Sparse lexicalised features and topic adaptation for SMT

Sparse Lexicalised Features and Topic Adaptation for SMT

Eva Hasler, Barry Haddow, Philipp Koehn

University of Edinburgh
Edinburgh, United Kingdom
e.hasler@ed.ac.uk, {pkoehn,bhaddow}@inf.ed.ac.uk

Abstract

We present a new approach to domain adaptation for SMT that enriches standard phrase-based models with lexicalised word and phrase pair features to help the model select appropriate translations for the target domain (TED talks). In addition, we show how source-side sentence-level topics can be incorporated to make the features differentiate between more fine-grained topics within the target domain (topic adaptation). We compare tuning our sparse features on a development set versus on the entire in-domain corpus and introduce a new method of porting them to larger mixed-domain models. Experimental results show that our features improve performance over a MIRA baseline and that in some cases we can get additional improvements with topic features. We evaluate our methods on two language pairs, English-French and German-English, showing promising results.

1. Introduction

In the field of statistical machine translation, domain adaptation is the task of tuning machine translation systems to produce optimal translations for a particular target domain by making the best possible use of the training data, given that we have, usually, a small amount of in-domain data and a larger amount of out-of-domain data. Most approaches to domain adaptation concentrate on either the language model or the translation model and ways to get more appropriate estimates for the respective probability distributions. Other approaches focus on acquiring more in-domain data as opposed to trying to make better use of existing training data.

In this paper, we focus on enhancing standard phrase-based machine translation systems with sparse features in order to bias our systems for the vocabulary and style of the target domain, the TED talks domain. We explore and compare several discriminative training approaches to include sparse features into small in-domain and larger mixed-domain systems. The idea is that sparse features can be added on top of baseline systems that are trained in the usual fashion, overlapping with existing features in the phrase table. This gives us flexibility to explore new feature sets which is particularly useful for training large systems from mixed-domain data. We show experimental results on data provided for the IWSLT 2012 shared task.

2. Training sparse features for domain adaptation

Adding sparse, lexicalised features to existing translation systems trained on in-domain or mixed-domain data is one way to bias translation systems towards translating a particular domain, in our case the TED talks domain. Our features are trained with the MIRA algorithm which is explained briefly in the following subsection. We compare the standard approach, e.g. tuning on a rather small development set, to the less common jackknife approach, details of which are given in subsection 2.4.

2.1. Training features with MIRA

Recently, the Margin Infused Relaxed Algorithm (MIRA) [6] has gained popularity as an alternative training method to Minimum Error Rate Training (MERT) [16], because it can deal with an arbitrary number of features. MIRA is an online large margin algorithm that enforces a margin between different translations of the same sentence. This margin can be tied to a loss function like BLEU [17] or another quality measure. Given that we can provide the learning algorithm with good oracle translations, the model learns to score hypothesis translations with higher BLEU scores better than translations with lower BLEU scores. MIRA updates the feature weights of a translation model by iterating though the training data, decoding one sentence at a time and performing weight updates for pairs of good and bad translation examples. Details about MIRA can be found in [12] or [3], for example.

We use a slightly modified version of the implementation described in [12] that selects hope and fear translations from a 30best list instead of running the decoder with hope and fear objectives. This has the effect that there is no need for dynamically computed sentence-level BLEU scores anymore because real sentence-level BLEU scores can be computed on the 30best list. [5] mentions that certain features, e.g. the language model, are very sensitive to larger weight changes and so we introduce a separate learning rate for core features (translation model, language model, word penalty and so on) in order to reduce fluctuations and keep MIRA training more stable. This learning rate is independent of the C parameter in the objective function solved by MIRA and is set to 0.1 for core features (1.0 for sparse features).

2.2. Feature sets

We experiment with two classes of indicator features, sparse phrase pair features and sparse word pair (or word translation) features. Word pair features capture translations of single source words to single target words, whereas phrase pair features capture translations of several words on the source side into several words on the target side. The class of phrase pair features depends on the decoder segmentation and can also include phrase pairs of length 1 on each side if such a phrase pair was extracted from the training data. Word pair features on the other hand depend on word alignment information and only contain word pairs that were connected by an alignment point in the training data.

Both of these feature classes were also extended with topic information acquired from topic models trained on the source side of the training corpus. The topic information is integrated as a source side trigger for a particular word or phrase pair, given a topic. Details about how these topic models were trained are given in section 2.3. Table 1 shows a pair of source sentence and hypothesis translation taken from a MIRA training run and examples of the features extracted from that sentence pair. The feature values indicate the number of times a feature occurred in a given sentence pair. The features in the first column capture general word or phrase translations while the features in the second column capture translations given a particular topic (here: topic 10). The features without topic information simply indicate whether a particular word or phrase translation should be favoured or avoided by the decoder, depending on whether they receive positive or negative weights during training. The features with topic information are triggered by the topic of the source sentence, that is, for a particular source sentence to be translated, only the features that were seen with the topic of that sentence will fire.

The TED domain is an interesting domain to try out these classes of features, because we can distinguish two different adaptation tasks: (1) adapting to the general vocabulary of TED talks as opposed to the vocabulary of out-of-domain texts (details in the experiments section), and (2) adapting to the vocabulary of subsets of TED talks that can be grouped into more fine-grained topics which we try to capture with topic models.

2.3. Training topic models

The topic models used for building enhanced word pair and phrase pair features are Hidden Topic Markov Models (HTMMs) [11] and were trained with a freely available toolkit. While topic modelling approaches like Latent Dirichlet Allocation assume that each word in a text was generated by a hidden topic and the topics of all words are assumed to be independent, HTMMs model the topics of words in a document as a Markov chain where all words in a sentence are assigned the same topic. This makes intuitively more sense than assigning several different topics within the same sentence and [11] show that HTMMs also yield lower model perplexity than LDA. The former characteristic makes HTMMs particularly suitable for our purpose. We are guaranteed that each word in a source phrase is assigned the same topic and therefore we do not have to figure out how to assign phrase topics given word topics.

Table 1: *Examples of en-fr word pair (wp) and phrase pair (pp) features, with and without topic information. Brackets indicate the phrase segmentation during decoding.*

input (topic 10): "[a language] [is a] [flash of] [the human spirit] [.]"	
hypothesis: "[une langue] [est une] [flash de] [l' esprit humain] [.] "	
reference: "une langue est une étincelle de l' esprit humain ."	
wp_a∼une=2	wp_10_a∼une=2
wp_language∼langue=1	wp_10_language∼langue=1
wp_is∼est=1	wp_10_is∼est=1
wp_flash∼ flash=1	wp_10_flash∼ flash=1
wp_of∼de=1	wp_10_of∼de=1
...	...
pp_a,language∼une,langue=1	pp_10_a,language∼une,langue=1
pp_is,a∼est,une=1	pp_10_is,a∼est,une=1
pp_flash,of∼flash,de=1	pp_10_flash,of∼flash,de=1
...	...

HTMMs compute $P(z_n, \psi_n | d, w_{i=1}, .., w_N)$ for each sentence, where z_n is the topic of sentence n, d is the document and w_i are words in sentence n. ψ_n determines the topic transition between words and can be non-zero only at sentence boundaries. When $\psi_n = 0$, the topic is identical to the previous topic, when $\psi_n = 1$, a new topic is drawn from a distribution θ_d. Once the sentence topic has been selected, all w_i are generated according to a multinomial distribution with topic-specific parameters. In order to assign topics to sentences in our training data, we derive a sentence topic distribution

$$\begin{aligned} P(topic|sentence) &= P(z_n|d, w_{i=1}, .., w_N) \\ &= P(z_n, \psi_n = 0|d, w_{i=1}, .., w_N) \\ &+ P(z_n, \psi_n = 1|d, w_{i=1}, .., w_N) \end{aligned} \quad (1)$$

We noticed that the distributions $P(topic|sentence)$ were quite peaked in most cases and therefore we tried to use a more compact representation. First, we selected the most likely topic according to the topic distribution and treated this as ground truth, ignoring all other possible topics. Alternatively, we selected the two most likely topics along with their probabilities, ignoring the second most likely topics with a probability lower than 30%. The topic probabilies were then used instead of the binary feature values in order to integrate the confidence of the topic model in its assigments. Experimental results were slightly better for the first representation without probabilities and therefore we chose this simpler presentation in all reported experiments.

In order to improve the quality of the topic models, we used stop word lists and lists of salient TED talk terms to clean the in-domain data before training the topic models.

Table 2: *Sample English and German HTMM topics and their interpretation in quotes.*

"cancer"	"ocean"	"body"	"universe"
cancer	water	brain	universe
cells	ice	human	space
body	surface	neurons	Earth
heart	Earth	system	light
blood	Mars	mind	stars
Krebs	Wasser	DNA	Erde
Patienten	Meer	Leben	Universum
Gehirn	Menschen	Licht	Planeten
Zellen	Ozean	Bakterien	Leben
Körper	Tiere	Menschen	Sonne

Table 3: *Topic assignment to training sentences with topic probabilities in brackets.*

"universe" (0.41)	"And physicists came and started using it sometime in the 1980s."
"universe" (0.47)	"And the miners in the early part of the last century worked, literally, in candle-light."
"ocean" (0.71)	"And today, you would see this inside the mine, half a mile underground."
"ocean"/"universe" (0.51/0.49)	"This is one of the largest underground labs in the world."
"universe" (0.99)	"And, among other things, they're looking for dark matter."
"universe" (1.00)	"There is another way to search for dark matter, which is indirectly."
"universe" (1.00)	"If dark matter exists in our universe, in our galaxy, then these particles should be smashing together..."

All TED talks come with a small set of keywords (∼300 in total) describing the content of the talk. The idea was to use the information contained in these keywords to select salient terms that frequently cooccur with the keywords. We first computed tf-idf for all words in each talk, normalised by the number of words in the talk. We then summed up the normalised tf-idf counts for each keyword, i.e. the counts of words in all documents associated with a particular keyword, and selected the top 100 terms for each keyword. This yielded ∼10500 terms for English and ∼11700 terms for German.

In cases where this filtering yielded empty sentences in the in-domain data (sentences with no salient terms), the topic information was replaced by "unk". We ran the topic training for 100 iterations and trained 30 topics over training, development and test sets. We modified the Moses decoder to accept topic information as XML mark-up and annotated all data with sentence-wise topics (and optionally the respective probabilities). Table 2 gives some examples of topics and their 5 most frequent terms for English and German as a source language, as we use topic triggers associated with the source sentence for our sparse features. The topic models represent topics as integers but here we have added labels to indicate the nature of the topics and we selected topics that map across the two languages. In general, the topics do not neccessarily map to equivalent topics in another language.

Table 3 shows a sequence of training sentences and their most likely topic (as well as the second most likely topic if applicable). We can see that for some of the sentences, the model assigns what we have labelled the "universe" topic with high probability while for others it is less certain or makes a transition to the "ocean" topic.

2.4. Jackknife setup

Training sparse features always involves a risk of overfitting on the tuning set, especially with highly lexicalized features that might occur only once in the tuning set. Therefore, training sparse features on the entire training set used to estimate the phrase table is expected to be more reliable. For discriminative training methods this means that the training set needs to be translated in order to infer feature values and compute BLEU scores. However, translating the same data that was used to train the translation system would obviously cause overfitting as well, thus the system needs to be adjusted to prevent this. In order to translate the whole training data without bias, we apply the jackknife method to split up the training data into n=10 folds. We create n subsets of the training data containing n-1 folds and leaving out one fold at a time. These subsets serve as training data for n systems that can be used to translate the respective left-out fold.

To use the jackknife systems for MIRA training, we modified the algorithm to accept n sets of decoder configuration files, input files and reference files. Instead of running n instances of the same translation system in parallel, we run n jackknife systems in parallel and average their weight vectors several times per epoch.

When applying the jackknife method to the TED in-domain data, we noticed a problem with this approach. Usually it would be good practice to create folds in a way that the resulting subsets of training data are as uniform as possible in terms of vocabulary to minimize the performance hit caused by the missing fold. However, the vocabulary of the TED data turned out to be quite repetitive within sentences belonging to the same talk. Thus, splitting up the data uniformly had the effect that each of the n systems had a certain amount of phrasal overlap with its left-out fold. This resulted in a preference for longer phrases, overly long translations on the test set and decreasing performance during MIRA training.

We were able to overcome the overfitting effect of line-wise data splits by splitting the data in a roughly talk-wise fashion instead. That is, the first x =corpus size/n lines were assigned to fold 1, the following x lines to fold 2 and so on. This way the folds were still the same size, but the training

data was much less likely to overlap with the left-out fold. The results on a held-out set during MIRA training (in particular the length penalty and overall length ratio) showed that this helped to prevent overfitting on the left-out fold.

3. Integrating features into mixed-domain models (retuning)

Tuning sparse features on top of large translation models can be time and memory-consuming. Especially the jackknife approach would cause immense overhead to tune with the mixed-domain data because we would need to train *n* different phrase tables that all include most of the in-domain data and all of the out-of-domain data[1]. Therefore, we wanted to investigate whether there is an alternative way of tuning our features on all of the in-domain data while also making use of the out-of-domain data. Tuning with the in-domain models allows for more flexibility in the training setup because the data set is relatively small. Since our goal is to translate documents of the TED talks domain, we assume that tuning sparse features only on the TED domain should provide the model with enough information to select the appropriate vocabulary. Hence we propose to port the tuned features from the in-domain models to the mixed-domain models. The advantage of this method is that features can be tuned on all the in-domain training data (jackknife) or in other ways that are feasible on a smaller in-domain model but might not scale well on a large mixed-domain model.

However, porting tuned feature weights from one model to another is not straightforward because the scaling of the core features is likely to be different. Therefore, to bring the sparse feature weights on the right scale to integrate them into the mixed-domain model, we perform a retuning step with MIRA. We take the sparse features tuned with the jackknife method and combine them into one aggregated meta-feature with a single weight. During decoding, the weight of the meta-feature is applied to all sparse features belonging to the same class (word pair or phrase pair features). In the retuning step, the core weights of the mixed-domain model are tuned together with the meta-feature weight.

An overview of our tuning schemes is given in figure 1. The training step denotes the entire training pipeline yielding the baseline models. Direct tuning refers to tuning with MIRA on a small development set and applies to both kinds of baseline models, while jackknife tuning only applies to in-domain models and retuning only to mixed-domain models.

4. Experiments

We evaluate our training schemes on English-French (en-fr) and German-English (de-en) translation systems trained on the data sets as advised for the IWSLT2012 TED task. As in-domain data we used the TED talks from the WIT[3] website[2] [2]. As out-of-domain data we used the Europarl, News Commentary and MultiUN [8] corpora and for en-fr also the 10^9 corpus taken from the WMT2012 release. An overview of all training data as well as development and test data is given in table 4 (sentence counts).

With this data we trained in-domain and mixed-domain baselines for both language pairs. For the mixed-domain baselines (trained on data from all domains), we used simple concatenations of all parallel training data, but trained separate language models for each domain and linearly interpolated them on the development set. All systems are phrase-based systems trained with the Moses toolkit [13]. Compound splitting and syntactic pre-reordering was applied to all German data. As optimizers we used MERT as implemented in the current version of Moses and a modified version of the MIRA implementation in Moses as described in section 2.1. We provide baseline results for tuning with both MERT and MIRA for comparison, though our model extensions are evaluated with respect to the MIRA baselines. Reported BLEU scores were computed using the mteval-v11b.pl script.

Figure 1: *In-domain (IN) and mixed-domain (IN+OUT) models with three tuning schemes for tuning sparse feature weights: direct tuning, jackknife tuning and retuning.*

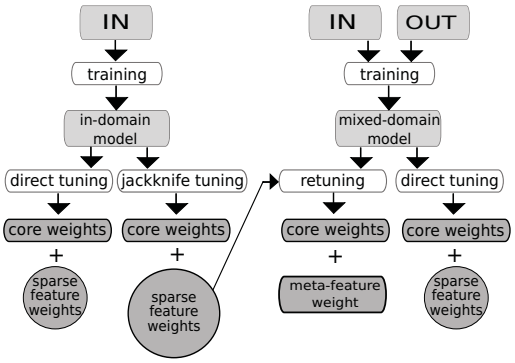


Table 4: *Sentence counts of in-domain (TED talks) and out-of-domain training data used in our systems.*

	en-fr	de-en
TED talks	140K (1029 talks)	130K (976 talks)
Europarl v7	2M	1.9M
News Commentary v7	137K	159K
MultiUN	12.9M	161K
10^9 corpus	22.5M	n/a
total	35.9M	2.3M
TED talks (monoling.)	143K	142K
dev2010	934 (8 talks)	900 (8 talks)
test2010.part1	898 (5 talks)	665 (5 talks)
test2010.part2	766 (6 talks)	900 (6 talks)

[1]Training the mixed-domain system for the en-fr language pair took more than a week.

[2]https://wit3.fbk.eu/mt.php?release=2012-03

All experiments except the jackknife experiments used the TED dev2010 set as development set (dev). The TED test2010 set was split into two parts, test2010.part1 and test2010.part2. For the in-domain experiments, one part was used to select the best weights found during MIRA training and the other part was used for evaluation, respectively. We refer to these sets as test1 and test2 to indicate which of the two parts was used as the test set. We note that test1 and test2 yield quite different BLEU scores for the baseline models. However, table 5 shows that the relative improvements achieved with MIRA are roughly proportional and thus we will report results on just one of the two sets for experiments on the mixed-domain baselines.

All MIRA experiments were initialized with the tuned weights of the MERT baselines. MIRA experiments on the dev set were run for 20 epochs, retuning experiments for 10 epochs and jackknife experiments on the entire training set for 2 epochs.

4.1. Results

We are evaluating the impact of our sparse features on the in-domain and mixed-domain systems. Tables 5 and 6 show the results on the in-domain system with BLEU scores reported on both parts of the test2010 set, using the respective other part as devtest set. Improvements over the MIRA baseline are marked in bold print and the relative changes are indicated in brackets. First we note that MIRA training improves the MERT baseline performance for the en-fr system by 0.8 BLEU on both test sets, but decreases performance for the de-en system by 0.3 BLEU. We believe that this divergence has to do with the changes in length ratio after MIRA training, as shown in table 7. For en-fr, translations get longer during MIRA training while for de-en they get shorter, incurring an increased brevity penalty according to the BLEU score.

Since MIRA has quite a different impact on the translation performance with the core features (translation model, reordering model, language model, word penalty, phrase penalty), we focus on the impact of sparse features with respect to the MIRA baselines. For en-fr, we observe that all sparse feature setups beat the MERT baseline and most of them beat the MIRA baseline. For the MIRA experiments on the dev set we notice that phrase pair features seem to perform better than word pair features on both test sets and sparse features with topic triggers seem to do better than sparse features without topic information. The results of the MIRA experiments using the jackknife method are in almost all cases better than the results trained on the small dev set. We get an increase of up to 1.3/0.2 BLEU (en-fr/de-en) over the MERT baseline and up to 0.5/0.7 BLEU (en-fr/de-en) over the MIRA baselines. This shows that the jackknife method is better suited to train sparse features than training on a small dev set. We still observe slightly better results for phrase pair features than for word pair features with the en-fr models, even though this observation is less conclusive than on the dev data.

Table 5: *In-domain baselines (IN) and results for sparse feature training on en-fr in-domain model, training on a development set (dev) and on all training data (jackknife).*

en-fr	BLEU(test1)	BLEU(test2)
MERT(dev) IN	*28.6*	*30.9*
MIRA(dev) IN	29.4	31.7
MIRA(dev)		
+ wp	29.2 (-0.2)	31.6 (-0.1)
+ wp + topics	**29.5** (+0.1)	**31.8** (+0.1)
+ pp	**29.6** (+0.2)	31.7 (+0.0)
+ pp + topics	**29.6** (+0.2)	**31.9** (+0.2)
MIRA(jackknife)		
+ wp	**29.7** (+0.3)	**32.2** (+0.5)
+ wp + topics	**29.5** (+0.1)	**32.1** (+0.4)
+ pp	**29.9** (+0.5)	**32.2** (+0.5)
+ pp + topics	**29.6** (+0.2)	**32.0** (+0.4)

Table 6: *In-domain baselines (IN) and results for sparse feature training on de-en in-domain model, training on a development set (dev) and on all training data (jackknife).*

de-en	BLEU(test1)	BLEU(test2)
MERT(dev) IN	*26.6*	*29.9*
MIRA(dev) IN	26.3	29.6
MIRA(dev)		
+ wp	**26.7** (+0.4)	**29.8** (+0.2)
+ wp + topics	**26.6** (+0.3)	**29.7** (+0.1)
+ pp	**26.5** (+0.2)	**29.7** (+0.1)
+ pp + topics	**26.4** (+0.1)	**29.8** (+0.2)
MIRA(jackknife)		
+ wp	**27.0** (+0.7)	**30.1** (+0.5)
+ wp + topics	**26.4** (+0.1)	**29.7** (+0.1)
+ pp	**26.8** (+0.5)	**30.0** (+0.4)
+ pp + topics	**26.4** (+0.1)	**29.8** (+0.2)

Tables 8 and 9 show results on the mixed-domain models, where we observe a similar divergence in performance between the MERT and MIRA baselines as on the in-domain models: a plus of 1.1 BLEU for en-fr and a minus of 0.4 BLEU for de-en. The first block of results refers to MIRA training on the dev2010 set as for the in-domain models (direct tuning), while the second block results from the retuning setup described in section 3 (retuning). The direct approach gains up to 0.5 BLEU for en-fr and up to 0.1 BLEU for de-en over the MIRA baselines, retuning with MIRA and jackknife features gains up to 0.5 BLEU for en-fr and up to 0.4 BLEU for de-en over the MIRA baselines. This is another indication that sparse features trained with the jackknife method can leverage information from the in-domain training data to help the model select appropriate words and phrases for the target domain. In some cases we can observe that topic

Table 7: *Changes to the length ratio (hypotheses/reference, in brackets) between MERT and MIRA tuning, indicated by (+) and (-).*

		BLEU(test1)	BLEU(test2)
en-fr	MERT(dev) IN	28.6 (0.969)	30.9 (0.963)
	MIRA(dev) IN	29.4 (0.987) (+)	31.7 (0.982) (+)
de-en	MERT(dev) IN	26.6 (0.987)	29.9 (1.001)
	MIRA(dev) IN	26.3 (0.955) (-)	29.6 (0.969) (-)

Table 8: *Mixed-domain baselines (IN+OUT) and results for sparse feature training on en-fr mixed-domain model: direct sparse feature tuning and retuning with MIRA using jackknife-trained features.*

en-fr	BLEU(test1)
MERT(dev) IN+OUT	*30.0*
MIRA(dev) IN+OUT	31.1
MIRA(dev), direct tuning	
+ wp	**31.6** (+0.5)
+ wp + topics	**31.4** (+0.3)
+ pp	**31.4** (+0.3)
+ pp + topics	**31.5** (+0.4)
MIRA(dev), retuning	
+ wp	**31.6** (+0.5)
+ wp + topics	31.1 (+0.0)
+ pp	**31.5** (+0.4)
+ pp + topics	**31.3** (+0.2)

features improve over simple features, even though they perform weaker in more of the cases. We suspect that sparsity issues need to be addressed to benefit more from these features. In general, the results show that features trained only on in-domain models can help to improve performance of much larger mixed-domain models. While for the in-domain models the results on both language pairs are similar w.r.t. the MIRA baselines, the results on mixed-domain models are clearly better for en-fr which can be considered an easier language pair for translation than de-en.

The feature sets ranged in size between around 5K-15K when training on a dev set and 60K-600K when training on all training data, depending on the particular feature type.

4.2. Topic features

For the en-fr in-domain systems trained on dev data, we see an improvement of topic features over simple sparse features. That these effects are not stronger might be due to the quite diverging distributions of topics across dev, devtest and test sets (see figure 2[3]). For example, the "universe" topic (topic 29) appears quite frequently in the training and dev data, but only twice in test2 and never in test1. For future experiments with sentence-level topic features it should be ensured that

[3]Training data counts were between 2252 and 7170 sentences per topic.

Table 9: *Mixed-domain baselines (IN+OUT) and results for sparse feature training on de-en mixed-domain model: direct sparse feature tuning and retuning with MIRA using jackknife-trained features.*

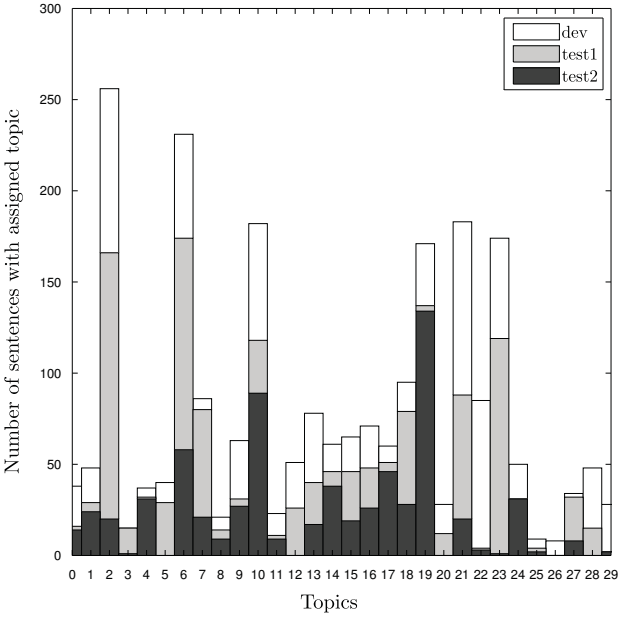
de-en	BLEU(test1)
MERT(dev) IN+OUT	*27.2*
MIRA(dev) IN+OUT	26.8
MIRA(dev), direct tuning	
+ wp	**26.9** (+0.1)
+ wp + topics	**26.9** (+0.1)
+ pp	**26.9** (+0.1)
+ pp + topics	26.7 (-0.1)
MIRA(dev), retuning	
+ wp	**27.1** (+0.3)
+ wp + topics	**27.2** (+0.4)
+ pp	**27.0** (+0.2)
+ pp + topics	**27.0** (+0.2)

topics are distributed more evenly across development sets.

Lexicalised features with topic triggers are even sparser than simple lexicalised features and therefore we would expect that they benefit particularly from jackknife training. However, our current results show the opposite tendency in that topic features seem to do worse than simple features under the jackknife setup. Table 10 gives an example of word pair features trained with the jackknife method, with and without topic information. It shows the features with the largest positive/negative weights (those with the highest discriminative power learned by the model) for translating the English source word "matter". Both models have learned that "matière" is the most appropriate French translation for the English word "matter". Both models penalize some translations of the other word sense like the French word "important". However, the model without topic information considers "importe" an almost equally likely translation, while the model with topic information penalizes all translations that do not preserve the physical word sense (as in "dark matter"). As mentioned above, the "universe" topic did not appear at all in test1, so the impact of features related to this topic has not been measured in the evaluation.

Table 11 shows jackknife-trained features for the source word "language". While with simple word pair features the most likely translation is "langage" (mode of speaking), the topic features express translation preferences according to the source topic. For example, given the "science" topic, the most likely translation is "langage", but given the "school" topic, the most likely translation is "langue". However, in table 1 we see that the input sentence is labelled with topic 10 ("science") but "language" is translated to "langue" in the reference translation. Thus, given the topic labelling the exptected translation with topic features would not match the reference translation, which is something that should be taken into account.

Figure 2: *Distribution of topics in dev, test1, test2.*



5. Related work

The domain adaptation literature can be broadly grouped into approaches adapting the language model and approaches adapting the translation model. Among the latter there has been work on mixture modeling of domain-specific phrase tables [9] and discriminative instance weighting [14] [10]. In similar spirit, [1] introduced a corpus-filtering technique that computes a bilingual cross-entropy difference to determine how similar a sentence pair is to an in-domain corpus and how dissimilar from a general-domain corpus. There has also been previous work on translation model adaptation using topics models. [19] employ HTMMs to train source-side topic models from monolingual in-domain data and the source side of parallel out-of-domain data. Phrase pairs are conditioned on in-domain topics via a mapping from in-domain to out-of-domain topics. Our approach is different in that we use parallel in-domain data and therefore do not need a mapping step. [7] extend previous work by [4] on lexical weighting conditioned on data provenance. They enhance lexical weighting features with topic model information to train separate word translation tables for every domain which can then be used to bias phrase selection based on source topics.

MIRA has been proposed for tuning machine translation systems with large features sets, for example by [20] and [3]. Recent work that compares tuning on a small development set versus tuning on the entire training data has been presented in [18]. The idea of using source triggers to condition word translation is somewhat related to the trigger-based lexicon models of [15], though they use context words as additional triggers and train their features with the EM algorithm.

Table 10: *Examples of en-fr jackknife-trained word pair features, with and without topic information (topic 29: "universe").*

sparse feature	feature weight
wp_matter∼matière	0.00170
wp_matter∼importe	0.00107
wp_matter∼important	-0.00037
wp_matter∼comptent	-0.00188
wp_29_matter∼matière	0.00431
wp_29_matter∼importent	-1.42913e-05
wp_29_matter∼importe	-0.00134
wp_29_matter∼important	-0.00172

Table 11: *Examples of en-fr jackknife-trained word pair features, with and without topic information (topic 10: "science", topic 27: "school").*

sparse feature	feature weight
wt_language∼langage	0.00444
wt_language∼langue	-0.00434
wt_10_language∼langage	0.01088
wt_10_language∼langue	-0.01071
wt_27_language∼langue	0.00792
wt_27_language∼langage	-0.00742

6. Conclusion

We presented a novel way of training lexicalised features for a domain adaptation setting by adding sparse word pair and phrase pair features to in-domain and mixed-domain models. In addition, we suggested a method of using topic information derived from HTMMs trained on the source language to condition the translation of words or phrases on the sentence topic. This was shown to yield improvements over simple sparse features on English-French in-domain models. We experimented with the jackknife method to use the entire in-domain data for feature training and showed BLEU score improvements for both language pairs. Finally, we introduced a retuning method for mixed-domain models that allows us to adapt features trained on the entire in-domain data to the mixed-domain models.

In the future, we would like to test our methods on hierarchical phrase-based or syntactic models. Other work in this field suggests that discriminative training yields larger gains with those types of models than with purely phrase-based models, so this would be an interesting comparison. We would also like to address the evaluation of topic features, which we believe requires a more controlled setting. Induced topics should be distributed more evenly across data sets and the quality of sentence topic labels should be taken into account.

7. References

[1] Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo In-Domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

[2] Cettolo, M., Girardi, C., and Federico, M. (2012). Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of EAMT*, pages 261–268, Trento, Italy.

[3] Chiang, D., Knight, K., and Wang, W. (2009). 11,001 new features for statistical machine translation. In *Proceedings of HLT: The 2009 Annual Conference of the NACL*, Stroudsburg, PA, USA. Association for Computational Linguistics.

[4] Chiang, D., DeNeefe, S., and Pust, M. (2011). Two easy improvements to lexical weighting. In *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies*, pages 455–460, Stroudsburg, PA, USA. Association for Computational Linguistics.

[5] Chiang, D. (2012). Hope and fear for discriminative training of statistical translation models. In *J. Machine Learning Research 13*, pages 1159–1187.

[6] Crammer, K. and Singer, Y. (2003). Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3(4-5):951–991.

[7] Eidelman, V., Boyd-Graber, J., and Resnik, P. (2012). Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju Island, Korea. Association for Computational Linguistics.

[8] Eisele, A. and Chen, Y. (2010). Multiun: A multilingual corpus from united nation documents. In *LREC'10*.

[9] Foster, G. and Kuhn, R. (2007). Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Stroudsburg, PA, USA. Association for Computational Linguistics.

[10] Foster, G., Goutte, C., and Kuhn, R. (2010). Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 451–459, Stroudsburg, PA, USA. Association for Computational Linguistics.

[11] Gruber, A., Rosen-Zvi, M., and Weiss, Y. (2007). Hidden Topic Markov Models. In *Journal of Machine Learnin Research, pp. 163-170*.

[12] Hasler, E., Haddow, B., and Koehn, P. (2011). Margin Infused Relaxed Algorithm for Moses. In *The Prague Bulletin of Mathematical Linguistics No. 96, 2011, pp. 69-78*, Prague.

[13] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *ACL 2007: proceedings of demo and poster sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

[14] Matsoukas, S., Rosti, A.-V. I., and Zhang, B. (2009). Discriminative corpus weight estimation for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, pages 708–717, Stroudsburg, PA, USA. Association for Computational Linguistics.

[15] Mauser, A., Hasan, S., and Ney, H. (2009). Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Conference on Empirical Methods in Natural Language Processing*, pages 210–217, Singapore.

[16] Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *ACL-2003: 41st Annual meeting of the ACL*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

[17] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *ACL 02: Proceedings of the 40th Annual Meeting on ACL*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

[18] Simianer, P., Riezler, S., and Dyer, C. (2012). Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of the 50th Annual Meeting of the ACL*. Association for Computational Linguistics.

[19] Su, J., Wu, H., Wang, H., Chen, Y., Shi, X., Dong, H., and Liu, Q. (2012). Translation model adaptation for statistical machine translation with monolingual topic information. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju Island, Korea. Association for Computational Linguistics.

[20] Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H. (2007). Online large-margin training for statistical machine translation. In *Proceedings of EMNLP-CoNLL*, pages 764–773, Prague. Association for Computational Linguistics.