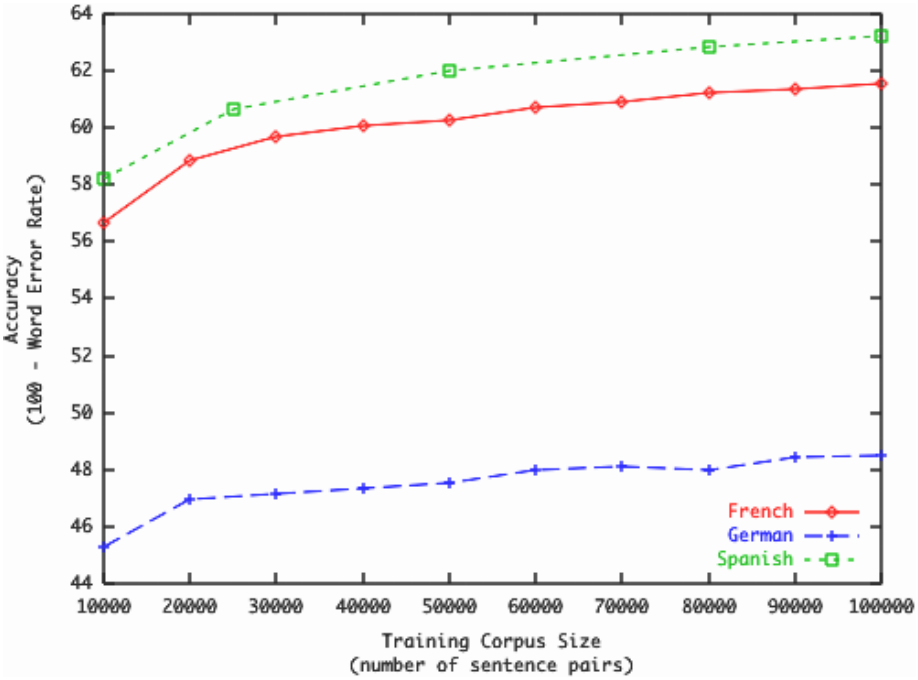


Figure 5.2: Accuracy vs training corpus size for French, German and Spanish

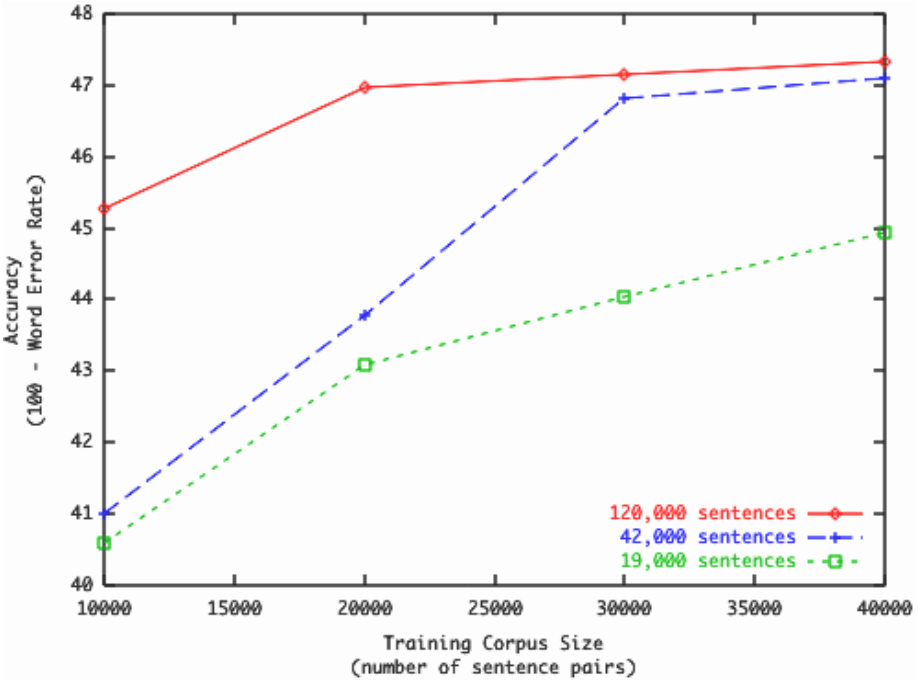


Figure 5.3: Effects of varying the size of the language model

ity when the size of the translation model training corpus was held steady, but the size of the language model was varied. The Fundamental Equation of Statistical Machine Translation selects the best translation as the product of translation model and language model probabilities, so increasing the size of the set used to train either component ought to improve performance. Figure 5.3 shows the quality of German to English translation for three different language models. Each language model is evaluated in conjunction with four translation models built from 10,000, 20,000, 30,000 and 40,000 sentence pairs. The graph indicates that increasing the amount of monolingual text used to train the language models does increase the quality of resulting translations.

I wanted to ensure that altering the language model probability would not skew the results of co-training. Therefore, I've held the language model training set size steady across all runs of co-training. Each language model training corpus is limited to the target language sentences from the human translated sets at round zero of co-training. This means that the results of co-training can be interpret as how co-training

is affecting translation model probabilities. This is desirable because language model probabilities are much easier to improve using unlabeled data; larger monolingual corpora simply need to be used to train them.

Other variables that were controlled for in addition to language model size were:

- decoding parameters – the ISI ReWrite Decoder has a number of parameters that can be set to speed the time it takes to translate a sentence, at the risk of not finding the optimal translation. Optimal translation was not possible because of the prohibitive amount of time needed to translate the hundreds of thousands of sentences used in co-training. Therefore the time-saving parameters were used, but kept the same for all experiments

- translation model training parameters – various parameters are available in GIZA++ for training translation models. These mainly involve the number of training iterations used to perform expectation maximization for each of the IBM Models. The values for the training parameters were kept the same for all experiments.

The actual parameters used are discussed in Appendix A.

The remainder of this chapter describes the co-training experiments that I performed.

5.2 Co-training Selection Techniques

Co-training was conducted using German, French, Spanish, Italian and Portuguese to English translation models. At round zero of the co-training the translation models were trained from small human-translated corpora. These initial translation models were used to translate the 63,000 sentences in the German-French-Spanish-Italian-Portuguese(-English) multilingual corpus for each of the source languages. English translations were selected from the five candidate translations for each sentence alignment. This was done using an oracle to divine the best translation from each set, by comparing them to the reserved English translation and choosing the one with the most favorable word error rate. The reason for using an oracle to choose the best translation was to provide a theoretical motivation that co-training would work under ideal

circumstances.

After the best English translations were produced for each sentence alignment in the German-French-Spanish-Italian-Portuguese multilingual corpus, and after it was broken into five bilingual corpora, the task remained to select which items from the bilingual corpora to include with the human-translated corpora for the next round of training. The co-training algorithm described in n Figure 4.2 does not describe how to select sentences for the next round of co-training. One possibility which is not adopted here would be to adapt the Abney (2002) Greedy Agreement Algorithm. The Greedy Agreement Algorithm would guarantee that co-training would result in improved translation provided that:

1. The re-write method for translation used in Brown et al. (1993) could be reformulated so that rewriting was carried out by a list of atomic classifiers

2. It were possible to prove the resulting set of rules satisfied weak rule dependence

3. The large quantity number of rules did not make the cost evaluation step practically impossible.

It might be possible to reformulate the string rewriting translation method as a list of atomic classifiers – for example, with each classifier associating a single word with its translation or a single source index with a target position. It might further be possible to show that such classifiers were only weakly dependent. However, the Greedy Agreement Algorithm would likely be infeasible since the cost evaluation step compares all pairwise combinations of new and old rules, and since there would likely be at least one rule per parameter in a translation model.

Instead of using agreement-based selection methods, I continued the use of the oracle and tried two methods for selecting examples for retraining: one based on selecting the most accurate translations, and the other based on maximizing the difference in accuracy between translations produced by different models.

The selection method that maximizes the difference between the accuracy of translation worked like this: each alignment of German-French-Spanish-Italian-Portuguese was translated into five English sentences. Each of those English translations was compared to the reference translation and assigned an accuracy scores. The item with

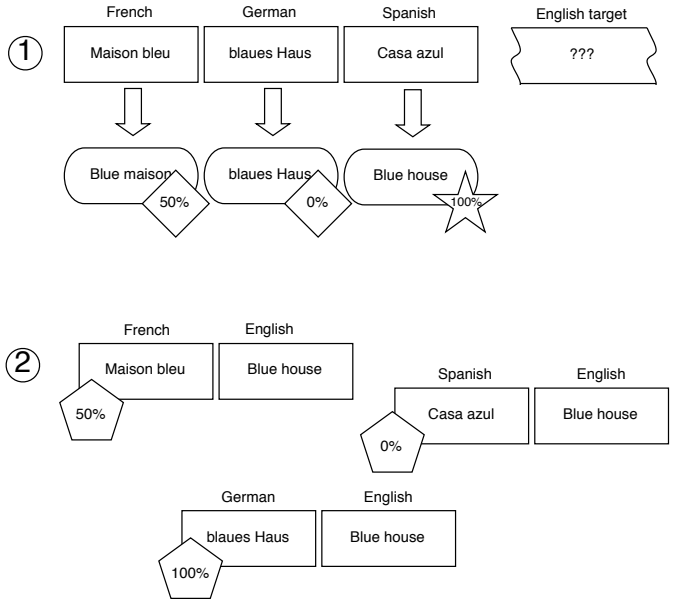


Figure 5.4: The difference in accuracy between each translation model's score and the top score is recorded

the best accuracy was noted, and then the difference between each of the scores and the best score was recorded for each item in each language. Figure 5.4 illustrates this. Items to be added to the bilingual corpora for the next round of retraining were selected on a per language basis. The new items were sorted based on their difference scores, and the ones with the highest scores were included. Therefore the German item from Figure 5.4 would likely be included in the next round of training for the German to English translation, but the Spanish item would not be. This method would have ideally selected those items which were maximally informative to each of the language models. However, Figure 5.5 shows disappointing results after using the method for one round of co-training.

I conducted an analysis of why the method did so poorly. I found that by choosing those items which maximized the difference in performance introduced a lot of noise, rather than choosing the most informative examples. Due to the automated fashion in which the multilingual corpus was created, a number of misalignments exist. For example an alignment might contain correct translations in German, Spanish, Italian,

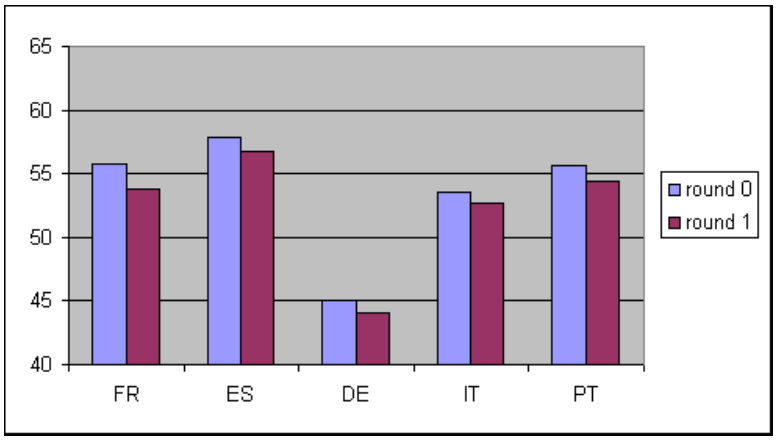


Figure 5.5: Results of the difference method after one round of co-training

and Portuguese, but contain a French sentence which was selected from the wrong area of the web page that it was drawn from. Therefore this French sentence is not a translation of other languages. When the French to English translates this sentence, it is comparable to the English reference translation. It is compared and receives a very low accuracy, and thus the difference between it and the highest scoring translation will be high. The difference metric is therefore likely to select this item to be added into the French-English bilingual corpus for the next round of training. The item added is not a translation; it is an English sentence paired with a random French sentence. This introduces noise into the training set, and results in worse translations. This bias towards misalignments is illustrated in Figure 5.6.

I reformulated my co-training experiment to address this problem. Rather than selecting examples which maximized the difference in performance, I simply selected those examples which had the highest accuracy overall. Misalignments were still occasionally introduced into the retraining, but the selection method was no longer biased towards them. The training data was therefore less noisy. Figure 5.7 gives the result of co-training using an oracle to select the best translation from the candidate translations produced by five translation models. Each translation model was initial trained on bilingual corpora consisting of anywhere between 16,000 to 20,000 human translated sentences. These translation models were used to translate 63,000 sentences, of which the top 10,000 were selected for the first round. At the next round 53,000 sentences

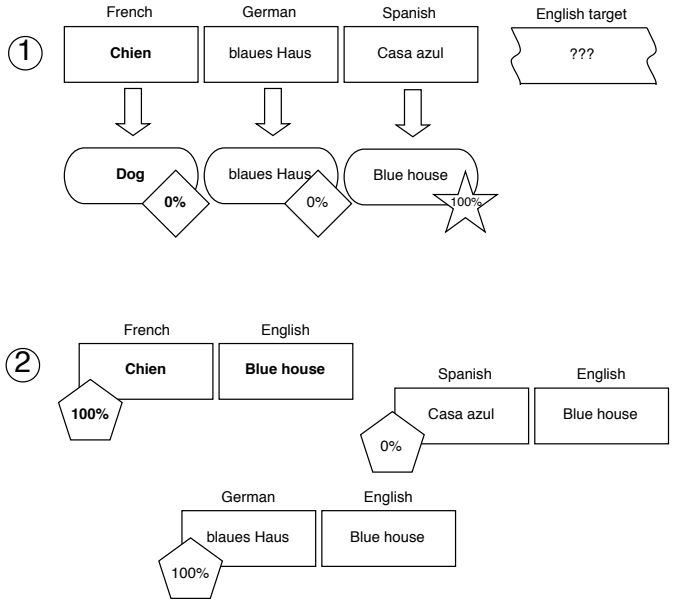


Figure 5.6: Bias towards selecting misalignments

were translated and the top 10,000 sentences were selected for the second round. The final candidate pool contained 43,000 translations and again the top 10,000 were selected. The graph indicates that some value may be gained from co-training. Each of the translation models improves over its initial training size at some point in the co-training. The German to English translation model improves the most – exhibiting a 2.5% improvement in accuracy.

The graph further indicates that co-training for machine translation suffers the same problem reported in Pierce and Cardie (2001): gains above the accuracy of the initial corpus are achieved, but decline as after a certain number of machine translations are added to the training set. This could be due in part to the manner in items are selected for each round. Because the best translations are transferred from the candidate pool to the training pool at each round the number of “easy” translations diminishes over time.

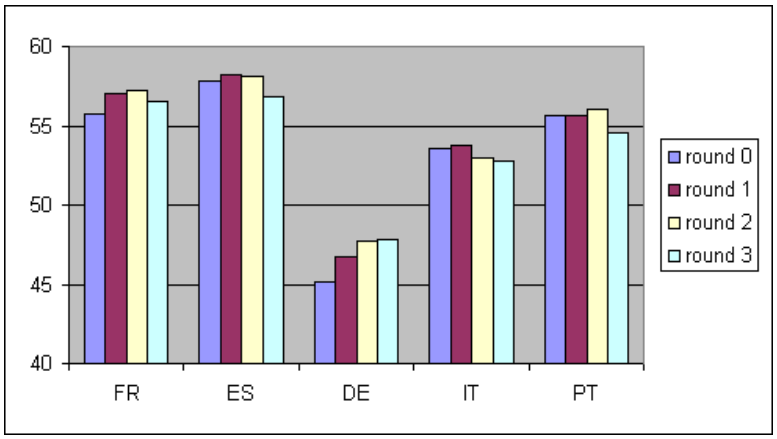


Figure 5.7: Co-training results

5.3 Practical Concerns

The actual effectiveness of co-training may actually be more positive than Figure 5.7 indicates. In their base noun phrase identification task Pierce and Cardie (2001) were able to report the best results in terms of co-training improving the accuracy of the classifier by trying many values for the parameters of the co-training algorithm. They varied the initial amount of labeled data from 10 to 5000 items, the candidate pool of unlabeled examples from 200 to 5000, and the growth size from 1 to 50. I was unable to perform that sort of experimentation when applying co-training to machine translation due to the prohibitive amount of time that each round of co-training required. Translating 60,000 sentences for five languages took six days using ten computers, and training the five translation models took an additional day.

I tried to select co-training parameters which would be likely to provide yield results: I choose initial translation model sizes at the point where the greatest increase was observed in the preliminary experiments; I opted for a large pool of untranslated items to draw from, so that the proportion of correctly translated sentences would be higher; and I used selection techniques which were likely to choose the best machine translations from the candidate pool. The careful selection of co-training parameters was necessary given the time constraints, and it did serve to illustrate that co-training can in principle help to improve the quality of machine translation.

If time allowed the parameter space to be explored it is possible that better improvement would be found. For example, if there were a wider range of sizes for the initial bilingual corpora, one might expect that the larger corpora would lead to greater gains in the smaller corpora. Limiting the amount of untranslated sentences at each round might have forced the selection algorithm to choose a set that more accurately represented the distribution of translations in the test set. Adding fewer sentences at each round of co-training would have caused less shift away from the human labeled data. And so on.

I explored one of these possibilities by conducting a co-training experiment with two translation models of vastly different size. This is explained in the next section.

5.4 Coaching Variation

I experimented with a variation on co-training for machine translation that I call "coaching". It employs two translation models of vastly different size. In this case I used a French to English translation model built from 60,000 human translated sentences and a German to English translation model that contained no human translated sentences. A German-English parallel corpus was created by taking a French-German parallel corpus, translating the French sentences into English and then aligning the translations with the German sentences. Figure 5.8 shows the performance of the resulting German to English translation model for various sized machine produced parallel corpora.

This graph illustrates that increasing the performance of translation models may be achievable using machine translations alone. Rather than the 2.5% improvement gained in co-training experiments wherein models of similar sizes were used, coaching achieves a 30% improvement by pairing translation models of radically different sizes.

I explored this method further by translation 100,000 sentences with each of the non-German translation models models from the co-training experiment. The result was a German-English corpus containing 400,000 sentence pairs. The performance of the resulting model matches the accuracy at round zero translation of the co-training experiment. Thus machine-translated corpora achieved achieved equivalent quality to human-translated corpora after two orders of magnitude more data was added.

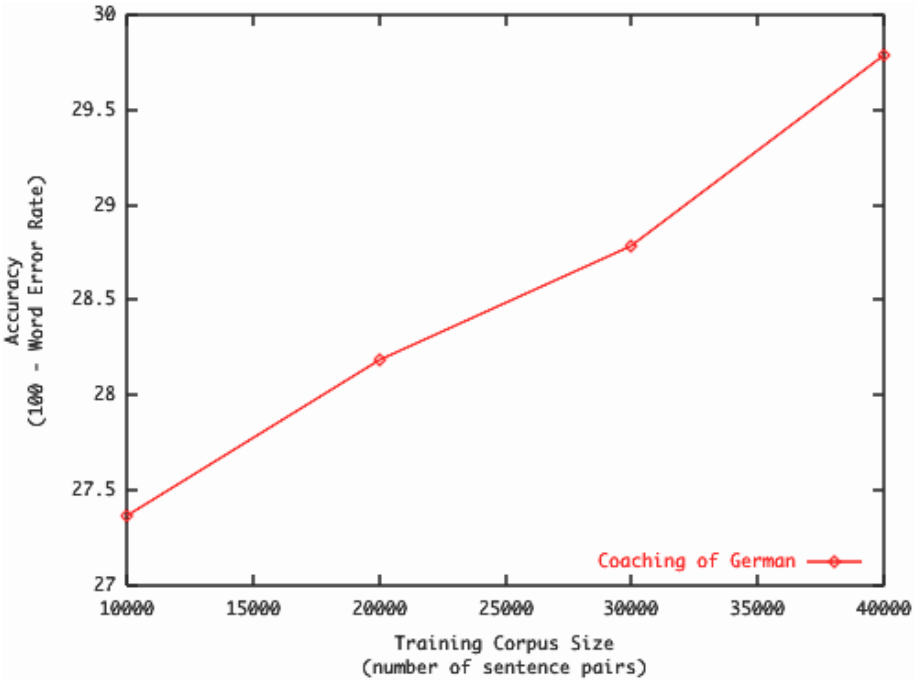


Figure 5.8: “Coaching” of German to English by a French to English translation model

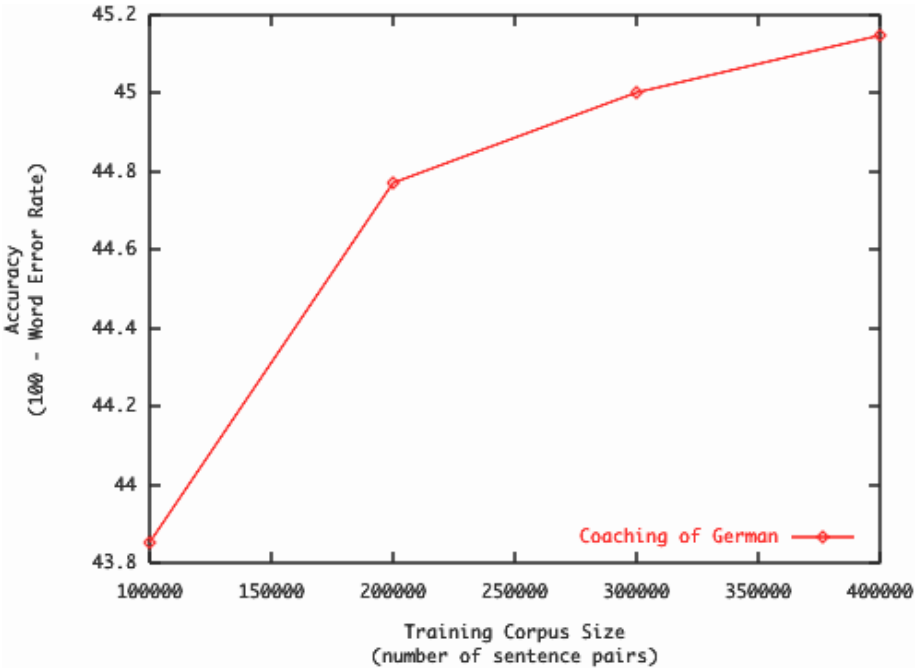


Figure 5.9: “Coaching” of German to English by multiple translation models

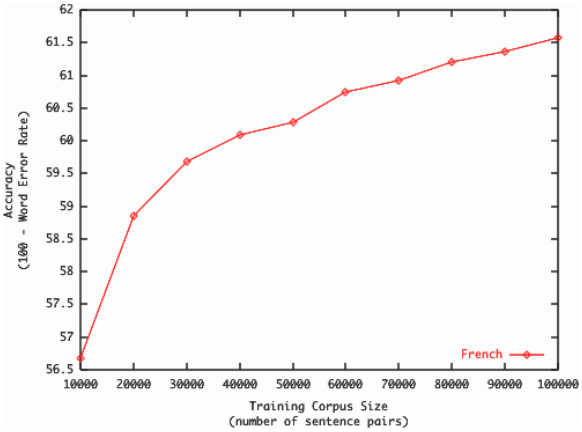
Chapter 6

Conclusion

This thesis presented a novel co-training method for statistical machine translation. Co-training is effective for a particular type of problem wherein the features used to label new items can be divided into distinct views, where each views contains sufficient information to perform the annotation. Many problems in natural language processing do not naturally divide into different views and have to be artificially constructed. Translation, on the other hand, has a very natural division of views onto the labels: labels are target translations, and views are source texts that can be used to produce those translations. Multiple views are achieved by using existing translations of the source text into other languages.

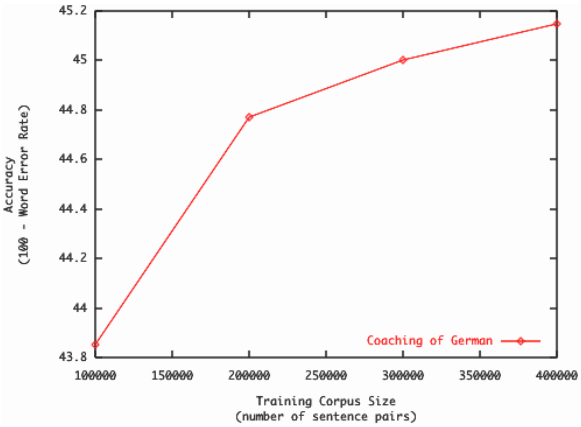
The use of multiple source documents to improve translation quality puts co-training for statistical machine translation in the category of multi-source translation (Kay (2000)). Previous work on multi-source translation (Och and Ney (2001)) has improved the quality of single translations only, by adapting the Brown et al. (1993) Fundamental Equation of Statistical Machine Translation to maximize the probability of a translation over multiple sources. The approach presented here is orthogonal to that work. Rather than adapting the process by which translations are selected, the procedure for training statistical translation models is altered. Just as in Och and Ney (2001) co-training uses multiple translation models to generate a set of candidate translations, and selects the best translation. Co-training for statistical machine translation goes one step further than that: it exploits the diversity and increased accuracy provided by the multiple source documents and integrates the resulting translations into

the training corpora. The motivation for increasing the size of parallel corpora is the fact that translation quality increases with size, with no signs of plateauing:



Because large parallel corpora are rare and difficult to assemble, the prospect of (semi-) automating their creation is appealing.

Experimental results suggest that translation models can experience at least small gains in accuracy from data sets augmented with machine translations. Better aligned multilingual corpora and further experimentation may yield even better results. Beyond the modest gains achieved through co-training similarly-sized translation models, the experiments conducted here give strong evidence for its being an effective technique for moving into the domain of new languages. Significant gains can be had in a special case of co-training, called “coaching”, wherein one translation model begins with no human translations (and therefore always abstains from adding items to the candidate pool) and the other translation model(s) contribute data to it:



This result has real-world implications: the cost associated with developing statistical machine translation systems for languages lacking parallel corpora may be lower than anticipated. Training on very large machine-translated parallel corpora can achieve quality equivalent to modestly sized human-translated corpora.

6.1 Future Work

There are a number of directions which this work could take. In the near-term I plan to try to improve the efficacy of co-training doing the following things:

- Develop a multilingual corpus with better alignments between sentences. The corpus that was used in this thesis was constructed from data previously used in Och and Ney (2001). Och and Ney assembled ten bilingual corpora aligning sentences in English with sentences in each of the non-English language of the European Union. I used the English sentences as keys to link translations across the languages, but found that there were a lot of misalignments. Building a multilingual corpus using the source documents would allow sentences to be aligned across all languages, thereby reducing the chance of introducing misalignments.

- Try different co-training parameters. Due to time constraints and limited computational resources, very few co-training parameters were tested. I plan to judicious vary the conditions under which co-training takes place in the hopes that different parameters will yield even more positive results.

- Experiment with sub-sentential units of training. Rather than including whole sentences from translation models, I plan to investigate the possibility of only including the phrases in which they are most confident. Though adding phrases would be less useful for estimating the distortion parameters of alignments, the fact that they contain fewer words would likely yield better translation estimation.

Appendix A

Software

This section briefly details the software used in this project. Three main software components were used:

- GIZA++, which is used to build translation models from parallel corpora
- CMU-Cambridge Language Modeling Toolkit, which is used to build language models from monolingual text
- ISI ReWrite Decoder, which is used to produce translations of new source language sentences, given a translation model and a language model of the target language.

Figure A.1 shows how each of these components interact to produce the translations.

A.1 GIZA++

GIZA++ (Och and Ney (2000)) is an extension of the program GIZA, which was a piece of the statistical machine translation toolkit developed by Al-Onaizan et al. at the 1999 summer workshop at the Center for Language and Speech Processing at Johns-Hopkins University (Al-Onaizan et al. (1999)). GIZA++ is a program for compiling translation models using aligned, bilingual corpora. It induces word-level alignments between sentences using EM, as described in Section 2.3.

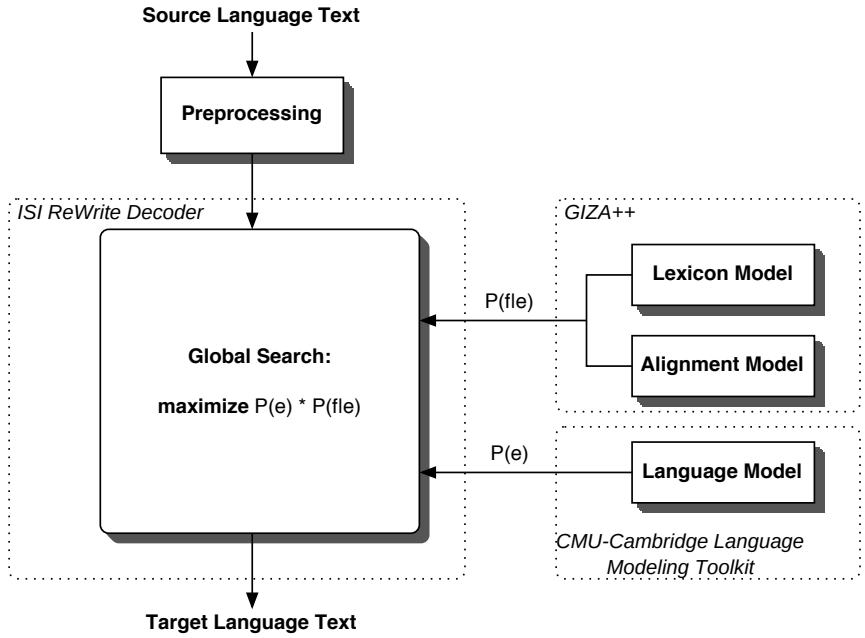


Figure A.1: Contributions of the software components to translation probabilities.

For my experiments I used the default training parameters, which included five iterations each of training for IBM Model 3 and Model 4. I generated 100 classes for each language files using the mkcls program, which is pointed to by the GIZA++ web page. I also note that I (unexpectedly) found varied performance on translation models depending on what CPU was used to train them. This was possibly due to a compiler error, but to be safe I constrained myself to train all translation models on the same machine.

A.2 CMU-Cambridge Language Modeling Toolkit

The CMU-Cambridge Language Modeling Toolkit (Clarkson and Rosenfeld (1997)) was used to create statistical language models for each language. All the models were backed-off tri-gram language models, using vocabularies with a capacity for one million words. Given the corpus statistics in Section 4.3 that means that the entire vocabulary for each language was retained.

A.3 ISI ReWrite Decoder

The ISI ReWrite Decoder uses the parameters for computing $P(e)$ provided by the CMU-Cambridge Language Modeling Toolkit, and the parameters for computing $P(f|e)$ provided by GIZA++ to translate new sentences. To translate a French sentence f, we seek the English sentence e which maximizes the product of those two terms. This process is called *decoding*. It is impossible to search through all possible sentences, but it is possible to inspect a highly relevant subset of such sentences. The ISI ReWrite Decoder does just that, and produces the single sentence from the subset that it inspects which best maximizes $P(e)P(f|e)$.

The decoding parameters that were used to reduce the search space, and thereby the time it took to produce each translation were to set the *M*axSwapSegmentSize to be 5 and then *M*axSwapDistance to be 10. The *M*axSwapSegmentSize is the maximum size for a phrase (or technically "cept" as defined in Brown et al. (1993)) that can be moved as a single unit. The *M*axSwapDistance is the maximum distance (in cepts) that a phrase can be moved from its original position. Neither of these optimization settings would have changed the effectiveness of co-training.

I originally set the *M*axTimePerSentence variable which limits the amount of time spent translating each sentence, but found that it caused variation based on the CPU that was being used. Since decoding took place on various machines, some with different models and most with different load demands, I repeated all experiments that had used this option without it.

Bibliography

Abney, S. (2002). Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F. J., Prudy, D., Smith, N., and Yarowsky, D. (1999). Statistical machine translation. Final Report, JHU Summer Workshop.

Al-Onaizan, Y., Germann, U., Hermjakob, U., Knight, K., Koehn, P., Marcu, D., and Kenji, Y. (2000). Translating with scarce resources. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Berger, A., Brown, P., Della Pietra, S., Della Pietra, V., Gillett, J., Lafferty, J., Mercer, R., Printz, H., and Ures, L. (1994). The candide system for machine translation. In *Proceedings of the 1994 ARPA Workshop on Human Language Technology*.

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory*. Morgan Kaufmann.

Brown, P., Della Pietra, S., Della Pietra, V., and Mercer, R. (1993). The mathematics of machine translation: Parameter estimation. *Compuatational Linguistics*, 19(2):263–311.

Callison-Burch, C. and Flournoy, R. (2001). A program for automatically selecting the best output from multiple machine translation engines. In *Proceedings of the Machine Translation Summit VIII*, Santiago de Compostela, Spain.

Charniak, E. (1996). Treebank grammars. In *Proceedings of AAAI-96*, Menlo Park, California.

Clarkson, P. and Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge toolkit. In *ESCA Eurospeech Proceedings*.

Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *Proceedings of the 1999 Conference on Empirical Methods in Natural Language Processing*.

Copestake, A., Flickinger, D., Malouf, R., Riehemann, S., and Sag, I. (1995). Translation using minimal recursion semantics. In *Proceedings of The Sixth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI95)*.

Corduneanu, A. and Jaakkola, T. (2001). Stable mixing of complete and incomplete information. AI Memo 2001-30, MIT Artificial Intelligence Laboratory.

Gale, W. and Church, K. (1993). A program for aligning sentence in bilingual corpora. *Compuatational Linguistics*, 19(1):75–90.

Kay, M. (2000). Triangulation in translation. Keynote at the MT 2000 conference, University of Exeter.

Knight, K. (1999). A statistical MT tutorial workbook. Prepared for the 1999 JHU Summer Workshop.

Koehn, P. and Knight, K. (2000). Estimating word translation probabilities from unrelated monolingual corpora using the EM algorithm. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Mitchell, T. (1999). The role of unlabeled data in supervised learning. In *Proceedings of the Sixth International Colloquium on Cognitive Science*, San Sebastian, Spain.

NieBen, S., Och, F. J., Leusch, G., and Ney, H. (2000). An evaluation tool for machine translation: Fast evaluation for MT research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, pages 39–45, Athens, Greece.

Nigam, K. and Ghani, R. (2000). Understanding the behavior of co-training. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*.

Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hong Kong.

Och, F. J. and Ney, H. (2001). Statistical multi-source translation. In *Proceedings of the Machine Translation Summit VIII*, pages 253–258, Santiago de Compostela, Spain.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 295–302, Philadelphia, PA.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). Bleu: a method for automatic evaluation of machine translation. IBM Research Report.

Pierce, D. and Cardie, C. (2001). Limitations of co-training for natural language learning from large datasets. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

Resnik, P. (1998). Parallel strands: A preliminary investigation into mining the web for bilingual text. In *Third Conference of the Association for Machine Translation in the Americas*.

Sankar, A. (2001). Applying co-training methods to statistical parsing. In *Proceedings of NAACL 2001*, Pittsburgh, PA.

Smith, N. (2002). From words to corpora: Recognizing translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, Philadelphia, Pennsylvania.

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*.