**RC25158 (W1105-029) May 11, 2011**
**Computer Science**

# IBM Research Report

## On-Demand Phrase Extraction: An On-Line String Similarity Approach to Machine Translation

**Juan M. Huerta**
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

**Research Division**
**Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich**

# On-demand Phrase Extraction: An on-Line String Similarity Approach to Machine Translation

**Juan M. Huerta**
IBM T. J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598
huerta@us.ibm.com

## Abstract

This paper describes a novel approach to creating, or synthesizing, unobserved translations from previously observed translated sentence pairs by matching and combining multiple matching sentences in a translation memory. Our approach balances high-accuracy with coverage. This approach yields a substantial BLEU score and coverage improvement over the basic best fuzzy-match baseline translation memory retrieval result.

## 1 Introduction

Translation memories (TM) are front end components responsible for detecting requests corresponding to previously produced translations. By directly retrieving human-provided translations, basic translation memories can increase the overall speed and accuracy of a translation system.

To avoid limiting the coverage of the TM exclusively to previously observed sentences many approaches to TMs have been proposed to combine relevant sentences in the TM to best match the query. The design of a TM requires a trade-off between high accuracy (e.g., exact matches) and coverage (i.e., the ability to combine previously seen translation).

We propose an approach in which the TM can generate, or synthesize, a previously unseen translation based on the on-demand optimal combination of multiple approximate matches and their corresponding word-level translation alignments. Similarly to most Translation Memory based systems, we want to have an approach that very high translation accuracy even if it has relatively high rejection rates (i.e., low recall) running in parallel to a SMT component. Our approach differs from previous approaches in several ways which we will outline in later sections, but the most salient difference is that our approach is statistic-model free and requires no model training or development.

To avoid the need to have statistical word and phrase reordering models, we propose a novel *carrier-sentence* editing approach in which a sentence with similar structure to the query is incrementally edited or modified with phrases extracted from the corpus on-line until it best resembles the query.

Analysis of experimental results using our approach and a large bank of translation pairs in a Software Documentation localization corpus shows that both the BLEU score and coverage in our approach are significantly better over the single match baseline translation memory result.

## 2 Algorithm

Our approach is based on a translation memory *T* and a query sentence $q_j$ to be translated. The TM *T* consists of the set of paired sentences $s_i^s$ and $s_i^t$ denoting the $i^{th}$ source and target sentence pair respectively. The cardinality of *T* is *N*. Our approach consists of a pre-decoding phase and 3 decoding phases: a fast match phase, a detailed match phase (with multiple steps), and a synthesis (or rewrite) phase.

*Pre-decoding Phase:* This step consists on creating word alignments between each source and corresponding target sentence pair in *T*.

*Fast Match Phase:* The fast match phase is optional and is meant to reduce the search space size for the detailed decoding phase. The fast match uses a previously constructed inverted index to rapidly identify the subset of $T$ that matches the query $q_j$. This subset is the Reduced Set (denoted by $R_j$) with cardinality $C_j$. The extent to which this step is beneficial depends on to the condition $C_j<<N$ being true, which is typically the case (i.e., for a typical query there is only a small portion of the TM which matches the query with a small edit distance).

*Detailed Match Phase:* The detailed match in turn has four steps.

*Step 1:* The detailed match phase begins with the computation of the string edit distance between the query sentence and each sentence in the matching set $T$ (or the fast match subset $R_j$ if available). In this step, in addition to the computation of the string edit distance scores, the set of mapping vectors are computed too. A mapping vector maps the words in $q_j$ to words in a given sentence of $T$ as deemed by its best String Edit Distance alignment path. The complexity of this step is $O(kmN)$ for length query $k$, average sentence length $m$ and no fast match phase. Typically $N>>k$ and $N>>m$.

*Step 2:* The mapping array is assembled by aggregating the $N$ mapping vectors into an $N$ *by* $k$ matrix and replacing the value of the indexes with 1's and 0's. The entries in the array will have value *0* if the query word does not map to a word in the corresponding candidate sentence and *1* otherwise.

We then carry out two discounting steps aimed at penalizing discontinuity. The first discounting penalizes non-adjacencies of source words (the source is the query) *in the target sentence* (the target is each sentence in the Memory) by multiplicative discount penalty for every position that separates two adjacent words in the target. The second discounting step, discounts each array entry if there exist gaps (uncovered words) in the source sentence. The goal of this discount is to promote clusters of adjacent source words. Both multiplicative discount base factors are determined empirically.

*Step 3:* Using the penalized values in the mapping array as observation scores we use Viterbi decoding to find the best path connecting the first column to the *kth* column, traversing all the words in the query across the rows in the mapping array, each row corresponding to the vector mapping of each sentence in $T$ (or in $R_j$). Transition scores for paths staying in the same sentence are set higher than switching scores. The resulting path from column 1 to column k determines the best combination of sentences and the words in the query they cover. In practice many of the $N$ sentences in $T$ have only zero-valued elements (i.e., they do not overlap with the query) and need not to be part of the Viterbi step. The subset of sentences in $T$ that are part of the best path is called the source set.

*Step 4:* This step consists of finding the carrier sentence which is the sentence in the source set whose overall structure most resembles the query. A simple way to implement this is to identify the sentence that contributed most to the best Viterbi path score. More advanced ways to achieve this could compare parse tree distances if these are pre-computed and available.

*Synthesis Phase:* The carrier sentence and its corresponding translation are concurrently edited (using the pre-computed word alignments) based on the best Viterbi path transitions and the source sentences by means of deletions (on the carrier) as well as insertion and substitutions (on the carrier using source sentences) steps. A contiguity constrain imposes that any substitution and insertion on the source is permissible if and only if the segment is also contiguous in its aligned translation.

Figure 1 below shows a very simple instance of how two source sentences are combined to produce the query. The first line is the word id's of the query "a connection to a sip container has been broken", the second line represents the Viterbi path (sentenceId:Index) showing that there are two source sentences, the next two lines illustrate the word alignment including the insertion operation of "sip container". Finally the two original source sentences are shown in their textual form.

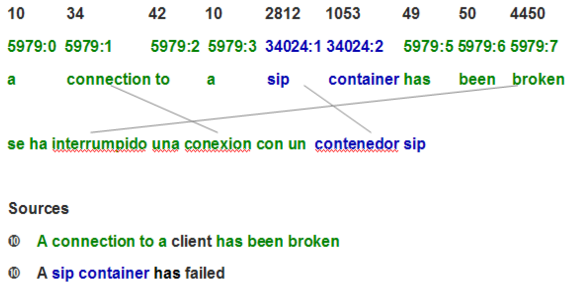

10 34 42 10 2812 1053 49 50 4450
5979:0 5979:1 5979:2 5979:3 34024:1 34024:2 5979:5 5979:6 5979:7
a connection to a sip container has been broken
se ha interrumpido una conexion con un contenedor sip
Sources
A connection to a client has been broken
A sip container has failed

**Figure 1.** Example of 2 source sentences synthesis

## 3 Related Work

We now present a brief overview of related work followed by our perspective on how our approach differs from these previous approaches.

Marcu (2001) combines a TM with a statistical model to improve over a pure SMT approach. Sumita (2001) proposed example based machine translation based on translation patterns and a bilingual dictionary. Langlais & Simard (2001) focused on subsentenial recall (phrases). Watanabe & Sumita (2003) proposed an example-based approach for Statistical Machine Translation (retrieval and modification of source) applying a greedy algorithm and a statistical model. Previously, Veale and Way (1997) relied on a transfer-template based bootstrapping approach. Hewavitharana et al (2005) proposed a hybrid approach (memory + statistical model) in which the TM component find similar sentences and edits them (similar to our approach), however it is based on phrases, it has a probabilistic phrase extraction model, and is not based on Viterbi to identify the optimal source sentence combination.

While our approach shares similarities with these previous approaches, it has the following fundamental differences:

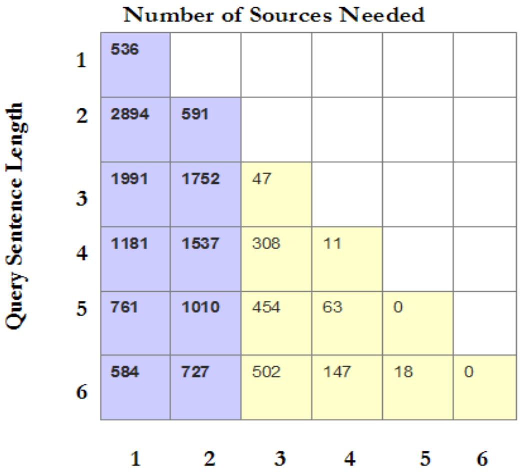
1. We do not rely on a probabilistic model of any kind. Nor we use a reordering model, or language model.
2. Our approach identifies first a *carrier sentence* based on Viterbi search and then performs operations on that sentence based on the set of matches sentences found. It does have a contiguity scoring approach which penalizes within sentence and across sentence transitions.
3. Is based on full sentence editing with word-level alignments and not in phrases, templates, bilingual dictionaries or translation models. No phrases are needed to be computed a-priori.
4. Our approach does not pre-compute a global phrase table. Rather, phrases are discovered and applied on-line giving our approach the opportunity to leverage phrases with arbitrarily large spans.

## 4 Corpus and Experiments

We conducted experiments using a Translation Memory originating from the English-Spanish set of sentences in the technical documentation of a family of software products. After basic filtering to remove unsuitable sentences (sentences containing variables) the TM comprised 1.14 M sentences.

We partitioned this set into 916k sentences for training (TM proper), and 5k for evaluation. (Additionally 200k sentences were left out for future large-scale testing). The TM had 15% redundant sentences in the source language because they originated from different product families and we preserved this redundancy as eventually we might want to have different product specific target translations. Word alignment was performed using GIZA (Och & Ney (2003)) using IBM Model 4. Figure 2 shows the size of the number of source sentences needed given the length of the query in a random subsample of the eval set. Up to 90% of these sentences are covered by 1, 2 or 3 sources.

Table 1 shows the summary of results each row represents a condition. For the baseline (v00), for each sentence in the eval set we plainly hypothesized the best match (even if it is not an exact match). Out of 5745 sentences, 779 were matched exactly (due to the redundancy in the TM). This means that 13.4% of the eval set (which is 13.4% of the translated set) was translated exactly. There were no rejections in v00 and the BLEU score of the translated set is 0.3418.



Number of Sources Needed

| Query Sentence Length | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| 1 | 536 | | | | | |
| 2 | 2894 | 591 | | | | |
| 3 | 1991 | 1752 | 47 | | | |
| 4 | 1181 | 1537 | 308 | 11 | | |
| 5 | 761 | 1010 | 454 | 63 | 0 | |
| 6 | 584 | 727 | 502 | 147 | 18 | 0 |

**Figure 2.** Constituents as function of Query Length.

|  | Description | # Sents. | # Skipped | % Skipped | # Transl. Exactly | Exactly as % of Total | Exactly as % of non-Skipped | F. measure | **BLEU n4r1** |
|---|---|---|---|---|---|---|---|---|---|
| v.00 | Baseline | 5745 | 0 | 0% | 779 | 13.6% | 13.6% | 0.23 | **0.3418** |
| v.01 | 1 source (no NULLS) | 5745 | 3802 | 66.2% | 459 | 8.0% | 23.6% | 0.27 | **0.4508** |
| v.02 | 1 source (allow NULLS) | 5745 | 3802 | 66.2% | 1003 | 17.5% | 51.6% | 0.41 | **0.6711** |
| v.03 | 2 sources | 5745 | 2137 | 37.2% | 1059 | 18.4% | 29.4% | 0.40 | **0.5914** |

**Table 1.** Translation Results for various experiment conditions

In conditions v01 and v02 for each sentence in the eval set we applied our technique only if the source set had cardinality 1, otherwise we skipped the sentence. In condition v01 we edited the carrier sentence but left all the NULL alignments out (a naïve approach), while in v02 we included NULLs within translated segments. In v01 and v02, out of 5745 sentences, 3802 were skipped and 459 and 1003 sentences were translated correctly, respectively. This means that 8.0% and 17.5% of the eval set (corresponding to 23.6% and 51.6 of the translated set) was translated exactly. The BLEU score of the translated set is 0.45 and 0.67 which includes all attempted (perfect+imperfect) translations (4-gram BLEU using 1 reference).

In condition v03, for each sentence in the eval set we applied our technique only if the source set had cardinality 2 (i.e., consisted of 2 source sentences only).

In v03 out of 5745 sentences, 2137 were skipped and 1059 sentences were translated correctly. Thus 18.4% of the eval set (29.4% of the translated set) was translated exactly. The BLEU score of the translated set is 0.59.

Interestingly, the BLEU score of the translated set increases in v02 and v01 but in v03 it decreases. We believe this is so because we are attempting more translations using more sources. The approach producing the large total number of perfectly translated sentences is v03.

Finally, we calculated an F measure taking the % sentences that were corrected perfectly (as percentage of the total) as precision and the % translated attempts (1-%skipped) as recall. We can see that the highest f-score is for condition v02.

## 5 Conclusion

We introduced a non-statistical approach to machine translation with high accuracy (and possibly selective, high rejection) intended to derive sentence translations based on a sentence retrieval and editing approach.

Our approach is based on carrier sentences and is free of pre-built phrase-tables, bilingual dictionaries and statistical models (fertility, reordering, or language models). By following the carrier sentence approach our method attempts produce a translation hypothesis if it can generate a hypothesis based on a small number of source sentences using only basic editing rules (linear synthesis). We saw that 1 or 2 sources is a reasonable limit.

In terms of computational complexity, the complexity of computing the string edit distance for every sentence in the memory is proportional to the size of the memory and thus it is important to pre-filter the memory in every query to identify potentially useful sentences and compute string edit distances only on that set.

Our experiments demonstrated that as the number of source sentences increases it is possible to cover more query sentences but at the same time more "distortion" is introduced in the translations. Thus future work should focus on improving the edit methods in order to keep the distortion low when using 2 or more source sentences.

## References

Hewavitharana S., Vogel S.,and Waibel A., (2005) Augmenting a Statistical Translation System with a Translation Memory, EAMT 2005

Koehn P., Och, F. J. and Marcu D.. (2003). Statistical phrase-based translation. In *Proc. of the 2003 Conf. of the NAACL on Human Language Technology*.

Langlais P., and Simard M. (2002). Merging Example-Based and Statistical Machine Translation: An Experiment. *In Proceedings of the 5th Conference of Association for Machine Translation in the Americas (AMTA)*.

Marcu, D. (2001). Towards a unified approach to memory- and statistical-based machine translation. *In Proc. of the 39th Meeting on ACL* Toulouse, France.

Och F. J., and Ney H., (2003). "A Systematic Comparison of Various Statistical Alignment Models", Computational Linguistics, volume 29, number 1, pp. 19-51 March 2003.

Sumita, E. (2001). Example-based machine translation using DP-matching between word sequences. In *Proceedings of the Workshop on Data-Driven Methods in Machine Translation - Volume 14* (Toulouse, France, July 07 - 07, 2001).

Veale T. and Way A.. (1997). Gaijin: A template-based bootstrapping approach to example based machine translation. In *Proceedings of "New Methods in Natural Language Processing"*, Sofia, Bulgaria.

Watanabe T., and Sumita E. (2003). Example based Decoding for Statistical Machine Translation. *In Proceedings of MT Summit IX*, New Orleans, LA, USA, September.