Detailed Analysis of different Strategies for Phrase Table Adaptation in SMT

Jan Niehues and Alex Waibel
Institute for Anthropomatics
Karlsruhe Institute of Technology, Germany
firstname.secondname@kit.edu

Abstract

This paper gives a detailed analysis of different approaches to adapt a statistical machine translation system towards a target domain using small amounts of parallel in-domain data. Therefore, we investigate the differences between the approaches addressing adaptation on the two main steps of building a translation model: The candidate selection and the phrase scoring. For the latter step we characterized the differences by four key aspects. We performed experiments on two different tasks of speech translation and analyzed the influence of the different aspects on the overall translation quality. On both tasks we could show significant improvements by using the presented adaptation techniques.

1 Introduction

Statistical machine translation (SMT) is currently the most promising approach to machine translation of large vocabulary tasks. The approach was first presented in (Brown et al., 1993) and has been used in many translation systems since then.

One drawback of this approach is that large amounts of training data are needed. Furthermore, the performance of the SMT system improves if this data is matching in topic and genre. Since this is not possible for many real-world scenarios, one approach to overcome this problem is to use all available data to train a general system and to adapt the system to the task at hand using in-domain training data.

Since parallel in-domain data is available for our scenario, we will focus on the adaptation of the translation model. When comparing the existing approaches to phrase table adaptation, the adaptation of the translation model can take place in one of two main steps: the candidate selection and the scoring of phrase pairs. We found two different approaches for candidate selection and analyzed their influence on the translation quality. Secondly, we analyzed the different techniques to adapt the phrase pair scoring and characterized them by using four key aspects.

By analyzing the steps separately, we are able to combine the techniques from the approaches in a new way and improve the translation quality even further.

The different strategies were evaluated on two different tasks of translating German to English: the translation of TED lectures[1] and computer science university lectures.

2 Related work

In recent years different methods were proposed to adapt translation systems to a specific domain. Some adapted only the language model inspired by similar approaches in speech recognition (Bulyko et al., 2007). The main advantage is that only monolingual in-domain data is needed.

In cases where also parallel in-domain data is available, the translation model can be adapted as well. Koehn and Schroeder (2007) proposed to use a log-linear combination of the in-domain and out-of-domain phrase table. We will refer to this approach as "*Log-Linear*".

In (Niehues et al., 2010), the translation model is

[1] http://www.ted.com

adapted by adding the in-domain relative frequencies to the general phrase table. If the phrase pair does not occur in the in-domain phrase table, they use a backoff value. We will refer to this method as "*Backoff*".

In (Niehues and Waibel, 2010) a factored translation model was used to adapt the translation model. They used the general scores together with the in-domain or out-of-domain relative frequencies. In addition, a word factor represents the part of the corpus where the phrase is extracted from. Then a Domain Sequence Model counts the number of in-domain phrase pairs or words ("*Factored*").

Another approach using the "*Fill-Up*" technique was described in (Bisazza et al., 2011). They used the in-domain and out-of-domain scores and an indicator feature. The out-of-domain scores were only used if no in-domain probabilities were available.

An approach based on mixture models was presented by Foster and Kuhn (2007) and Banerjee et al. (2011). They used linear and log-linear, language model and translation model adaptation. Furthermore, they optimized the weights for the different domains on a development set and the weights are set according to text distance measures.

Matsoukas et al. (2009) also adapt the system by changing the weights of the phrase pairs. In their approach this is done by assigning discriminative weights for all sentences in the parallel corpus.

3 Translation model

When parallel in-domain data is available, we want to adapt the translation model to be able to encode the domain specific knowledge without losing the information learned from the much bigger parallel corpus. If we compare the different approaches of translation model adaptation, we see that two main aspects of the model can be adapted to better match the specific domain. The first aspect is the candidate selection, where we determine for every possible source phrase, which translations to consider during translation.

The second aspect we can adapt is the selection of the phrase pair scores. There are different possibilities to encode the additional information gained only from the in-domain data.

In our experiments we used a phrase-based machine translation system. The phrase pairs were extracted from an automatically generated word alignment.

Before translation, a set of candidate translations $T(\bar{f}_i)$ is selected for every source phrase $\bar{f}_i$. For several source phrases there are many different translations, which we cannot all consider during decoding due to runtime limitations. Furthermore, most of them have low probabilities and will not lead to good translations of the sentence. Therefore, we limit the number of translations for every source phrase to a maximum of n by a combination of histogram and beam pruning. We used at most 10 translations for every source phrase. We rank the phrase pairs in order to be able to select the top translations by scoring them using some initial weights that were determined heuristically. These weights are independent from the weights generated by MERT.

In the baseline phrase table we use four different scores $\Phi_s((\bar{f}_i, \bar{e}_i))$ for every phrase pair $(\bar{f}_i, \bar{e}_i)$. The relative frequencies in both directions and the lexical probabilities in both directions. We use modified Kneser-Ney smoothing for the relative frequencies as described in (Foster et al., 2006). This leads to the following definition of the translation model when translating the source sentence $f = \bar{f}_1^I$ into the target sentence $e = \bar{e}_1^I$ using the phrase pairs $((\bar{f}_1, \bar{e}_1),(\bar{f}_2, \bar{e}_2),..,(\bar{f}_I, \bar{e}_I))$

$$log(p(\bar{e}_1^I|\bar{f}_1^I)) = \sum_{i=1}^{I} log(p(\bar{e}_i|\bar{f}_i)) \quad (1)$$

$$= \sum_{i=1}^{I}\sum_{s=1}^{S} \lambda_s log(\Phi_s((\bar{f}_i, \bar{e}_i))) - log(Z) \quad (2)$$

The weights used for the four scores during the actual decoding are optimized using MER training on the development data.

In our scenario there are three different phrase tables. One trained only on the in-domain data providing the candidate translations $T_{IN}(\bar{f}_i)$ for a given source phrase $\bar{f}_i$, one trained on the out-of-domain data ($T_{OUT}(\bar{f}_i)$) and one trained on all data ($T_{ALL}(\bar{f}_i)$). Consequently, for a given translation task there are 3 different sets of phrase pairs, which

were determined by candidate selection as described above. Each of these sets contain at most n translations for a given source phrase due to pruning as mentioned before. In the next section we compare different approaches on how to combine these three sets into a new one. Afterwards, we will describe the different features used as phrase table scores.

4 Candidate Selection

As described in the previous section, the first step in the translation process is to determine the phrase pairs which the decoder can then use to build the translation.

The first approach here is to use the phrase pairs which are selected from the phrase table trained on all data $T(\bar{f}_i) = T_{ALL}(\bar{f}_i)$. In this case we would not adapt the component of the candidate selection at all. This approach was used by the backoff and factored approaches and will be referred to as *NoAdapt*.

A second approach is to use the union of the candidates selected from the in-domain and the out-of-domain phrase table $T(\bar{f}_i) = T_{IN}(\bar{f}_i) \cup T_{OUT}(\bar{f}_i)$. This was done in the log-linear and fill-up approaches. We will refer to this method as *UnionOut*. Instead of the out-of-domain phrase table, we could alternatively use the phrase table trained on all data and combine its candidate phrase pairs with the in-domain phrase pair selection $T(\bar{f}_i) = T_{IN}(\bar{f}_i) \cup T_{ALL}(\bar{f}_i)$. We will refer to this method as *UnionAll*.

A third approach is to mainly rely on the phrase pairs of the in-domain phrase table. Only if there are no or not enough candidate translations for one source phrase, we will fill up the candidate list by the ones suggested by the out-of-domain phrase table or the general phrase table, respectively. This leads to the definition of the translation candidates set as: $T(\bar{f}_i) = T_{IN}(\bar{f}_i) \cup T^k_{ALL}(\bar{f}_i)$, where $T^k_{ALL}(\bar{f}_i)$ are the top k translation of $T_{ALL}(\bar{f}_i)$ and $k = n - |T_{IN}(\bar{f}_i)|$, where n is the maximum number of translation candidates and therefore k is always larger than or equal to 0. In our case n is 10. In contrast to the *Union* approaches, we will still have at most 10 translations for every source phrase. Analog to the *UnionAll* and *UnionOut* approach, we will refer to this approach as *PaddingAll* and *PaddingOut*.

In a last approach, we allow only to fall back the candidates by the out-of-domain, if there are no translations at all for the source phrase in the in-domain phrase table. In this case we will consider all out-of-domain candidates for this source phrase.

$$T(\bar{f}_i) = \begin{cases} T_{IN}(\bar{f}_i): & T_{IN}(\bar{f}_i) \neq \emptyset \\ T_{OUT}(\bar{f}_i): & else \end{cases}$$

We will refer to this approach as *SourcePadding*.

5 Selecting Scores for Phrase Table

The other step in the translation model that can be adapted is the scores of the phrase pairs. Here the approaches differ in four key aspects. For the adapted translation model the definition of the translation probability needs to be changed slightly in the following way:

$$\begin{aligned} log(p(\bar{e_i}|\bar{f}_i)) &= \sum_{s=1}^{S} \vec{\lambda}_s log(\overrightarrow{\Phi_f(\bar{e_i}|\bar{f}_i)}) \\ &+ \sum_{f=1}^{S'} \lambda_f log(\Phi_f(\bar{e_i}|\bar{f}_i)) \\ &- log(Z_{\bar{f}_i}) \end{aligned}$$

Instead of having only one value for the four different phrase table features, there may be now several ones from the different phrase tables as indicated by the vectors. In this case, the log function should be applied separately for each component. Furthermore, in some approaches their might be some additional features S'.

The first key aspect is the usage of the scores trained on all data. Although these scores are not adapted to the target domain, they are often more reliable, since they are calculated on a bigger amount data. Therefore, they might be useful for smoothing the adapted features. We can use them as a log-linear smoothing by defining $\Phi_s =< \Phi_s^{All}, \Phi_s^{Adapted} >$ or we can ignore them by using only the adapted features ($\Phi_s =< \Phi_s^{Adapted} >$) and therefore do not need to train an additional phrase table on the whole data. While the backoff and factored approaches extend the features of the general phrase table by the adapted ones, the general ones are not used at all in the log-linear and fill-up approaches.

Secondly, when adding the in-domain scores to the ones calculated on all data, it might not be necessary to use all adapted scores, but just adding one or two of the four scores might be sufficient. Therefore, we will analyze for which scores we need an adapted version and for which of them we can just use the score from the general phrase table. In this case, for some of the scores, the features will then be defined as $\Phi_s =< \Phi_s^{All} >$ and not $\Phi_s =< \Phi_s^{All}, \Phi_s^{Adapted} >$.

Thirdly, the out-of-domain scores and the in-domain scores cannot be calculated for all phrase pairs, but only for the ones that occur in the corresponding corpus. Therefore, the approaches suggested different ways to handle unknown probabilities.

Log-Linear As it was done in the log-linear combination, we can use either the in-domain or the out-domain scores and then use different scaling factors for each of them. That means both in-domain and out-of-domain phrase pairs have 8 scores. For in-domain phrase pairs the four out-of-domain phrase table features are set to one and for the out-of-domain phrase pairs the in-domain features are set to one. For phrase pairs from the in-domain corpus this leads to the definition $\Phi_s^{Adapted} =< \Phi_s^{IN}, 1 >$ and for the out-of-domain phrase pairs we get the definition: $\Phi_s^{Adapted} =< 1, \Phi_s^{OUT} >$

Backoff In the backoff method the in-domain scores are used for in-domain phrase pairs. For phrase pairs that only occur in the out-of-domain phrase table each score is set to the worst value that occurs in the in-domain phrase table for this score. In this case, for all phrase pairs from the in-domain corpus, we get the definition: $\Phi_s^{Adapted} = \Phi_s^{IN}$ and for all the other phrase pairs:

$$\Phi_s^{Adapted} = min_{(\bar{e_i}|\bar{f_i})}\Phi_s^{IN}(\bar{e_i}|\bar{f_i}))$$

Indicator An indicator feature signals whether the phrase stems from the in-domain or from the out-of-domain phrase table. As the first four scores we use the probabilities from the in-domain and out-of-domain phrase table, respectively, and the last one being the indicator feature. This additional feature will have the value 1 for all in-domain phrase pairs and $exp(1)$ for all out-of-domain phrase pairs.

The fourth and last aspect is the treatment of phrase pairs which can be assigned both in-domain and out-of-domain scores, because they occur both in the in-domain and in the out-of-domain corpus. In this case, the backoff and fill-up approach suggest to use only the in-domain scores, while the other two approaches add the phrase pair to the phrase table twice, once with in-domain and once with out-of-domain scores.

An overview over the different aspects of the four approaches to phrase table adapation as mentioned in the related work is given in Table 1.

6 Results

After analyzing the different approaches we will now evaluate their effects on translation quality. We perform experiments on two different German-to-English speech translation tasks. First, we describe the SMT system and then we run some baseline experiments to demonstrate the characteristics of the data. Afterwards, we will evaluate the influence of the candidate selection and aspects of the phrase scoring.

We performed significance tests following (Zhang and Vogel, 2004). All results that are significantly better than the baseline system at a level of 0.05 are marked by a star(*).

6.1 System Description

The translation system was trained on the European Parliament corpus, News Commentary corpus and the BTEC corpus. As parallel in-domain data, the TED talks were used in addition. The data was preprocessed and compound splitting was applied. Afterwards the discriminative word alignment approach as described in (Niehues and Vogel, 2008) was applied to generate the alignments between source and target words. The phrase table was built using the scripts from the Moses package (Koehn et al., 2007). The language model was trained on the target side of the parallel data using the SRILM toolkit (Stolcke, 2002). In addition we used a bilingual language model as described in (Niehues et al., 2011).

Table 1: Different phrase table adaptation approaches

Approach	Candidate Selection	Score Selection				Number of scores
		General	Adapted	Unknown Prob.	Unique	
Log-Lin	UnionOut		all	Log-Lin		8
Backoff	NoAdapt	X	2	Backoff	X	6
Factored	NoAdapt	X	2	Indicator		7
Fill-Up	UnionOut		all	Indicator	X	5

Reordering was performed as a preprocessing step using POS information generated by the TreeTagger (Schmid, 1994). We used the reordering approach described in (Rottmann and Vogel, 2007) and the extensions presented in (Niehues and Kolss, 2009) to cover long-range reorderings, which are typical when translating between German and English.

An in-house phrase-based decoder was used to generate the translation hypotheses and the optimization was performed using MER training.

We used TED talks as development and test data. In addition, we tested the systems on transcribed university lectures from the computer science(CS) department. Each test set contains at least 30K words.

6.2 Baseline

In a first series of experiments we show the influence of the in-domain and out-of-domain data. We tested the systems on both tasks using three different configurations. The first condition uses no language model adaptation, the other two conditions use a language model adapted by log-linear or linear combination of the in-domain and out-of-domain data.

As shown in Table 2 using only the small, in-domain parallel data leads to quite good quality translations despite of the size of the parallel data. This is especially true for the TED task. In contrast, when using the much bigger out-of-domain data, we get a worse performance unless language model adaptation is used. If both corpora are combined, we could improve on the CS task. On the TED task, we can only improve by using some kind of language model adaptation.

Table 3: Number of Phrase pairs

	TED		CS	
In	140K	40%	109K	31%
Out	335K	96%	338K	97%
All	348K	100%	347K	100%
UnionOut	425K	122%	408K	118%
UnionAll	413K	118%	399K	115%
PaddingOut	366K	105 %	363K	105%
PaddingAll	364K	104%	361K	104%
SourcePadding	250K	72%	258K	74%

6.3 Candidate Selection

In the next series of experiments, we analyzed the influence of the candidate selection.

Before considering the translation quality itself, we analyzed the size of the phrase tables generated by the different methods.

In Table 3 the number of phrase pairs selected by the different methods for both test sets are presented. The in-domain phrase table contains 30 to 40% of the phrase pairs that are in the general phrase table and the out-of-domain around 95%.

If we take the Union of in-domain and out-of-domain (UnionOut) or in-domain and all data phrase table (UnionAll) the size increases by around 20%. By using the Padding method, the phrase table increases only by around 5% compared to the phrase table trained on all data. If we only use out-of-domain phrase pairs for those source phrases, which did not occur in the in-domain corpus, the phrase table size is reduced by around 30% compared to the aforementioned phrase table.

After looking at the phrase table sizes, we measured the quality of the translations generated using these phrase tables. In this case, we used the

Table 2: Baseline results (case-insensitive BLEU)

System	No LM Adaptation			Log-Lin. LM Adaptation			Linear LM Adaptation		
	Dev	Test		Dev	Test		Dev	Test	
		TED	CS		TED	CS		TED	CS
Only In	**27.11**	**26.30**	25.11	27.02	26.17	23.82	27.05	26.13	24.30
Only Out	25.30	24.65	24.86	26.42	26.10	24.97	26.38	26.33	25.32
All data	26.32	25.39	**25.15**	**27.45**	**26.56**	**25.43**	**27.50**	**26.70**	**25.43**

scores as described in the Backoff method. Since all phrase tables use the same features, we performed first experiments without running separate optimizations for the different methods. The results for the TED translation task and the translations of the CS lectures are shown in Table 4.

For the TED task the method using the Union of the two phrase tables and the Padding method are performing best. This leads to improvements between 0.05 and 0.3 BLEU points compared to the system using the phrase pairs selected from the general phrase table. They are also the only methods that are significantly better than the baseline system on the TED task using log-linear LM Adaptation. The SourcePadding method leads to slightly worse results than the best two methods. The differences are bigger, if the language model is also adapted towards the target domain.

To see the performance also for the cases where the in-domain and test condition do not match perfectly, we also test the system on a set of computer science lectures.

Again, the Union and Padding method perform best. But in this case, the Union methods outperform the Padding technique. Furthermore, the SourcePadding technique performs worse than using the phrase pairs extracted from the general phrase table.

For all methods it does not matter whether we combine the in-domain phrase table with the out-of-domain phrase table or a phrase table trained on all data.

We also performed experiments, where we optimized the weights for every phrase table separately. The results are summarized in Table 5.

Performing individual optimizations for every configuration introduces additional random noise, so that no clear picture can be seen. For the TED task, the difference between Union, Padding and SourcePadding is mostly below 0.2 BLEU points. For the CS task, the situation is a little different. Here again, SourcePadding the is worse than the other two and Union produces in most cases the best translation.

To summarize the results of these experiments, it seems to be important to keep all phrase pairs from the in-domain phrase table. Not to adapt the candidate selection performs in many experiments worse than the Union or the Padding approach. Furthermore, especially in the case where the in-domain and test data do not perfectly match, i.e. with CS lectures, it seems also to be important to keep all phrase pairs from the bigger phrase table. Therefore, we will use the UnionAll method for the following experiments,

6.4 Selecting Scores for the Phrase Table

After analyzing the influence of the candidate selection, the remaining experiments concentrate on the different features that can be used as scores for the phrase table entries. In the first group of experiments we analyzed which features need to be adapted.

In all system we used the four scores from the phrase table trained on all data. In addition, we used some of the in-domain scores. We used the in-domain features as described in the backoff method. The results for the TED and CS task are shown in Table 6.

For the TED task, more improvements can be gained by using the relative frequencies for adaptation than by using the lexical probabilities. Furthermore, adapting both relative frequencies is mostly better than adapting only one.

If we adapt all four features, no additional gains can be reached over only adapting the relative frequencies. But the systems perform similar. Over all, an additional 0.6 to 1.3 BLEU points can be gained by using the in-domain phrase scores.

Table 4: Candidate Selection (No Optimization) (BLEU)

System	No LM Adaptation Test		Log-Lin. LM Adaptation Test		Linear LM Adaptation Test	
	Ted	CS	Ted	CS	Ted	CS
NoCSAdapt	26.77	25.97	27.16	26.64	27.33	26.39
UnionOut	26.78	**26.05**	27.49*	**26.82**	**27.41**	**26.44**
UnionAll	26.78	26.04	27.49*	26.81	27.40	26.42
PaddingOut	**26.80**	25.95	**27.50***	26.74	27.40	26.33
PaddingAll	**26.80**	25.94	**27.50***	26.73	27.40	26.32
SourcePadding	26.76	25.81	27.35	26.57	27.21	26.26

Table 5: Candidate Selection (BLEU)

System	No LM Adaptation			Log-Lin. LM Adaptation			Linear LM Adaptation		
	Dev	Test		Dev	Test		Dev	Test	
		Ted	CS		Ted	CS		Ted	CS
NoCSAdapt	28.03	26.77	25.97	28.40	27.16	26.64	28.30	27.33	26.39
UnionOut	28.14	26.81	25.58	28.43	27.23	26.74*	28.61	27.21	**26.58**
UnionAll	**28.34**	**27.03**	**26.44***	**28.69**	27.38	**26.79**	28.43	27.38	26.26
PaddingOut	28.20	26.96	25.77	28.68	**27.46**	26.73	**28.66**	27.59*	26.35
PaddingAll	28.19	26.74	25.69	28.53	27.37	26.20	28.61	27.48	25.93
SourcePadding	28.13	26.80	25.81	28.49	27.43	25.51	28.45	**27.62**	25.97

Table 6: Feature Selection (BLEU)

System	No LM Adaptation			Log-Lin. LM Adaptation			Linear LM Adaptation		
	Dev	Test		Dev	Test		Dev	Test	
		TED	CS		TED	CS		TED	CS
No	26.51	25.59	25.41	27.56	26.72	26.36	27.65	26.95	25.61
rel. Freq 1	28.04	26.73*	25.83*	28.50	27.18*	26.70*	28.37	27.40*	25.99*
rel. Freq 2	28.28	26.85*	25.99*	28.44	27.24*	25.18	**28.56**	**27.59***	25.86
rel. Freq	**28.34**	**27.03***	**26.44***	**28.69**	27.38*	**26.79***	28.43	27.38*	**26.26***
Lex 1	27.87	26.33*	25.52	28.40	26.98	26.73*	28.17	27.16	25.88
Lex 2	27.73	26.61*	25.88*	28.42	27.22*	26.06	28.41	26.98	26.17*
Lex	27.47	26.53*	25.55	28.22	26.86	25.43	28.08	26.75	25.49
All	28.28	26.98*	25.32	28.46	**27.42**	26.17	28.43	27.46*	25.79

For the CS lectures the picture is not as clear. It is not obvious which of the in-domain features is the most important one. But again, using both in-domain relative frequencies leads to the best performance. In this task, this feature selection is clearly better than using all in-domain features. The additional improvement for this task is between 0.4 and 1.0 BLEU points.

In conclusion, using in-domain phrase scores generated significantly better BLEU scores for all tasks. If we use both relative frequencies we get significant improvements on all six conditions. And in both tasks, these improvements are bigger than the ones gained by selecting translation candidates in a different way.

After dealing with the number of adapted features, we focus on the other aspects mentioned in Section 5.

If the phrase pair occurs both in the in-domain and out-of-domain corpus, we can calculate the adapted scores according to definition for the in-domain or out-of-domain phrase pairs for all approaches except the backoff. We can then either use only the ones generated by the in-domain scores or add the phrase pair to the translation model twice with the different scores. In some preliminary experiments, we could not find any significant difference between the two approaches. Therefore, we did not perform any additional experiments on this task and always used two phrase table entries, one based on the in-domain scores, and one based on the out-of-domain scores. Now we concentrated on the other two aspects: whether to include the general scores in the in-domain phrase table entry in addition to the in-domain scores and how to deal with unknown probabilities.

Since the number and type of features is different for all the experiments, a separate optimization had to be run each time. In all the experiments we used the UnionAll method as candidate selection and use two sets of features for one phrase pair if the phrase occurs in the in-domain and out-of-domain corpus. The results are shown in Table 7.

The first system in the table uses no adapted features at all. The next two systems use only the adapted features using the indicator and log-linear method to handle unknown probabilities. The remaining six systems use both, the general scores and the adapted ones. Out of these systems, the first three systems use all adapted scores, while the last three use only the adapted relative frequencies.

If we first look at the TED translation task, the results for the different features are quite similar. The maximal average difference between the different approaches is 0.25 BLEU points. The best result is achieved with the General + Log-Lin combination. The reason for this may be that this approach uses the most features, so there are more dimensions for adapting to the target domain.

The influence of the general scores and the effect of using two or four adapted scores on the translation quality is not clear for this task.

If we now take a look at the translation quality in the task of CS lecture translation, the picture is a little different. First of all, not all features could improve over the baseline system using no in-domain features and the translation quality of the approaches differ more. So it may be harder to gain improvements when using phrase table adaptation, if the test domain does not match the in-domain data perfectly.

On the other hand, using the scores from the general phrase table helps in all cases. In addition, most of the time it was better to use only the relative probabilities for the adaptation and not all four phrase table scores.

The systems General+Indicator and General+rel.Backoff could significantly improve over the baseline system on all conditions.

7 Conclusion

In this paper we analyzed different approaches to perform phrase table adaptation. We compared their way of dealing with different aspects in the translation model adaptation. The comparison was done on two different tasks of speech translation. In a first step, we compared different ways of selecting the candidate phrase pairs and it could be shown that the best performance is reached by the Union or Padding approaches. When the in-domain and test condition do not match very well, the Union approach performed better.

Afterwards we analyzed different methods to select scores for the phrase pairs. First, some approaches use the general and adapted scores, while others only use the adapted ones. In our experi-

Table 7: Feature Combination (BLEU)

System	No LM Adaptation			Log-Lin. LM Adaptation			Linear LM Adaptation		
	Dev	Test		Dev	Test		Dev	Test	
		TED	CS		TED	CS		TED	CS
No	26.51	25.59	25.41	27.56	26.72	26.36	27.65	26.95	25.61
Log-Lin	28.26	27.01*	25.53	28.63	27.51*	25.87	28.18	**27.84***	25.55
Indicator	28.23	27.06*	25.15	28.31	**27.77***	26.34	28.33	27.36	25.06
General + Backoff	28.28	26.98*	25.32	28.46	27.42	26.17	28.43	27.46*	25.79
General + Log-Lin	**28.52**	**27.27***	25.60	28.60	27.59*	26.07	**28.68**	27.74*	**26.52***
General + Indicator	28.34	26.83*	26.17*	28.40	27.37*	26.72*	28.31	27.58*	25.93*
General + rel. Backoff	28.34	27.03*	26.44*	**28.69**	27.38*	**26.79***	28.43	27.38*	26.26*
General + rel. Log-Lin	28.23	26.92*	26.15*	28.61	27.52*	25.81	28.40	27.34*	26.15*
General + rel. Indicator	28.40	27.13*	**26.67***	28.48	27.53*	26.60	28.53	27.54*	26.18*

ments, it turned out that for not perfectly matching training and test condition it was best to include the general scores. Furthermore, the approaches differ in how to handle unknown probabilities. While it was best to use the log-linear approach for the TED task, on the CS task the backoff or indicator feature approach performed best.

Overall, for some aspects there seems to be a best method for both tasks, while for other aspects which method performs best depends on how well the test and in-domain training data matches.

8 Acknowledgements

This work was partly achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

References

Pratyush Banerjee, Sudip Kumar Naskar, Johann Roturier, Andy Way, and Josef van Genabith. 2011. Domain Adaptation in Statistical Machine Translation of User-Forum Data using Component-Level Mixture Modelling. In *13th Machine Translation Summit*, Xiamin,China.

Arianna Bisazza, Nick Ruiz, and Marcello Federico. 2011. Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation. In *International Workshop on Spoken Language Translation (IWSLT)*, San Francisco, USA.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Ivan Bulyko, Spyros Matsoukas, Richard Schwartz, Long Nguyen, and John Makhoul. 2007. Language Model Adaptation in Machine Translation from Speech. In *ICASSP 2007*, Honolulu, USA.

George Foster and Roland Kuhn. 2007. Mixture-Model Adaptation for SMT. In *ACL 2007*, Prague, Czech Republic.

George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable Smoothing for Statistical Machine Translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, Sydney, Australia.

Philipp Koehn and Josh Schroeder. 2007. Experiments in Domain Adaptation for Statistical Machine Translation. In *Second Workshop on Statistical Machine Translation*, Prague, Czech Republic.

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 *ACL 2007, Demonstration Session*, Prague, Czech Republic.

Spyros Matsoukas, Antti-Veikko I. Rosti, and Bing Zhang. 2009. Discriminative Corpus Weight Estimation for Machine Translation. In *Conference on Empirical Methods on Natural Language Processing (EMNLP 2009)*, Singapore.

Jan Niehues and Muntsin Kolss. 2009. A POS-Based Model for Long-Range Reorderings in SMT. In *Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece.

Jan Niehues and Stephan Vogel. 2008. Discriminative Word Alignment via Alignment Matrix Modeling. In *Proc. of Third ACL Workshop on Statistical Machine Translation*, Columbus, USA.

Jan Niehues and Alex Waibel. 2010. Domain Adaptation in Statistical Machine Translation using Factored Translation Models. In *Proceedings of EAMT*, St. Raphael, France.

Jan Niehues, Mohammed Mediani, Teresa Herrmann, Michael Heck, Christian Herff, and Alex Waibel. 2010. The KIT Translation system for IWSLT 2010. In *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, Paris, France.

Jan Niehues, Teresa Herrmann, Stephan Vogel, and Alex Waibel. 2011. Wider Context by Using Bilingual Language Models in Machine Translation. In *Sixth Workshop on Statistical Machine Translation (WMT 2011)*, Edinburgh, UK.

Kay Rottmann and Stephan Vogel. 2007. Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model. In *TMI*, Skövde, Sweden.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, Manchester, UK.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proc. of ICSLP*, Denver, Colorado, USA.

Ying Zhang and Stephan Vogel. 2004. Measuring Confidence Intervals for MT Evaluation Metrics. In *TMI 2004*.