Two Approaches to Matching in Example-Based Machine Translation

Sergei Nirenburg, Constantine Domashnev and Dean J. Grannes

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA

Abstract

This paper describes two approaches to matching input strings with strings from a translation archive in the example-based machine translation paradigm - the more canonical "chunking + matching + recombination" method and an alternative method of matching at the level of complete sentences. The latter produces less exact matches while the former suffers from (often serious) translation quality lapses at the boundaries of recombined chunks. A set of text matching criteria was selected to reflect the trade-off between utility and computational price of each criterion. A metric for comparing text passages was devised and calibrated with the help of a specially constructed diagnostic example set. A partitioning algorithm was developed for finding an optimum "cover" of an input string by a set of best-matching shorter chunks. The results were evaluated in a monolingual setting using an existing MT post-editing tool: the distance between the input and its best match in the archive was calculated in terms of the number of keystrokes necessary to reduce the latter to the former. As a result, the metric was adjusted and an experiment was run to test the two EBMT methods, both on the training corpus and on the working corpus (or "archive") of some 6,500 sentences.

The growth rate of theoretical studies of language structure and use stubbornly remains higher than the improvement rate of large-scale applications. It has been repeatedly proved that large-scale realistic NLP applications carry prohibitive price tag of large-scale, routine acquisition of knowledge about language and about the world, collected in computational grammars, lexicons and domain models. Strategically, there are several ways of dealing with this problem:

- biting the bullet and going through a massive knowledge acquisition effort, either general-purpose (e.g., the CYC project, Lenat et al., 1990) or domain-specific (e.g., the KBMT-89 project, Goodman and Nirenburg, 1992
- seeking ways of bringing down the price of knowledge acquisition by studying ways of automatically or semi automatically extracting relevant information from machine-readable dictionaries (morphological, syntactic and some semantic information, e.g., in the work of Wilks et al., 1990) or text corpora (for instance, collocations cf. Smadja, 1991)
- seeking ways of avoiding the need for massive knowledge acquisition by rejecting the entire established NLP paradigm in favor of knowledge-free, linguistics- and AI-independent approaches.

This last option has been energetically promulgated in the important NLP application of machine translation (MT). The two basic "non-traditional" approaches to MT are

- statistical MT, which seeks to carry out translation based on complex cooccurrence and distribution probability calculations over very large aligned bilingual text corpora, and
- a more modest approach, called example-based MT. which is the topic of this paper.

1. Example-Based MT

The basic idea of EBMT is simple (cf. Nagao, 1984): given an input passage *S* in a source language and a bilingual text archive, where text passages *S′* in the source language are stored, aligned with their translations into a target language, passages *T′*, *S* is compared with the source-language "side" of the archive. The "closest" match for passage *S'* is selected and the translation of this closest match, the passage *T′* is accepted as the translation of *S*.

The appeal of the basic idea of EBMT is so high that it has been suggested as the basis for tackling additional tasks such as source language analysis (e.g., Jones. 1992; Furuse and Iida, 1992), source-to-target language transfer (e.g., Grishman and Kosaka. 1992; Furuse and Iida, 1992; Watanabe. 1992) and generation (e.g., Somers, 1992). This marks the advent of hybrid rule-based and example-based MT systems. The hybridization route is chosen in the hope that the resulting systems will have fewer practical shortcomings than the pure rule-based systems (a high complexity of processing plus a high price of knowledge acquisition) or the pure EBMT systems (a very ungraceful degradation curve when matches are bad).

Among the crucial tasks in EBMT are a) selecting the most appropriate *length* of the passages *S* and *S′* and b) establishing the *metric* of similarity between *S* and *S′.* In what follows we analyze these tasks, in turn.

The longer the matched passages, the lower the probability of a complete match (see also the discussion in Nomiyama, 1992. p. 715). The shorter the passages, the greater the probability of ambiguity (one and the same *S′* can correspond to more than one passage *T'*) and the greater the danger that the resulting translation will be of low quality, due to passage boundary friction and incorrect chunking. This is easy to see when a passage is exactly one word long (the minimum reasonable length), since words are typically ambiguous, and word-for-word translation has been repeatedly shown to be inadequate in many ways. In practice, if passage length in a particular EBMT system is user-settable, the optimum passage length should be chosen in accordance with the expected level of similarity of an input text with the resident archive - the greater the expected similarity, the greater the passage length. Alternatively, the optimum length of passages can be derived dynamically, by analyzing the available parallel corpora. This, however, requires an additional procedure — determining the best "cover" of an input text by the passages.

A practical constraint on the selection of passage length is that the more flexibility is allowed in this respect, the more difficult it is to develop a working system (as. for instance, has been experienced by Kaji et al., 1992 who, in order to implement an EBMT algorithm based on a pre-acquired set of translation equivalence patterns, had to forgo the flexibility of matching expressions of the kind suggested by Sato and Nagao, 1990). Indeed, a major hurdle introduced by using passages shorter than a complete sentence is the necessity of finding the optimum cover of an entire passage by a string of matched fragments (see, e.g., Sato and Nagao, 1990, where the passage length parameter is made a component of the similarity metric, see below). This task is exacerbated if there are many possible covers see Maruyama and Watanabe, 1992). The simplest solution, therefore, is to select a single external criterion for partitioning the text into passages. For instance, sentence boundaries can be declared as passage delimiters. In this paper we report experiments on matching both on complete sentences and on passages whose length is determined dynamically.

Turning to the subject of matching an input string with an archive, we first observe that the simplest similarity metric is a complete match, with a Boolean outcome. If an S completely matches an S', then T' is used, otherwise, failure is announced. Even this kind of a system can be very useful in practice, specifically, for what is known in the translation industry as revision control - to allow a translator to work only on the differing sections of two versions of essentially the same document. In the general case, however, an EBMT system cannot rely on complete matches. Therefore, the metric must be able to support more complex judgments of similarity between the input passage and the archive than a simple Boolean outcome. In order to come up with an advanced similarity metric,[1] it is necessary to

1. devise a set of comparison criteria for individual words;
2. assign an individual "penalty" score to outcomes corresponding to each of the comparison criteria;
3. devise a formula for the combination of evidence obtained about single-word comparisons to derive an overall score for an input passage;
4. conduct a calibration experiment to check whether the matching metric built on the chosen criteria produces, on a set of candidate matches, a ranking similar to an independently motivated "canonical" ranking order of matches; if the metric fails to do so. adjust penalty scores and the combination of evidence formula and rerun the experiment;

Two major sources of comparison criteria can be tapped — *string composition comparison* and *relaxation of the concept of matching.* As to the former, two strings S and S' can either fully match or differ as follows:

1. S can be completely included in S' (contiguously or discontiguously);
2. S' can be completely included in S (contiguously or discontiguously):
3. (the general case of the above two), S and S' can have a common (contiguous or discontiguous) substring (possibly, empty).

For each of the above cases we suggest the heuristic that the quality of a match is proportional to a measure of contiguity of matching.

To relax the concept of matching, we allowed an equivalence class of strings to match against the input string instead of the otherwise one-for-one matching. Naturally, a relaxed match is less valuable than a complete match, so that while its probability is greater, its contribution to the overall quality of the match (at the passage level) is smaller. In other words, relaxed matches carry a penalty. The latter is determined individually for each type of relaxation. A possible set of match relaxation criteria (that is. the composition of the equivalence class for comparisons) is presented in Table 1. The table lists: a) equivalence classes for matching, b) an a *priori* set of penalty factors assigned to matches against elements of a given class (see Section 3 for a discussion of the experimental verification of these factors) and c) the background knowledge and tools necessary to support a particular matching criterion.

[1] For reasons of immediate inapplicability, we disregard both statistical and connectionist methods of distance determination. See McLean, 1992, for an experiment on connectionist determination of the distance.

	Equivalence Class	Penalty Factor	Background Knowledge and Tools
1	Exact match	0	
2	Morphological Paradigm of each word	2	A morphological analyzer and a corresponding dictionary
3	Union of morphological paradigms for all parts of speech of each word	3	As above plus a part-of-speech tagger
4	Set of all synonyms of each word (appropriate word sense)	3	a dictionary of synonyms or a thesaurus plus a semantic analyzer with a corresponding dictionary and ontological domain model
5	Set of all synonyms of each word (all senses)	4	a dictionary of synonyms or a thesaurus
6	Set of all hyperonyms of each word (appropriate word sense)	4	a thesaurus plus a semantic analyzer with a corresponding dictionary and ontological domain model
7	Set of all hyperonyms of each word (all senses)	5	an hierarchically organized lexical database or a thesaurus
8	Set of all hyponyms of each word (appropriate word sense)	4	a thesaurus plus a semantic analyzer with a corresponding dictionary and ontological domain model
9	Set of all hyponyms of each word (all senses)	5	an hierarchically organized lexical database or a thesaurus
10	Set of all antonyms of each word (appropriate word sense)	6	a dictionary of antonyms plus a semantic analyzer with a corresponding dictionary and ontological domain model
11	Set of all antonyms of each word (all senses)	7	a dictionary of antonyms
12	Set of part-of-speech tags for each word	9	a part-of-speech tagger and a tagged example archive (can be tagged during operation, too)

Table 1: Some Equivalence Classes for Matching Strings.

The computational price of using equivalence sets 4, 6, 8 and 10 is prohibitive (if one already has a semantic analyzer, it can be put to better use than just supporting EBMT!). The utility of comparing part-of-speech symbol strings (set 12) is very limited (what use would be the match of "Queen of England" with "rate of exchange," both *N Prep N?*) even though this heuristic was used by Furuse and Iida. 1992 and Sumita and Iida (1991), as was synonymy and hyperonymy based on a thesaurus-based hierarchy of semantic markers. The thesaurus-based definition of synonymy is more relaxed than the one intended in equivalence class 4 (it might include *pen, pencil* and *calligraphy,* as having the marker *writing,* in an equivalence set, whereas the latter would not list *calligraphy),* though it might be weakly more selective than the one used in equivalence class 5, which might include both *pencil* and *pigsty* as members of the synonym class of *pen.*

2. The Matching Metric

We decided to use a matching metric based on string composition discrepancies (all three classes above) and match relaxation (using equivalence classes 1, 2, 5 and 7[2]*).* Sentence boundaries were used as passage delimiters, and match relaxation occurred at the level of single words.[3]

The following distance metric was posited originally:

$$S - S' = 10W + 10w + 5H + 4Y + 3M + 0C, \quad (1)$$

where W stands for the number of words in *S* but not in *S';* w, for the number of words in *S'* but not in S; H, for the number of words which matched on the hyper- or hyponym equivalence set; Y, for the number of words which matched on the synonym equivalence set; M for the number of words that matched relative to the morphological paradigm and C for the complete matches. Individual penalties from Table 1 were used, where applicable; the flat penalty of 10 was suggested for each mismatch.

As the source of synonyms, antonyms, hyperonyms and hyponyms, we used the WordNet lexical system developed at Princeton University. For the initial calibration experiment we have used a (small) set of 50 diagnostic sentences selected to assure a relatively high hit rate. The sentence *The top money funds currently provide the opportunity for a high return on an initial investment* was selected as S. The following sentence from the calibration set illustrates how that set was constructed: *Money market funds may currently give the common investor the opportunity for a very high acquisition on an initial investment of a few thousand dollars.* We intermittently used additions, deletions and substitutions of synonyms, hyperonyms and hyponyms.

A central task was to devise a method of determining the "canonical" ranking order of possible matches. As our original desire was to see how EBMT can fit into a practical MT environment, we decided to use an independently motivated "control" metric for checking the quality of the EBMT metric we use. To do so, we decided to use the CMU Translator's Workstation (TWS) (Nirenburg et al., 1992). This TWS includes, among other facilities, a post-editing editor which makes it easy for the user to substitute a word or a phrase in the text by their synonyms, hyperonyms or hyponyms, as well as delete and move text passages using at most two mouse clicks; see Figure 1 for an illustration.

The "control" metric was put together as follows: for each sentence in the example set, we calculated the number of keystrokes it required in the TWS editor to make it completely match the input sentence. Every word deletion was counted as 3 keystrokes; every word substitution using a TWS editor-supported synonym and hyperonym list was also counted as 3 keystrokes; for manual insertion, the number of keystrokes was counted directly.[4]

---

[2] We decided that the use of antonyms is not a clear indicator, as it also requires the appearance of a negation morpheme in one of the comparands, which the program, in this case, erroneously, will count as a separate mismatch. The use of a procedure that would look up antonyms when a negation morpheme is present in the string is too computationally expensive relative to the expected benefit.

[3] Having a phrasal lexicon would improve the matching performance but will significantly add to the computational price of the enterprise, and we simply did not have access to such a lexicon at the moment.

[4] We understand that using the same procedure on a *translation* of each *S'* should be more direct; however, we assumed that the discrepancies

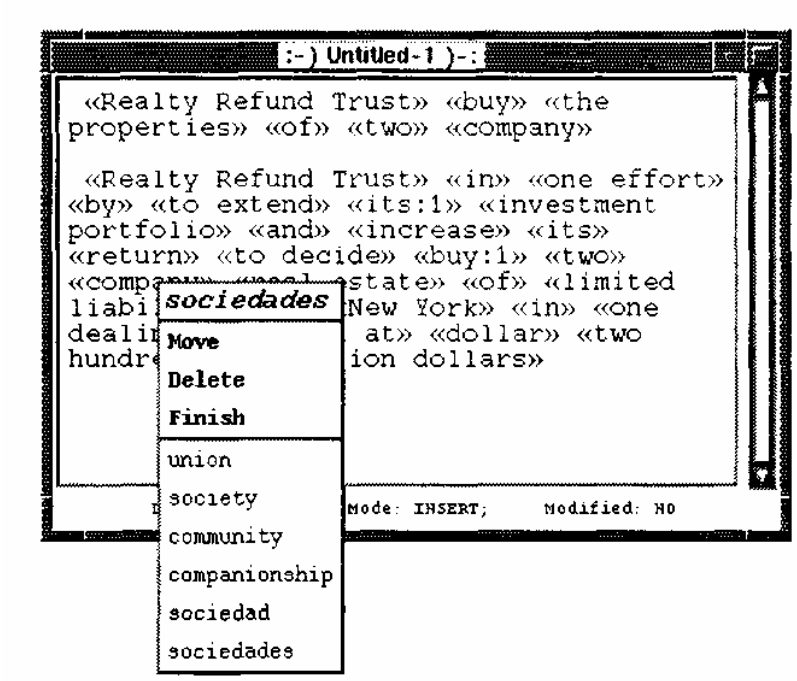

Figure 1: A typical post-editing tool menu. In order to substitute a region from the menu for the one in the text, the user must release the mouse button while the desired substitution is highlighted.

As a result of the calibration test, our distance metric was calibrated to raise the penalty for the case when a word appears in *S* but not in *S′,* reflecting the intuition that it costs more to type a word than to delete it:

$$S - S' = 20W + 10w + 5H + 4Y + 3M + 0C, \quad (2)$$

Of the 16 possible sets of comparison criteria, 6 most closely corresponded to the canonical ordering of the 10 best matches. The distances between the 10 best matches in the calibration set and the input passage ranged from 4 to 23. In terms of the number of keystrokes, the distance contained in the range from 3 to 13. We used one of the 6 best criteria sets to run the comparison of the same input sentence against a corpus of 150,000 words of *The Wall Street Journal* articles. The ten best matches in this test came in with the distances in the range of 120 to 130. They would have required between 82 and 101 keystrokes to be reduced to the original sentence.

The immediate utility of our comparison metric for the type of application we tested can be measured as follows: if the number of keystrokes necessary to "correct" the best match is less than 1/2 of the number of keystrokes necessary to type in the entire passage from scratch, we could consider the approach applicable. (We assume that reading and comparing a best match to the input takes about 1/2 of the time necessary to type in the entire sentence.)

The best match in the test run would have required 82 keystrokes to convert it to the original passage. There are 97 characters in the input sentence. The results of using our method, thus, have not reached our self-imposed threshold, though they were useful for calibration purposes.

---

between *S* and S' and *T* and an acceptable translation *S* are monotonic.

3. The Partitioning Algorithm

Will the quality of EBMT improve if the length of the passage S, instead of being defined by sentence boundaries, is determined dynamically, by the EBMT program itself? In other words, will it be better (and if so, under which conditions) for the EBMT program to accept the burden of a recombination step at target text "generation" time in order to improve overall quality of result? To find an answer to this question, we have developed an algorithm to determine, for any input sentence, an optimum partition into substrings which best match source-language strings in the archive. The partitioning algorithm, by itself, does not, strictly speaking, require the presence of an aligned corpus since it uses only the source-language part of the translation archive. The target-language part is needed only when this algorithm is embedded into an actual EBMT system.

We present only a high-level description of the partitioning algorithm. A complete description treating special cases and a detailed analysis of complexity will be reported separately.

For a source language sentence *S* and each sentence *S′* from the source language corpus, the algorithm first determines all the matching substrings under the following conditions:

- all substrings in each of the sentences are contiguous;
- the matching on each candidate string is performed using the same metric as described above where a match could occur between a word and an equivalence class.

Each match is, thus, given a score, and matches of poor quality (those below an empirically determined threshold) are discarded. With the remaining matches, the algorithm determines the best complete cover of the input passage by substrings from the archive. The result of this procedure is a set of tuples consisting of the boundaries of the substring of the input passage and the number of the sentence from the archive which contains the best match for this substring.

The partitioning algorithm consists of two procedures - one for generating a set of candidate partitions (we will call it *matcher)* and the other, for selecting a single (best) outcome (we will call it *selector).* The matcher takes as input a passage and a corpus in the same text.[5] Each word from the input string is compared, in turn, with the words in the archive, sentence by sentence. The input word is considered to match a word from the archive if the latter belongs to an equivalence class of the former, defined as described in Section 2. The results of this procedure are recorded in a data structure (DS-1) indexed by words in the archive, where for each word the ordinal number of a matching word (or words) in the input string are recorded, together with the score of the corresponding match.

At this point, we start collecting matched substrings of length greater than 1. First, we find all the substrings of the elements in DS-1 whose indices are consecutive (that is, all contiguous substrings). Next, we check whether the lists of their matching words from the input string also form a contiguous substring. The longest contiguous matching string for each archive string in DS-1 is recorded, together with its newly calculated cumulative matching score, the ordinal number of the archive sentence with which the match occurred and the starting position of this substring in the input string, in another data structure, DS-2. DS-2 is indexed by substring length.

Next, the algorithm looks for the best partition of the input string into a string of available matched substrings. We proceed from the assumption that the longer the component substrings, the higher the quality of a partition.[6] The algorithm selects the longest matched substring with the highest score (from DS-2) as a component of the partition. The algorithm is then recursively applied to the remaining substrings of the input string. The result of this process is an optimum partition relative to the basic "longest first" assumption.

The above algorithm requires, for the input string of length *n* on the order of n2 steps. More concretely, for the average sentence length of 22, the number of steps will be 250.

---

[5] In an actual EBMT application, the corpus will be the source-language part of a text archive consisting of aligned passages in two or more languages. We assume that the alignment takes place at the level of sentence.

[6] Experimental results reported in this paper support this assumption. In future, we will make a more complex assumption which will take into account the combined impact of string length and comparison score.

4. Experimenting with the Two EBMT Approaches

Having a) designed and calibrated a matching metric and b) devised a partitioning algorithm, we carried out an experiment to test the metric and the algorithm. Using the same sentence as in the calibration exercise as input, we matched it against both the calibration test suite and the corpus, using both the method of matching on complete sentences and the partitioning method.

We had three different metric settings for the matcher for each mode:

- a graduated scale using equivalence classes, in which morphological variants added a cost of 2 to the score, synonyms added 3, while hyperonyms and hyponyms added 5.
- a flat scale where matches on any members of the equivalence class containing morphological variants, synonyms, hyperonyms and hyponyms were considered equally good.
- exact matches with no equivalence classes.

For the complete sentence method we tabulated a) the sum total of the calibration match scores on each individual word in the input string (in Table 2) and b) the number of keystrokes required to reduce the best-matching sentence found to the input sentence (in Table 3).

For the partitioning method we tabulated a) the number of keystrokes required to reduce the closest match to the input (in Table 4) and b) the number of segments (partitions) needed to cover the input in the best match (in Table 5).

The number of keystrokes is interesting in that it indicates how much work may be involved in post-editing. The number of segments is also an important measure, as the fewer the segments, the larger they are, and hence, the more likely they are to be correct, without requiring much post-editing.

Calibrated Score	Graduated Scale	Any Match	Exact Match
Short Corpus	4	0	10
Long Corpus	129	129	118

Table 2: Score of Best-Scoring Matches for Complete Sentence Method

Keystrokes	Graduated Scale	Any Match	Exact Match
Short Corpus	3	3	3
Long Corpus	104	104	111

Table 3: Number of Keystrokes of Best-Scoring Matches for Complete Sentence Method

Keystrokes	Graduated Scale	Any Match	Exact Match
Short Corpus	3	3	3
Long Corpus	7	7	0

Table 4: Number of Keystrokes of Best-Scoring Matches for Partition Method

Based on the above results, we can calculate a final estimate of the quality of the match, the "distance" between the source passage and the passage returned by the system. We devise this measure as the number of keystrokes necessary to reduce, using the CMU TWS, the passage returned by the system to the input passage *divided by* the number of characters in the input string,

Segments	Graduated Scale	Any Match	Exact Match
Short Corpus	1	1	2
Long Corpus	11	11	11

Table 5: Score of Best-Scoring Matches for Partition Method

$$Distance = \frac{number\ of\ keystrokes}{number\ of\ characters\ in\ input}$$

(When the target language is connected to the system, the measure of distance will become more complex, as at least the following parameters will have to he factored in - a) a penalty for segmentation of the matched string (the greater the number of partitions, the worse the total quality of the match); b) a penalty for the presence of very short segments in the matched string (translation of short segments will be more ambiguous than long segments) and c) a penalty for unmatched words.)

For our results, the distances are tabulated in Table 6.

Distances	Graduated Scale	Any Match	Exact Match
Complete Sentence / Short Corpus	0.04	0.04	0.04
Complete Sentence / Long Corpus	1.3	1.3	1.4
Partitioning / Short Corpus	0.04	0.04	0.04
Partitioning / Long Corpus	0.1	0.1	0

Table 6: Score of Best-Scoring Matches for Partition Method

5. Discussion

The results suggest that the use of equivalence classes may have a smaller impact on the outcome than initially expected. However, before making a final judgment on this issue, we intend to conduct more extensive experiments, with more input sentences, different corpora and, possibly, automatically generated specialized test suites. An additional piece of experimental evidence corroborating our initial decision to use equivalence classes is the fact that exact matching has been shown to lead to the decrease in the length of segments, which is, as we have already mentioned, a negative factor.

The partitioning algorithm produced some interesting results. Under the experiments in which the graduated scale and the "lenient" scale (morphological variants, synonyms, hyperonyms and hyponyms all counted as well as an exact match) were used, the cover turned out to be the same: the sentence "the top money funds currently provide the opportunity for a high return on an initial investment" was covered as, "the top" "money funds" "currently" "provide the" "opportunity" "for" "a" "high" "return" "on an" "initial investors". The 7 keystrokes shown in Table 4 are required to turn "investors" to "investment". However, when the constraints were tightened, and the words had to match exactly, the cover changed to: "the top" "money funds" "currently" "provide" "the opportunity" "for a" "high" "return" "on an" "initial" "investment". By changing the cover, the sentence was made to match exactly, without being penalized for requiring more segments. This possible cover existed for the earlier two experiments, but, because of the greedy nature of the cover algorithm, once the other (less optimal) cover was found, it was accepted as sufficient.

5.1. Future Directions

There are a large number of dimensions along which the work reported here can and will develop.

First of all, we need to move to a full-fledged EBMT environment, which means working with a real bilingual archive. The immediate problem to be solved then is the problem of alignment of the archive. If the full-sentence comparison method is used, it is sufficient to have the archive aligned at sentence level. If, however, the partitioning method is used, it becomes necessary to obtain alignment at the sub-sentential level. This latter task is, in fact, exactly the goal of the full-fledged statistical MT approaches. Results in text alignment have been achieved at IBM (e.g. Brown et al., 1990) and AT&T (e.g., Gale and Church, 1991). In the short run, the quality of sub-sentential alignment does not promise high-enough fidelity to support EBMT in a stable fashion. Because of this (and unconditionally for the full-sentence comparison method) a practical EBMT environment will have to involve a user interface, similar to the CMU TWS, to allow the human user to correct system output.

A second avenue of improvement is upgrading the matching and partitioning algorithms and metrics. We have immediate plans to improve our partitioning algorithm by a) optimizing the choice of the longest substring; and b) accepting discontinuous substrings as candidates for partitioning. Among the possibilities for improving the metric are: a) diversifying the treatment of open- and close-class lexical items: a match on the latter can be considered less significant than a match on the former; b) allowing a bonus for "clustered" matches, where a contiguous subset of a string matches an example, compared with the match on a similar number of discontiguous words; c) further calibrating the ratios in the metric definition by repeatedly modifying them and running the calibration test on a large number of ratio combinations, to choose the one which leads to the optimum correspondence with the results obtained using the "control" metric; d) augmenting the set of comparison criteria, for instance, by including membership in the same semantic class (as did Sumita and Iida, 1991), though this criterion is inherently weaker than synonymy or direct hyperonymy: this enhancement presupposes the availability of a thesaurus or another source of semantic class markers.

Yet another major area of improvement has to do with creating a complete experimental set-up which will allow for fast and abundant calibration of all the parameters of the EBMT environment, which would allow us to adapt the system to a particular set of texts and archives. This experimental testbed will also serve as an evaluation testbed for the quality of the EBMT system itself.

Additional studies must be conducted to calculate the optimum tradeoffs of EBMT utility and robustness versus the complexity of requisite static and dynamic knowledge sources. EBMT researchers should always remember the lesson of the development of the field of qualitative reasoning in AI, which set out to simplify the very intricate and involved theories in physics and other natural sciences by relying on commonsense reasoning and ended up with a set of theories whose formulation was arguably even more intricate and difficult to use for reasoning than the original ones. For EBMT to succeed, it should be shown not to rely on an extensive apparatus of linguistic and domain-oriented language analysis, which forms the basis of the "traditional" rule-based MT, which EBMT set out to supplant.

5.2. Related Work

The only detailed description of a proposal for solving the partitioning problem has been reported by Maruyama and Watanabe (1992). They describe an algorithm for determining the best partitioning ("cover") of a source language sentence in an environment which, though not strictly EBMT, is closely related to EBMT. The input to their algorithm is a) a source language sentence and b) a set of pairs consisting of a substring of the input sentence and the translation of this substring into a target language. The output of their algorithm is essentially, a string of source language substrings which completely covers the entire input sentence. (The substitution of translations for these substrings is, then, a trivial step.) The task which we set out to perform is more general than the one described in *op. cit.,* along each possible dimension of comparison. Our assumptions have consistently been less constraining than those used by Maruyama and Watanabe. To name just a few, in addition to the operations carried out by Maruyama and Watanabe's system, we also a) determine the set of substrings that best match the input - instead of stipulating the prior existence of the set of best matches; b) allow a fuzzy, incomplete match between input and corpus; and c) do not proceed from the assumption (as do Maruyama and Watanabe) that each substring is relatively short. In a later report we will analyze

these differences in greater detail.

Bibliography

Brown. P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R.L. and Roossin P.S. 1990. A statistical approach to language translation, *Computational Linguistics,* vol 16, 79-85.

Goodman, K. and S. Nirenburg. 1992. **KBMT-89:** A **Case Study in Knowledge-Based Machine Translation.** San Mateo, CA: Morgan Kaufmann.

Gale, W. and K. Church. 1991. Identifying word correspondence in parallel text. Proceedings of the DARPA NLP Workshop.

Grishman, R. and M. Kosaka. 1992. Combining rationalist and empiricist approaches to machine translation. Proceedings of TMI-92. Montreal. 263-74

Jones, D. 1992. Non-hybrid example-based machine translation architectures. Proceedings of TMI-92. Montreal. 163-71.

Lenat, D. and R. Guha. 1990. **Building Large Knowledge-Based Systems.** Reading, MA: Addison-Wesley.

Maruyama, H. and H. Watanabe. 1992. Tree cover search algorithm for example-based translation. Proceedings of TMI-92. Montreal. 173-84.

McLean, I. 1992. Example-based machine translation using connectionist matching. Proceedings of TMI-92. Montreal. 35-43.

Nagao, M. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In: A. Elithorn and R. Banerji (eds.) **Artificial and Human Intelligence.** NATO Publications.

Nirenburg, S., P. Shell, A. Cohen, P. Cousseau, D. Grannes and C. McNeilly. 1992. Multi-Purpose Development and Operation Environments for Natural Language Generation. Proceedings of the Third Conference on Natural Language Applications. Trento, Italy. April.

Nomiyama, H. 1992. Machine translation by case generalization. Proceedings of COLING-92, Nantes, 714-20.

Sato, S. and M. Nagao. 1990. Towards memory based translation. Proceedings of COLING-90, Helsinki, Volume 3, 247-52.

Sadler, V. 1989. **Working With Analogical Semantics.** Dordrecht: Foris.

Smadja, F. 1991. From N-Grams to Collocations: An Evaluation of Xtract. Proceedings of 29th ACL Meeting. Berkeley.

Somers, H. 1992. Interactive multilingual text generation for a monolingual user. Proceedings of TMI-92. Montreal. 151-61.

Sumita, E. and H. Iida. 1991. Experiments and prospects of example-based machine translation. Proceedings of 29th ACL Meeting. Berkeley.

Watanabe, H. 1992. A similarity-driven transfer system. Proceedings of Coling-92, Nantes, 770-76.

Wilks, Y., D. Fass, C Guo, J. McDonald, T. Plate, and B. Slator. 1990. Providing Machine Tractable Dictionary Tools. Machine Translation, Volume 5, Number 2, 99-155.