A Poor Man's Translation Memory Using Machine Translation Evaluation Metrics

Michel Simard
National Research Council Canada
283 Alexandre-Taché
Gatineau QC Canada J8X 3X7
michel.simard@nrc.ca

Atsushi Fujita
Future University Hakodate
116-2 Kameda-nakano-cho,
Hakodate, Hokkaido, 041-8655, Japan
fujita@fun.ac.jp

Abstract

We propose straightforward implementations of translation memory (TM) functionality for research purposes, using machine translation evaluation metrics as similarity functions. Experiments under various conditions demonstrate the effectiveness of the approach, but also highlight problems in evaluating the results using an MT evaluation methodology.

1 Introduction

A translation memory (TM) is a computer application for assisting translators, which manages a database of translations, i.e. a collection of pairs of corresponding segments in the source and target languages. Given a new segment of text to translate, the system looks up the database for exact or approximate ("fuzzy") matches; if the search is successful, the system returns the target-language version of the best match (or matches), which the user is then free to keep, modify or discard, as fit.

Although TM systems have been commercially available since the 1980's, they have not been the object of much attention from the research communities. In practice, many researchers in the field of machine translation (MT) tend to view the TM as a low-tech expedient, doomed to be replaced by MT systems sooner or later. Yet, as a computer-assisted translation (CAT) tool, the TM has a number of advantages over MT. For one, under normal circumstances, the output of a TM is always a fluent, human-quality translation. It may not be the translation of the right sentence, but it is generally a valid translation of something (which is more than most MT can claim). To determine to what extent the TM proposal is appropriate, the translator can examine the source sentence of which it is the translation and evaluate how it differs from the current sentence (some tools actively assist the user in this task by highlighting differences and similarities). Furthermore, meta-information is typically attached to translations in the database, allowing the user to determine where it comes from, who produced it, and from there establish whether or not it is reliable. In other words, the TM technology is one that users can understand and trust. As a result, TMs have succeeded where most of MT has failed so far: in establishing themselves as a must-have item in translators' toolboxes.

Recently, however, this situation appears to be changing, as we see a surge of interest for MT among the translation community. This renewed enthusiasm is fueled in part by the increased quality of the output of MT systems, but also by the availability of reliable free software, most notably the Moses system (Koehn et al., 2007). Yet, because of the inherent qualities of TM outlined above, and their solid entrenchment in the translation community, it is unlikely that MT will completely replace TM, at least not in the near future. Instead, we will probably see them co-exist for some time. This phenomenon is already visible, as many work environments for translators now incorporate both TM and MT technology. See, for example, the products of SDL Trados[1], Multicorpora[2] or the Google Transla-

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

[2] http://www.multicorpora.com

tor Toolkit[3].

As researchers tackle the question of properly integrating the two technologies, TMs themselves become objects of scientific inquiry. One of the obstacles to this however is that, because TM technology was essentially developed by industrial actors, nobody really knows how they work. In particular, the details of the similarity function at the heart of TMs are well-kept commercial secrets.

One possibility when experimenting with TM functionalities is to rely on one of the commercial implementations, and treat the TM as a "black box" component. But this is unwieldy because in most commercial systems, the TM functionality is only accessible through a complex GUI; furthermore, this raises issues with regard to reproducibility of experiments. The alternative is to build a TM system from scratch. Based on hints about the inner workings of TMs (Baldwin, 2001; Somers, 2003), some researchers have developed their own in-house implementations (Simard and Isabelle, 2009; Koehn and Senellart, 2010a; He et al., 2010).

In what follows, we propose a straightforward approach to building and evaluating baseline research TM systems, based on well-known methods and widely available tools. In particular, our proposal hinges on MT evaluation metrics and software, which we use not only to assess the effectiveness of our TM systems, but also as active components in the implementation: our plan is to use MT evaluation metrics as TM similarity functions.[4]

It should be pointed out that we do not aim at building a full-blown, operational TM environment. In particular, at this stage, we are not concerned with time or space requirements. Our goal is merely to provide researchers with the necessary machinery to experiment with TM technology. We therefore focus exclusively on the basic TM functionality: finding the best matching segments from a database of existing translations.

We begin by formalizing the notion of translation memory in Section 2, present the MT evaluation metrics we use in Section 3 and how we use them in Section 4; we then present experimental results under different conditions in Section 5.

2 Translation Memories

Conceptually, a translation memory consists of:

- a **database** D, containing pairs $\langle s, t\rangle$, where s is source-language segment of text (typically a sentence) and t is its translation in the target language;
- a **similarity function** f; and
- a **filtering threshold** α

Given a new sentence to translate q (the *query*), the system searches D for best match, i.e. the pair $\langle \hat{s}, \hat{t}\rangle$ whose similarity $x = f(q, \hat{s})$ is maximal; if $x \geq \alpha$, then the system outputs the target-language counterpart $\hat{t}$ of $\hat{s}$, otherwise nothing.[5]

Function f measures the similarity between two source-language strings. Typically, it produces a value between 0 and 1, where 0 means "completely different" and 1 means "identical"; α can then be in the range $[0, 1]$.

While we do not know exactly how f is implemented in commercial systems, it is generally acknowledged that most systems are based on variants of the Levenshtein distance (minimum number of edit operations required to transform one string into the other), normalized over the length of the query (Baldwin, 2001; Somers, 2003); for example:

$$f_{\textit{Levenshtein}}(q, s) = 1 - \min\left[1, \frac{\text{count\_edits}(q, s)}{|q|}\right]$$

The *count_edits()* function and the length of q can be computed over characters or words; Equal weights are normally assigned to all edit operations (insertions, deletions and substitutions), but these can be changed. Variants also exist that take into account local inversions of single items (character or word "swaps").

3 MT Evaluation Metrics

Since the advent of the BLEU MT evaluation metric, countless proposals have been made for automatically evaluating the quality of MT output. Most

[3] http://translate.google.com/toolkit

[4] Interestingly, MT evaluation metrics have also recently been used for paraphrase identification (Madnani et al., 2012).

[5] In practice, many different source segments s_i can be equally similar to q, and each s_i can be paired with multiple translations t_{ij} in D. For simplicity, we assume that when this happens, one of these pairs is selected at random.

(if not all) existing metrics rely on some measure of similarity between MT output t and one or more reference translations r (for simplicity, in what follows, we consider single-reference evaluation only). The more the MT output resembles the references, the better the score. In this study, we focus on four well-known MT evaluation metrics: WER, BLEU, NIST and Meteor. We briefly review these here:

WER (Word Error Rate): This metric has been used extensively in speech recognition. It is computed as the word-based Levenshtein distance between t and r, divided by the number of words in the reference: $|r|$. Several variants exist, most notably **TER** (Translation Edit – or Error – Rate), in which local swaps of sequences of words are allowed. TER itself has two variants: **HTER**, in which the reference translation is manually produced by minimally post-editing the MT output under evaluation (Snover et al., 2006); and **TERp**, which also relies on a table of paraphrases to detect semantically equivalent matches (Snover et al., 2009).

BLEU: (Papineni et al., 2002) The mother of all MT evaluation metrics: BLEU measures n-gram precision, i.e. the proportion of word n-grams of t that are also found in r. These n-gram precisions p_n are calculated separately for values of n ranging from 1 to N (typically $N = 4$), and then combined using a geometric mean. The score is scaled by a *brevity penalty* if the candidate translations are shorter than the references, $\mathrm{BP} = \min(1, e^{1-\frac{|r|}{|t|}})$.

$$\mathrm{BLEU}_N = \mathrm{BP} \cdot \exp\left[\frac{1}{N}\sum_{n=1}^{N}\log p_n\right] \qquad (1)$$

NIST: (Doddington, 2002) A variant of BLEU, in which n-gram precisions are averaged with harmonic rather than geometric mean; it also uses a slightly different brevity penalty

$$\mathrm{BP} = \exp\left[\beta \log \min(\frac{|t|}{|r|}, 1)\right] \qquad (2)$$

and, more importantly, weights n-gram matches by how informative they are; the informativeness of an n-gram $w_1...w_n$ is estimated as $\log \frac{\mathrm{count}(w_1...w_{n-1})}{\mathrm{count}(w_1...w_n)}$ where counts are typically obtained from the reference translations.

Meteor: (Denkowski and Lavie, 2011) Based on a one-to-one alignment between words of t and r, giving preference to alignments with less crossing alignments, Meteor computes unigram precision P and recall R, which are then combined in a weighted harmonic mean, $F_\alpha = PR/(\alpha P + (1-\alpha)R)$, and scaled by a reordering penalty, which counts the number of chunks t and r would need to be broken into to allow them to be rearranged with no crossing alignments, $P_{\beta,\gamma} = 1 - \gamma(\text{chunks}/\text{matches})^\beta$.

$$\text{Meteor}_{\alpha,\beta,\gamma} = F_\alpha \times P_{\beta,\gamma}$$

Word-alignments are not restricted to surface-similar forms, as Meteor can rely on a lemmatiser and other linguistic resources (Wordnet and paraphrase tables) to account for semantic equivalence. In this study, we experiment with this metric used in two different modes[6]:

- without any linguistic resources; we refer to this as **Vanilla-Meteor** (or VMeteor). In this mode, the metric behaves more like its earlier versions (Banerjee and Lavie, 2005).
- with all linguistic resources; we refer to this simply as **Meteor**.

4 MT Evaluation Metrics as TM Similarity Functions

Since automatic MT evaluation methods are based on text-similarity metrics, it seems natural to use them in TMs as well. The idea is to replace the reference r and the system output t in the evaluation metrics (Section 3) with the TM's query q and source-language segment s (Section 2), respectively.

The four evaluation metrics described above are well-known to the MT community, and represent different perspectives on textual similarity. Given the large number of existing metrics, we clearly could have tested many others. Ours nevertheless seemed like a natural choice.

First, the motivation for using a Levenshtein-based MT evaluation metric such as WER as TM similarity function is quite straightforward: as far as anyone knows, this is essentially what is already

[6] In all our experiments, Meteor is used with parameter values recommended for "ranking" tasks.

used in commercial TMs. Therefore, WER serves as a sort of baseline for our work.[7]

Then, n-gram-based metrics such as BLEU are interesting in their own right, because they take a different view to text similarity. According to Papineni et al. (2002), lower-order n-grams account for *adequacy* in MT evaluation, while higher-order n-grams account for *fluency*. Adequacy and fluency are classical measures of MT quality (White et al., 1993); such an approach intuitively also makes sense in a TM application. Also, the brevity penalty (BP) guarantees that the proposed translations will be of length comparable to that of the query.

WER and BLEU measure similarity in a "flat" way: all edit operations in WER have the same weight; all n-grams in BLEU are counted as equal. The NIST metric introduces n-gram weighting, which is intended to reflect how informative each n-gram is. In principle, it allows placing more emphasis on content (or adequacy) than on form (fluency). From the point of view of TM, this is a departure from pure surface similarity, and closer to the idea of *relevance*, as defined for information retrieval.

Vanilla-Meteor balances precision against recall, thus eliminating the need for a brevity penalty; furthermore, its non-Vanilla form is more "linguistically-informed" than other metrics discussed here, taking us even further away from surface resemblance, and in the direction of *semantic* similarity, through the use of stemming, WordNet and paraphrases.

Our implementation of the TM functionality is based on a straightforward, exhaustive search strategy: we compare each query q against the source segment of *all* pairs $\langle s, t \rangle$ in the database D, and output the one with the highest similarity. Much more efficient implementations are of course possible, based on a two-pass strategy, where an efficient search method – for example (Koehn and Senellart, 2010b) – produces a reduced set of candidates, which are then reranked using one of the similarity functions proposed here.

Open-source implementations exist in the public domain for all our MT evaluation metrics. Using these directly within an exhaustive TM search is certainly not optimal, but it is usually feasible. For instance, the publicly available Meteor software has options `-nBest` and `-oracle`, which allow to compare the reference (in our case, the query) to multiple system translations (in our case: candidate TM source-language matches) and output the one that produces the highest score. We used that implementation of Meteor for our experiments, but produced our own implementations of BLEU, NIST and WER. For reference, finding the best match for a sentence in the ECB corpus (see Section 5.1 for details) takes approximately 0.5 second using our BLEU-based TM implementation, 1.2 seconds for NIST and 1.5 for WER. The same operation takes 43 seconds using Vanilla-Meteor, and almost 5 times longer with Meteor (210 seconds); this difference can be explained by the time required to load linguistic data (paraphrase table, etc.).

In practice, we found that better results were obtained by measuring similarity over lower-cased texts. The differences with true-cased texts are minimal, however; in the end, it's probably a matter of user-preference. In all that follows, we assume lower-cased source-language texts[8].

Both BLEU and NIST are designed to evaluate document-level MT quality, and are not well adapted to finer-grained evaluation; for example, if t and r do not have at least one 4-gram in common, then the product in Equation (1) goes to zero, and therefore the whole BLEU score. To compensate for this problem, it is common to use a "smoothed" version of the score, in which 1 is added to all n-gram counts (Lin and Och, 2004).

BLEU and Meteor naturally produce scores comprised between 0 and 1, and can be used directly as similarity functions in a TM. NIST and WER are not as well-behaved: WER can produce scores larger than 1, when the number of edits required to convert t into r (or s into q) is larger than the number of words in r. In a TM setting, where such matches are unlikely to be useful, this problem is easily remedied by capping the value at 1. Also, because WER

[7] TER would arguably have been a better choice, being a well-established metric in the MT community. In practice, however, when attempting to use existing TER implementations for our purposes, we ran into a number of technical difficulties; furthermore, WER and TER are known to behave very similarly (Cer et al., 2010).

[8] Note, however, that all target-language evaluations were performed on true-cased texts.

is a distance metric rather than a similarity metric, we take 1 minus that value.

$$f_{\mathrm{WER}}(q, s) = 1 - \min(1, \mathrm{WER}(s, q))$$

NIST also produces values that are unbounded in the positive. The solution we propose is to divide the outcome by the largest possible obtainable value for the current query, i.e. the value that would be produced by an exact match:

$$f_{\mathrm{NIST}}(q, s) = \frac{\mathrm{NIST}(s, q)}{\mathrm{NIST}(q, q)}$$

5 Experiments

5.1 Data

We performed experiments to assess the performance of each MT evaluation metric as TM similarity function. Experiments were done under different conditions, corresponding to different corpora and language pairs. While most of the metrics we use are language-agnostic, Meteor relies on language-specific resources which are not available in all languages. For this reason, we restricted our experiments to English, French, German and Spanish.

Our four datasets are drawn from the Europarl v.6 (Koehn, 2005), OPUS corpus (Tiedemann, 2009) (corpora ECB, featuring content from the *European Central Bank* and EMEA, from the *European Medicines Agency*) and the JRC-Acquis v.2.2 (Steinberger et al., 2006). All bilingual corpora are available aligned at the sentence level. From each corpus, we randomly sampled 1000 pairs of segments, to be used as test data; the rest was used to build translation memories. All experiments were performed "in-domain", i.e. for any given experiment, test and TM data always come from the same corpus. Table 1 provides additional details.

Corpus	Lang.	TM ("Train")		Test
		segments	words	words
Europarl	en-fr	1.8M	50.4M	28 817
	en-es	1.8M	49.2M	28 365
	en-de	1.7M	48.0M	26 715
ECB	en-fr	194k	5.7M	30 471
	en-es	114k	3.1M	28 054
	en-de	111k	3.0M	27 426
EMEA	en-fr	753k	9.1M	16 514
JRC-Acquis	en-fr	329k	6.9M	19 260

Table 1: Experimental Data

5.2 Evaluation Methodology

Little attention has so far been devoted to the evaluation of TM systems. Gow (2003) proposes a general methodology, but which is very much user-centered, and that focuses on complete CAT environments rather than specifically on the TM functionality; Baldwin and Tanaka (2000) and Whyman and Somers (1999) both propose methods that are based on recall and precision, but these have not been widely used or evaluated.

Instead, because our work primarily focuses on integration with MT systems, we opt for the approach proposed in Simard and Isabelle (2009), in which TM systems are evaluated as if they were MT systems: test sentences are submitted to the TM, with the filtering threshold α set to zero (Section 2), thus effectively inhibiting output filtering; the target segments of the best matches are then compared to the reference translations, using any standard MT evaluation metric. In practice, in this study, we use the metrics described in Section 3, i.e. the same metrics used as similarity functions.

5.3 Results

We tested each TM similarity function on each dataset, and measured performance using each MT evaluation metric. Table 2 summarizes the results of these experiments: for each evaluation metric and dataset, it reports the best performing similarity function. (For lack of space, we do not report the detailed outcome of each individual experiment, but results of all English-French experimental conditions can be seen in Table 4.)

The most striking aspect of these results is the direct link between the evaluation metric and the best similarity function for BLEU and WER: in general, if BLEU is used to measure performance, then BLEU comes out as the best similarity function, and likewise for WER. Vanilla-Meteor and Meteor behave somewhat similarly, always preferring one of the Meteor family. Interestingly, these two metrics always agree with one another, usually preferring Vanilla-Meteor when English is the source language and Meteor when English is target. The preferences of NIST are not as clear – we discuss this further in Section 5.4.

Corpus	**Language**	**Evaluation Metric**				
		WER	BLEU	NIST	VMeteor	Meteor
Europarl	en-de	WER	BLEU	BLEU	Meteor	Meteor
	en-es	WER	BLEU	NIST	VMeteor	VMeteor
	en-fr	WER	BLEU	NIST	VMeteor	VMeteor
	de-en	WER	BLEU	NIST	Meteor	Meteor
	es-en	WER	VMeteor	Meteor	Meteor	Meteor
	fr-en	WER	BLEU	NIST	Meteor	Meteor
ECB	en-de	WER	BLEU	BLEU	VMeteor	VMeteor
	en-es	WER	BLEU	BLEU	VMeteor	VMeteor
	en-fr	WER	BLEU	BLEU	VMeteor	VMeteor
	de-en	WER	BLEU	BLEU	Meteor	Meteor
	es-en	WER	VMeteor	VMeteor	Meteor	Meteor
	fr-en	WER	BLEU	BLEU	Meteor	Meteor
EMEA	en-fr	WER	BLEU	BLEU	VMeteor	VMeteor
JRC-Acquis	en-fr	WER	BLEU	BLEU	VMeteor	VMeteor

Table 2: Best performing TM similarity function $f(q, s)$ according to each evaluation metric, under all tested conditions.

To those with a background in statistical machine translation, who are familiar with the general approach of Minimum Error-Rate Training (Och, 2003), this may seem like a very natural outcome at first sight. After all, using any given metric as a similarity function is somewhat like optimizing the behavior of the system for that evaluation metric. The subtle difference here is that in a TM, similarity is measured in the source language. If we take the example of BLEU, this means that maximizing n-gram precision relative to the source-language query somehow results in the n-gram precision being maximized in the target-language as well.

One possible explanation lies in the way each metric accounts for length differences in the sequences under comparison. Figure 1 plots the length of individual queries against that of the source-language segment of the corresponding TM best match; it can be seen that BLEU and NIST (blue and green dots, respectively) tend to produce TM best matches whose source segment length is very close to that of the query (average length ratios are given in Table 3). This contrasts with WER (red dots), which naturally favors segments that are much shorter than the query, and with the Meteor metrics (purple and black), which tend to produce source segments that are much longer than the query. On the target side, shorter TM matches such as those produced by the WER similarity function will be penalized at evaluation time by the BLEU and NIST brevity penalties (BP – see eq. 1). Meteor's longer matches will naturally be penalized by precision-based evaluation metrics.

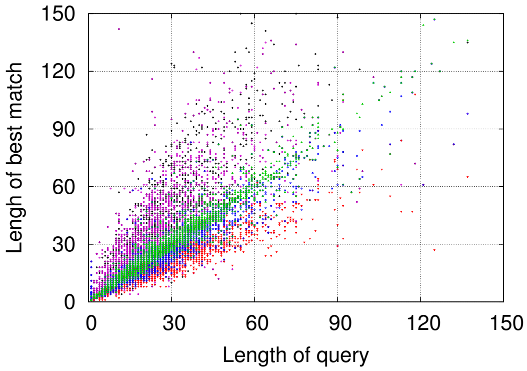

Figure 1: Length (in words) of TM best match source segment $\hat{s}$, as a function of length of query q (all English-French conditions), for each tested TM similarity functions: BLEU similarity function is reported in blue, NIST is in green, WER is in red, Vanilla-Meteor is in purple and Meteor is in black.

The Meteor metrics highlight another aspect of the relationship between similarity functions and evaluation metrics: using WordNet and paraphrase tables as additional resources to measure source-language similarity may allow retrieving better source-language matches from the TM; however, this will likely not come out in the evaluation, unless the evaluation metric also accounts for semantic relatedness in the target-language. As a similarity function, English-Meteor arguably exploits the

Similarity Function	Source Ratio	Target Ratio
WER	0.85 ± 0.18	0.88 ± 0.58
BLEU	1.01 ± 0.60	1.04 ± 0.91
NIST	1.03 ± 0.15	1.06 ± 0.63
Vanilla-Meteor	1.31 ± 0.85	1.43 ± 2.01
Meteor	1.36 ± 0.85	1.48 ± 2.07

Table 3: Macro average of length ratio of TM best match $\langle \hat{s}, \hat{t} \rangle$ relative to query q and reference r in all English-French experiments, for each tested TM similarity function. Source ratio is $|\hat{s}|/|q|$; Target ratio is $|\hat{t}|/|r|$.

richest set of linguistic resources to find semantic equivalents to the query. Intuitively, we expect the corresponding target-language segments output by the TM to share the same kind of relationship with the reference translations. However, if such equivalences are not realized into surface-similar tokens, they are unlikely to be picked up by the evaluation metric. This appears to be true even for French, Spanish and, to a lesser extent, German versions of Meteor, possibly because they rely on poorer linguistic resources, or because these resources are not "aligned" with those used for English.

The choice of a similarity function has a major effect on the overall performance of the translation memory, as measured using a machine translation evaluation methodology. For example, on the JRC-Acquis corpus, there is a difference of almost 10 BLEU points between the best and the worst performing functions (see Table 4). However, it is relevant to ask where and why these differences occur.

To better understand what is going on, we reintroduce the α filtering threshold (Section 2) and examine how TM performance varies as we filter out the segments for which good matches cannot be found in the TM. As α is set higher, less queries find matches in the TM and source coverage goes down; at the same time, performance on the filtered material improves. This can be seen in Figure 2, where we plot WER against source coverage for each TM similarity function, on English-French conditions. A striking feature of this graph is that performance differs the most at high-coverage levels, i.e. when TM outputs are proposed even for low-similarity matches. In a real-life TM application, weakly matching segments are seldom useful: they are queries for which the system manages to find a segment in the TM that has a few words in common with the query (most often function words), while being approximately the same size as the query. This is the kind of material that the translator typically does not want to see. By contrast, in the low-coverage areas, where only the best matching segments from the TM are retained, all metrics display very comparable performances.

$f(q,s)$	Evaluation Metric				
	WER	BLEU	NIST	VMeteor	Meteor
Europarl Corpus (en-fr)					
WER	**84.21**	6.85	1.857	13.71	15.28
BLEU	89.45	**9.85**	3.004	17.26	19.04
NIST	92.54	9.34	**3.088**	18.16	20.11
VMeteor	117.74	8.22	2.731	**20.69**	**22.93**
Meteor	130.20	6.88	2.425	19.84	22.45
ECB Corpus (en-fr)					
WER	**61.92**	38.93	7.026	45.43	45.63
BLEU	66.29	**42.05**	**7.426**	48.50	48.66
NIST	68.63	40.62	7.189	48.43	48.68
VMeteor	78.58	37.51	6.659	**49.73**	**49.88**
Meteor	80.11	36.42	6.480	49.49	49.77
EMEA Corpus (en-fr)					
WER	**75.28**	13.55	3.388	23.95	25.46
BLEU	81.60	**15.35**	**3.999**	25.69	27.39
NIST	84.59	14.63	3.934	26.19	28.02
VMeteor	95.28	14.63	3.867	**28.68**	**30.44**
Meteor	98.05	13.84	3.688	28.42	30.32
JRC-Acquis Corpus (en-fr)					
WER	**53.50**	39.70	6.137	48.80	48.93
BLEU	58.57	**42.47**	**6.731**	51.65	51.88
NIST	60.99	41.35	6.436	51.63	51.88
VMeteor	83.11	33.95	5.457	**52.46**	**52.70**
Meteor	84.79	33.02	5.326	52.03	52.45

Table 4: TM performance on English-French conditions, under each tested similarity function (rows) and evaluation metric (columns).

5.4 Discussion: Metric Tuning

As mentioned earlier, one notable exception to the direct relationship between the choice of a TM similarity function and the outcome in evaluation is NIST. In most test conditions, using NIST for similarity does not maximize output quality, as measured by the NIST metric (see Tables 2 and 4).

To better understand what is going on here, it is instructive to also examine *source-language performance*, i.e. measure the global similarity between the source language test set (the queries q) and the source segments $\hat{s}$ from the TM's best matches. This is shown in Table 5 for English-French conditions. Here, we expect the relationship between similar-

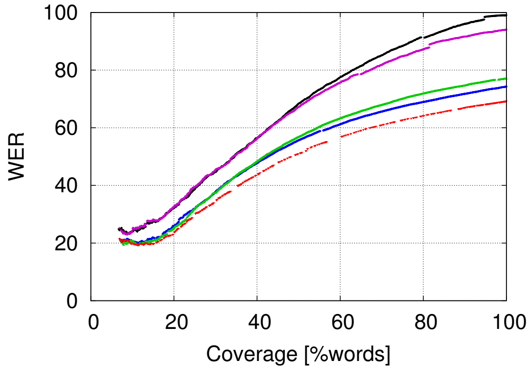

Figure 2: WER of filtered TM proposals (all English-French conditions), for each tested TM similarity functions: BLEU similarity function is reported in blue, NIST is in green, WER is in red, Vanilla-Meteor is in purple and Meteor is in black.

$f(q,s)$	Evaluation Metric				
	WER	BLEU	NIST	VMeteor	Meteor
Europarl Corpus (en-fr)					
WER	**65.80**	14.10	3.065	13.53	14.60
BLEU	78.84	**21.75**	**4.534**	16.07	17.27
NIST	83.69	16.31	4.495	17.61	18.84
VMeteor	112.63	12.82	3.649	**19.57**	20.84
Meteor	127.49	9.58	2.941	17.07	**22.37**
ECB Corpus (en-fr)					
WER	**41.06**	52.66	8.884	34.84	35.04
BLEU	48.48	**57.85**	**9.388**	37.33	37.49
NIST	50.13	54.41	9.057	36.91	37.21
VMeteor	63.34	49.82	8.259	**38.30**	38.47
Meteor	68.19	47.01	7.813	37.56	**38.52**
EMEA Corpus (en-fr)					
WER	**56.56**	22.89	4.514	19.60	20.33
BLEU	66.85	**28.80**	**5.465**	21.60	22.25
NIST	71.73	23.96	5.282	21.96	22.71
VMeteor	82.60	23.77	5.081	**24.04**	24.58
Meteor	87.47	21.41	4.665	22.79	**25.24**
JRC-Acquis Corpus (en-fr)					
WER	**41.84**	47.08	6.816	32.18	32.66
BLEU	49.53	**51.96**	**7.474**	34.28	34.75
NIST	52.51	48.86	7.156	33.76	34.39
VMeteor	68.13	43.12	6.372	**35.07**	35.50
Meteor	70.76	41.18	6.114	34.10	**35.86**

Table 5: Source-language TM performance on English-French conditions, under each tested similarity function (rows) and evaluation metric (columns).

ity function and evaluation metric to hold strictly; in practice, it does for all metrics, except NIST. The explanation lies in the fact that TM performance is measured globally, while the TM seeks to maximize similarity locally, at the segment level. Both NIST and BLEU behave non-compositionally, i.e. global scores can not be computed by combining local scores. Therefore, for these metrics, collecting locally maximal solutions does not guarantee a global optimum. In our experiments, BLEU does not appear to be globally affected by local optimization, but NIST is, possibly due to differences between geometric and harmonic means of n-gram precision, but also to its brevity penalty harshly penalizing the shorter matches.

NIST has a parameter β which can be adjusted to weight the relative importance of the brevity penalty in the score. By default, β is set so that BP $= .5$ when $|t|/|r| = 2/3$ (see eq. (2)). This setting may be optimal for maximizing correlation with human judgment of MT output, but it is not necessarily appropriate for the current task or datasets.

More generally, this suggests that it could make sense to "tune" MT metrics to fit the particular requirements of the TM task, or the final evaluation scheme. Most metrics have parameters that affect their behavior and can be changed: for example, for BLEU and NIST, it is possible to change the maximum n-gram size N over which precisions are computed; in WER, the cost of individual edit operations can be changed; Meteor has numerical parameters which are tuned for different applications.

All metrics under consideration here can be coerced into producing K-best lists of matches, which means that a general optimization scheme such as Minimum-Error Rate Training (MERT) could be applied in a relatively straightforward manner to optimize numerical parameters with regard to a given evaluation metric[9].

Meteor also relies on a paraphrase table to discover semantic similarities. One possible way of optimizing the performance of that metric as a TM similarity function is to provide it with domain-specific paraphrases. In a preliminary experiment along this line, we created domain-specific paraphrase tables from each TM using the technique described in (Fujita et al., 2012), and used these with Meteor instead of the provided table. In practice, in-domain paraphrases do not lead to measurable gains or losses

[9] Here, we overlook the integral nature of some parameters, such as N for BLEU and NIST, which raises problems for standard optimization techniques. Then again, using MERT to optimize N for BLEU is clearly overkill.

Example 1

Query	**This is** ***the process we*** *are* ***commencing.***
Meteor	I suggest that we perhaps continue *the work we have started.*
CMeteor	**This is the point** at which **we** must **start**.
WER	This is the stage we are at.
BLEU	This is the stage we are at.
NIST	This is the stage we are at.
VMeteor	We are in the process of revising this regulation.

Example 2

Query	A lysodren patient card **is included** ***at the end of this leaflet.***
Meteor	*At the end of this leaflet.*
CMeteor	Detailed instructions for subcutaneous injection **are provided at the end of this leaflet**.
WER	Listed at the end of this leaflet.
BLEU	Ingredients are listed at the end of this leaflet.
NIST	Listed at the end of this leaflet (see section 6).
VMeteor	At the end of this leaflet.

Figure 3: Examples of source segments from TM matches found using each tested similarity function. "Meteor" refers to Meteor using the standard paraphrase table (matching words in *italics*), while "CMeteor" is with in-domain paraphrases (matching words in **bold**).

in performance. It should be pointed out that measuring this sort of improvement is problematic: the in-domain paraphrases theoretically allow finding more useful matches in the TM, but the translation of these are often also realized as target-language domain-specific paraphrases, which are not properly acknowledged by the evaluation metrics. Nevertheless, the approach seems promising, and Figure 3 gives examples of queries for which the in-domain version (labeled *CMeteor*) provides matches that are potentially more useful for the translator than those found with other similarity functions.

6 Conclusion

We have shown how machine translation evaluation metrics can effectively be used as translation memory similarity functions. Each metric has its own characteristics and potential benefits.

In terms of efficiency, metrics based on n-gram precision such as BLEU and NIST are less computationally expensive than classic edit-distance-based metrics such as WER, or metrics that rely on linguistic resources, such as Meteor. In practice, they are easy to implement and produce results comparable to WER (on which existing commercial systems are believed to be based), especially in high-similarity situations, where it counts for real-life TM usage.

Most metrics can be tuned, to optimize performance of TM systems for specific text domains. Optimization methods commonly used in statistical machine translation could easily be adapted to this task. One related aspect that we have not yet examined is the combination of different metrics into a single similarity function (Liu and Gildea, 2007). In a similar vein, preliminary experiments suggest that customizing linguistic resources such as paraphrase tables could help in better leveraging the contents of the TM when appropriate metrics are used, such as Meteor or TERp. Extracting domain-specific paraphrases is one possible avenue, but in a TM perspective, it would be interesting to extend similarity to other semantic relations besides synonymy, e.g. antonymy, hyponymy, etc. These are areas we hope to explore further in the near future.

When evaluating the performance of TM systems using MT evaluation metrics, in general, we find that whichever metric is used as TM similarity function will likely obtain the best score under that evaluation metric. This suggests that existing MT evaluation metrics are not appropriate for evaluating TM performance. In fact, it is unclear whether it is actually possible to measure TM performance in an unbiased way using fully automatic methods. Human-based evaluation may well be the only credible alternative, and is what we plan to resort to in future experiments.

References

T. Baldwin and H. Tanaka. 2000. The Effects of Word Order and Segmentation on Translation Retrieval Performance. In *Proceedings of the 18th International Conference on Computational Linguistics*, volume 1, pages 35–41.

T. Baldwin. 2001. Low-cost, High-performance Translation Retrieval: Dumber is Better. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 18–25.

S. Banerjee and A. Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

D. Cer, C. D. Manning, and D. Jurafsky. 2010. The Best Lexical Metric for Phrase-Based Statistical MT System Optimization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 555–563.

M. Denkowski and A. Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*, pages 85–91.

G. Doddington. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the Second International Conference on Human Language Technology*, pages 138–145.

A. Fujita, P. Isabelle, and R. Kuhn. 2012. Enlarging Paraphrase Collections through Generalization and Instantiation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 631–642.

F. Gow. 2003. *Metrics for Evaluating Translation Memory Software*. Ph.D. thesis, University of Ottawa.

Y. He, Y. Ma, A. Way, and J. van Genabith. 2010. Integrating N-best SMT Outputs into a TM System. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 374–382.

P. Koehn and J. Senellart. 2010a. Convergence of Translation Memory and Statistical Machine Translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31.

P. Koehn and J. Senellart. 2010b. Fast Approximate String Matching with Suffix Arrays and A* Parsing. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180.

P. Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Tenth Machine Translation Summit*, volume 5.

C. Y. Lin and F. J. Och. 2004. Orange: a Method for Evaluating Automatic Evaluation Metrics for Machine Translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 501–507.

D. Liu and D. Gildea. 2007. Source-language Features and Maximum Correlation Training for Machine Translation Evaluation. In *Proceedings of the 2007 Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 41–48.

N. Madnani, J. Tetreault, and M. Chodorow. 2012. Re-examining Machine Translation Metrics for Paraphrase Identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190.

F. J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

M. Simard and P. Isabelle. 2009. Phrase-based Machine Translation in a Computer-assisted Translation Environment. In *Proceedings of the Twelfth Machine Translation Summit*, pages 120–127.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

M. G. Snover, N. Madnani, B. Dorr, and R. Schwartz. 2009. TER-Plus: Paraphrase, Semantic, and Alignment Enhancements to Translation Edit Rate. *Machine Translation*, 23(2):117–127.

H. Somers. 2003. Translation Memory Systems. *Benjamins Translation Library*, 35:31–48.

R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufiş, and D. Varga. 2006. The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 24–26.

J. Tiedemann. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In *Proceedings of Recent Advances in Natural Language Processing*, pages 237–248.

J. S. White, T. A. O'Connell, and L. M. Carlson. 1993. Evaluation of Machine Translation. In *Human Language Technology: Proceedings of a Workshop (ARPA)*, pages 206–210.

E. K. Whyman and H. L. Somers. 1999. Evaluation Metrics for a Translation Memory System. *Software-Practice and Experience*, 29(14):1265–84.