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

Michel Simard † Atsushi Fujita ‡

† National Research Council Canada
michel.simard@nrc.ca

‡Future University Hakodate
fujita@fun.ac.jp

FUTURE UNIVERSITY HAKODATE

The Goal

Implement the **core Translation Memory (TM) functionality**: given a source-language sentence, find the best matching translation in a bilingual corpus. Our goal is to build a reasonable **research** prototype, focusing on output quality rather than time/space efficiency.

Motivations

- Much recent work on post-editing and computer-assisted translation (CAT) ignores the fact that **most translators are already using CAT tools**, the most common of which are TM systems. In the short term, new tools are not going to supplant TMs, but rather complement them.
- The problem for researchers who wish to experiment with TM technology is that existing systems are big and unwieldy: their basic functionality is often concealed behind a thick layer of GUI, and their algorithms are not documented (they are trade secrets). This makes it **difficult to use commercial systems in a research setting**, or even to reverse-engineer their functionality.
- The obvious alternative is to **re-implement the TM functionality** from scratch.

Translation Memory

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 **core TM functionality** consists in finding the best match for q in D, i.e. the pair $\langle \hat{s}, \hat{t}\rangle$ with maximum similarity $x = f(q, \hat{s})$; if $x \geq \alpha$, then the system outputs the target-language counterpart $\hat{t}$ of $\hat{s}$, otherwise nothing.

Similarity 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]. It is generally acknowledged that commercial TM systems use **variants of the Levenshtein distance**, e.g.:

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

MT Evaluation Metrics

Implementing the core TM functionality requires that we come up with a **similarity function**. As it turns out, one sub-field of MT research that has churned out many such functions is **MT evaluation**: Many (if not all) of the MT evaluation metrics proposed in recent years rely on measuring the similarity between a machine translation output and one or more reference translations. In this study, we examine five different evaluation metrics:

- **WER** – Word-error rate is based on **word-level Levenshtein distance**. As far as anyone knows, this is essentially what is used in commercial TM systems, and serves as **baseline** for this study.
- **BLEU** – Papineni et al. (2002) : based on ***n*-gram precision**, it implements the idea of accounting separately for *adequacy* (low-order *n*-grams) and *fluency* (high-order *n*-grams).
- **NIST** – Doddington (2002) : Adds a notion of IR-style **relevance** to the mix.

We also consider Meteor, under two different conditions:

- **VMeteor** ("Vanilla" Meteor) – Banerjee & Lavie (2005): considers **lexical recall**, while de-emphasizing match length.
- **Meteor** – Denkowski & Lavie (2011) : linguistic resources (stemmer, WordNet, paraphrases) bring us closer to **semantic similarity**.

Implementation

We implement an **exhaustive search strategy** : for every pair $\langle s, t\rangle$ in the TM D, measure the similarity between q and s, using MT evaluation metric X's similarity function:

$$\langle \hat{s}, \hat{t}\rangle = \underset{\langle s,t\rangle \in D}{\arg\max}\, f_X(q, s)$$

- query q is used in place of the reference translation and s is used in place of the machine translation output.
- BLEU and NIST do not behave well when applied to single sentences: we use smoothed versions of these functions, as in Lin & Och (2004).
- NIST and WER do not produce values strictly between 0 and 1, their values needs to be normalized.

While it is sometimes possible to use public domain MT evaluation software directly (e.g. Meteor, VMeteor), it is often easier and more efficient to reimplement the distance functions based on the published descriptions (BLEU, WER and NIST).

Evaluation Methodology

We opt for the evaluation approach proposed in Simard & 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, thus effectively inhibiting output filtering. The target segments of the best matches are then compared to the reference translations, using standard MT evaluation metrics. In practice, in this study, we use the same metrics that were used as similarity functions.

Data

We perform experiments to assess the performance of each MT evaluation metric as TM similarity function. Experiments were done using English, French, German and Spanish data, drawn from **Europarl** v.6 (Koehn, 2005), the **OPUS** corpus (Tiedemann, 2009) and the **JRC-Acquis** v.2.2 (Steinberger, 2006). From each corpus, we randomly sampled 1000 pairs of segments, to be used as test data; the rest was used to build translation memories.

Corpus	Language	TM ("Train") segments	TM ("Train") words	Test 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

Results

This table reports which similarity function $f(q, s)$ performs best, according to each MT evaluation metric. **Who wins the race depends heavily on who is keeping the score!**

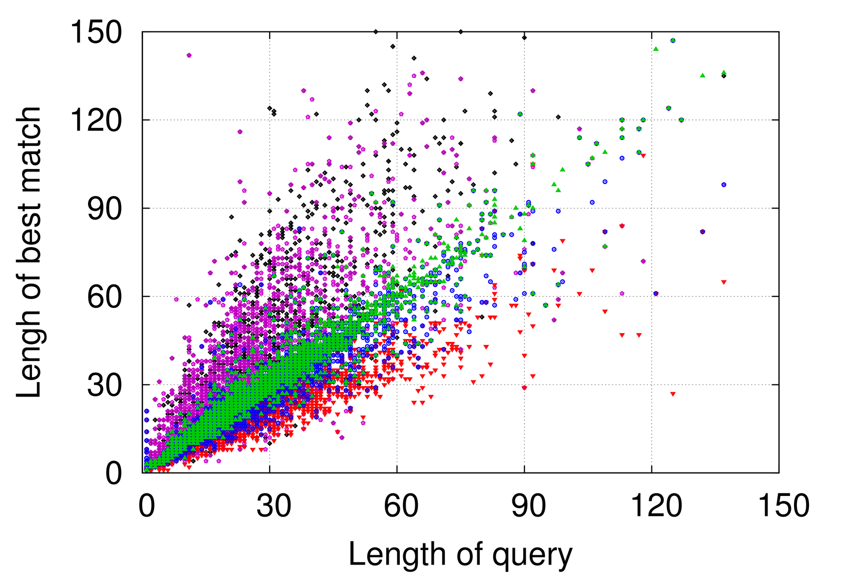
Corpus	Language	Evaluation Metric: WER	BLEU	NIST	VMeteor	Meteor
Europarl	en-de	WER	BLEU	BLEU	Meteor	Meteor
	en-es	WER	BLEU	NIST	VMeteor	VMeteor
	en-fr	WER	BLEU	NIST	VMeteor	VMeteor
	de-en	WER	BLEU	NIST	Meteor	Meteor
	es-en	WER	VMeteor	Meteor	Meteor	Meteor
	fr-en	WER	BLEU	NIST	Meteor	Meteor
ECB	en-de	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

- When WER is used to measure performance, WER always comes out as the best similarity function;
- BLEU generally exhibits similar behaviour.
- VMeteor and Meteor always prefer one of the Meteor family; both metrics also always agree with one another, usually preferring VMeteor when English is the source language and Meteor when English is target.
- NIST has low self-esteem: this is because **local optimization** (finding the best match for each sentence) doesn't guarantee a global maximum.

TM query:match size ratios

Length ratios between source language query and TM best match indirectly impacts target language ratio as well. It thus plays a potentially important role in measured performance.

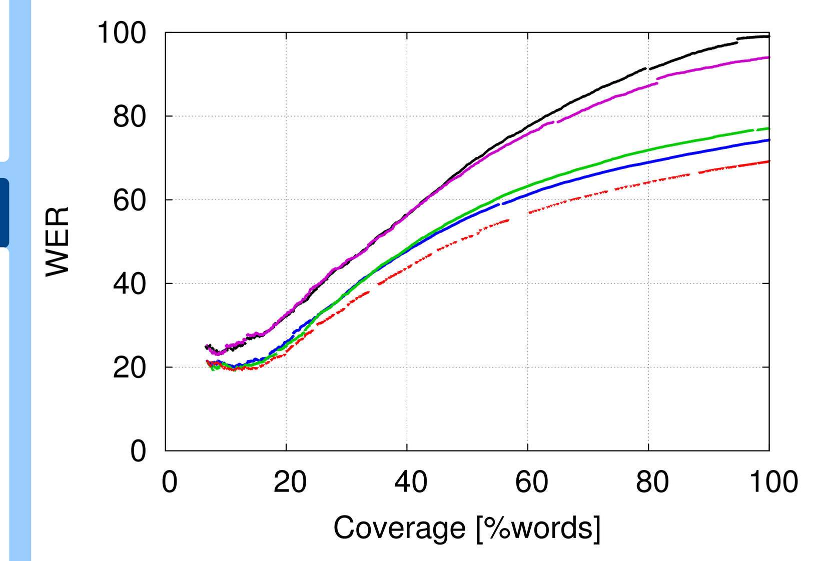
Similarity Function	Source Ratio $\lvert\hat{s}\rvert/\lvert q\rvert$	Target Ratio $\lvert\hat{t}\rvert/\lvert r\rvert$
WER	0.85 ± 0.18	0.88 ± 0.58
BLEU	1.01 ± 0.60	1.04 ± 0.91
NIST	1.03 ± 0.15	1.06 ± 0.63
VMeteor	1.31 ± 0.85	1.43 ± 2.01
Meteor	1.36 ± 0.85	1.48 ± 2.07



- BLEU and NIST tend to produce TM best matches whose source segment length is very close to that of the query.
- WER naturally favors segments that are much shorter 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.
- Although the Meteor metrics (Meteor and VMeteor) de-emphasize length-similarity, they tend to produce source segments that are much longer than the query, which will naturally be penalized by precision-based evaluation metrics.

TM performance VS coverage

TM filtering has a direct impact on performance: As threshold α is set higher, less queries find matches in the TM, but performance on the filtered material improves.



- 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: This is the kind of material that the translator typically does not want to see.
- in the **low-coverage** areas, where only the best matching segments from the TM are retained, all metrics display very comparable performances.

Custom Paraphrase Tables for Meteor

One way of optimizing the performance of Meteor as a TM similarity function is to provide it with **domain-specific paraphrases**.

- **CMeteor** ("Custom" Meteor): Using the method of Fujita et al. (2012), we extract paraphrases from each TM to create domain-specific paraphrase tables, and use these with Meteor instead of the standard tables (no parameter tuning).
- In practice, domain-specific paraphrases do not lead to measurable gains or losses in performance: 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.

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.

Conclusions

MT evaluation metrics can be used effectively as translation memory similarity functions. Each metric has its own characteristics and potential benefits, but evaluation is problematic.

- 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, especially in high-similarity situations, where it counts for real-life TM usage.
- **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 (Snover et al. 2009). 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.
- 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.