Dynamic Translation Memory: Using Statistical Machine Translation to improve Translation Memory Fuzzy Matches

Ergun Biçici[1] and Marc Dymetman[2]

[1] Koç University (Istanbul, Turkey)
ebicici@ku.edu.tr
[2] Xerox Research Centre Europe (Grenoble, France)
marc.dymetman@xrce.xerox.com

Abstract. Professional translators of technical documents often use Translation Memory (TM) systems in order to capitalize on the repetitions frequently observed in these documents. TM systems typically exploit not only complete matches between the source sentence to be translated and some previously translated sentence, but also so-called *fuzzy matches*, where the source sentence has some substantial commonality with a previously translated sentence. These fuzzy matches can be very worthwhile as a starting point for the human translator, but the translator then needs to manually edit the associated TM-based translation to accommodate the differences with the source sentence to be translated. If part of this process could be automated, the cost of human translation could be significantly reduced. The paper proposes to perform this automation in the following way: a phrase-based Statistical Machine Translation (SMT) system (trained on a bilingual corpus in the same domain as the TM) is combined with the TM fuzzy match, by extracting from the fuzzy-match a large (possibly gapped) bi-phrase that is dynamically added to the usual set of "static" bi-phrases used for decoding the source. We report experiments that show significant improvements in terms of BLEU and NIST scores over both the translations produced by the stand-alone SMT system and the fuzzy-match translations proposed by the stand-alone TM system.

1 Introduction

Translation Memory (TM) systems [1, 2] have become indispensable tools for professional translators working with technical documentation. Such documentation tends to be highly repetitive, due to several factors, such as multiple versioning of similar products, importance of maintaining consistent terminology and phraseology, and last but not least, simplification of the translation process itself. TM systems typically exploit not only *complete matches* between the source sentence to be translated and some previously translated sentence, but also so-called *fuzzy matches* [3], where the source sentence has some substantial commonality with a previously translated sentence. These fuzzy matches can be extremely useful as a starting point for a human translator, but the translator then needs to manually edit the associated TM-based translation in order to accommodate the differences with the source sentence under translation. Typically, the amount of editing required depends on the *fuzzy-match level*, which can be defined as the percentage of shared words between the new source sentence and the previously translated source sentence. Actually, some translation services use these fuzzy-match levels as a basis for estimating the cost of translations both to the freelance translators they employ and to their end-customers; below a certain fuzzy-match level (say 50%) the translation cost of a sentence depends only on the number of words, whereas for a sentence close to a complete match ($\sim$100% fuzzy match), the cost is typically a small fraction of this price.

If part of the manual editing for the TM-based translations could be automated, then the cost of translation could be significantly reduced. This paper proposes such an automation, and describes a hybrid TM-SMT system that works along the following lines. We start by training a phrase-based Statistical Machine Translation (SMT) system on the bilingual corpus associated with the TM. Then, to translate a given source sentence S, the system first retrieves a fuzzy-match S' for S from the TM along with its associated target sentence T', and then uses these for *biasing* the translation candidates produced by the SMT system towards translations "compatible" with the pair (S', T').

Our approach capitalizes on the ability of the underlying SMT system to use *non-contiguous*, or *gapped*, phrases, where the fuzzy-match pair (S', T') is used to dynamically augment the collection of bi-phrases normally used by the SMT system for translating S, by adding a bi-phrase (P_S, P_T), where P_S is a (possibly non-contiguous) common subsequence of S and S', and where P_T is the (possibly non-contiguous) subsequence of T' which is detected to be aligned with P_S.

Using this approach, our hybrid translation system achieves significant improvements in BLEU and NIST scores over both the translation proposed by the stand-alone SMT system and the translation proposed by the stand-alone TM-based system. While these results do not necessarily translate directly into increased usability by the human translators, they do suggest important potential cost reductions for users of Translation Memory tools.

The remainder of the paper is organized as follows. In Section 2 we briefly describe the specificities of the underlying phrase-based SMT system that we use. In Section 3 we detail the ways in which we detect and use fuzzy-matches in order to construct a dynamic bi-phrase which is then added to the "static" biphrases normally used by the SMT system. In Section 4 we present our experimental results. In Section 5 we describe related research, and finally we conclude.

2 The SMT system MATRAX

The phrase-based SMT system MATRAX [4], developed at Xerox, was used in all experiments. MATRAX is based on a fairly standard log-linear model of the

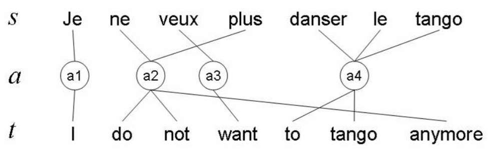

Fig. 1. A MATRAX translation making use of the set of bi-phrases $a = \{$*'Je'/'I'*, *'veux'/'want'*, *'danser le tango'/'to tango'*, *'ne ◊ plus'/'not ◊ ◊ ◊ anymore'* $\}$.

form:

$$\Pr(t, a|s) = 1/Z_s \exp \sum_{m=1}^{M} \lambda_m \phi_m(s, t, a),$$

where the notation is as follows: s is the source sentence to be translated, t a candidate target sentence, and a an "alignment" between the two, namely a decomposition of (s, t) into a set of "bi-phrases" (see Figure 1); the ϕ_m's are real-valued *features* each representing the assesment of a translation candidate (s, t, a) relative to some particular dimension, such as whether a is composed of high-probability bi-phrases (estimated on a large bilingual corpus), or whether the target sentence t is well-formed according to a certain language model, and so forth. The λ_m's are weights balancing the contributions of each aspect, trained from a small bilingual development corpus, and finally Z_s is a normalization factor. When given a test source sentence s to translate, the so-called *decoder* attempts to find the pair (t, a) which maximizes $\Pr(t, a|s)$ ["Viterbi decoding"], and outputs the translation t.

One original aspect of MATRAX is the use of non-contiguous bi-phrases. Most existing phrase-based models (see [5, 6]) depend on phrases that are sequences of contiguous words on either the source or the target side (e.g. *'prendre feu'* / *'catch fire'*). By contrast, MATRAX considers pairs of non-contiguous phrases, such as *'ne ... plus'/'not ... anymore'*, where words in the source and target phrases may be separated by gaps, to be filled at translation time by lexical material provided by some other such pairs. One motivation behind this approach is that, basically, the fact that the source expression *'ne ... plus'* is a good predictor of *'not ... anymore'* does not depend on the lexical material appearing inside the source expression, an insight which is generally unexploitable by models based on contiguous phrases.

These bi-phrases are estimated on the basis of a large training corpus of aligned bi-sentences (s, t). As a first step in this process, the training procedure invokes GIZA++ [7], which has the effect of producing a matrix describing probable *word-level* alignments between the sentences s and t (see section 3.5 for details about this process). Then, through a certain technique of "non-negative matrix factorization" [8], words that show strong affinities are grouped together into bi-phrases. One particularity of this approach is that it inherently produces gapped bi-phrases. These bi-phrases are then stored in a bi-phrase database, that we will refer to in the sequel as the "static bi-phrase library", along with some intrisic features of each bi-phrase $(\tilde{s}, \tilde{t})$, the most important of which is ϕ_{phr}, which is an estimate of the conditional probability $\Pr(\tilde{t}|\tilde{s})$.

3 Dynamic Translation Memory (DTM)

3.1 Experimental Setup

Before going into the details of our hybrid SMT-TM system, *Dynamic Translation Memory (DTM)*, it will be convenient to introduce our experimental setup. We start from a Background Translation Memory (TMB) data base provided to us by Xerox's GKLS (Global Knowledge and Language Services), a branch of the company that produces translations for internal and external customers. The translation memory tool used is TRADOS [9] and the TMB contains around 400,000 English-to-French translation units (typically sentences) in the automotive domain.

We start by randomly partitioning the translation memory into two subsets TM_1 and TM_2, representing respectively 90% and 10% of the translation units in the TMB. TM_1 will from now on be our "effective" translation memory, while TM_2 will serve as a reference for testing the performance of the different systems.

More precisely, we will consider various subsets TM_{FML}, where TM_{FML} is the set of bi-sentences (S, T) in TM_2 such that S has a fuzzy-match *relative to* TM_1 at a certain Fuzzy-Match Level (FML) (e.g. TM_{90-95}). We will then take T as the reference translation, and compare the system performances relative to that reference, for different fuzzy-match levels.

3.2 DTM Overview

The overview of our hybrid system is given below:

1. We first take TM_1 as our bilingual corpus and train the SMT system MATRAX on it;
2. As a by-product of this training, we produce, for each bi-sentence (S', T') in TM_1, a word-alignment matrix where each word pair is associated with a number representing the strength of the association;
3. When given a test source sentence S to translate, we then:
   (a) Attempt to find a fuzzy-match S' for S among the source sentences in TM_1;
   (b) Compute a (possibly gapped) source phrase P_S such that P_S is a subsequence of both S and S';
   (c) Retrieve the word-alignment matrix precomputed for (S', T'), where T' is the target sentence associated with S' in TM_1;
   (d) Use the word alignment matrix to recover a (possibly gapped) target phrase P_T which: (i) is a subsequence of T', and (ii) is strongly aligned with P_S according to the word alignment matrix;

(e) Start calling the MATRAX decoder on S, which has the consequence of retrieving from MATRAX's static bi-phrase library a collection of (typically small) candidate bi-phrases for decoding S;
(f) Add to that collection the "dynamic" bi-phrase (P_S, P_T), and also assign to it a strong "weight" through its bi-phrase feature ϕ_{phr};
(g) Resume the decoder call with the updated bi-phrase collection, and obtain a translation T.

3.3 Example

We will now detail some aspects of this process, using as an illustration the translation of the test sentence S: *'Install the upper arm front retaining bolt in three stages.'* While this sentence is not found literally in TM_1, there is a fuzzy-match for it there (at a fuzzy match level of 82%), namely S': *'Install the lower arm front retaining bolt in two stages.'*

Anticipating on the procedure that we will describe next, here are the translation results for S obtained by three systems: TRADOS (stand-alone TM-based translation system), MATRAX (stand-alone SMT system), and MATRAX-DTM (our hybrid system):

TRADOS *'Poser la vis de fixation avant de bras inférieur en deux passes.'*
MATRAX *'Poser le bras supérieur avant vis de fixation dans trois passes.'*
MATRAX-DTM *'Poser la vis de fixation avant de bras supérieur en trois passes.'*

Here, while the MATRAX-DTM translation is correct, the TRADOS translation is identical with the target T' associated with the fuzzy match S' in TM_1 and does not account for the differences between *'lower'*/*'upper'* (*'inférieur'*/*'supérieur'*) or *'two'*/*'three'* (*'deux'*/*'trois'*); as for the MATRAX translation, while it does get the translations of *'upper'* and *'three'* correctly, it confuses the order of several words and also uses the preposition *'dans'* while in this context the preposition *'en'* is called for.

3.4 Fuzzy-matcher

While we initially thought of using TRADOS's internal fuzzy matcher to provide us with a fuzzy match S' for S, we finally preferred to implement our own matcher, for two main reasons: (i) this gives us greater flexibility, and in particular we use our matcher for producing a subsequence of the source sentence which is later useful for the dynamic bi-phrase extraction; (ii) TRADOS does not directly provide to the end-user the source sentences from the TM that it uses for producing approximate targets in case of fuzzy matches, but only the targets themselves (such as the one shown in the previous section); furthermore in certain cases TRADOS appears to be able to use two different translation units in order to produce the approximate targets, and in such cases it would not make sense to ask for *one* TRADOS fuzzy-match source S'.[3]

MATRAX-DTM initially indexes the translation units in TM_1 for faster retrieval, using an inverted word index. The candidates for the fuzzy match are selected among the first N (e.g. N = 100) source sentences in TM_1 having the largest number of words in common with the given source sentence, S, to be translated. However, while the first of these candidate sentences has by construction more words in common with S than any other sentence in the TM, it may not have these words in the same order as S. So, in a second pass, we perform a LCS (Longest Common Subsequence) procedure between S and each of the N candidates, and retain the S' for which the corresponding subsequence is the longest. The LCS procedure is the standard dynamic program for performing this task, but we modify it in order to allow control of the maximal size of gaps that are allowed to occur in the subsequence (which is taken to be 2 in the current experiments). Note that if N was taken to be the number of units in TM_1, the procedure would guarantee finding the best match according to the LCS criterion, but that taking N = 100 is only a reasonable heuristic approximation to that.

In the case of our example S, this procedure finds S': *'Install the lower arm front retaining bolt in two stages.'*, along with its translation T': *'Poser la vis de fixation avant de bras inférieur en deux passes.'* The procedure also identifies as the longest common subsequence between S and S' the sequence *'Install the* $\diamondsuit$ *arm front retaining bolt in* $\diamondsuit$ *stages .'*, where the symbol $\diamondsuit$ indicates a gap. This gapped word sequence will be used as the source phrase P_S of the dynamic bi-phrase (P_S, P_T) that we are going to build next.

In order to complete the process (that is, in order to compute P_T), we now need to consider the word alignment matrix associated with the pair (S', T').

3.5 Word-Alignment Matrix

As we explained earlier, we started the whole process by training MATRAX on TM_1, considered as an aligned bilingual corpus.[4]

One of the steps in this training consists in using GIZA++ [7] for producing word alignments for all the bi-sentences (S', T') in TM_1. This is done in the following way:

- Use GIZA++ to produce the top 100 forward word alignments for (S', T'), as well as the top 100 backward word alignments for (S', T'); as is well known the forward (resp. backward) alignments are non-symmetrical, as they only allow 1-to-n alignment configurations,.

[3] This is an instance of a certain "sophistication" of commercial TM-based systems, that we could easily emulate in DTM, by considering several S' instead of one.

[4] One difference between a TM and a bilingual corpus is that the TM does not accurately represent the actual statistics of duplicate source sentences in the translations actually produced, and thus using the TM as a bilingual corpus might diverge somewhat from the corresponding statistics of the domain, but this does not appear to be a serious problem.

– Combine these alignments in a $|S| \times |T|$ matrix of counts; this joint word alignment matrix is now "symmetrical" relative to source and target, with each entry being an integer between 0 and 200.

The alignment matrices thus obtained as a by-product of MATRAX training are stored along with the TM_1 translation units. At translation time, when retrieving the fuzzy-match pair (S', T'), we then also retrieve the associated alignment matrix shown in Figure 2.

NIL	poser	la	vis	de	fixation	avant	de	bras	infÃ©rieur	en	deux	passes	.	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	NIL
0	196	1	0	1	0	0	2	0	1	1	0	0	0	Install
0	1	179	2	2	1	1	3	2	2	3	2	2	0	the
0	1	2	2	2	1	1	2	2	183	2	1	2	1	lower
0	0	1	1	2	0	2	3	185	2	1	1	2	1	arm
0	0	0	0	2	2	189	2	2	2	1	0	0	0	front
0	0	1	4	90	184	2	4	2	2	1	1	2	1	retaining
0	0	2	182	3	3	2	3	2	2	2	2	2	1	bolt
0	1	3	1	2	1	1	3	1	2	176	2	1	0	in
0	0	0	1	1	1	1	1	1	1	2	182	2	1	two
0	0	1	1	2	2	1	2	2	2	1	2	182	2	stages
0	1	0	1	1	1	1	2	1	1	0	1	2	181	.

Fig. 2. Word alignment matrix for (S', T').

3.6 Bi-phrase extraction

Our task is now the following. We are given a gapped source phrase P_S, along with a word alignment matrix for the pair (S', T'), and we need to extract from that matrix a gapped target phrase P_T "maximally" aligned with P_S. The algorithm we use to do that is related to the competitive linking method of Melamed [10], and works in the following way:

1. We initialize the set TargetWords to $\emptyset$;
2. We start by identifying the cell C with the maximal count in the matrix;
3. If C is associated with a source word belonging to P_S, then we include the corresponding target word occurrence in TargetWords, otherwise we don't;
4. We eliminate all cells belonging to the same line as C from further consideration, as well as all cells belonging to the same column as C;
5. We now identify the cell C' with maximal count among the cells still under consideration in the matrix, and iterate the process until all cells have been eliminated;

6. We finally collect all the word occurrences belonging to TargetWords in their sequential order to produce a gapped sequence P_T.

Performing this procedure on our example, we obtain the bi-phrase (P_S, P_T): (*'Install the ♢ arm front retaining bolt in ♢ stages .'*, *'Poser la vis de fixation avant de bras ♢ en ♢ passes .'*)

This "dynamic" bi-phrase is then added to the collection of bi-phrases retrieved by Matrax from its static bi-phrase library. In order to favor the use of the dynamic bi-phrase over that of other "standard" bi-phrases which may be in competition with it, we assign a strong (a priori defined) value to its associated feature ϕ_{phr}, which is an estimate of the conditional probability $\Pr(P_T|P_S)$.[5] Standard decoding is then resumed with this extended collection of bi-phrases, producing the translation: *'Poser la vis de fixation avant de bras supérieur en trois passes.'*, which is seen by inspection to be composed from three bi-phrases: (P_S, P_T), plus the static bi-phrases (*'upper'*,*'supérieur'*) and (*'three'*,*'trois'*).

One interesting aspect of this example is the complexity of the reorderings at the level of word alignments in the translation unit (S', T'), as can be seen informally in Figure 3.

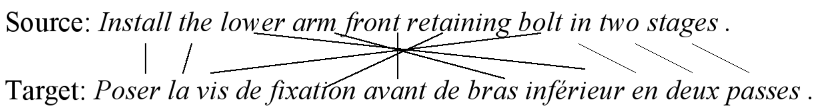

Fig. 3. A word aligned example bisentence.

Phrase-based SMT systems such as MATRAX are usually reluctant to generate such complex reorderings, as typically one of the feature functions used strictly controls the amount of reordering between biphrases of the source and targets sentences (i.e. the reordering feature function used in MATRAX [4]). The result of this reluctance can be seen in Figure 4 where the corresponding translations of MATRAX and MATRAX-DTM for the source sentence is given. As it is apparent from the translations, MATRAX is only able to simulate local organization whereas MATRAX-DTM can exploit global organization. The amount of reorderings allowed in MATRAX is simply not enough in this case and it is likely that increasing it will degrade the quality of the translations as it will also open possibilities for a soup of words. In such cases MATRAX-DTM is likely to perform better than MATRAX, because it can rely on larger bi-phrases which encapsulate complex re-ordering constraints.

---

[5] In the current implementation, the value of that feature is fixed arbitrarily, once for all for all dynamic bi-phrases, and the feature is not distinguished from the standard feature ϕ_{phr} used for static bi-phrases; A more principled approach would consist in having a specific $\phi_{phr'}$ feature for the dynamic bi-phrases, for which the associated log-linear weight would be trained independently from the weight of the standard feature.

Test Source: *Install the upper arm front retaining bolt in three stages .*

MATRAX Translation: *Poser le bras supérieur avant vis de fixation dans trois passes .*

MATRAX-DTM Translation: *Poser la vis de fixation avant de bras supérieur en trois passes .*

Fig. 4. Corresponding translations for a given source sentence.

4 Experimental Results

We present in Table 1 the results we have obtained for different fuzzy-match levels.

Table 1. Translation performance for different fuzzy-match levels.

		Translation Systems		
Level of Fuzzies	**Score**	**MATRAX**	**MATRAX-DTM**	**TRADOS**
30 − 40	NIST	6.8033	6.9568	3.2830
	BLEU	0.1944	0.2118	0.0905
40 − 45	NIST	7.0222	7.2083	3.9443
	BLEU	0.2307	0.2527	0.1307
50 − 74	NIST	8.5695	9.4795	6.5578
	BLEU	0.3181	0.4015	0.2652
74 − 85	NIST	9.2462	10.9127	8.7225
	BLEU	0.3951	0.5504	0.4154
80 − 95	NIST	9.7141	12.3632	10.1338
	BLEU	0.4301	0.6597	0.5118
90 − 95	NIST	9.4954	12.2955	10.2812
	BLEU	0.4478	0.7140	0.5633
95 − 100	NIST	9.2833	12.7308	11.8645
	BLEU	0.4053	0.7106	0.6516
30 − 100	NIST	9.4000	11.8356	9.5668
	BLEU	0.3672	0.5572	0.4410

The results show that for all fuzzy-match ranges, the DTM system MATRAX-DTM performs markedly better than both the SMT system and the TM-based system. We plot the NIST [11] and BLEU [12] performances of the three systems for different fuzzy match levels respectively in the first and second panels of Figure 5.

In both of these panels, it is again visible that MATRAX-DTM performs much better than both MATRAX and TRADOS. We also observe that TRADOS' performance is inferior to MATRAX's on lower-order fuzzies but it catches up in roughly the range $50-80$ and performs better for higher-order fuzzies. We also see that TRADOS is a little better than MATRAX for the overall range $30-100$, in terms of both the NIST and the BLEU performance, and that the difference is larger for the BLUE scores; here the difference between BLEU and NIST may be due to the fact that MATRAX is optimized using the NIST scoring function, thus possibly advantaging NIST scores at test time.

In the third panel of Figure 5, we finally show a comparison of MATRAX-DTM with the other two systems, in terms of relative percentage gains in terms of BLEU and NIST. We see that with increasing fuzzy match levels, the difference between the performance of MATRAX-DTM and that of MATRAX goes up whereas it goes down relative to that of TRADOS.

5 Related Work

Several researchers have proposed ways to associate Translation Memories and Statistical Machine Translation systems. One of the early works in this area is [13], whose authors used an IBM-4 SMT model to *construct* a "Translation Memory" of source/target subsentential segments from a large bilingual corpus. Their adapted decoder then exploits these segments to produce better translations than available to the original word-based decoder. Today, this system would probably be recognized more as an early manifestation of phrase-based translation (the TM segments constructed in [13] would now be called "bi-phrases"!) than as an approach for improving the usability of standard Translation Memories. In a somewhat similar spirit, [14] also produce subsentential bi-segments, but this time using an Example-Based approach which attempts to detect useful source and target segments based on a syntactic chunker, and then using these bi-segments in a statistical decoder. The authors of a more recent article [15] experiment with different combinations of EBMT (Example-Based Machine Translation) and Phrase-Based SMT (PB-SMT) techniques, focussing on the potential gains of jointly using bi-segments obtained both by EBMT and by PB-SMT techniques, whether the translations are actually produced using an EBMT or a PB-SMT decoder. While by and large the papers just cited concentrate more on ways to extract reliable subsentential segments from a bilingual corpus, in [16], the authors are somewhat closer to the spirit of the present paper as they focus on improving the translation of close matches (one word edit-distance at most) between a test source sentence and a TM sentence, where the TM is a collection of short sentences in a "Basic Travel Expression Corpus". When such a close match is found, phrase-based techniques are used for computing the translation of the replacer-word as well as for identifying the subsegment of the TM target, corresponding with the source word to be replaced, which should be overwritten with that translation. When no such close match is found in the TM, then the test sentence is given to a standard PB-SMT system for translation. While the paper reports no significant improvement of the combined system relative to the PB-SMT system in terms of BLEU and NIST, it does report some improvements in terms of human evaluation.

6 Conclusions

We have presented the hybrid TM-SMT system DTM which is able to improve the translation of sentences having a TM fuzzy match by dynamically extracting from the fuzzy match a highly predictive bi-phrase which is added to the collection of static bi-phrases used by the underlying SMT system. The approach relies on the fact that the underlying SMT system handles non-contiguous phrases on a par with standard contiguous phrases, and so is able to accommodate the new dynamic bi-phrase, which is typically large and non-contiguous.

We describe experiments, performed on texts from a large industrial TM data base, that show that the DTM system considerably improves the BLEU and NIST scores compared to both the underlying SMT and TM-based systems, for all fuzzy-match ranges, with the SMT component of DTM being especially useful at low fuzzy-match levels, and its TM fuzzy-match component being especially useful at high fuzzy-match levels. We believe that these results suggest useful potential applications of this technology for professional translators in technical domains, for whom the use of Translation Memory tools is now commonplace.

Acknowledgments Thanks to Brian Hudson and Felix Ustorf at GKLS for providing access to the translation memory used in the experiments and for their useful advice.

References

1. Zetzche, J.: Translation memory: state of the technology. Multilingual **18** (2007) 34–38
2. Lagoudaki, E.: Translation memories survey 2006. Technical report, Imperial College London (2006)
3. Sikes, R.: Fuzzy matching in theory and practice. Multilingual **18** (2007) 39–44
4. Simard, M., Cancedda, N., Cavestro, B., Dymetman, M., Gaussier, É., Goutte, C., Yamada, K., Langlais, P., Mauser, A.: Translating with non-contiguous phrases. In: HLT/EMNLP. (2005)
5. Och, F.J., Ney, H.: The alignment template approach to statistical machine translation. Computational Linguistics **30** (2004) 417–449
6. Lopez, A.: A survey of statistical machine translation. Technical report (2007) University of Maryland technical report, UMIACS-TR-2006-47.
7. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics **29** (2003) 19–51
8. Goutte, C., Yamada, K., Gaussier, E.: Aligning words using matrix factorisation. In: ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics. (2004) 502–509
9. Trados: Trados translator's workbench (2007) URL: http://www.translationzone.com/en/products/sdltrados2007/.
10. Melamed, I.D.: Models of translational equivalence. Computational Linguistics **26** (2000) 221–249

11. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the second international conference on Human Language Technology Research, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2002) 138–145
12. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2001) 311–318
13. Marcu, D.: Towards a unified approach to memory- and statistical-based machine translation. In: Proceedings of ACL '01. (2001)
14. Langlais, P., Simard, M.: Merging example-based and statistical machine translation: an experiment. In: Proceedings of AMTA '02. (2002)
15. Groves, D., Way, A.: Hybrid data-driven models of machine translation. Machine Translation **19** (2005) 301–323
16. Hewavitharana, S., Vogel, S., Waibel, A.: Augmenting a statistical translation system with a translation memory. In: Proceedings of EAMT '05. (2005)

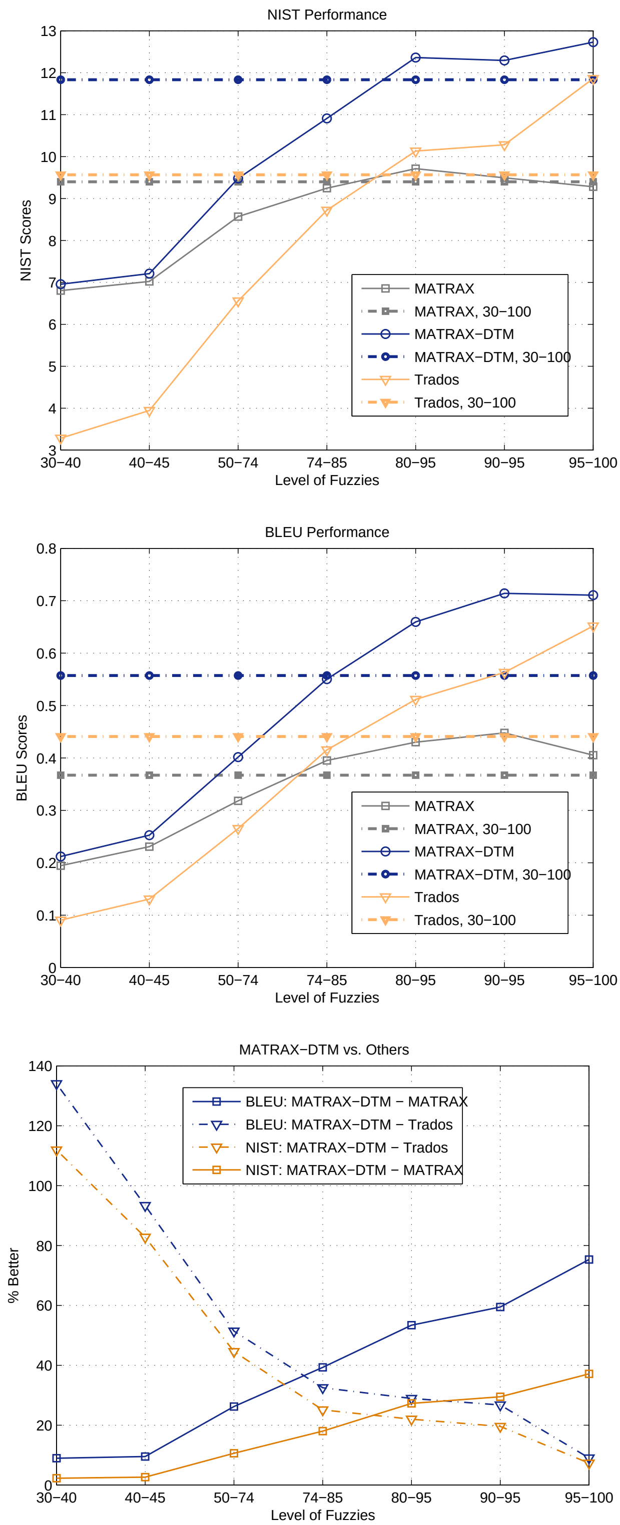

Fig. 5. The first two panels show NIST and BLEU performance of the three systems for different fuzzy match levels; the third panel compares the performance of MATRAX-DTM relative to each of the other two systems.