Gaming Fluency: Evaluating the Bounds and Expectations of Segment-based Translation Memory

John Henderson and **William Morgan**
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
{jhndrsn,wmorgan}@mitre.org

Abstract

Translation memories provide assistance to human translators in production settings, and are sometimes used as first-pass machine translation in assimilation settings because they produce highly fluent output very rapidly. In this paper, we describe and evaluate a simple whole-segment translation memory, placing it as a new lower bound in the well-populated space of machine translation systems. The result is a new way to gauge how far machine translation has progressed compared to an easily understood baseline system.

The evaluation also sheds light on the evaluation metric and gives evidence showing that gaming translation with perfect fluency does not fool BLEU the way it fools people.

1 Introduction and background

Translation Memory (TM) systems provide roughly concordanced results from an archive of previously translated materials. They are typically used by translators who want computer assistance for searching large archives for tricky translations, and also to help ensure a group of translators rapidly arrive at similar terminology (Macklovitch et al., 2000). Several companies provide commercial TMs and systems for using and sharing them. TMs can add value to computer assisted translation services (Drugan, 2004).

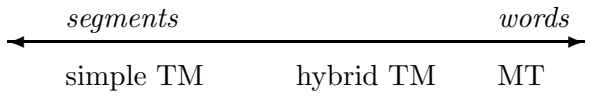
Machine Translation (MT) developers make use of similar historical archives (parallel texts, bitexts), to produce systems that perform a task very similar to TMs. But while TM systems and MT systems can appear strikingly similar, (Marcu, 2001) key differences exist in how they are used.

TMs often need to be fast because they are typically used interactively. They aim to produce highly readable, fluent output, usable in document production settings. In this setting, errors of omission are more easily forgiven than errors of commission so, just like MT, TM output must look good to users who have no access to the information in source texts.

MT, on the other hand, is often used in assimilation settings, where a batch job can often be run on multiple processors. This permits variable rate output and allows slower systems that produce better translations to play a part. Batch MT serving a single user only needs to run at roughly the same rate the reader can consume its output.

Simple TMs operate on an entire translation segment, roughly the size of a sentence or two, while more sophisticated TMs operate on units of varying size: a word, a phrase, or an entire segment (Callison-Burch et al., 2004). Modern approaches to MT, especially statistical MT, typically operate on more fine-grained units, words and phrases (Och and Ney, 2004). The relationship between whole segment TM and MT can be viewed as a continuum of translation granularity:

Proceedings of the ACL Workshop on Building and Using Parallel Texts, pages 175–182,
Ann Arbor, June 2005.



Simple TM systems, focusing on segment-level granularity, lie at one extreme, and word-for-word, IBM-model MT systems on the other. Example-Based MT (EBMT), phrase-based, and commercial TM systems likely lie somewhere in between.

This classification motivates our work here. MT systems have well-studied and popular evaluation techniques such as BLEU (Papineni et al., 2001). In this paper we lay out a methodology for evaluating TMs along the lines of MT evaluation. This allows us to measure the raw relative value of TM and MT as translation tools, and to develop expectations for how TM performance increases as the size of the memory increases.

There are many ways to perform TM segmentation and phrase extraction. In this study, we use the most obvious and simple condition—a full segment TM. This gives a lower bound on real TM performance, but a lower bound which is not trivial.

Section 2 details the architecture of our simple TM. Section 3 describes experiments involving different strategies for IR, oracle upper bounds on TM performance as the memory grows, and techniques for rescoring the retrievals. Section 4 discusses the results of the experiments.

2 A Simple Chinese-English Translation Memory

For our experiments below, we constructed a simple translation memory from a sentence-aligned parallel corpus. The system consists of three stages. A source-language input string is rewritten to form an information retrieval (IR) query. The IR engine is called to return a list of candidate translation pairs. Finally a single target-language translation as output is chosen.

2.1 Query rewriting

To retrieve a list of translation candidates from the IR engine, we first create a query which is a concatenation of all possible ngrams of the source sentence, for all ngram sizes from 1 to a fixed n.

We rely on the fact that the Chinese data in the translation memory is tokenized and indexed at the unigram level. Each Chinese character in the source sentence is tokenized individually, and we make use of the IR engine's phrase query feature, which matches documents in which all terms in the phrase appear in consecutive order, to create the ngrams. For example, to produce a trigram + bigram + unigram query for a Chinese sentence of 10 characters, we would create a query consisting of eight three-character phrases, nine two-character phrases, and 10 single-character "phrases". All phrases are weighted equally in the query.

This approach allows us to perform lookups for arbitrary ngram sizes. Depending on the specifics of how *idf* is calculated, this may yield different results from indexing ngrams directly, but it is advantageous in terms of space consumed and scalability to different ngram sizes without reindexing.

This is a slight generalization of the successful approach to Chinese information retrieval using bigrams (Kwok, 1997). Unlike that work, we perform no second stage IR after query expansion. Using a segmentation-independent engineering approach to Chinese IR allows us to sidestep the lack of a strong segmentation standard for our heterogeneous parallel corpus and prepares us to rapidly move to other languages with segmentation or lemmatization challenges.

2.2 The IR engine

Simply for performance reasons, an IR engine, or some other sort of index, is needed to implement a TM (Brown, 2004). We use the open-source Lucene $v1.4.3$, (Apa, 2004) as our IR engine. Lucene scores candidate segments from the parallel text using a modified *tf-idf* formula that includes normalizations for the input segment length and the candidate segment length. We did not modify any Lucene defaults for these experiments.

To form our translation memory, we indexed all sentence pairs in the translation memory corpora, each pair as a separate document. We

Source
库林斯说："他随时都可能亮相，这一切取决于他的感觉。 目前，他训练的侧重点是防守，同时也练习投篮。 他准备略为提高强度，以检验自己的身体是否能适应。 就膝盖而言，他说至少比手术前好了百分之百。"
TM output
However , everything depended on the missions to be decided by the Security Council . The presentations focused on the main lessons learned from their activities in the field . It is wrong to commit suicide or to use ones own body as a weapon of destruction . There was practically full employment in all sectors .
One reference translation (of four)
Doug Collins said, "He may appear any time. It really depends on how he feels." At present, his training is defense oriented but he also practices shots. He is elevating the intensity to test whether his body can adapt to it. So far as his knee is concerned, he thinks it heals a hundred percent after the surgery."

Table 1: Typical TM output. Excerpt from a story about athlete Michael Jordan.

indexed in such a way that IR searches can be restricted to just the source language side or just the target language side.

2.3 Rescoring

The IR engine returns a list of candidate translation pairs based on the query string, and the final stage of the TM process is the selection of a single target-language output sentence from that set.

We consider a variety of selection metrics in the experiments below. For each metric, the source-language side of each pair in the candidate list is evaluated against the original source language input string. The target language segment of the pair with the highest score is then output as the translation.

In the case of automated MT evaluation metrics, which are not necessarily symmetric, the source-language input string is treated as the reference and the source-language side of each pair returned by the IR engine as the hypothesis.

All tie-breaking is done via *tf-idf*, i.e. if multiple entries share the same score, the one ranked higher by the search engine will be output.

Table 1 gives a typical example of how the TM performs. Four contiguous source segments are presented, followed by TM output and finally one of the reference translations for those source segments. The only indicator of the translation quality available to monolingual English speakers is the awkwardness of the segments as a group. By design, the TM performs with perfect fluency at the segment level.

3 Experiments

We performed several experiments in the course of optimizing this TM, all using the same set of parallel texts for the TM database and multiple-reference translation corpus for evaluation. The parallel texts for the TM come from several Chinese-English parallel corpora, all available from the Linguistic Data Consortium (LDC). These corpora are described in Table 2. We discarded any sentence pairs that seemed trivially incomplete, corrupt, or otherwise invalid. In the case of LDC2002E18, in which sentences were aligned automatically and confidence scores produced for each alignment, we dropped all pairs with scores above 9, indicating poor alignment. No duplication checks were performed. Our final corpus contained approximately 7 million sentence pairs and contained 3.2 GB of UTF-8 data.

Our evaluation corpus and reference corpus

come from the data used in the NIST 2002 MT competition. (NIST, 2002). The evaluation corpus is 878 segments of Chinese source text. The reference corpus consists of four independent human-generated reference English translations of the evaluation corpus.

All performance measurements were made using a fast reimplementation of NIST's BLEU. BLEU exhibits a high correlation with human judgments of translation quality when measuring on large sections of text (Papineni et al., 2001). Furthermore, using BLEU allowed us to compare our performance to that of other systems that have been tested with the same evaluation data.

3.1 An upper bound on whole-segment translation memory

Our first experiment was to determine an upper bound for the entire translation memory corpus. In other words, given an oracle that picks the best possible translation from the translation memory corpus for each segment in the evaluation corpus, what is the BLEU score for the resulting document? This score is unlikely to approach the maximum, BLEU =100 because this oracle is constrained to selecting a translation from the target language side of the parallel corpus. All of the calculations for this experiment are performed on the target language side of the parallel text.

We were able to take advantage of a trait particular to BLEU for this experiment, avoiding many of BLEU score calculations required to assess all of the 878×7.5 million combinations. BLEU produces a score of 0 for any hypothesis string that doesn't share at least one 4-gram with one reference string. Thus, for each set of four references, we created a Lucene query that returned all translation pairs which matched at least one 4-gram with one of the references. We picked the top segment by calculating BLEU scores against the references, and created a hypothesis document from these segments.

Note that, for document scores, BLEU's brevity penalty (BP) is applied globally to an entire document and not to individual segments. Thus, the document score does not necessarily increase monotonically with increases in scores of individual segments. As more than 99% of the segment pairs we evaluated yielded scores of zero, we felt this would not have a significant effect on our experiments. Also, the TM does not have much liberty to alter the length of the returned segments. Individual segments were chosen to optimize BLEU score, and the resulting documents exhibited appropriately increasing scores. While there is no efficient strategy for whole-document BLEU maximization, an iterative rescoring of the entire document while optimizing the choice of only one candidate segment at a time could potentially yield higher scores than those we report here.

3.2 TM performance with varied Ngram length

The second experiment was to determine the effect that different ngram sizes in the Chinese IR query have on the IR engine's ability to retrieve good English translations.

We considered cumulative ngram sizes from 1 to 7, i.e. unigram, unigram + bigram, unigram + bigram + trigram, and so on. For each set of ngram sizes, we created a Lucene query for every segment of the (Chinese) evaluation corpus. We then produced a hypothesis document by combining the English sides of the top results returned by Lucene for each query. The hypothesis document was evaluated against the reference corpora by calculating a BLEU score.

While it was observed that IR performance is maximized by performing bigram queries (Kwok, 1997), we had reason to believe the TM would not be similar. TMs must attempt to match short sequences of stop words that indicate grammar as well as more traditional content words. Note that our system performed neither stemming nor stop word (or ngram) removal on the input Chinese strings.

3.3 An upper bound on TM N-best list rescoring

The next experiment was to determine an upper bound on the performance of *tf-idf* for different result set sizes, i.e. for different (maximum)

LDC Id	Description	Pairs
LDC2002E18	Xinhua Chinese-English Parallel News Text v. 1.0 beta 2	64,371
LDC2002E58	Sinorama Chinese-English Parallel Text	103,216
LDC2003E25	Hong Kong News Parallel Text	641,308
LDC2004E09	Hong Kong Hansard Parallel Text	1,247,294
LDC2004E12	UN Chinese-English Parallel Text v. 2	4,979,798
LDC2000T47	Hong Kong Laws Parallel Text	302,945
	Total	7,338,932

Table 2: Sentence-aligned parallel corpora used for the creation of the translation memory. The "pairs" column gives the number of translation pairs available after trivial pruning.

numbers of translation pairs returned by the IR engine. This experiment describes the trade-off between more time spent in the IR engine creating a longer list of returns and the potential increase in translation score.

To determine how much IR was "enough" IR, we performed an oracle experiment on different IR query sizes. For each segment of the evaluation corpus, we performed a cumulative 4-gram query as described in Section 4.2. We produced the n-best list oracle's hypothesis document by selecting the English translation from this result set with the highest BLEU score when evaluated against the corresponding segment from the *reference* corpus. We then evaluated the hypothesis documents against the *reference* corpus by computing BLEU scores.

3.4 N-best list rescoring with several MT evaluation metrics

The fourth experiment was to determine whether we could improve upon *tf-idf* by applying automated MT metrics to pick the best sentence from the top n translation pairs returned by the IR engine. We compared a variety of metrics from MT evaluation literatures. All of these were run on the tokens in the source language side of the IR result, comparing against the single pseudo-reference, the original source language segment. While many of these metrics aren't designed to perform well with one reference, they stand in as good approximate string matching algorithms.

The score that the IR engine associates with each segment is retained and marked as *tf-idf* in this experiment. Naturally, BLEU (Papineni et al., 2001) was the first choice metric, as it was well-matched to the target language evaluation function. ROUGE was a reimplementation of ROUGE-L from (Lin and Och, 2004). It computes an F-measure from precision and recall that are both based on the longest common subsequence of the hypothesis and reference strings. WER-G is a variation on traditional word error rate that was found to correlate very well with human judgments (Foster et al., 2003), and PER is the traditional position-independent error rate that was also shown to correlate well with human judgments (Leusch et al., 2003). Finally, a random metric was added to show the BLEU value one could achieve by selecting from the top n strictly by chance.

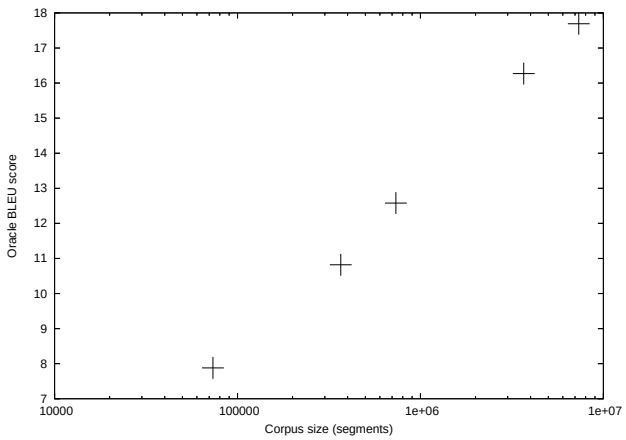
After the individual metrics are calculated for these segments, a uniform-weight log-linear combination of the metrics is calculated and used to produce a new rank ordering under the belief that the different metrics will make predictions that are constructive in aggregate.

4 Results

4.1 An upper bound for whole-sentence TM

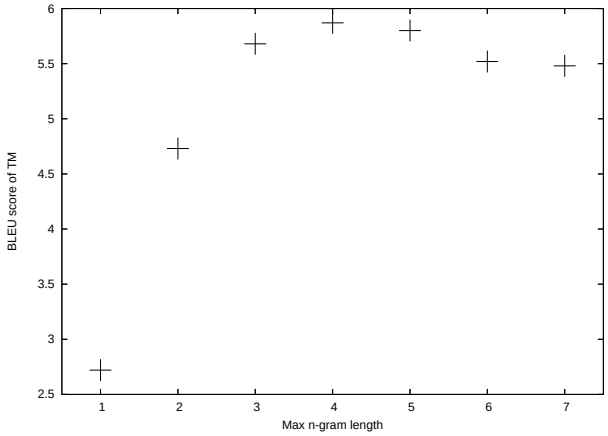
Figure 1 shows the maximum possible BLEU score that can an oracle can achieve by selecting the best English-side segment from the parallel text. The upper bound achieved here is a BLEU score of 17.7, and this number is higher than the best performing system in the corresponding NIST evaluation.

Note the log-linear growth in the resulting



Size	BLEU
73389	7.88
366947	10.82
733893	12.58
3669466	16.27
7338932	**17.69**

Figure 1: Oracle bounds on TM performance as corpus size increases.



Ngrams in query	BLEU
1	2.72
1,2	4.73
1,2,3	5.68
1,2,3,4	**5.87**
1,2,3,4,5	5.80
1,2,3,4,5,6	5.52
1,2,3,4,5,6,7	5.48

Figure 2: BLEU scores for different cumulative ngram sizes, when retrieving only the first translation pair.

BLEU score of the TM with increasing database size. As the database is increased by a factor of ten, the TM gains approximately 5 points of BLEU. While this trend has a natural limit at 20 orders of magnitude, it is unlikely that this amount of text, let alone parallel text, will be a indexed in the foreseeable future. This rate is more useful in interpolation, giving an idea of how much could be gained from adding to corpora that are smaller than 7.5 million segments.

4.2 The effect of ngram size on Chinese *tf-idf* retrieval

Figure 2 shows that our best performance is realized when IR queries are composed of cumulative 4-grams (i.e. unigrams + bigrams + trigrams + 4-grams). As hypothesized, while longer sequences are not important in document retrieval in Chinese IR, they convey information that is useful in segment retrieval in the translation memory. For the remainder of the experiments, we restrict ourselves to cumulative 4-gram queries.

Note that the 4-gram result here (BLEU of 5.87) provides the baseline system performance measure as well as the value when the segments are reranked according to *tf-idf*.

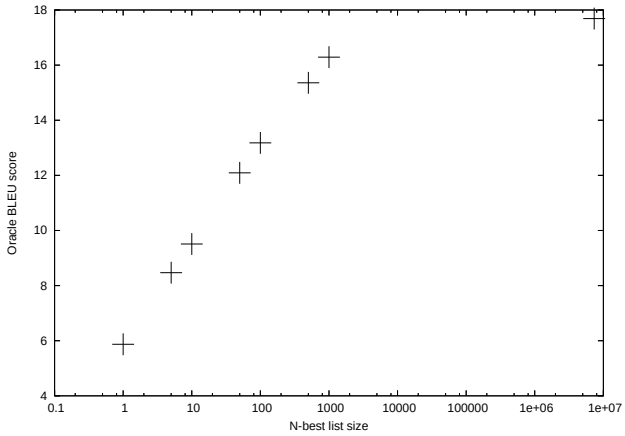
4.3 Upper bounds for *tf-idf*

Figure 3 gives the n-best list rescoring bounds. The upper bound continues to increase up to the top 1000 results. The plateau achieved after 1000 IR results suggests that is little to be gained from further IR engine retrieval.

Note the log-linear growth in the BLEU score the oracle achieves as the n-best list extends on the left side of the figure. As the list length is increased by a factor of ten, the oracle upper bound on performance increases by roughly 3 points of BLEU. Of course, for a system to perform as well as the oracle does becomes progressively harder as the n-best list size increases.

Comparing this result with the experiment in section 4.1 indicates that making the oracle choose among Chinese source language IR results and limiting its view to the 1000 results given by the IR engine incurs only a minor reduction of the oracle's BLEU score, from 17.7 to

16.3. This is one way to measure the impact of crossing this particular language barrier and using IR rather than exhaustive search.



Size	BLEU score
1	5.87
5	8.47
10	9.51
50	12.09
100	13.18
500	15.36
1000	16.29
7338932	**17.69**

Figure 3: BLEU scores for different (maximum) numbers of translation pairs returned by IR engine, where the optimal segment is chosen from the results by an oracle.

4.4 Using automated MT metrics to pick the best TM sentence

Each metric was run on the top 1000 results from the IR engine, on cumulative 4-gram queries. Each metric was given the (Chinese) evaluation corpus segment as the single reference, and scored the Chinese side of each of the 1000 resulting translation pairs against that reference. The hypothesis document for each metric consisted of the English side of the translation pair with the best score for each segment. These documents were scored with BLEU against the reference corpus. Ties (e.g. cases where a metric gave all 1000 pairs the same score) were broken with *tf-idf*.

Results of the rescoring experiment run on an n-best list of size 100 are given in Table 3. Choosing from 1000 pairs did not give better results. Choosing from only 10 gave worse results. The random baseline given in the table represents the expected score from choosing randomly among the top 100 IR returns. While the scores of the individual metrics aside from PER and BLEU reveal no differences, BLEU and the combination metric performed better than the individual metrics.

Metric	BLEU
BLEU	6.20
WER-G	5.90
ROUGE	5.88
tf-idf	5.87
PER	5.72
random	3.32
log(*tf-idf*)	
+log(BLEU)	
+log(ROUGE)	
-log(WER-G)	
-log(PER)	**6.56**

Table 3: BLEU scores for different metrics when picking the best translation from 100 translation pairs returned by the IR engine.

Surprisingly, *tf-idf* was outperformed only by BLEU and the combination metric. While we hoped to gain much more from n-best list rescoring on this task, reaching toward the limits discovered in section 4.3, the combination metric was less than 0.5 BLEU points below the lower range of systems that were entered in the NIST 2002 evals. The BLEU scores of research systems in that competition roughly ranged between 7 and 15. Of course, each of the segments produced by the TM exhibit *perfect fluency*.

5 Discussion

The maximum BLEU score attained by a TM we describe (6.56) would place it in last place in the NIST 2002 evals, but by less than 0.5 BLEU. Successive NIST competitions have exhibited impressive system progress, but each year there have been newcomers who score near (or in some cases lower than) our simple TM baseline.

We have presented several experiments that quantitatively describe how well a simple TM performs when measured with a standard MT evaluation measure, BLEU. We showed that the translation performance of a TM grows as a log-linear function of corpus size below 7.5 million segments. We showed, somewhat surprisingly, only 1000 IR returns need be evaluated by a rescorer to get within 1 BLEU point of the maximum possible score attainable by the TM.

In future work, we expect to validate these results with other language pairs. One question is: how well does this simple IR query expansion address segmented languages and languages that allow more liberal word order? Supervised training of n-best reranking schemes would also determine how far the oracle bound can be pushed. The computationally more expensive reranking procedure that attempts to optimize BLEU on the entire document should be investigated to determine how much can be gained by better global management of the brevity penalty.

Finally, we believe it's worth noting the degree to which high fluency of the TM output could potentially mislead target-language-only readers in their estimation of the system's performance. Table 1 is representative of system output, and is a good example of why translations should not be judged solely on the fluency of a few segments of target language output.

References

Apache Software Foundation, 2004. *Lucene 1.4.3 API*. `http://lucene.apache.org/java/docs/api/`.

Ralf D. Brown. 2004. A modified burrows-wheeler transform for highly-scalable example-based translation. In *Machine Translation: From Real Users to Research, Proceedings of the 6th Conference of the Association for Machine Translation (AMTA-2004)*, Washington, D.C., USA.

Chris Callison-Burch, Colin Bannard, and Josh Schroeder. 2004. Searchable translation memories. In *Proceedings of ASLIB Translation and the Computer 26*.

Joanna Drugan. 2004. Multilingual document management and workflow in the european institutions. In *Proceedings of ASLIB Translation and the Computer 26*.

George Foster, Simona Gandrabur, Cyril Goutte, Erin Fitzgerald, Alberto Sanchis, Nicola Ueffing, John Blatz, and Alex Kulesza. 2003. Confidence estimation for machine translation. Technical report, JHU Center for Language and Speech Processing.

K. L. Kwok. 1997. Comparing representations in chinese information retrieval. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 34–41, New York, NY, USA. ACM Press.

G. Leusch, N. Ueffing, and H. Ney. 2003. A novel string-to-string distance measure with applications to machine translation evaluation. In *Proc. of the Ninth MT Summit*, pages 240–247.

Chin-Yew Lin and Franz Josef Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland, August.

E. Macklovitch, M. Simard, and Ph. Langlais. 2000. Transsearch: A free translation memory on the world wide web. In *Second International Conference On Language Resources and Evaluation (LREC)*, volume 3, pages 1201–1208, Athens Greece, jun.

Daniel Marcu. 2001. Towards a unified approach to memory- and statistical-based machine translation. In *ACL*, pages 378–385.

NIST. 2002. The NIST 2002 machine translation evaluation plan (MT-02). NIST web site. `http://www.nist.gov/speech/tests/mt/doc/2002-MT-EvalPlan-v1.3.pdf`.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4).

K. Papineni, S. Roukos, T. Ward, and W-J. Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. Technical Report RC22176 (W0109-022), IBM Research Division, Thomas J. Watson Research Center.