To Cache or not to Cache?
Experiments with Adaptive Models in Statistical Machine Translation

Jörg Tiedemann
Department of Linguistics and Philology
Uppsala University, Uppsala/Sweden
jorg.tiedemann@lingfil.uu.se

Abstract

We report results of our submissions to the WMT 2010 shared translation task in which we applied a system that includes adaptive language and translation models. Adaptation is implemented using exponentially decaying caches storing previous translations as the history for new predictions. Evidence from the cache is then mixed with the global background model. The main problem in this setup is error propagation and our submissions essentially failed to improve over the competitive baseline. There are slight improvements in lexical choice but the global performance decreases in terms of BLEU scores.

1 Motivation

The main motivation of our submission was to test the use of adaptive language and translation models in a standard phrase-based SMT setting for the adaptation to wider context beyond sentence boundaries. Adaptive language models have a long tradition in the speech recognition community and various approaches have been proposed to reduce model perplexity in this way. The general task is to adjust statistical models to essential properties of natural language which are usually not captured by standard n-gram models or other local dependency models. First of all, it is known that repetition is very common especially among content words (see, for example, words like "honey", "milk", "land" and "flowing" in figure 1). In most cases a repeated occurrence of a content word is much more likely than its first appearance, which is not predicted in this way by a static language model. Secondly, the use of expressions is related to the topic in the current discourse and the chance of using the same topic-related expressions again in running text is higher than a mixed-topic model would predict.

In translation another phenomenon can be observed, namely the consistency of translations. Polysemous terms are usually not ambiguous in their context and, hence, their translations become consistent according to the contextual sense. Even the choice between synonymous translations is rather consistent in translated texts as we can see in the example of subtitle translations in figure 1 (taken from the OPUS corpus (Tiedemann, 2009)).

The 10 commandments	**Kerd ma lui**
To some land flowing with milk and **honey**! Till ett land fullt av mjölk och **honung**. I've never tasted **honey**. Jag har aldrig smakat **honung**. ... But will sympathy lead us to this land flowing with milk and **honey**? Men kan sympati leda oss till detta mjölkens och **honungens** land?	Mari **honey** ... Mari, **gumman** **Sweetheart**, where are you going? **Älskling**, var ska du? ... Who was that, **honey**? Vem var det, **gumman**?

Figure 1: Repetition and translation consistency

Ambiguous terms like "honey" are consistently translated into the Swedish counterpart "honung" (in the sense of the actual substance) or "gumman" (in the metaphoric sense). Observe that this is true even in the latter case where synonymous translations such as "älskling" would be possible as well. In other words, deciding to stick to consistent lexical translations should be preferred in MT because the chance of alternative translations in repeated cases is low. Here again, common static translation models do not capture this property at all.

In the following we explain our attempt to integrate contextual dependencies using cache-based adaptive models in a standard SMT setup. We have already successfully applied this technique to a domain-adaptation task (Tiedemann, 2010).

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 189–194,
Uppsala, Sweden, 15-16 July 2010. ©2010 Association for Computational Linguistics

Now we would like to investigate the robustness of this model in a more general case where some in-domain training data is available and input data is less repetitive.

2 Cache-based Adaptive Models

The basic idea behind cache-based models is to mix a large static background model with a small local model that is dynamically estimated from recent items from the input stream. Dynamic cache language models have been introduced by (Kuhn and Mori, 1990) and are often implemented in the form of linear mixtures:

$$\begin{aligned} P(w_n|history) &= (1-\lambda)P_{background}(w_n|history) + \\ & \lambda P_{cache}(w_n|history) \end{aligned}$$

The background model is usually a standard n-gram model taking limited amount of local context from the history into account and the cache model is often implemented as a simple (unsmoothed) unigram model using the elements stored in a fixed-size cache (100-5000 words) to estimate its parameters. Another improvement can be achieved by making the importance of cached elements a function of recency. This can be done by introducing a decaying factor in the estimation of cache probabilities (Clarkson and Robinson, 1997):

$$P_{cache}(w_n|w_{n-k}..w_{n-1}) \approx \frac{1}{Z}\sum_{i=n-k}^{n-1} I(w_n = w_i)e^{-\alpha(n-i)}$$

This is basically the model that we applied in our experiments as it showed the largest perplexity reduction in our previous experiments on domain adaptation.

Similarly, translation models can be adapted as well. This is especially useful to account for translation consistency forcing the decoder to prefer identical translations for repeated terms. In our approach we try to model recency again using a decay factor to compute translation model scores from the cache in the following way (only for source language phrases f_n for which a translation option exist in the cache; we use a score of zero otherwise):

$$\phi_{cache}(e_n|f_n) = \frac{\sum_{i=1}^{K} I(\langle e_n, f_n\rangle = \langle e_i, f_i\rangle) * e^{-\alpha i}}{\sum_{i=1}^{K} I(f_n = f_i)}$$

The importance of a cached translation option exponentially decays and we normalize the sum of cached occurrences by the number of translation options with the same foreign language item that we condition on.

Plugging this in into a standard phrase-based SMT engine is rather straightforward. The use of cache-based language models in SMT have been investigated before (Raab, 2007). In our case we used Moses as the base decoder (Koehn et al., 2007). The cache-based language model can be integrated in the decoder by simply adjusting the call to the language modeling toolkit appropriately. We implemented the exponentially decaying cache model within the standard SRILM toolkit (Stolcke, 2002) and added command line arguments to Moses to switch to that model and to set cache parameters such as interpolation, cache size and decay. Adding the translation model cache is a bit more tricky. For this we added a new feature function to the global log-linear model and implemented the decaying cache as explained above within the decoder. Again, simple command-line arguments can be used to switch caching on or off and to adjust cache parameters.

One important issue is to decide when and what to cache. As we explore a lot of different options in decoding it is not feasible to adapt the cache continuously. This would mean a lot of cache operations trying to add and remove hypotheses from the cache memory. Therefore, we opted for a context model that considers history only from previous sentences. Once decoding is finished translation options from the best hypothesis found in decoding are put into language and translation model cache. This is arguably a strong approximation of the adaptive approach. However, considering our special concern about wider context across sentence boundaries this seems to be a reasonable compromise between completeness and efficiency.

Another issue is related to the selection of items to be cached. As discussed earlier repetition is most likely to be found among content words. Similarly, translation consistency is less likely to be true for function words. In the best case one would know the likelihood of specific terms to be repeated. This could be trained on some development data possibly in connection with word classes instead of fully lexicalized parameters in order to overcome data sparseness and to improve generality. Even though this idea is very tempting it would require a substantial extension of our model and would introduce language and domain-specific parameters. Therefore, we just added a simplistic approach filtering tokens by their length in characters instead. Assuming that longer items are more likely to be content words we simply set a threshold to decide whether to add a term to the cache or not. This threshold can be adjusted using command-line arguments.

Finally, we also need to be careful about noise in the cache. This is essential as the caching approach is prone to error propagation. However, detecting noise is difficult. If there would be a notion of noise in translation hypotheses, the decoder would avoid it. In related work (Nepveu et al., 2004) have studied cache-based translation models in connection with interactive machine translation. In that case, one can assume correct input after post-editing the translation suggestions. One way to approach noise reduction in non-interactive MT is to make use of transition costs in the translation lattice. Assuming that this cost (which is estimated internally within the decoder during the expansion of translation hypotheses) refers to some kind of confidence we can discard translation options above a certain threshold, which is what we did in the implementation of our translation model cache.

3 Experiments

We followed the setup proposed in the shared translation task. Primarily we concentrated our efforts on German-English (de-en) and English-German (en-de) using the constrained track, i.e. using the provided training and development data from Europarl and the News domain. Later we also added experiments for Spanish (es) and English using a similar setup.

Our baseline system incorporates the following components: We trained two separate 5-gram language models for each language with the standard smoothing strategies (interpolation and Kneser-Ney discounting), one for Europarl and one for the News data. All of them were estimated using the SRILM toolkit except the English News LM for which we applied RandLM (Talbot and Osborne, 2007) to cope with the large amount of training data. We also included two separate translation models, one for the combined Europarl and News data and one for the News data only. They were estimated using the standard tools GIZA++ (Och and Ney, 2003) and Moses (Koehn et al., 2007) applying default settings and lowercased training data. Lexicalized reordering was trained on the combined data set. All baseline models were then tuned on the News test data from 2008 using minimum error rate training (MERT) (Och, 2003). The results in terms of lower-case BLEU scores are listed in table 1.

		n-gram scores			
	BLEU	1	2	3	4
de-en baseline	21.3	57.4	27.8	15.1	8.6
de-en cache	21.5	58.1	28.1	15.2	8.7
en-de baseline	15.6	52.5	21.7	10.6	5.5
en-de cache	14.4	52.6	21.0	9.9	4.9
es-en baseline	26.7	61.7	32.7	19.9	12.6
es-en cache	26.1	62.6	32.7	19.8	12.5
en-es baseline	26.9	61.5	33.3	20.5	12.9
en-es cache	23.0	60.6	30.4	17.6	10.4

Table 1: Results on the WMT10 test set.

In the adaptation experiments we applied exactly the same models using the feature weights from the baseline with the addition of the caching components in both, language models and translation models. Cache parameters are not particularly tuned for the task in our initial experiments which could be one reason for the disappointing results we obtained. Some of them can be integrated in the MERT procedure, for example, the interpolation weight of the translation cache. However, tuning these parameters with the standard procedures appears to be difficult as we will see in later experiments presented in section 3.2. Initially we used settings that appeared to be useful in previous experiments. In particular, we used a language model cache of 10,000 words with a decay of $\alpha = 0.0005$ and an interpolation weight of 0.001. A cache was used in all language models except the English News model for which caching was not available (because we did not implement this feature for RandLM). The translation cache size was set to 5,000 with a decay factor of 0.001. The weight for the translation cache was set to 0.001. Furthermore, we filtered items for the translation cache using a length constraint of 4 characters or more and a transition cost threshold (log score) of -4.

The final results of the adaptive runs are shown in table 1. In all but one case the cache-based result is below the baseline which is, of course, quite disappointing. For German-English a small improvement can be observed. However, this may be rather accidental. In general, it seems that the adaptive approach cannot cope with the noise added to the cache.

3.1 Discussion

There are two important observations that should be mentioned here. First of all, the adaptive approach assumes coherent text input. However, the WMT test-set is composed of many short news headlines with various topics involved. We, therefore, also ran the adaptive approach on individual news segments. The results are illustrated in figure 2.

Basically, the results do not change compared to the previous run. Still, cache-based models perform worse on average except for the German-English test-set for which we obtained a slight but insignificant improvement. Figure 2 plots the BLEU score differences between standard models and cached models for the individual news items. We can see a very blurred picture of these individual scores and the general conclusion is that caching failed. One problem is that the individual news items are very short (around 20 sentences each) which is probably too little for caching to show any positive effect. Surprising, however, is the negative influence of caching even on these small documents which is quite similar to the runs on the entire sets. The drop in performance for English-Spanish is especially striking. We have no explanation at this point for this exceptional behavior.

A second observation is the variation in individual n-gram precision scores (see table 1). In all but one case the unigram precision goes up which indicates that the cache models often improve lexical choice at least in terms of individual words. The first example in figure 2 could be seen as a slight improvement due to a consistent lexical choice of “missile” (instead of “rocket”).

The main problem, however, in the adaptive approach seems to appear in local contexts which might be due to the simplistic language modeling cache. It would be interesting to study possibilities of integrating local dependencies into the cache models. However, there are serious problems with data sparseness. Initial experiments with a bigram LM cache did not produce any improvements so far.

baseline	until the end of the journey , are , in turn , technical damage to the rocket .
cache	until the end of the journey , in turn , technical damage to the missile .
reference	but near the end of the flight there was technical damage to the missile .
baseline	iran has earlier criticism of its human rights record .
cache	iran rejected previous criticism of its human rights record .
reference	iran has dismissed previous criticism of its human rights record .
baseline	facing conservationists is accused of extortion
cache	facing conservationists is accused of extortion
reference	Nature protection officers accused of blackmail
baseline	the leitmeritz-polizei accused the chairman of the bürgervereinigung ” naturschutzgemeinschaft leitmeritz ” because of blackmail .
cache	the leitmeritz-polizei accused the chairman of the bürgervereinigung ” naturschutzgemeinschaft leitmeritz ” because of extortion .
reference	The Litomerice police have accused the chairman of the Litomerice Nature Protection Society civil association of blackmail.

Table 2: German to English example translations.

Another crucial problem with the cache-based model is of course error propagation. An example which is probably due to this issue can be seen in table 2 in the last two translations (propagation of the translation option “extortion”). This problem is difficult to get around especially in case of bad baseline translations. One possible idea would be to implement a two-pass procedure to run over the entire input first only to fill the cache and to identify reliable evidence for certain translation options (possibly focusing on simple translation tasks such as short sentences). Then, in the second pass the adaptive model can be applied to prefer repetition and consistency according to the parameters learned in the first pass.

3.2 Parameter Optimization

Another question is if the cache parameters require careful optimization in order to make this approach effective. An attempt to investigate the influence of the cache components by simply varying the interpolation weights gave us the following results for English-German (see table 3).

fixed cache TM parameters		fixed cache LM parameters	
λ_{LM}	BLEU	λ_{TM}	BLEU
0.1	14.12	0.1	12.75
0.01	14.39	0.01	13.04
0.005	14.40	0.005	13.57
0.001	14.44	0.001	14.42
0.0005	14.43	0.0005	14.57

Table 3: Results for English to German with varying mixture weights.

Looking at these results the tendency of the scores

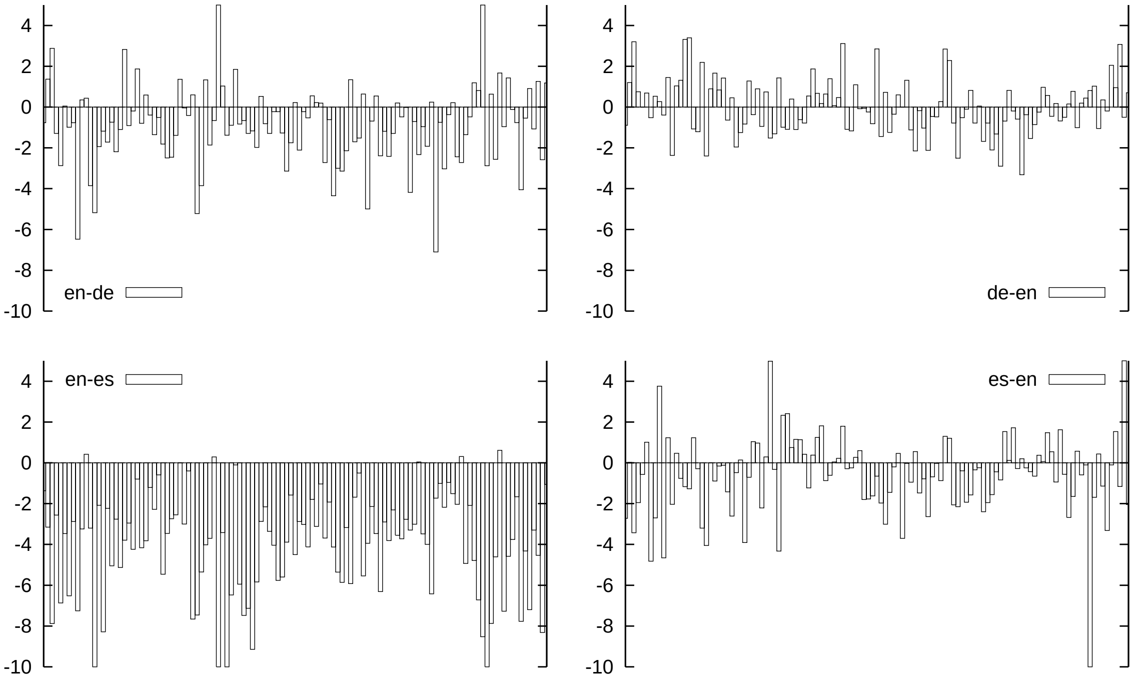

Figure 2: BLEU score differences between a standard model and a cached model for individual news segments from the WMT test-set.

seems to suggest that switching off caching is the right thing to do (as one might have expected already from the initial experimental results). We did not perform the same type of investigation for the other language pairs but we expect a similar behavior.

Even though these results did not encourage us very much to investigate the possibilities of cache parameter optimization any further we still tried to look at the integration of the interpolation weights into the MERT procedure. The weight of the TM cache is especially suited for MERT as this component is implemented in terms of a separate feature function within the global log-linear model used in decoding. The LM mixture model, on the other hand, is implemented internally within SRILM and therefore not so straightforward to integrate into standard MERT. We, therefore, doubled the number of LM's included in the SMT model using two standard LM's and two LM's with cache (one for Europarl and one for News in both cases). The latter are actually mixtures as well using a fixed interpolation weight of $\lambda_{LM} = 0.5$ between the cached component and the background model. In this way the cached LM's benefit from the smoothing with the static background model. Individual weights for all four LM's are then learned in the global MERT procedure. Unfortunately, other cache parameters cannot be optimized in this way as they do not produce any particular values for individual translation hypotheses in decoding.

We applied this tuning setup to the English-German translation task and ran MERT on the same development data as before. Actually, caching slows down translation quite substantially which makes MERT very slow. Due to the sequential caching procedure it is also not possible to parallelize tuning. Furthermore, the extra parameters seem to cause problems in convergence and we had to stop the optimization after 30 iterations when BLEU scores seemed to start stabilizing around 14.9 (in the standard setup only 12 iterations were required to complete tuning). Unfortunately, the result is again quite disappointing (see table 4).

Actually, the final BLEU score after tuning is even lower than in our initial runs with fixed cache parameters taken from previous unrelated experiments. This is very surprising and it looks like that MERT just failed to find settings close to the global optimum because of some strong local suboptimal points in the search space. One would expect that it should be possible to obtain at least the

BLEU on dev-set (no caching)	15.2
BLEU on dev-set (with caching)	14.9
Europarl LM	0.000417
News LM	0.057042
Europarl LM (with cache)	0.002429
News LM (with cache)	-0.000604
λ_{TM}	0.000749
BLEU on test-set (no caching)	15.6
BLEU on test-set (with caching)	12.7

Table 4: Tuning cache parameters.

same score on the development set which was not the case in our experiment. However, as already mentioned, we had to interrupt tuning and there is still some chance that MERT would have improved in later iterations. At least intuitively, there seems to be some logic behind the tuned weights (shown in table 4). The out-of-domain LM (Europarl) obtains a higher weight with caching than without and the in-domain LM (News) is better without it and, therefore, the cached version obtains a negative weight. Furthermore, the TM cache weight is quite similar to the one we used in the initial experiments. However, applying these settings to the test-set did not work at all.

4 Conclusions

In our WMT10 experiments cache-based adaptive models failed to improve translation quality. Previous experiments have shown that they can be useful in adapting SMT models to new domains. However, they seem to have their limitations in the general case with mixed topics involved. A general problem is error propagation and the corruption of local dependencies due to over-simplified cache models. Parameter optimization seems to be difficult as well. These issues should be investigated further in future research.

References

P.R. Clarkson and A. J. Robinson. 1997. Language model adaptation using mixtures and an exponentially decaying cache. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 799–802, Munich, Germany.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *ACL '07: Proceedings of the 45th Annual Meeting of the ACL*, pages 177–180, Morristown, NJ, USA.

Roland Kuhn and Renato De Mori. 1990. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583.

Laurent Nepveu, Lapalme, Guy, Langlais, Philippe, and George Foster. 2004. Adaptive Language and Translation Models for Interactive Machine Translation. In *Proceedings of the 9th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 190–197, Barcelona, Spain.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA.

Martin Raab. 2007. *Language Modeling for Machine Translation*. VDM Verlag, Saarbrücken, Germany.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the 7th international conference on spoken language processing (ICSLP 2002)*, pages 901–904, Denver, CO, USA.

David Talbot and Miles Osborne. 2007. Randomised language modelling for statistical machine translation. In *ACL '07: Proceedings of the 45th Annual Meeting of the ACL*, Prague, Czech Republic. Association for Computational Linguistics.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia.

Jörg Tiedemann. 2010. Context adaptation in statistical machine translation using models with exponentially decaying cache. In *ACL 2010 Workshop on Domain Adaptation for Natural Language Processing (DANLP)*.