Linear Mixture Models for Robust Machine Translation

Marine Carpuat, Cyril Goutte and **George Foster**
Multilingual Text Processing
National Research Council
Ottawa, ON K1A0R6, Canada
firstname.lastname@nrc.ca

Abstract

As larger and more diverse parallel texts become available, how can we leverage heterogeneous data to train robust machine translation systems that achieve good translation quality on various test domains? This challenge has been addressed so far by repurposing techniques developed for domain adaptation, such as linear mixture models which combine estimates learned on homogeneous subdomains. However, learning from large heterogeneous corpora is quite different from standard adaptation tasks with clear domain distinctions. In this paper, we show that linear mixture models can reliably improve translation quality in very heterogeneous training conditions, even if the mixtures do not use any domain knowledge and attempt to learn generic models rather than adapt them to the target domain. This surprising finding opens new perspectives for using mixture models in machine translation beyond clear cut domain adaptation tasks.

1 Introduction

While machine translation tasks used to be defined by drawing training and test data from a single well-defined domain, current systems have to deal with increasingly heterogeneous data, both at training and at test time. As larger and more diverse parallel texts become available, how can we leverage heterogeneous data to train statistical machine translation (SMT) systems that achieve good translation quality on various test domains?

So far, this challenge has been addressed by repurposing techniques developed for more clear-cut domain adaptation scenarios, such as linear mixture models (Koehn and Schroeder, 2007; Foster and Kuhn, 2007; Sennrich, 2012b). Instead of estimating models on the whole training corpus at once, linear mixture models are built as follows: (1) partition the training corpus into homogeneous domain-based component, (2) train one model per component, (3) linearly mix models using weights learned to adapt to the test domain, (4) replace resulting model in translation system.

In this paper, we aim to gain a better understanding of the benefits of linear mixture models in heterogeneous data conditions, by examining key untested assumptions:

- Should mixture component capture domain information? Previous work assumes that training data should be organized into domains. When manual domain distinctions are not available, previous work uses clustering approaches to approximate manual domain distinctions (Sennrich, 2012a). However, it is unclear whether it is necessary to use or mimic domain distinctions in order to define mixture components.

- Mixture models are usually assumed to improve translation quality by giving more weight to parts of the training corpus that are more relevant to the test domain. Is this intuition still valid in our more complex heterogeneous training conditions? If not, how do mixture models affect translation probability estimates?

In order to answer these questions, we propose to study several variants of linear mixture models that reflect different modeling assumptions and different levels of domain knowledge. We first

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 499–509,
Baltimore, Maryland USA, June 26–27, 2014.

consider two methods for setting mixture weights: adaptation to the test domain via maximum likelihood, and uniform mixtures that make no assumption about the domain of interest (Section 2). Then, we will describe a wide range of techniques that can be used to define mixture components (Section 3). Again, these techniques reflect opposite modeling assumptions: manually defined domains and automatic clusters attempt to organize heterogeneous training sets into homogeneous groups that represent distinct domains, while random samples capture no domain information and simply provide different views of the training set.

We present an empirical investigation of all the variations outlined above using a strong system trained on large and diverse training corpora, for two language pairs and two distinct test domains. Our results show that linear mixtures reliably and robustly improve the quality of machine translation (Section 5). While they were originally developed for domain adaptation tasks, linear mixtures that have no domain knowledge can perform as well as traditional mixtures meant to perform domain adaptation. This suggests that improvements do not stem from domain modeling per se, but from better generic estimates from the heterogeneous training data. Further analysis shows that the linear mixture estimates are very different from estimates obtained using more explicit smoothing schemes (Section 6).

2 Linear Mixtures for Translation Models

Does domain knowledge yield better translation quality when learning linear mixture weights for the translation model of a phrase-based MT system? We leave the study of linear mixtures for language and reordering models for future work.

2.1 Maximum Likelihood Mixtures

In the standard domain adaptation scenario, the linear mixture combines translation probabilities learned on distinct sub-domains in the training corpus. The conditional translation probability of phrase t given s is defined as:

$$p(t|s) = \sum_{k=1}^{K} \lambda_k p_k(t|s) \tag{1}$$

where $p_k(t|s)$ is a conditional translation probability learned on subset k of the training corpus. Note that for all phrase pairs (s,t) that are not observed in component k of the training corpus, we will have $p_k(t|s) = 0$. As a result, the resulting distributions are not normalized.

The weights λ_k are learned to adapt the translation model to a development set, which represents the domain of interest. First, we extract all phrase pairs from the development set, using the same technique used to extract phrases from the training set as part of standard phrase-based MT training. This yields a joint distribution $\tilde{p}(s,t)$, which can be used to define a maximum likelihood objective:

$$\hat{\lambda} = \text{argmax}_{\lambda} \sum_{s,t} \tilde{p}(s,t) \log \sum_{k=1}^{K} \lambda_k p_k(s|t). \tag{2}$$

We use the Expectation Maximization algorithm to solve this maximization problem.

2.2 Uniform Mixtures

We will consider uniform mixtures where all components are weighted equally:

$$p(t|s) = \frac{1}{K} \sum_{k=1}^{K} p_k(t|s). \tag{3}$$

In contrast with maximum likelihood mixtures, uniform mixtures are not meant to adapt the translation model to a specific test domain. Instead, they combine estimates learned on various subsets of the data in the hope of obtaining a better estimate of the translation probability distributions from the (possibly heterogeneous) training domain as a whole.

2.3 Why Not Use Loglinear Mixtures?

In current machine translation systems, there are two straightforward ways to combine estimates from heterogneous training data: linear and loglinear mixtures. We argue that linear mixtures are a better model for combining domain-specific probabilities, since they sum translation probabilities, while loglinear mixtures multiply probabilities. In a loglinear mixture, a translation candidate t for a phrase s will only be scored highly if all components agree that it is highly probable. In contrast, in a linear mixture, t can be a top translation candidate overall even if it is not a preferred translation in some of the components. When the training data is very heterogeneous, linear mixtures are therefore preferable.

Previous work provides empirical evidence supporting this. For instance, Foster et al. (2010) found that linear mixtures outperform log linear mixtures when adapting a French-English system to the medical domain, as well as on a Chinese-English NIST translation task.

2.4 Estimating Conditional Translation Probabilities

Within each mixture component, we extract all phrase-pairs, compute relative frequencies, and use Kneser-Ney smoothing (Chen et al., 2011) to produce the final estimate of conditional translation probabilities $p_k(t|s)$. Per-component probabilities are then combined in Eq. 1 and 3. Similarly, baseline translation probabilities are learned using Kneser-Ney smoothed frequencies collected on the entire training set.

3 Defining Mixture Components

We assume that the heterogeneous training corpus can be split into basic elements that will be organized in various ways to define the K mixture components. Basic components could be documents or sets of sentences defined along various criteria. Sennrich (2012a) show that using isolated sentences as basic elements might not provide sufficient information, as smoothing component assignments using neighboring sentences benefits translation quality. In our experiments, basic elements are sets of parallel sentences which share the same provenance, genre and dialect, as we will see in Section 4.

We consider four very different ways of defining mixture components by grouping the basic corpus elements: (1) manual partition of the training corpus into domains, (2) automatically learning homogeneous domains using text clustering algorithms, (3) random partitioning, (4) sampling with replacement.

3.1 Manually Defined Domains

Heterogeneous training data is usually grouped into domains manually using provenance information. In most previous work, such domain distinctions are very clear and easy to define. For instance, Haddow (2013) uses European parliament proceedings to improve translation of text in the movie subtitles and News Commentary domains; Sennrich (2012a) aims to translate Alpine Club reports using components trained on European parliament proceedings and movie subtitles. Foster et al. (2010) work with a slightly different setting when defining mixture components for the NIST Chinese-English translation task: while there is no single obvious "in-domain" component in the NIST training set, homogeneous domains can still be defined in a straightforward fashion based on the provenance of the data (e.g., Hong Kong Hansards vs. Hong Kong Law vs. News articles from FBIS, etc.). We take a similar approach in our experiments. However, we will see that since our training data is very heterogeneous, we take into account other dimensions beyond provenance, such as genre and dialect information (Section 4).

3.2 Induced Domains Using Automatic Clustering Algorithms

We propose to use automatic text clustering techniques to organize basic elements into homogeneous clusters that are seen as sub-domains. In our experiments, we apply clustering algorithms to the target (English) side of the corpus only.

Each corpus element is transformed into a vector-space format by constructing a tf.idf vector representation. After indexing, we filter out stop-words as well as words occuring in a single document. We then weight each word token by the log of its frequency in the document, combined with an inverse document frequency (Salton and McGill, 1983) followed by a normalization to unit length. The cosine similarity between each pair of elements is obtained by simply computing the scalar product, resulting in a $N \times N$ similarity matrix, where N is the number of corpus elements.

For clustering, we used Ward's hierarchical clustering algorithm (Ward, 1963). We start with one cluster per corpus element, i.e. N clusters. From the similarity matrix, we identify the two most similar clusters and merge them into a single one, resulting in $N-1$ clusters. The similarity matrix is updated using Ward's method to form a $(N-1)\times(N-1)$ similarity matrix. The process is repeated on the new set of clusters, until we reach the target number of clusters K.

3.3 Random Partitioning

We consider random partitions of the training corpus. They are generated by using a random number generator to assign each basic element to one of K clusters. Resulting components therefore do not capture any domain information. Each com-

Arabic-English Training Conditions			
	segs	src	en
train	8.5M	262M	207M
Test Domain 1: Webforum			
	segs	src	en
dev (tune)	4.1k	66k	72k
web1 (eval)	2.2k	35k	38k
web2 (eval)	2.4k	37k	40k
Test Domain 2: News			
	segs	src	en
dev (tune)	1664	54k	51k
news (eval)	813	32k	29k

Table 1: Statistics for Arabic-English data: Number of segments (segs), source tokens (src) and English tokens (en) for each corpus. For English dev and test sets, word counts averaged across 2 references.

Chinese-English Training Conditions			
	segs	src	en
train	11M	234M	253M
Test Domain 1: Webforum			
	segs	src	en
dev (tune)	2.7k	61k	77k
web1 (eval)	1.4k	31k	38k
web2 (eval)	1.2k	29k	36k
Test Domain 2: News			
	segs	src	en
dev (tune)	1.7k	39k	24k
news (eval)	0.7k	19k	19k

Table 2: Statistics for Chinese-English data: Number of segments (segs), source tokens (src) and English tokens (en) for each corpus. For English dev and test sets, word counts averaged across 4 references.

ponent can potentially be as heterogeneous as the full training set.

3.4 Random Sampling with Replacement

All previous techniques assume that the training corpus should be partitioned into distinct clusters. We now consider mixture components that break this assumption, and simply represent several, possibly overlapping, views of the training corpus. They are defined by sampling basic corpus elements uniformly with replacement. This approach simply requires defining a number of samples K and the size n of each sample. We set the sample size n to the average size of the manual clusters. We do not fix K in advance: in order to provide a fair comparison with corpus partitioning techniques where components achieve coverage of the entire training set by definition, we keep generating samples until all basic elements have been used, and use all resulting K components.

When using uniform linear mixtures, this approach is similar to bootstrap aggregating (bagging) for regression (Breiman, 1996), where a more stable model is learned by averaging K estimates obtained by sampling the training set uniformly and with replacement.

4 Experiment Settings

We evaluate our linear mixture models on two different language pairs, Arabic-English and Chinese-English, and two different test domains.

4.1 Training Conditions

We use the large-scale heterogeneous training conditions defined in the DARPA BOLT project. Data statistics for both language pairs are given in Tables 1 and 2. Training corpora cover a wide variety of sources, genres, dialects, domains, topics.

For instance, for the Arabic task, the training corpus is originally bundled into 48 files representing different provenance and epochs. The data spans 15 genres (defined based on data provenance, they range from lexicon to newswire, United Nations, and many variants of web data such as webforum, weblog, newsgroup, etc.) and 4 automatically tagged dialects (Egyptian, Levantine, Modern Standard Arabic, and untagged). The distribution along each of these dimensions is very unbalanced, and each corpus file often contains text in more than one genre, epoch or dialect.

As a result, we divide the large training corpus into basic elements, based on the available metadata. We define basic corpus elements as a subset of sentences from the same provenance (i.e. corpus file), dialect and genre. For Arabic, splitting the original 48 files along these dimensions yields 82 basic elements. Similarly, the Chinese data was split into a set of 101 basic elements, using genre, dialects, as well as time span information to split the original files. Figure 1 shows the wide range of component sizes in the Arabic and Chinese collection. For Arabice, notice that several components are very small, from 6 lines and 90 words to 5.3 million lines and 137M words.

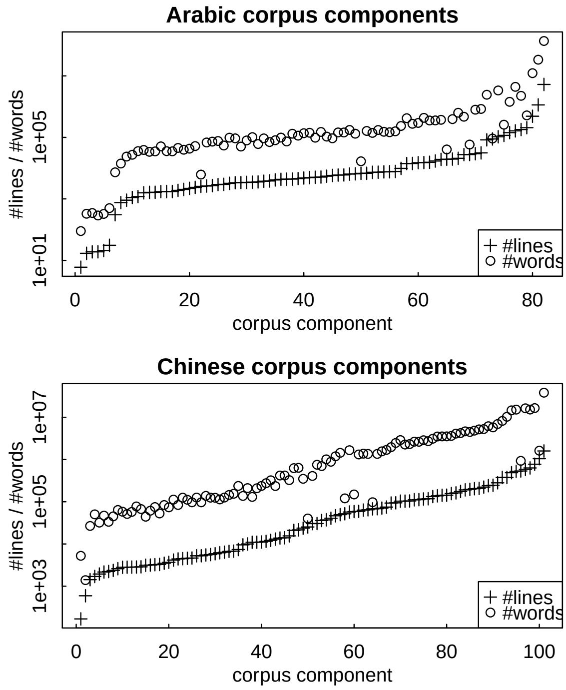

Figure 1: Sizes of the 82 Arabic-English (top) and 101 Chinese-English (bottom) corpus components.

4.2 Definition of Mixture Components

Manual partitions were created first by the system developers, based on intuitions on the nature of the test domain and manual inspection of the training data. The main goal was to group data into components that are large enough to reliably estimate translation probabilities, but small enough to be homogeneous. This resulted in $K_m = 10$ clusters for Arabic, and $K_m = 17$ for Chinese.

Automatic partitions are created as described in Section 3. Preliminary experiments with the hierarchical agglomerative clustering algorithm showed that the number of clusters used did not have a big impact on translation quality,[1] so we will only present results that use the same number of clusters as in the manual partitions (10 for Arabic and 17 for Chinese).

Results for random partitions are averaged across experiments run with four random seeds.

4.3 Test Domains

We consider two test domains, as described in Tables 1 and 2: webforum and news.

The webforum test domain is defined by development test sets made available through BOLT. It contains very informal text drawn from online discussion of various topics. Taking these data sets as the definition of the target domain, there is no single obvious in-domain section of the training corpus. For instance, for Arabic, the dev set sentences are almost exclusively written in the Egyptian dialect. Therefore, Egyptian webforum data is presumably the closest to the test domain, but Egyptian weblogs or mixed-dialect broadcast conversations could potentially be useful as well.

We also test the Arabic and Chinese systems on the news domain. The goal of these experiments is to evaluate the robustness of linear mixtures across different test domains. We use publicly available test sets from the NIST evaluation. The dev set used to learn maximum likelihood mixtures and tune the translation system is the NIST section of the 2006 test set. We evaluate system performance on the newswire section of the NIST 2008 test set.

4.4 Machine Translation System

We use an in-house implementation of a Phrase-based Statistical Machine Translation system (Koehn et al., 2007) to build strong baseline systems for both language pairs. Translation hypotheses are scored according to the following features:

- 4 phrase-table scores: Kneser-Ney smoothed phrasal translation probabilites and lexical weights, in both translation directions (Chen et al., 2011)[2]
- 6 hierarchical lexicalized reordering scores (Galley and Manning, 2008)
- a word penalty, and a word-displacement distortion penalty
- a Good-Turing smoothed 4-gram language model trained on the Gigaword corpus, Kneser-Ney smoothed 5-gram models trained on the English side of the training corpus, and an additional 5-gram model trained on monolingual webforum data.

Weights for these features are learned using a batch version of the MIRA algorithm (Chiang, 2012). Phrase pairs are extracted from several word alignments of the training set: HMM, IBM2, and IBM4. Word alignments are kept constant across all experiments.

We apply our linear mixture models to both translation probability scores, in each direction. The reordering and language models are not

[1] We tried $K = \{2, 4, \ldots, 18, 20\}$ for Arabic and $K = \{12, 14, \ldots, 20\}$ for Chinese, plus all basic components.

[2] The Arabic-English system uses 6 additional binary features which fire if a phrase-pair was generated by one of the 3 word alignment methods in each translation direction.

Test domain	Webforum	
Arabic eval	Forum1	Forum2
Linear mix	**39.67**	**40.60**
Loglinear mix	37.53	38.80
Chinese eval	Forum1	Forum2
Linear mix	**30.17**	**26.86**
Loglinear mix	27.65	23.78

Table 3: Impact of mixture type on translation quality as measured by BLEU.

adapted. Note that systems used to translate the web1 and web2 test sets are always tuned on the webforum tuning set, while systems used to translate data in the news domain are tuned on a news development set. The relevant tuning set is also used for learning maximum likelkihood mixtures when appropriate.

5 Findings: Impact on Translation Quality

5.1 Linear vs. Loglinear Mixtures

Before focusing exclusively on linear mixtures, we confirm that they outperform loglinear mixtures. This comparison was conducted on the webforum domain, using manually defined domains as components. For linear mixtures, we trained the weights using maximum likelihood. Loglinear mixture weights are trained by MIRA. Table 3 shows that linear mixtures yield consistently and significantly higher BLEU scores than loglinear mixtures, which is consistent with existing results (Foster et al., 2010, inter alia).

5.2 Impact of Mixture Components

We now focus on linear mixtures and measure the impact on translation quality of the various component types described in Section 3. In all cases, mixtures weights are estimated by maximum likelihood. Results are summarized in Table 4 for both Arabic and Chinese.

The main result is that all mixture models considered significantly improve on the "no mix" baseline for both languages. Directly using the 101 basic elements for Chinese and the 82 basic elements for Arabic significantly improves on the baseline. Grouping the basic elements into coarser clusters can further improve BLEU. For Arabic, automatic partitioning (randomly or by clustering) yields better BLEU scores than manual partitioning, while the manual and cluster-based domains yield the highest BLEU scores for Chinese.

Test domain	**Webforum**		**News**
Arabic eval	*web1*	*web2*	*news*
Cluster domains	40.11	40.60	**57.95**
Random partition	**40.43**	**40.63**	57.78
Random sample	39.94	40.36	57.85
Manual domains	39.67	40.60	57.63
Basic elements	39.83	40.63	57.57
No mix	38.64	39.21	56.59
Chinese eval	*web1*	*web2*	*news*
Cluster domains	29.82	26.34	**37.22**
Random partition	29.50	26.21	36.83
Random sample	29.47	26.17	36.70
Manual domains	**30.17**	**26.86**	36.90
Basic elements	29.29	26.25	36.17
No mix	28.61	25.63	35.96

Table 4: Impact of mixture component definition on BLEU score: there is no clear benefit to explicitly modeling domains.

5.3 Impact of Mixture Weights

Does domain knowledge yield better translation quality when learning linear mixture weights? We answer this question by comparing the translation quality obtained with maximum likelihood vs. uniform mixtures. The maximum likelihood weights are set once per domain, using the relevant domain development set, while the uniform mixture is the same across all test domains.

Table 5 shows that maximum likelihood weights generally have a slight advantage over uniform weights, especially in the Webforum domain. On "basic elements" in Arabic, the gain is a massive 5 BLEU points, which we attribute to the fact that, as shown in Figure 1, there are many more very small components in Arabic. Those get a disproportionate influence in the uniform mixture, hurting the overall performance. On the other hand, the uniform mixture performs better in the News domain. This might be explained by the fact that the tune and test sets are more distant in News than in Webforum, as suggested by the fact that the tuning BLEU scores are not as good at predicing test BLEU rankings in the news domain as in the webforum domain.

Overall, the difference in performance between the best linear mixture and the "no mix" baseline is 1.4 to 1.6 BLEU on Arabic, and 0.7 to 1.3 BLEU on Chinese. By comparison, the delta between the two weight setting approaches (maximum likelihood vs. uniform), depending on the partitioning technique, is below 0.4 BLEU for Arabic (except for Basic elements, +3.6 BLEU) and below 0.57 BLEU for Chinese. It is therefore clear that the gain from using linear mixtures is much larger than the influence of the mixture weight setting, except in the one specific case discussed above.

Taken together, these results show that linear mixtures can reliably and robustly improve the quality of machine translation. But surprisingly, linear mixtures that have no domain knowledge (random partition + uniform weights) can sometimes perform as well as traditional mixtures meant to perform domain adaptation. This suggests that improvements cannot be only explained by improved domain modeling.

Test domain	**Webforum**		**News**
Arabic eval	*web1*	*web2*	*news*
Cluster domains	**40.11**	**40.60**	57.95
w/ uniform mix	39.63	40.15	**58.21**
Random partition	**40.43**	**40.63**	57.78
w/ uniform mix	40.31	40.15	**58.18**
Random sample	**39.94**	40.36	58.06
w/ uniform mix	39.88	**40.56**	**58.65**
Manual domains	39.67	**40.60**	57.63
w/ uniform mix	**39.93**	40.18	**58.00**
Basic elements	**39.83**	**40.63**	57.57
w/ uniform mix	34.84	35.82	**58.46**
No mix	38.64	39.21	56.59
Chinese eval	*web1*	*web2*	*news*
Cluster domains	**29.82**	**26.34**	37.22
w/ uniform mix	29.44	25.94	**37.47**
Random partition	**29.50**	**26.21**	36.83
w/ uniform mix	29.43	25.89	**36.95**
Random sample	**29.47**	**26.17**	**36.70**
w/ uniform mix	28.47	25.54	36.61
Manual domains	**30.17**	**26.86**	36.90
w/ uniform mix	29.25	26.36	**36.95**
Basic elements	**29.29**	**26.25**	36.17
w/ uniform mix	29.23	25.81	**36.63**
No mix	28.61	25.63	35.96

Table 5: Impact of linear mixture weights on translation quality as measured by BLEU: using domain knowledge when setting weights has an unreliable impact.

6 Findings: Impact on Translation Probability Estimates

Thus far, all our experiments have measured the impact of different types of linear mixtures on overall translation quality. But what is the impact of these various estimations methods on the learned phrasal translation probability distributions themselves? More specifically, how do translation probabilities estimated using linear mixtures differ from global "no mix" estimates? If linear mixtures do not only capture domain knowledge as suggested by Section 5, do they simply perform a form of smoothing? If so, how does this implicit smoothing compare to more explicit smoothing schemes for translation probabilities?

6.1 How do linear mixtures affect translation probabilities?

Let us compare translation probabilities estimated directly on the entire corpus $P_{nomix}(t|s)$, with linear mixtures $p_{mix}(t|s) = \sum_{k=1}^{K} \lambda_k p_k(t|s)$. The difference between $p_{mix}(t|s)$ and $p_{nomix}(t|s)$ is hard to represent analytically in the general case, but studying a few particular cases can help us gain a better understanding.

First, we observe that linear mixtures scale down the contribution of component-specific source phrases. Assume that the phrase s occurs only once in the training corpus, with translation t. By definition, there is a single mixture component k such that $p_{mix}(t|s) = \lambda_k$, which is likely to be smaller than $p_{nomix}(t|s) = 1$. In the slightly more general case where s occurs more than once, but always in the same component k, then $p_{mix}(t|s) = \lambda_k p_{nomix}(t|s)$, which has no impact on the ranking of translation candidates for s, but yields a smaller feature value for the decoder.

Second, let us consider the case of very frequent "general language" phrases. They should have roughly the same translation distributions in all mixture components: If the $p_k(t|s)$ distributions are the same in each component, the λ_k values learned do not matter, they have no impact on $p_{mix}(t|s) = p_{nomix}(t|s)$.

In between these extremes, the impact of linear mixtures depends on the frequency and ambiguity of translation candidates t across mixture components. For instance, let us assume that the mixture components are somehow defined such that they partition the translate candidates t of a phrase s into separate clusters. In that case, for each t, there

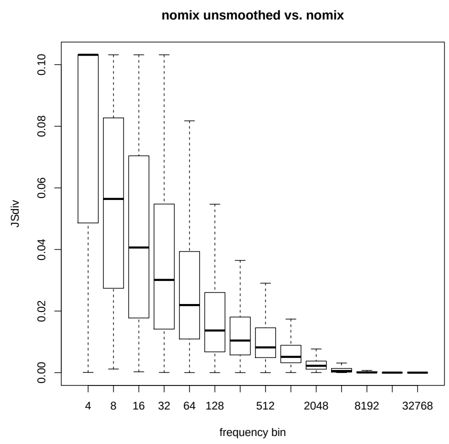

Figure 2: Comparing translation probability distributions with and without Kneser Ney smoothing for Chinese phrase-tables: boxplots of Jensen-Shannon divergences binned by source phrase frequency. For instance, the box and whisker at x = 8 represent the distribution of the values of Jensen-Shannon divergence between the unsmoothed and smoothed translation probability distribution for all Chinese phrases seen between 5 and 8 times during phrase extraction.

is a k such that $p_k(t|s) = p_{nomix}(t|s)$. The ranking of translation candidates t for s according to $p_{mix}(t|s)$ can be very different from $p_{nomix}(t|s)$, as controlled by the λ values used.

6.2 Smoothing Effects

As a basis for comparison, let us analyze the difference between unsmoothed relative frequencies and smoothed translation probabilities using a conventional smoothing scheme. We focus on the Kneser-Ney smoothing scheme (Chen et al., 2011), since it is used to smooth translation probabilities in the 'nomix" baseline as well as in all mixture components.

For seen phrase pairs (with $f(s,t) > 0$), the difference between Kneser-Ney estimates $p_{kn}(t|s)$ and relative frequency estimates $p_{rf}(t|s)$ can be written as:

$$p_{rf}(t|s) - p_{kn}(t|s) = \frac{D}{f(s)} - \frac{D * n(s) * p_b(t)}{f(s)} \tag{4}$$

where D is a discount coefficient, $f(s)$ is the raw frequency for source phrase s, $n(s)$ is the number of translation candidates for s in the phrase-table, $p_b(t)$ is a back-off distribution proportional to $n(t)$. The first term is a discount that increases when s is rare, while the second term adds some probability mass back, based on the frequency and degree of ambiguity of the target phrase t. Therefore, Kneser-Ney smoothing has primarily a discount effect, applied on rare source phrases. In addition, for more frequent and ambiguous phrases, the relative frequency can be adjusted up or down depending on how ambiguous s and t are.

Overall, there are some similarities between the impact of Kneser-Ney smoothing and linear mixtures, since one can expect that the translation distributions will diverge more from global relative frequencies for rare phrases than for frequent phrases. However, the discounting / down-scaling effects are controlled by very different parameters in linear mixtures than in Kneser-New smoothing. In order to better understand these differences in practice, an empirical analysis is required.

6.3 Empirical Comparison

How do linear mixtures and smoothing affect translation probabilities $p(t|s)$ in practice? We use the Jensen-Shannon divergence (Lin, 1991) to quantify the distance between (a) various mixture model estimates and (b) the global smoothed relative frequency estimates used in our baseline "no mix" experiments. In addition, we also compare the Kneser-Ney smoothed translation probabilities with unsmoothed relative frequencies, in order to highlight the difference between standard smoothing techniques and linear mixture models.

Figures 2 and 3 show the distributions of divergence values by source phrase frequencies for Chinese-English phrase-tables. The divergence from the global estimate is the largest for rare phrases in all cases, as expected based on previous Sections. However, the Figures also highlight the different behavior of linear mixtures compared to Kneser-Ney smoothing. The divergence values are much higher overall for the linear mixtures than for smoothing (note that the difference in range on the y axis in Figure 2 vs. Figure 3). In addition, linear mixtures have a large impact on translation probabilities not only on the rarest source phrases but also on relatively frequent phrases: in Figure 3, the median Jensen-Shannon divergence remains high for source phrases extracted up to 128 times from the training set[3], while the median value drops significantly as the frequency range in-

[3]Recall that we use multiple word alignment methods, so extraction counts are summed across all alignment methods.

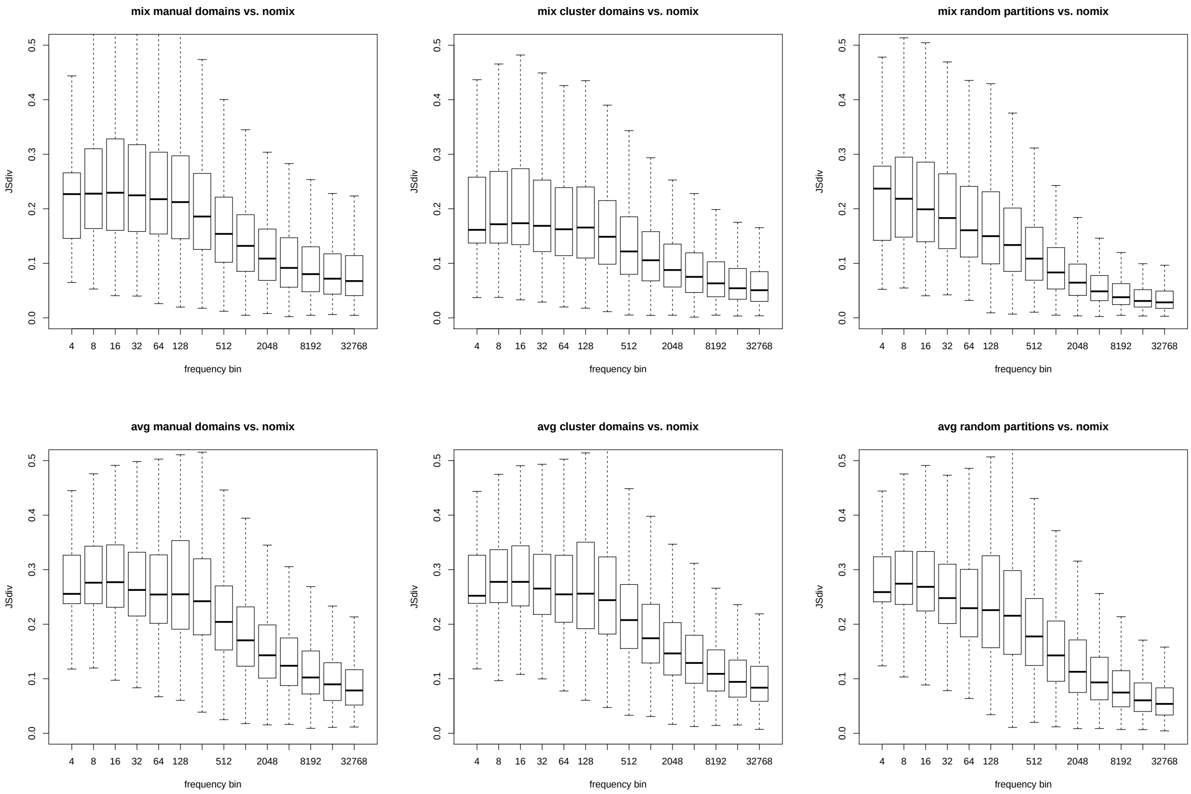

Figure 3: Comparing translation probability distributions of mixtures vs. "nomix" on Chinese webforum data, including EM weights (top row) and uniform weights (bottom row).

creases in Figure 2. In addition, uniform mixtures have an even higher impact on frequent phrases than mixtures based on EM weights.

Furthermore, the nature of mixture components used has a visible impact on the divergence distributions in Figure 3: random partitions yield lower divergences for very frequent source phrases.

Overall, the linear mixtures result in very different translation probability distributions than global estimates, including smoothed estimates. This suggests that standard smoothing techniques can be improved when learning from heterogeneous training data, and that mixture components are beneficial even when they do not explicitly capture domain distinctions.

7 Related work

Most previous work on domain adaptation in machine translation presupposes a clear-cut distinction between in-domain and out-of-domain data (Koehn and Schroeder, 2007; Foster and Kuhn, 2007; Duh et al., 2010; Bisazza et al., 2011; Haddow and Koehn, 2012; Sennrich, 2012b; Haddow and Koehn, 2012; Clark et al., 2012, among many others). We focused instead on a different less-studied question: how can we leverage training data drawn from a wide variety of sources, genres, time periods, to translate a domain represented by a small development set?

Many approaches focus on mapping the test domain to a single subset of the training data. In contrast, we show that the test domain can be flexibly represented by a mixture of many components. Yamamoto and Sumita (2007) cluster the parallel data using bilingual representations, and assign data to a single cluster at test time. Wang et al. (2012) show how to detect a known domain at test time in order to configure a generic translation system with domain-specific feature weights. Others select a subset of training data that is relevant to the test domain, using e.g., IR techniques (Hildebrand et al., 2005) or language model cross-entropy (Axelrod et al., 2011).

Closer to this work, Sennrich (2012a) proposes a sentence-level clustering approach to automatically recover domain distinctions in a heteorogeneous corpus obtained by concatenating data from a small number of very distant domains. The target domain was Alpine Club reports, while out of domain data sets comprised European parliament proceedings and movie subtitles. We address training conditions where the dimensions for organizing the training data are not as clear-cut, and show that partitions that do not attempt to mimick domain distinctions can improve translation quality. It would be interesting to see whether our conclusion holds in these more artificial training settings, and whether sentence-level corpus organization could help translation quality in our settings.

Finally, recent work shows that linear mixture weights can be optimized for BLEU, either directly (Haddow, 2013), or by simulating discriminative training (Foster et al., 2013). In this paper, we limited our studies to maximum likelihood and uniform mixtures, however, the various mixture component definitions proposed here can also be applied when maximizing BLEU.

8 Conclusion

We have presented an extensive study of linear mixtures for training translation models on very heterogeneous data on Arabic-English and Chinese-English translation tasks. In addition, we evaluated the robustness of our models across two distinct domains on the Arabic-English task.

Our results show that linear mixtures reliably and robustly improve the quality of machine translation. Improvements on the mixture-free baseline system range from 0.7 to 1.6 BLEU points depending on the components and weights used. While linear mixture translation models were originally proposed for domain adaptation tasks, we showed that linear mixtures that have no domain knowledge can perform as well or better than traditional mixtures meant to perform domain adaptation. This suggests that improvements with linear mixture models do not only stem from giving more weight to sections of the training data that are relevant to the test domain, as is assumed in a standard domain adaptation task. Improvements also come from averaging better generic estimates from the heterogeneous training data. In other words, in heterogeneous training settings, linear mixture models improve translation quality even though they do not perform domain adaptation. Finally, we show that while linear mixtures can be viewed as a smoothing technique, linear mixture estimates do not diverge from global estimates in the same way as Kneser-Ney smoothed translation probabilities. In particular, while smoothing primarily has a large discounting effect for rare source phrases, linear mixtures yield differences in translation probabilities for phrases with a wider range of frequencies.

These surprising results encourage us to rethink the use of mixture models, and opens up new ways of conceptualizing learning from heterogeneous data beyond domain adaptation. In future work, we will extend this study by varying the granularity of basic elements used to define mixture components, including sentences and phrases, and will explore how they compare with more general smoothing techniques.

Acknowledgments

This research was supported in part by DARPA contract HR0011-12-C-0014 under subcontract to Raytheon BBN Technologies. The authors would like to thank the reviewers and the PORTAGE group at the National Research Council.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362.

Arianna Bisazza, Nick Ruiz, and Marcello Federico. 2011. Fill-up versus interpolation methods for phrase-based SMT adaptation. *International Workshop on Spoken Language Translation (IWSLT).*

Leo Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

Boxing Chen, Roland Kuhn, George Foster, and Howard Johnson. 2011. Unpacking and transforming feature functions: New ways to smooth phrase tables. In *Proceedings of Machine Translation Summit*.

David Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *Journal of Machine Learning Research*, 13(1):1159–1187, April.

Jonathan H. Clark, Alon Lavie, and Chris Dyer. 2012. One system, many domains: Open-domain statistical machine translation via feature augmentation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas*.

Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Analysis of translation model adaptation in statistical machine translation.

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

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

George Foster, Boxing Chen, and Roland Kuhn. 2013. Simulating discriminative training for linear mixture adaptation in statistical machine translation. In *Proceedings of the XIV Machine Translation Summit*, pages 183–190.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 848–856.

Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on SMT systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432.

Barry Haddow. 2013. Applying pairwise ranked optimisation to improve the interpolation of translation models. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 342–347.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. In *European Association for Machine Translation*.

Philipp Koehn and Josh Schroeder. 2007. Experiments in Domain Adaptation for Statistical Machine Translation. In *Workshop on Statistical Machine Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Trans. Inf. Theor.*, 37(1):145–151, September.

Rico Sennrich. 2012a. Mixture-modeling with unsupervised clusters for domain adaptation in statistical machine translation. In *16th Conference of the European Association for Machine Translation (EAMT)*.

Rico Sennrich. 2012b. Perplexity minimization for translation model adaptation in statistical machine tra. In *Thirteenth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Wei Wang, Klaus Macherey, Wolfgang Macherey, Franz Och, and Peng Xu. 2012. Improved domain adaptation for statistical machine translation. In *10th biennial conference of the Association for Machine Translation in the Americas (AMTA)*.

Hirofumi Yamamoto and Eiichiro Sumita. 2007. Bilingual cluster based models for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 514–523.