Integrating a Large, Monolingual Corpus as Translation Memory into Statistical Machine Translation

Katharina Wäschle and **Stefan Riezler**
Department of Computational Linguistics
Heidelberg University
69120 Heidelberg, Germany
{waeschle, riezler}@cl.uni-heidelberg.de

Abstract

Translation memories (TM) are widely used in the localization industry to improve consistency and speed of human translation. Several approaches have been presented to integrate the bilingual translation units of TMs into statistical machine translation (SMT). We present an extension of these approaches to the integration of partial matches found in a large, monolingual corpus in the target language, using cross-language information retrieval (CLIR) techniques. We use locality-sensitive hashing (LSH) for efficient coarse-grained retrieval of match candidates, which are then filtered by fine-grained fuzzy matching, and finally used to re-rank the n-best SMT output. We show consistent and significant improvements over a state-of-the-art SMT system, across different domains and language pairs on tens of millions of sentences.

1 Introduction

A translation memory (TM) is a computational tool used by professional translators to speed up translation of repetitive texts. At its core is a database, in which source and target of previously translated segments of text are stored. TMs are capable of retrieving not only exact, but also partial matches, where only a certain percentage of source words overlap with the query, called fuzzy matches. A computer-assisted translation (CAT) tool presents possible matches found in the database to a user, if the match is considered similar enough to the current source sentence. Even if the presented target sentence is not a perfect translation, a fuzzy match can be a good starting point for the translation of the current sentence and reduce translation time and effort. Furthermore, the approach can help with translation consistency and terminology control. In contrast to statistical machine translation (SMT), TM tools are widely used in the translation industry, since the results presented to the translator are fluent translations. They are especially successful for translation of texts from repetitive domains, e.g. technical documents such as IT manuals, that are the predominant use case in the localization industry.

The idea of combining the strengths of TM and SMT tools has been successfully explored in recent years. In this paper, we extend these approaches to the integration of a large, monolingual corpus in the target language as a TM into an SMT system using cross-language information retrieval (CLIR). Our approach utilizes locality-sensitive hashing (LSH) as an efficient coarse retrieval technique to select candidate translations. In a next step, search is performed at a finer-grained level using distance metrics customary in CAT. Given a match, our model re-ranks the n-best list output by an SMT decoder using features modeling the closeness of the hypothesis and the target of the TM match. Since our approach does not rely on an alignment between source and target side of the TM match, we are able to search for potential matches in large, monolingual corpora that might only be available in the target language. We show consistent and significant improvements on different domains (IT, legal, patents) for different language pairs (including Chinese, Japanese, English, French, and German), achieving results comparable to or better than using a target-language reference of source-side matches.

2 Related Work

Work on integrating MT and SMT can be divided into approaches at the sentence level that decide whether to pass SMT or TM output to the user (He et al., 2010a,b), and approaches that merge both techniques at a sub-sentential level (Smith and Clark, 2009; Koehn and Senellart, 2010; Zhechev and van Genabith, 2010; Wang et al., 2013). While the goal of the former is to improve human translation effort in a CAT environment, the second line of research aims to improve SMT performance.

Biçici and Dymetman (2008) were among the first to propose a combined system. They start by identifying matching subsequences between the current sentence and a fuzzy match retrieved from a translation memory. Source and target of the match together with the corresponding alignment are used to construct a non-contiguous bi-phrase, which is added to the SMT grammar with a strong weight. The decoder is then run as usual using the augmented grammar. The approaches of Koehn and Senellart (2010), Zhechev and van Genabith (2010), and Ma et al. (2011), force the SMT system to translate only the unmatching segments of the source, either by restricting translation or by adding a very high feature weight to rules or biphrases extracted from the TM match. While all presented approaches make use of the alignment between source and target of the fuzzy match, our approach uses only the target side to restrict the translation, making it possible to use matches that can be found in a target-only corpus.

The use of TM matches to generate additional features for SMT has been explored by Simard and Isabelle (2009), Wang et al. (2013), Wang et al. (2014) and Li et al. (2014). Our re-ranking approach is very similar, with the novelty of using not only matches found by querying the source side of the corpus, but also the target.

The idea of directly searching for translations in a monolingual target language corpus has been explored by Dong et al. (2014). They retrieve target side translation candidates using a lattice representation of possible translations of a source sentence. The system is successfully applied to the task of identifying parallel sentences, but no SMT experiments are reported.

3 Integrating monolingual TM into SMT

Our integrated model uses a coarse-to-fine approach for integrating TM information into an SMT system: First, efficient retrieval is done using locality-sensitive hashing on large corpora. Second, a more fine-grained search for the best match is performed for a given sentence. Lastly, a reranking step uses this information to re-score the n-best list output of an SMT decoder.

3.1 Coarse-grained retrieval using LSH

In order to be able to use large corpora as translation memory, a fast method is needed to retrieve matches. In CAT practice, the goodness of a TM match is calculated using the so-called fuzzy match score (Sikes, 2007),

$$\mathrm{FMS}(s_1, s_2) = 1 - \frac{\mathrm{LD}(s_1, s_2)}{\max(|s_1|, |s_2|})$$

which is based on the Levenshtein distance LD, i.e. the minimum number of operations[1] needed to transform the sequence s_1 into the sequence s_2. Levenshtein distance can be computed with dynamic programming in $O(mn)$ time. However, computing edit distance against a corpus of tens of millions of sentences is too slow for real-time use, especially for long sentences that appear e.g. in patent data. This leads us to a two-step approach with a coarse pre-retrieval that delivers candidates for good fuzzy matches for a given sentence in milliseconds. For a smaller candidate set we can then compute the exact fuzzy match score.

MinHash (Broder, 1997) is a way to estimate the similarity of two documents by reducing the dimensionality of the document signature using sampling. It is an instance of locality-sensitive hashing, where similar items hash to the same bucket, which makes comparison extremely fast, since only hashes have to be compared. It is usually employed for tasks such as near-duplicate detection of websites, but can be applied to our task as well. MinHash approximates the Jaccard similarity of two sets X and Y,

$$\mathrm{JC}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

by generating signatures of each set, from which the Jaccard similarity can be estimated. The signature is gained by repeatedly hashing each member

[1] Allowed operations are removal, insertion, substitution or transposition.

of the set and storing only the minimal resulting hash. By representing each sentence as a set of n-grams we can use this technique to efficiently approximate the n-gram overlap of two sentences. n-gram overlap has been found to be a good predictor of TM match quality (Bloodgood and Strauss, 2014). In our experiments we used 3-grams to represent sentences in corpora with high average sentence length (legal, patent) and 1-grams for data sets featuring short sentences (IT).

To efficiently estimate the Jaccard Similarity from the MinHash signatures, we apply the banding technique described in Rajaraman and Ullman (2012, Chapter 3), where similar items are likely to get hashed to the same bucket. When setting the similarity threshold t, which regulates how similar two items have to be in order to become candidates, we are faced with a effectiveness-efficiency trade-off (Ture et al., 2011), where we find false positives, which slow down the second retrieval step, and also false negatives, which will cost overall performance. We set the t for each dataset on a held-out development set by choosing a setting in which a candidate match is returned for at least 90% of the sentences. We then compute the actual Jaccard Similarity for the set of match candidates returned by the hashing step and rank them accordingly. We take the 100 best matches for each query q_i and choose the best match from them in the fine-grained step described in the following.

3.2 Fine-grained matching

In the standard bilingual case, choosing the best TM match amounts to selecting the sentence pair (s, t) from the coarse candidate set $\mathrm{LSH}(q_i)$ that achieves the highest fuzzy match score FMS of the (source) query q_i against the source side $s_{i,j}$ of the TM pair, and returning its target side $t_{i,j}$.

$$(s,t)_{i,best} = \underset{(s,t)_{i,j} \in \mathrm{LSH}(q_i)}{\mathrm{argmax}} \mathrm{FMS}(q_i, s_{i,j}).$$

For the target-language scenario, however, this step is not straightforward. We want to select a target sentence t from a set of target-only candidates given a query q_i in the source language, however, in order to do this, we require a cross-language similarity score CLIR. To generate a target candidate set with coarse retrieval we use the 1-best translation $Tr(q_i)$ by an SMT decoder trained on bilingual data as a query[2].

$$t_{i,best} = \underset{t_{i,j} \in \mathrm{LSH}(Tr(q_i))}{\mathrm{argmax}} \mathrm{CLIR}(q_i, t_{i,j}).$$

To determine the best match among the candidates in a fine-grained way, we investigate three different cross-language techniques.

1-best FMS. This model uses as a selection criterion the fuzzy match score of the candidate $t_{i,j}$ given the most likely translation hypothesis produced for the query q_i by an SMT model, $Tr(q_i)$.

$$\mathrm{CLIR}(q_i, t_{i,j}) = \mathrm{FMS}(Tr(q_i), t_{i,j})$$

This corresponds to a direct translation baseline in cross-language information retrieval.

In addition to this simple model, we explore two methods that operate on the full translation hypergraph of the query. Both techniques are similar to the translation retrieval technique presented by Dong et al. (2014). They perform Viterbi search on a translation lattice of the input sentence that is enriched, besides the default SMT features, with n-gram features that indicate the overlap status between the current state in the lattice and a given TM match. We adopt this approach for the hypergraph built by the cdec decoder (Dyer et al., 2010). As a cross-lingual similarity measure we then compute the Viterbi score on the query hypergraph $Hg(q_i)$ for each match candidate $t_{i,j}$, i.e.

$$\mathrm{CLIR}(q_i, t_{i,j}) = \max_{p \in Hg(q_i)} \sum_{e \in p} w_{\mathrm{SMT}} \cdot \phi_{\mathrm{SMT}}(e(q_i)) + w_{\text{n-gr}} \cdot \phi_{\text{n-gr}}(e(q_i), t_{i,j}))$$

where p is a path through the hypergraph, e the set of edges on the path, ϕ are feature values of an edge, w the corresponding weights, and $\cdot$ denotes the vector dot product. We explore two different ways to incorporate n-gram features $\phi_{\text{n-gr}}$ in addition to the SMT feature set ϕ_{SMT}.

Unigram oracle. Since n-gram features are non-local and the size of the hypergraph grows when adding n-gram features for orders higher than $n = 1$ (Chiang, 2007), we restrict our first model to unigram precision and a brevity penalty feature; the latter is only active at goal state. In this way, two additional features are inserted into the log-linear model, using the TM match candidate as an oracle.

[2] We also tested query constructions involving a larger set of possible translations, but found that using the 1-best translation prediction of the baseline system yielded superior results.

Additional language model. To be able to include higher-order n-gram matches, we add the match candidates as an additional language model to the decoder.This approach makes use of the fact that `cdec` handles the extension of the hypergraph to accommodate for the non-local higher order n-grams. Cube pruning (Chiang, 2007) is used to make the search feasible.

In both cases, we keep the weights of the SMT features fixed, which have been optimized for translation performance on a development set, and only adjust the additional weights in relation. This is done by pairwise ranking (Hopkins and May, 2011). The gold standard ranking of the TM candidates is given by $\mathrm{FMS}(t_{i,j}, r_i)$ with respect to the reference r_i for q_i. The learning goal is to adjust the weights of the n-gram features so as to rank the TM match highest that has the smallest distance to the reference. Note, that we do not optimize the translation performance of the derivation, which corresponds to the Viterbi path. This could potentially replace the re-ranking step and we plan to explore this option in the future.

3.3 Re-ranking SMT output

To incorporate the retrieved TM match into the SMT pipeline we use a simple re-ranking model on the n-best list output by the baseline SMT system and select the best hypothesis $\hat{h}$ under this model. We balance information from the SMT model and the TM by computing a linear interpolation of SMT model score SMT and fuzzy match score FMS between hypothesis h and best TM target match $t_{i,best}$. We also add a confidence-weighted version of the FMS score using the retrieval score (CL)IR between TM match and original query q_i as confidence measure:

$$\begin{aligned}\hat{h} = \operatorname*{argmax}_{h \in H(q_i)} & w_1 \times \mathrm{SMT}(h) \\ & + w_2 \times \mathrm{FMS}(h, t_{i,best}) \\ + w_3 \times ((\mathrm{CL})\mathrm{IR}&(q_i, t_{i,best}) \times \mathrm{FMS}(h, t_{i,best})).\end{aligned}$$

We experimented with more features, including n-gram overlap and a brevity penalty, but found that they did not add any information that was not already present in the model. We learn weights for the different components of the score by pairwise ranking using PRO (Hopkins and May, 2011). This time the gold-standard ranking is induced on the n-best list of SMT outputs by TER match against the reference.

domain	sentences	vocabulary size	
		src	tgt
acquis (en-fr)	1M	121K	140K
oo3 (en-zh)	50K	6K	8K
ntcir (jp-en)	1.6M	96K	185K
pattr (en-de)	10.1M	728K	679K

Table 1: Statistics for experimental data.

	acquis	oo3	ntcir	pattr
RR	16.85	5.98	16.9	5.85
SL	27.27	6.48	33.91	33.55

Table 2: Test set repetition rates (RR) and average sentence length (SL) in tokens.

4 Experiments

Since translation memories are most effective on text that has a certain amount of repetition, we evaluate our approach on typical localization data, from the IT, legal and intellectual property domains[3] (Table 1). All corpora are freely available for research purposes. We report repetition rate (Cettolo et al., 2014) and average sentence length in Table 2 and show the number of matches for each fuzzy match interval in Table 3. Among the freely available corpora, only the JRC-Acquis corpus has been used previously in combinations of TM and SMT (Koehn and Senellart, 2010; Li et al., 2014). Most works in this area report results on TM data from industrial partners that are not publicly available. Usually, these datasets feature a large proportion of fuzzy matches in high ranges, e.g. between 80% and 100%, which makes it possible for the combined systems to achieve a large boost in score. Our reported results are in a smaller range, but achieved on data with much less high-percentage matches. We manage to gain improvements in performance from matches with an associated fuzzy match score between 10% and 80%.

[3] Europarl has been used as a dataset by (Koehn and Senellart, 2010), but performance of the enriched SMT system actually dropped below the baseline, showing that less repetitive corpora are badly suited for the TM adaptation methods.

We prepared an English-Chinese corpus of IT manuals from the OPUS[4] corpus (Tiedemann, 2012), the OpenOffice 3 (OO3) data. We only kept pairs that contained at least one Chinese character[5]. The Chinese side was segmented using the Stanford Word Segmenter (Tseng et al., 2005) with the Penn Treebank standard. Development and test sets were created by randomly sampling 1,000 sentence pairs each and remaining pairs used for training. We used English-French legal data from the JRC-Acquis corpus[6] (Steinberger et al., 2006) and sampled dev, devtest and test set from documents published in 2000. The remaining years were used for training. We evaluated our approach on two patent data sets; English-German data from the PatTR[7] corpus (Wäschle and Riezler, 2012) and Japanese-English data from the NTCIR[8] challenge (Utiyama and Isahara, 2007). We used NTCIR-10 dev, test[9] and training set. Held-out data sets for PatTR were sampled from documents from 2006, the remaining data formed the training set.

	acquis	oo3	ntcir	pattr
0-10%	5	0	15	17
10-20%	68	4	118	121
20-30%	89	3	200	205
30-40%	56	10	167	187
40-50%	51	3	95	88
50-60%	70	13	58	56
60-70%	52	14	28	19
70-80%	59	15	17	28
80-90%	109	29	8	18
90-99%	136	21	1	8
100%	292	500	6	19

Table 3: Number of test sentences with source side fuzzy match score in a certain range.

We trained a baseline SMT system using the `cdec` decoder (Dyer et al., 2010) and the accompanying tools, i.e. `fast align` (Dyer et al., 2013) on each data set. A 6-gram language model was trained with SRILM (Stolcke, 2002) on the target side of the training data. The weights of the log-linear model were optimized with MIRA (Watanabe et al., 2007) on a held-out development set reserved for this purpose (dev). We employed the baseline model to produce query translations and hypergraphs for the cross-lingual retrieval of target matches as well as to produce 500-best lists, which we re-ranked according to our model given the best match found after fine-grained retrieval. Retrieval and re-ranking parameters were optimized on an additional held-out (devtest) set. All presented results were obtained on a third (test) data set. To compare source and different target retrieval methods in a fair setting, we used the bilingual data from training the SMT model as translation memory, restricted to the target side for target retrieval. To evaluate our target retrieval approach in more a realistic setting, we furthermore set up an experiment for the English-German patent task, where SMT training data and monolingual TM deviate. We assume that we have parallel data from patent claims and the task is to translate text from a different genre, patent descriptions, for which only data in the target language available as well as a small amount of bitext to tune parameters on – a typical domain adaptation scenario. The available monolingual data is used to extend both the language model as well as the target-language TM (Table 4).

	genre	size (sent.)
parallel train (en-de)	cl.	6M
dev/devtest/test (en-de)	desc.	1K (each)
LM-train (de)	cl.+descr.	16.2M
TM (de)	cl.+desc.	16.2M

Table 4: Data for domain adaptation scenario.

We report BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) evaluation scores. Statistical significance of all results was assessed following the method described in Clark et al. (2011) using the source code provided by the authors[10].

4.1 Results

Results in Table 5 show that adding the TM information always improves over the baseline, up to 1.23 BLEU and -3.77 TER. Improvements in TER (the optimized metric) are always significant at $p < 0.05$. Both source and target-side match re-

[4] http://datahub.io/de/dataset/opus

[5] We tested for Chinese characters by checking if they were in the Unicode range `[0x4E00, 0x9FFF]`.

[6] `https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis`

[7] `http://www.cl.uni-heidelberg.de/statnlpgroup/pattr/`

[8] `http://research.nii.ac.jp/ntcir/ntcir-10/`

[9] We merged, shuffled and again split up the data into three sets, to generate a devtest set.

[10] `https://github.com/jhclark/multeval`

	acquis		oo3		ntcir		pattr	
	BLEU	TER	BLEU	TER	BLEU	TER	BLEU	TER
baseline	61.43%	28.16%	36.04%	50.83%	24.52%	66.52%	26.89%	57.51%
+src-rr	62.62%	**26.63%**	**36.65%**	**50.01%**	**25.51%**	**62.75%**	27.11%	57.04%
	+1.19%	**-1.57%**	**+0.61%**	**-0.82%**	**+0.99%**	**-3.77%**	+0.22%	-0.47%
+tgt-FMS-rr	**62.92%**	26.79%	36.26%	50.13%	25.23%	63.59%	**27.31%**	**56.78%**
	+1.48%	-1.37%	+0.22%*	-0.70%	+0.71%	-2.93%	**+0.42%**	**-0.73%**
+tgt-Oracle-rr	62.23%	27.56%	36.16%	50.17%	24.55%	66.20%	27.03%	57.25%
	+0.80%	-0.60%	+0.12%	-0.66%	+0.03%*	-0.31%	+0.13%*	-0.26%
+tgt-LM-rr	62.29%	27.45%	36.09%	50.15%	24.63%	66.11%	26.98%	57.29%
	+0.85%	-0.71%	+0.05%*	-0.67%	+0.11%*	-0.41%	+0.09%*	-0.21%

Table 5: BLEU and TER difference to baseline for TM integration on by source-side matching and re-ranking (+src-rr) and variants of target-side matching and re-ranking (+tgt-*-rr). All improvements, except marked with *, are significant w.r.t the baseline at $p < 0.05$. Best results in **bold face**.

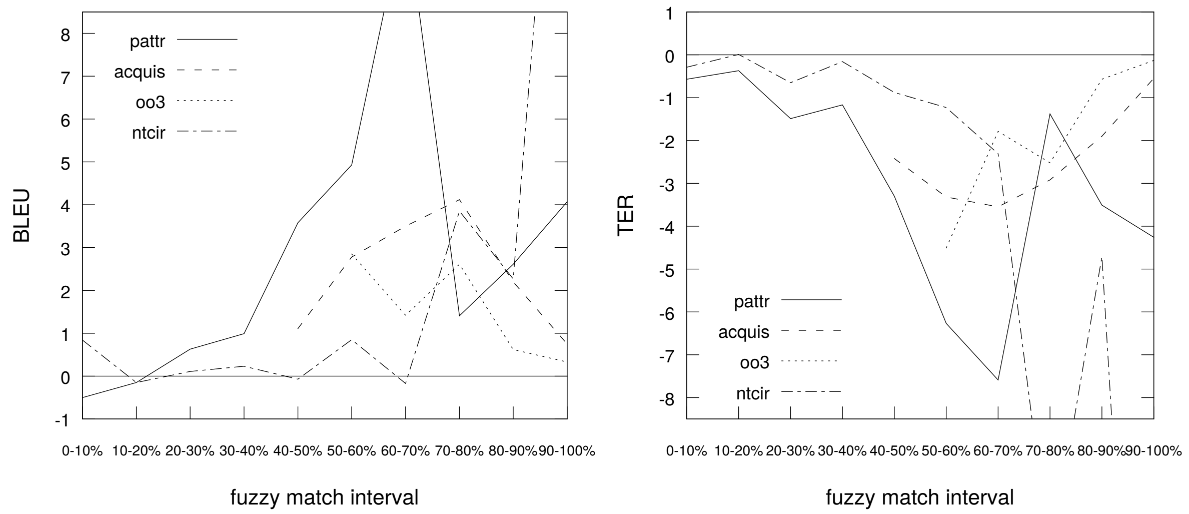

Figure 1: Δ BLEU and Δ TER between baseline and system output on different fuzzy match intervals

trieval beat the baseline. Re-ranking using target-side only matches beats source-side retrieval on two datasets. n-gram based models for choosing the best target match always perform worse than the fuzzy-match-score based models.

Figure 1 shows detailed results on the different fuzzy match intervals, in particular the difference between +tgt-FMS-rr system and the baseline. It is interesting to note that the highest gains are achieved in the 70-80% range, while previous research reports highest gains in the 95-100% range. This is apparently dependent on the data set, but it also suggests, that the baseline SMT system is already very good in the high match range, at least for short sentences. For ntcir we achieve extremely high numbers in the 90-100% range and for pattr in the 60-70% but these scores are achieved on very few examples (7 and 14, respectively) and therefore cannot be expected to be stable. The difference between the datasets is probably due to the average sentence length – shorter sentences with a perfect match in the TM are easier to reproduce for the SMT system than longer ones, due to the smaller number of translation options. It is also remarkable, that for ntcir and pattr datasets even extremely low-range matches are beneficial. While there are some drops in terms of BLEU, TER always goes down, even on 0-10% matches. Having established that target-side retrieval performs comparably to source retrieval, we evaluate our approach in the domain adaptation setting, where additional monolingual data for the TM is available. Results are given in Table 6. We find significant improvements over the competitive baseline with an adapted language model without adding any bilingual data.

	BLEU	TER
baseline	21.58%	62.54%
+tgt-FMS-rr	**21.81%**	**62.18%**
	+0.23%	**-0.36%**

Table 6: Results for domain adaptation scenario.

Figure 2 compares translation output between baseline and the +tgt-FMS-rr extension, showing that the system is able to correct syntactical errors, but also, that some changes consist only of swapping translations for a term, where both translations would be correct choices. In this case, the translation both gains and loses from this phenomenon with regard to the reference. We assume that this holds for the whole test set: in some cases out system will randomly pick the right (used by the reference) translation; sometimes adding a match will change a correct translation. Since overall our system improves significantly over the baseline, meaningful changes are made frequently.

5 Conclusion

We present an approach to integrate large corpora as translation memories into an SMT system, which yields consistent and significant improvements over baseline results on IT, legal and patent data. In contrast to previous approaches, the discriminative model is light-weight and needs no phrase-segmentation or alignment between TM source and target, allowing for the integration of partial matches found in the target language. Results with target-language matches are comparable to using a target reference of source-side matches.

In future work, we would like to extend our approach to multiple fuzzy matches for one source sentence that cover different spans of the input, as proposed in Li et al. (2014). Furthermore, we would like to conduct experiments on a translation memory gained from real-world industrial data with post-editing feedback.

References

Biçici, E. and Dymetman, M. (2008). Dynamic translation memory: Using statistical machine translation to improve translation memory fuzzy matches. In *Computational Linguistics and Intelligent Text Processing*, pages 454–465.

Bloodgood, M. and Strauss, B. (2014). Translation memory retrieval methods. In *EACL*.

Broder, A. Z. (1997). On the resemblance and containment of documents. In *Compression and Complexity of Sequences*, pages 21–29.

Cettolo, M., Bertoldi, N., and Federico, M. (2014). The repetition rate of text as a predictor of the effectiveness of machine translation adaptation. In *AMTA*.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *ACL-HLT*.

Dong, M., Cheng, Y., Liu, Y., Xu, J., Sun, M., Izuha, T., and Hao, J. (2014). Query lattice for translation retrieval. In *COLING*.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of IBM model 2. In *NAACL-HLT*.

Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., Setiawan, H., Eidelman, V., and Resnik, P. (2010). cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *ACL*.

He, Y., Ma, Y., van Genabith, J., and Way, A. (2010a). Bridging SMT and TM with translation recommendation. In *ACL*.

He, Y., Ma, Y., Way, A., and Van Genabith, J. (2010b). Integrating n-best SMT outputs into a TM system. In *COLING*.

Hopkins, M. and May, J. (2011). Tuning as ranking. In *EMNLP*.

Koehn, P. and Senellart, J. (2010). Convergence of translation memory and statistical machine translation. In *AMTA*.

Li, L., Way, A., and Liu, Q. (2014). A discriminative framework of integrating translation memory features into SMT. In *AMTA*.

Ma, Y., He, Y., Way, A., and van Genabith, J. (2011). Consistent translation using discriminative learning: A translation memory-inspired approach. In *ACL*.

source	*in one particular embodiment , the aliphatic hydroxy carboxylic acids bear the hydroxyl group and the carboxyl group on the same carbon atom .*
baseline	*in einer besonderen ausführungsform die aliphatischen hydroxycarbonsäuren , die die hydroxylgruppe und die carboxylgruppe an ein und demselben kohlenstoffatom tragen .*
+tgt-FMS-rr	*in einer besonderen ausführungsform* ***tragen*** *die aliphatischen hydroxycarbonsäuren die hydroxygruppe und die carbonsäuregruppe* ***am gleichen c - atom*** *.*
tm match	*in einer besonderen ausführungsform des erfindungsgemäßen verfahrens tragen die aliphatischen hydroxycarbonsäuren die hydroxy - und carbonsäuregruppe am selben c - atom .*
reference	*in einer besonderen ausführungsform tragen die aliphatischen hydroxycarbonsäuren die hydroxy - und carboxyl gruppe am gleichen c - atom .*

Figure 2: Example system output on patent test set: With the TM match, the syntax of the output has been corrected: the subordinate clause has been removed and the verb *tragen* placed correctly in the main clause. *kohlenstoffatom* became *c - atom*, which is both correct, but the latter is the term used in the reference; on the other hand, *carboxylgruppe* was correctly output by the baseline, but changed to *carbonsäuregruppe* – correct, but not the term used in the reference.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830.

Rajaraman, A. and Ullman, J. D. (2012). *Mining of massive datasets*. Cambridge University Press.

Sikes, R. (2007). Fuzzy matching in theory and practice. *MultiLingual*, 18(6):39–43.

Simard, M. and Isabelle, P. (2009). Phrase-based machine translation in a computer-assisted translation environment. *MT Summit XII*.

Smith, J. and Clark, S. (2009). Ebmt for SMT: A new EBMT-SMT hybrid. In *EBMT*.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *AMTA*.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., and Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *LREC*.

Stolcke, A. (2002). SRILM – an extensible language modeling toolkit. In *ICSLP*.

Tiedemann, J. (2012). Parallel data, tools and interfaces in OPUS. In *LREC*.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., and Manning, C. (2005). A conditional random field word segmenter for SIGHAN bakeoff 2005. In *SIGHAN*.

Ture, F., Elsayed, T., and Lin, J. (2011). No free lunch: brute force vs. locality-sensitive hashing for cross-lingual pairwise similarity. In *SIGIR*.

Utiyama, M. and Isahara, H. (2007). A japanese-english patent parallel corpus. *MT Summit XI*.

Wang, K., Zong, C., and Su, K.-Y. (2013). Integrating translation memory into phrase-based machine translation during decoding. In *ACL*.

Wang, K., Zong, C., and Su, K.-Y. (2014). Dynamically integrating cross-domain translation memory into phrase-based machine translation during decoding. In *COLING*.

Wäschle, K. and Riezler, S. (2012). Analyzing parallelism and domain similarities in the MAREC patent corpus. In *IRFC*.

Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H. (2007). Online large-margin training for statistical machine translation. In *EMNLP*.

Zhechev, V. and van Genabith, J. (2010). Seeding statistical machine translation with translation memory output through tree-based structural alignment. In *SSST*.