RC25416 (WAT1310-015) October 9, 2013
Computer Science

IBM Research Report

Domain-Adaptive Translation Models Based on Bilingual Data Clustering

Fei Huang, Bing Xiang
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 208
Yorktown Heights, NY 10598
USA

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Domain-Adaptive Translation Models Based on Bilingual Data Clustering

Fei Huang
huangfe@us.ibm.com

Bing Xiang
bxiang@us.ibm.com

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

Abstract

We present a novel approach for SMT domain adaptation with different level of bilingual data clustering. We first merge bilingual corpora into topic-relevant clusters based on multiple features that capture corpus similarities. After initial corpus clustering, we select the most representative phrase pairs and sentence pairs for each cluster, then apply a refined sentence-level clustering using this seed data. As each sentence pair is re-assigned to the most likely cluster, the seed data for each cluster keeps growing, with the models being updated iteratively. At decoding time, for each input sentence we select the most relevant top K clusters, and combine their phrase tables with the baseline phrase table using dynamic weights. Experiments show 1.0-2.0 points of gain in BLEU on various test sets over an English-to-Chinese baseline system built with general models. Similar improvement is also observed on a Chinese-to-English MT system.

1 Introduction

Statistical Machine Translation (SMT) requires a large amount of parallel text for training. This training data is typically collected from various sources in different domains: some are newswire from news agencies and websites, while others are legal documents or proceedings from legislative council meetings such as the United Nations Assembly; some are formal, while others may cover informal text such as weblog/newsgroup (WB/NG) or even spoken language transcriptions from broadcast news or broadcast conversations (BN/BC); some are domain-specific data collected in-house while others may be data provided by a third party. Putting together all the bilingual data for the same language pair, the sizes of the different corpora vary significantly: some corpora include several million sentence pairs while others may only have a few thousand sentence pairs (especially for manually translated data, given the high cost of human translation). If all the data are used with equal weights to train an SMT system, the larger corpora will dominate the MT models. If only domain-specific data are selected to train an MT model, one needs to know the target domain beforehand, in addition to a possible data sparseness problem for certain domains.

Domain adaptation for SMT has been intensively investigated in the last several years, and most work has focused on either data selection and weighting, or model interpolation (see Section 2 for a detailed discussion of related work). Our approach tries to combine both of these approaches: we select domain-similar data (sentence pairs) from different corpora to form domain-specific clusters, and the translation models from each cluster are combined based on testset-specific weights selected on the fly. In particular, this work focuses on translation model adaptation, since language model adaptation has already been widely explored.

More specifically, our proposed approach groups training data into clusters at two levels: a coarse corpus clustering followed by a refined sentence clustering. The corpus clustering groups corpora based on their domain similarity using various features defined in section 3.1. Bilingual sentence pairs from the same corpus are grouped into the same cluster, even though they may cover different topics[1]. The sentence-level clustering allows these sentences to be re-grouped into different clusters, based on the similarity of a sentence pair to different clusters. This approach allows quick clustering of training data based on their domain similarity, followed by several iterations of more refined sentence re-clustering. Experimental results show that our approach improves translation quality by 1.0-2.0 BLEU points on various test sets over an English-to-Chinese baseline system built with general models, and ~1.0 BLEU point gain over a Chinese-to-English baseline system.

The rest of this paper is organized as follows: in section 2, we discuss related work. In section 3, we describe our bilingual corpus clustering. In section 4 we present our iterative sentence clustering. In section 5 we show how to apply our clustered models to SMT. Experiments on clustering and machine translation are presented in section 6, which is followed by the conclusion.

2 Related Work

A significant amount of work has been proposed for domain adaptation in SMT in the past. These work mostly fall into the categories of data selection and weighting or model interpolation.

For example, in (Hildebrand et al., 2005), an information retrieval technique is utilized to select comparable sentence pairs from the parallel corpus. The selected sentences are added to the original corpus to re-train the system. Another work that uses information retrieval methods is presented in (Lu et al., 2007), with both offline data optimization and online model optimization. The offline method adapts the training data by redistributing the weight of each training sentence pair, where the weights are obtained based on the information retrieval model. Then the online method adapts the translation model further by redistributing the weight of each predefined sub-model.

In (Koehn and Schroeder, 2007), two translation models are used in decoding within a log-linear framework. One is trained on in-domain data, with the other trained on out-of-domain data. An interpolation of in-domain and out-of-domain language models is also conducted, with the weights tuned so that the perplexity of the development set from the target domain is optimized.

Some previous work has focused on adapting the word alignment prior to phrase extraction. For example, in (Civera and Juan, 2007), a mixture extension of HMM alignment model is proposed. The mixture distinguishes which bilingual sentence pairs in the training data should contribute more to learn a given HMM component in the mixture.

Another mixture-model domain adaptation approach is presented in (Foster and Kuhn, 2007). Many alternatives are investigated in the mixture-model framework, comparing linear versus log-linear mixtures, cross-domain versus dynamic adaptation, etc. Different text distance metrics are also proposed and compared.

In (Yamamoto and Sumita, 2007), bilingual cluster based translation models are proposed. The bilingual sentence pairs in the training corpus are clustered such that the total source and target entropy is minimized through multiple iterations. The entropy is calculated by applying the language models from the previous step to the sentences in the current clusters at each iteration.

A discriminative corpus weighting approach is proposed in (Matsoukas et al., 2009). It assigns a weight to each sentence pair in the parallel training corpus to optimize a discriminative objective function (expected TER) on a designated tuning set. In this way, the translation model can be adapted to the target domain. But it requires that the tuning set used for weights optimization must share the same characteristics as the test set. Another work along the line of discriminative weighting is introduced in (Foster et al., 2010). It extends the work of (Matsoukas et al., 2009) with a finer granularity, learning weights on individual phrase pairs instead of sentences.

In (Axelord et al., 2011), three cross-entropy-based methods were utilized to select sentences that are relevant to the target domain, from a general-domain large parallel corpus. The performance was further improved by combining the domain-adapted models with a true in-domain model.

Most recently, an ensemble decoding approach was proposed in (Razmara et al., 2012), which combines a number of translation systems dynamically at the decoding step. The model combination was applied using various mixture

[1] For example, a newswire corpus may cover wide range of topics, including technical, political and sports articles.

operations. It shows better performance over the mixture model introduced in (Foster et al., 2010). Another recent work on domain adaptation is in (Su et al., 2012), which adapts the translation model by utilizing in-domain monolingual topic information.

Our work differs from all of the previous methods outlined above by introducing two-level data clustering (corpus-level and sentence-level) during training and then sentence-level dynamic model combination at decoding time, as will be described in the next few sections.

3 Bilingual Corpus Clustering

Our bilingual data includes 105 English-Chinese corpora from 50+ sources. These corpora cover a wide range of topics, and contain very unbalanced amount of data (5K ~ 10M sentence pairs in each corpus). Starting from bilingual sentence pairs from different corpora, we run automatic word alignments on all the training data and extract bilingual phrase tables for each corpus. We also build source and target LMs for each corpus, using the corresponding source and target side text in the bilingual corpora.

3.1 Features for Corpus Clustering

To cluster corpora with similar topics, we employ the following features to measure the topical similarity between any two corpora. The features are defined over both bilingual phrase tables and monolingual text to improve the identification of topic relevancy. We use the following list of features:

1. Symmetric Kullback-Leibler (KL) distance between their phrase tables;
2. Source phrase overlap ratio between their phrase table;
3. Source LM perplexity;
4. Target LM perplexity.

Given phrase tables T_a , T_b from corpora (*a*, *b*), the symmetric KL distance between T_a and T_b is

$$d(T_a, T_b) = KL(T_a, T_b) + KL(T_b, T_a), \qquad (1)$$

where

$$KL(T_a, T_b) = \frac{1}{|S|} \sum_{s \in S} \sum_{t \in T(s)} p_{T_a}(t \mid s) \log \frac{p_{T_a}(t \mid s)}{p_{T_b}(t \mid s)}$$

- *S* is the set of common source phrases in T_a and T_b;
- *T*(*s*): the union of target phrase translations for common source phrase *s* in T_a and T_b.
- $p_T(t \mid s)$ is the conditional probability of a target phrase *t* given a source phrase *s*, in phrase table *T*. If phrase pair (*s*, *t*) does not exist in the phrase table, $p(t \mid s)$ is set to a default value inversely proportional to the phrase table size.

As the *KL* function is asymmetric, the symmetric KL distance is the sum of $KL(T_a, T_b)$ and $KL(T_b, T_a)$. A small KL distance indicates that the corpus pair shares a similar domain.

The source phrase overlap ratio is defined as

$$r(T_a, T_b) = \frac{|S|}{|S_a|} \times \frac{|S|}{|S_b|} \qquad (2)$$

where S$_a$ and S_b are the set of source phrases in T_a and T_b, respectively, and *S* is the set of common source phrases, i.e. $S = S_a \cap S_b$. |*S*| is the size of a phrase set *S*. This feature computes the percentage of common source phrases in each phrase table so a higher overlap ratio suggests that two corpora are close.

The source and target LM perplexity features are straightforward. Given the source and target text t_a and t_b from corpora *a* and *b*, we build a 5-gram LM, m_a, using t_a, then compute the log probability and perplexity of corpus *b*'s text, t_b, with model m_a.

$$P(t_b \mid m_a) = 2^{-\frac{1}{N}\sum_{i=1}^{N} \log P_{m_a}(w_i)} \qquad (3)$$

where w_i is the *i*-th word in t_b, and *N* is the number of words in t_b. A lower perplexity indicates that the two corpora have similar domains.

The first two features utilize bilingual information (source and target sentence pairs and the corresponding word alignments). In particular, the KL distance measures the similarity between phrase pairs, and the source phrase overlap ratio measures the similarity between source phrases. The last two features exploit only monolingual information. The distance between two corpora, *a* and *b*, is defined as the following weighted sum of feature scores:

$$\begin{aligned} D(a,b) = {} & w_{KL} d(T_a, T_b) + w_r (1 - r(T_a, T_b)) + \\ & w_s (P_s(t_b \mid m_a) + P_s(t_a \mid m_b)) + \qquad (4) \\ & w_t (P_t(t_a \mid m_b) + P_t(t_b \mid m_a)) \end{aligned}$$

where w_{KL}, w_r, w_s, w_t are the weights for symmetric KL distance, source phrase overlap ratio, sum of source LM perplexities and sum of target LM perplexities, respectively. All the feature values are normalized between (0, 1] so that they are comparable. The feature weights are tuned to maximize the purity score defined by Eq. (5) so that clusters are more homogeneous in terms of source and genre.

$$purityScore = \frac{1}{C}\sum_{i=1}^{C}\frac{\max_j S_{ij}}{\sum_j S_{ij}} \quad (5)$$

where C is the number of clusters, and S_{ij} is the number of corpora from the same source j in cluster i.

3.2 Clustering Algorithm

The clustering algorithm we use for corpus-level clustering is similar to k-means. Assume we want to cluster *M* corpora into *C* clusters. The number of clusters is determined empirically, based on the change in intra-cluster distances, as described in Section 6.1.

The clustering algorithm is as follows:

1. Randomly select *C* corpora from the *M* corpora as the seeds for the clusters.
2. For each corpus, assign it to a specific cluster if the distance between the corpus and the cluster seed is the shortest. The distance is the weighted sum of 4 features in Eq. (4).
3. Find the cluster medoid within each cluster. It is defined as the corpus that has the minimum sum of distances to all other corpora in the same cluster.
4. Re-assign corpora to the clusters. For each corpus, assign it to a specific cluster if the distance between the corpus and the cluster medoid is the shortest.
5. Go back to step 3 for more iterations.

The algorithm stops when there is no further change in the clusters or the maximum number of iterations is reached. We set the maximum number of iterations to be 10.

4 Iterative Sentence Clustering

Corpus clustering provides preliminary grouping of corpora based on their domains. Sentence-level clustering allows sentences from the same corpus to be grouped into different clusters, if they cover different domains. Similar to the corpus clustering, we compute the domain similarity between a sentence pair and each of the clusters, and group the sentence pair to the closest cluster.

4.1 Features for Sentence Clustering

Some features in corpus clustering can still be used for sentence clustering, for example, source and target LM perplexities. The difference is, t_b will be the source and target sentences, instead of the text from corpus *b*. However, the other two features, the phrase table KL distance and source phrase overlap ratio, can no longer be used because the phrase table extracted from the sentences pair alignments will be tiny. Here we use a new feature, sentence phrase pair translation probability, to capture the similarity between two phrase tables (T_s, T_c) , where T_s is extracted from a bilingual sentence pair and T_c from a bilingual corpus cluster.

$$s(T_s, T_c) = \frac{1}{|(s,t)|}\prod_{(s,t)\in T_s} P_{T_c}(t \mid s) \quad (6)$$

This feature uses the cluster phrase table to compute the conditional probability of all phrase pairs in the sentence phrase table. If any phrase pair is unseen in the cluster phrase table, a default probability is used. Higher probability indicates more similarity between the sentence pair and the cluster.

For each sentence pair, we compute all three features with respect to each cluster models. The sentence pair is assigned to a cluster only if that cluster is the closest as measured by at least 2 out of 3 features.

4.2 Seed Model for Sentence Clustering

In section 4.1, the source and target LMs as well as the corpus phrase table are obtained from seed sentence pairs instead of all the bitext in the cluster. The seed data is considered as the most representative set of sentence pairs in the cluster. We now describe how to select the seed data: we build source and target LMs using all of the bitext in a cluster, then compute each sentence pair's source and target perplexities. The top *N%* of the sentence pairs with minimum perplexity are selected as seed data, where *N* is manually specified.

4.3 Iterative Clustering Algorithm

The following algorithm is used for iterative sentence clustering:

1. Starting with corpus clusters, build source and target LMs for each cluster using all its bitext;
2. Compute the perplexity for each sentence pair in the cluster, and select the smallest top *N%* as seed data. These data are considered as "assigned";
3. Build source and target LMs and phrase tables from the seed data and their word alignments;
4. For each "unassigned" sentence pair, compute the source and target LM perplexities and sentence phrase pair likelihood using all the cluster's seed models;
5. A sentence pair is grouped to cluster *k* by majority voting: *k* is the closest cluster measured by at least 2 features;
6. The sentence pair will stay in the original cluster if none of the three features agree;
7. Repeat from step 1 with the new clusters.

Assignment in step 5 is a hard decision: a sentence pair will only be grouped into one of the clusters. Another option is soft clustering: a sentence pair can be grouped into cluster *k* with probability r/3 if r feature(s) choose *k* as the closest cluster. We report results using both hard and soft sentence clustering in the Section 6.

Note that with more iteration in sentence clustering, the *N* grows. As a result, more and more sentence pairs will be included as seed data.

5 Dynamic TM Combination

Given the phrase tables (aka TM) extracted from each cluster's sentence pairs and their word alignments, how should they be combined in statistical machine translation? During decoding, for each input sentence, we select the top-*K* clusters that are most relevant in terms of topic similarity, then combine these cluster-specific TMs with the baseline model TM (i.e. the phrase table extracted from all data after pruning). Notice that the top-*K* clusters and their weights are dynamically determined according to the input sentence. The phrase translation probability is therefore

$$p(t \mid s) = \frac{C_0(s,t) + \sum_{i \in topK} w_i C_i(s,t)}{\sum_{t'} [C_0(s,t') + \sum_{i \in topK} w_i C_i(s,t')]} \tag{7}$$

where $C_0(s,t)$ is the co-occurrence frequency of phrase pair (s,t) in the baseline TM, i is the id of each of the top-K clusters, $C_i(s,t)$ is the phrase pair's co-occurrence frequency in cluster i. and w_i is the weight of cluster i's phrase table. In this paper the weight is estimated based on the source LM perplexity. We train a 5-gram source LM for each cluster, and compute the perplexity of the test sentence. The weight for cluster i is defined as:

$$w_i = \frac{perp_{\min}}{perp_i} \tag{8}$$

where $perp_{min}$ is the minimum perplexity across all clusters, and $perp_i$ is the perplexity for cluster i. As low perplexity indicates a cluster and the test set share a similar domain, that cluster is assigned higher weight. The closest cluster's weight is always assigned to 1.0. We sort the clusters based on their weights, and only keep the top-*K* clusters for TM combination. In our experiments, we set *K*=3.

In the mixture-model domain adaptation approach of (Foster and Kuhn, 2007), all of the mixture models are dynamically combined for each document. Our approach conducts sentence-level adaptation, thus it better captures topic differences among sentences within one document. Additionally, we combine the baseline TM with the most relevant top-*K* cluster TMs, in order to maintain a balance of domain relevance and good coverage in the combined model.

6 Experiments

Our experimental setup is English-Chinese translation. We have 105 English-Chinese corpora from political, technical, legal, financial and other domains, with genres covering newswire, weblog, transcription of broadcast news and broadcast conversations. Some of the data was collected by the LDC, and some is from in-house collections. The sizes of the corpora vary significantly, ranging from several thousands to several millions of sentence pairs. In total, there are over 20 million sentence pairs.

We ran automatic word alignments (HMM alignments in both directions and Chinese-English MaxEnt alignment (Ittycheriah and Roukos, 2005)) on all the bitext, extracted a phrase table and trained source and target LMs for each corpus. We computed the distance between any two corpora, i.e., the weighted sum of the four features described in section 3.1. We compute cluster distance for all 105 corpus pairings.

6.1 Corpus Clustering

As mentioned earlier, we determine the number of clusters based on the change in the intra-cluster distances. The intra-cluster distance is defined as the average of the distance between each cluster medoid and all other corpora in the same cluster. Figure 1 shows the decrease of the average intra-cluster distance as the number of clusters increases. There is no appreciable change when the number of clusters is greater than 10 so we choose 10 as the number of clusters in this work.

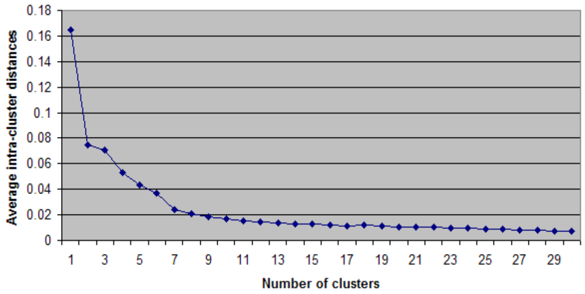

Figure 1: The average intra-cluster distances with different number of clusters.

Table 1 shows the 10 clusters obtained from corpus clustering, as well as the number of sentences and a representative corpus from each cluster. As each domain's training data is of different size, the cluster sizes vary a lot, from 0.2M to 5M sentence pairs. Still, we observed a clear domain-specific corpora grouping in each cluster: cluster 0 is mostly composed of in-house technical data translations between English and traditional Chinese, cluster 1 and 6 are the technical data translations between English and simplified Chinese. Some clusters are mostly formal text, such as LDC newswire, legal domain (HK law) and the United Nations proceedings, while others may cover informal text (weblog/newsgroup) or spoken languages (BC/BN).

Cluster-domain	**#of sentences**	**#of corpora**
Tech-ZH-Traditional	3.1M	17
Tech-ZH-Simplified	2.4M	10
Financial news	1.3M	8
LDC newswire	1.7M	18
HK Hansard	2.6M	3
HK law	1.4M	2
Tech-ZH-Simplified	3.2M	19
BC/BN	0.9M	13
Weblog/Newsgroup	0.2M	5
United Nation	5.0M	10

Table 1: Sample 10 clusters obtained from corpus clustering.

Cluster Domain	**Weights on TMB**
Tech-ZH-Traditional	**0.87**
Tech-ZH-Simplified	**0.61**
Financial news	0.25
LDC newswire	0.24
HK Hansard	0.26
HK law	0.1
Tech-ZH-Simplified	**1**
BC/BN	0.15
Weblog/Newsgroup	0.19
United Nation	0.24

Table 2: Sample cluster weights with source LM perplexity on the technical manual test set.

Table 2 shows each cluster's weight, estimated from source LM perplexity on a technical manual test set. All of the tech-domain clusters have large weights, while legal and spoken language domain clusters have the smallest weights (0.1 for HK law and 0.15 for BC/BN).

To evaluate the effect of the proposed corpus clustering on machine translation, we conducted several MT experiments. Our phrase-based decoder is similar to the one described in (Koehn et. al. 2003), where various features are combined within a log-linear framework. These features include source-to-target phrase translation score based on relative frequency, source-to-target and target-to-source word-to-word translation scores, a 5-gram language model score, distortion model scores and word count. We extract phrase translation pairs from each corpus cluster, and build the corresponding translation models (TMs). During decoding, unlike some of the earlier work, we do not rely on development sets to tune the TM mixture weights. Instead, each TM's weight is determined based on the corresponding source LM perplexity on each test sentence as described in Section 5. Therefore, we achieve sentence-level dynamic TM combination (the base general TM combined with top-K topic-relevant TMs).

For English-Chinese MT experiments, we selected three test sets: two are from technical domains focusing on technical manual translation (**Manual**) and online technical support and customer service (**eSupp**), respectively. The third test set is a general domain newswire test data from NIST-**MT08** which covers a wide range of domains by itself. The translation quality is measured by automatic metrics such as BLEU score (Papineni et. al., 2001). All the translation experiments use the same set of features and weights, thus the change in translation quality is solely due to different weighted TM combinations.

Table 3 shows translation results using our dynamic TM combination based on clustered corpora (**CorpCls**). Compared with a general MT system (**General**), which is trained using all of the data with equal weights, the corpus-clustering model improved translation by 0.5-1.0 BLEU points. Further analysis shows that larger gain is obtained when the testset is more homogenous, i.e., the sentences in the test set belong to the same domain (such as **Manual**). When the test set covers a wide range of domains (such as **MT08**), we see relatively small gains since none of the clusters match the test set perfectly.

We also compare with the results from a manual ad-hoc corpus clustering (**Manual Cls**), where all technical documents are grouped into one cluster (9.7M sentences), the UN corpus are grouped into another cluster (5M sentences), and the remaining data (about 7M sentences) are grouped together. With the same dynamic TM combination algorithm, this simple clustering scheme outperforms the baseline general system on technical document translations (**Manual** and **eSupp**), but performs worse on the news test set. Compared to our dynamic corpus-based clustering, it is worse on all three test sets.

	Manual	**eSupp**	**MT08**
# of Sentences	582	600	1859
General Baseline	30.77	30.43	34.70
Manual Cls	31.51	30.99	34.01
Corp Cls	**31.71**	**31.18**	**35.18**
Sent Cls	**32.80**	**31.73**	**35.70**

Table 3: English-Chinese MT comparison on BLEU scores: general model vs. corpus clustering model vs. sentence clustering model on different test sets. Manual Cls is a corpus-based clustering scheme done by human.

We applied the same strategy on Chinese-English MT as well. Trained with the same English-Chinese bilingual data (just swap the source and the target), the corpus clustering MT models (with the same clustering configuration) obtains 1.0 BLEU points improvement over the general baseline on both GALE newswire and web-blog test sets[2], as shown in Table 4.

	GALE10-NW	**GALE10-WB**
# of Sentences	1155	1239
General Baseline	14.81	23.35
Corp Cls	**15.76**	**24.38**

Table 4: Chinese-English MT comparison on BLEU scores: general model vs. corpus clustering model on newswire and weblog test sets.

6.2 Sentence clustering

Sentence clustering follows the corpus clustering. Any sentence pair can be grouped to the closest cluster which may or may not include its original corpus. In fact, most sentence pairs from a corpus will stay in the original cluster because these sentence pairs are used to build the cluster's seed model. We run a few iterations of sentence clustering after corpus clustering. Table 5 shows the redistribution of the sentence pairs from cluster 0 (Tech-ZH-Traditional) after the 1st iteration of

[2] The nw testset has single reference translation while the wb testset has 4 reference translations which is why the BLEU score for the wb test set is so much higher than that for the nw test set.

sentence clustering: most sentences stay in cluster 0. For those sentence pairs changing clusters, cluster 1 and 6 absorb the most because they share similar domains with cluster 0 (all are in technical domain).

As similar sentence pairs will be grouped together after each iteration, the number of sentence pairs changing clusters will decrease. It is observed that after the first iteration, 8.5% of the sentence pairs change clusters, and after the second iterations, only 1.6% of the sentence pairs changed clusters. With smaller number of sentence pairs changing clusters, the effect on each cluster's translation model diminish as well. Therefore we stopped sentence cluster after the 2nd iteration.

Cluster Domain	# of sentences
Tech-ZH-Traditional	1952198
Tech-ZH-Simplified	167139
Financial news	1181
LDC newswire	845
HK Hansard	1619
HK law	3315
Tech-ZH-Simplified	156022
BC/BN	1330
Weblog/Newsgroup	1814
United Nation	1494

Table 5: The distribution of sentence pairs from **Cluster 0** (Tech-ZH-Traditional) after the 1st iteration of sentence clustering.

Table 6 shows the translation results with hard and soft sentence clustering on the technical **Manual** test set, compared with the baseline and corpus clustering results. Sentence clustering improves over the corpus clustering by an additional 0.4 BLEU points (for soft clustering) and 1.1 BLEU points (for hard clustering). Due to the better match between the top-1 cluster and the test set, hard clustering does not spread relevant training data across multiple clusters, like soft clustering does. Overall, sentence clustering improved over the general system baseline by 2 BLEU points, and improved over corpus clustering by 0.5-1.0 BLEU point, as seen in the last row (**SentCls**) in Table 3. Table 7 shows some translation outputs using different models. One may notice the improved translations, especially on technical terms, by using the cluster-specific models.

We also compare our results with the LM-based sentence selection methods as proposed in (Axelrod et al., 2011). In our experiments replicating their approach, the best result is obtained by selecting domain-relevant sentences based on in-domain LM perplexities, where the source and target LMs are trained with 30K in-domain sentences. As shown in Table 6, our hard sentence clustering approach outperforms the LM-based sentence selection approach by 1 BLEU point.

	Tech Manual
General	0.3077
Corpus-clustering	0.3171
Soft Sentence-clustering	0.3207
Hard Sentence-clustering	**0.3280**
LM-based sentence selection (Axelrod et al., 2011)	0.3175

Table 6: MT comparison on BLEU scores: general model vs. corpus clustering vs. sentence clustering model on technical manual translation test set.

6.3 Dynamic TM combination

(Foster and Kuhn 2007) proposed several weighting schemes to combine TM mixtures, including tf/idf, LSA, perplexity and EM-based mixture weights. Based on their experiments, the overall difference of different weighting schemes is small. We compared our TM combination method with the EM-based weighting scheme. The result, as shown in Table 8, indicates that combining the base TM with top-K cluster-specific TMs outperforms merging all cluster-specific TMs with EM-trained weights on eSupp and MT08 news test set. The EM combination scheme works better on the technical manual test set. Further analysis shows that the EM-trained weights have sharper distribution: the top-1 cluster TM takes most of the total weights. When the test set is homogeneous and is a good match to the top-1 cluster (as is the case for the technical manual translation), it would perform better. Otherwise, adding the base TM in the mixture provides better domain coverage.

Source	Xtools Compare Merge Modeler Client runtime **component (组件)**.
Baseline	Xtools Compare Merge Modeler Client 执行时期**元件**。
CorpCls	Xtools Compare Merge Modeler Client 运行时**组件**。
SentCls	Xtools Compare Merge Modeler Client 运行时**组件**。
Source	Add WAS 5.1 ND + Deployment Manager **Profile (概要文件)**
Baseline	新增 WAS 5.1 ND + Deployment Manager **设定档**
CorpCls	新增 WAS 5.1 ND + Deployment Manager **设定档**
SentCls	新增 WAS 5.1 ND + Deployment Manager **概要文件**

Table 7: MT translation comparison: general model vs. corpus clustering vs. sentence clustering model. Changed translations are highlighted with bold fonts, and correct translations are inserted after the corresponding English phrases in the source sentences.

	Manual	**eSupp**	**MT08**
Base+top-K perp.	33.08	33.23	39.06
EM(Foster and Kuhn 2007)	34.27	32.72	37.41

Table 8 [3] : MT comparison with different TM combination schemes.

7 Conclusion and discussion

We presented a novel unsupervised approach to iteratively cluster training data at the corpus and sentence level using multiple features. Corpus clustering groups bilingual training corpora according to their domains, topics and genres, while sentence clustering further refines these corpora clusters, allowing some out-of-domain sentences joining other clusters. Such data clustering enables building domain-specific translation models. For example, it is possible to train word translation lexicons, alignments and phrase tables with different pruning strategies for each cluster.

The features for corpus clustering can be pre-computed for each corpus, and the number of corpora is typically no more than a few hundred. So the corpus clustering does not require huge computation cost but still achieves significant improvement over the general baseline system as observed in our experiments. For the sentence clustering, the distance between each "unassigned" sentence pair to every cluster must be computed in each iteration, which is computationally more expensive but leads to refined data clusters and improved translation quality. It is suitable for the corpora including heterogeneous data sources, such as data from web crawling.

During decoding, the combination of the top *K* cluster-specific phrase tables and the baseline phrase table shows improvement of 1.0-2.0 BLEU points on various test sets over a general English-Chinese baseline MT system. Similar improvement is also observed on a Chinese-English MT system when translating newswire and weblog test sets. According to our experiment results, such combination strategy outperforms the weighted combination of all the cluster-specific TMs as described in (Foster and Kuhn 2007).

For the future work, we would like to explore new features to capture the similarity between corpora and sentences. We would also investigate more efficient algorithms to reduce the computation cost for sentence clustering, for example, selecting documents instead of sentences as the clustering unit.

[3] This experiment is conducted on an improved English-Chinese MT system, so the numbers are not directly comparable with those in Table 3.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Edinburgh, Scotland, UK. July.

Jorge Civera and Alfons Juan. 2007. Domain adaptation for statistical machine translation with monolingual resources. Proceedings of the Second Workshop on Statistical Machine Translation. Prague, Czech Republic. June.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. Proceedings of the Second Workshop on Statistical Machine Translation. Prague, Czech Republic. June.

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

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Adaptation of the translation model for statistical machine translation based on information retrieval. Proceedings of the 10th EAMT Conference, Budapest, May.

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for Arabic-English machine translation. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT '05). Association for Computational Linguistics, Stroudsburg, PA, USA, 89-96.

Philipp Koehn, Franz Josef Och, and Daniel March. 2003. Statistical phrase-based translation. In Proceedings of the Human Language Technology Conference of the NAACL, pages 127-133, Edmonton, May. NAACL.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. Proceedings of the Second Workshop on Statistical Machine Translation. Prague, Czech Republic. June.

Yajuan Lu, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. Proceedings of the 2007 Conferences on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Prague, Czech Republic.

Spyros Matsoukas, Antti-Veikko Rosti, and Bing Zhang. 2009. Discriminative corpus weight estimation for machine translation. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Singapore.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. Proceedings of ACL, pp. 311-318.

Majid Razmara, George Foster, Baskaran Sankaran, and Anoop Sarkar. 2012. Mixing multiple translation models in statistical machine translation. Proceedings of ACL, pp. 940-949.

Jinsong Su, Hua Wu, Haifeng Wang, Yidong Chen, Xiaodong Shi, Huailin Dong, and Qun Liu. 2012. Translation model adaptation for statistical machine translation with monolingual topic information. Proceedings of ACL, pp. 459-468.

Hirofumi Yamamoto and Eiichiro Sumita. 2007. Bilingual cluster based models for statistical machine translation. Proceedings of the 2007 Joint Conferences on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Prague, Czech Republic.