

# Toward Multi-Engine Machine Translation

Sergei Nirenburg and Robert Frederking

Center for Machine Translation  
Carnegie Mellon University  
Pittsburgh, PA 15213

## ABSTRACT

Current MT systems, whatever translation method they at present employ, do not reach an optimum output on free text. Our hypothesis for the experiment reported in this paper is that if an MT environment can use the best results from a variety of MT systems working simultaneously on the same text, the overall quality will improve. Using this novel approach to MT in the latest version of the Pangloss MT project, we submit an input text to a battery of machine translation systems (engines), collect their (possibly, incomplete) results in a joint chart-like data structure and select the overall best translation using a set of simple heuristics. This paper describes the simple mechanism we use for combining the findings of the various translation engines.

## 1. INTRODUCTION

A number of proposals have come up in recent years for hybridization of MT. Current MT projects — both “pure” and hybrid, both predominantly technology-oriented and research-oriented are single-engine projects, capable of one particular type of source text analysis, one particular method of finding target language correspondences for source language elements and one prescribed method of generating the target language text.

It is common knowledge that MT systems, whatever translation method they at present employ, do not reach an optimum output on free text. In part, this is due to the inherent problems of a particular method — for instance, the inability of statistics-based MT to take into account long-distance dependencies or the reliance of most transfer-oriented MT systems on similarities in syntactic structures of the source and the target languages. Another crucial source of deficiencies is the size and quality of the static knowledge sources underlying the various MT systems — particular grammars, lexicons and world models. Thus, in knowledge-based MT the size of the underlying world model is typically smaller than necessary for secure coverage of free text.

Our hypothesis for the experiment reported in this paper is that if an MT environment can use the best results from a variety of MT systems working simultaneously on the same text, the overall quality will improve. Using this novel approach to MT in the latest version of the Pangloss MT project, we submit an input text to a battery of machine translation systems (engines), collect their (possibly, incomplete) results in a joint chart-like data structure and select the overall best translation using a set of simple heuristics.

## 2. INTEGRATING MULTI-ENGINE OUTPUT

The MT configuration in our experiment used three MT engines:

- a knowledge-based MT (KBMT) system, the mainline Pangloss engine[1];
- an example-based MT (EBMT) system (see [2, 3]; the original idea is due to Nagao[4]); and
- a lexical transfer system, fortified with morphological analysis and synthesis modules and relying on a number of databases — a machine-readable dictionary (the Collins Spanish/English), the lexicons used by the KBMT modules, a large set of user-generated bilingual glossaries as well as a gazetteer and a list of proper and organization names.

The results (target language words and phrases) were recorded in a *chart* whose initial edges corresponded to words in the source language input. As a result of the operation of each of the MT engines, new edges were added to the chart, each labeled with the translation of a segment of the input string and indexed by this segment's beginning and end positions. The KBMT and EBMT engines also carried a quality score for each output element. Figure 1 presents a general view of the operation of our multi-engine MT system.

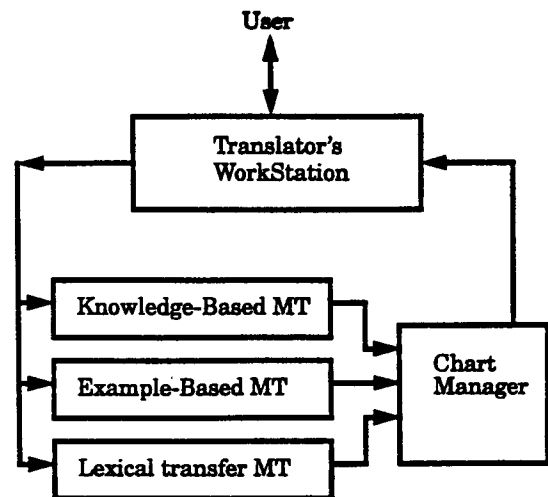


Figure 1: Structure of a multi-engine MT system

In what follows we illustrate the behavior of the system using the example Spanish sentence: *Al momento de su venta a Iberia, VIASA contaba con ocho aviones, que tenían en promedio 13 años de vuelo* which can be translated into English as *At the moment of its sale to Iberia, VIASA had eight airplanes, which had on average thirteen years of flight (time)*. This is a sentence from one of the 1993 ARPA MT evaluation texts.

The initial collection of candidate partial translations placed in the chart for this sentence by each individual engine are shown in Figures 2, 3, 4, 5, and 6. The chart manager selects the overall best cover from this collection of candidate partial translations by providing each edge with a normalized positive *quality score* (larger being better), and then selecting the best combination of edges with the help of the *chart-walk* algorithm.

Position		Input (Spanish)	Output (English)
Left	Right		
0	1	Al momento	"A moment"
2	4	de su venta	"sale."
5	6	a Iberia	Iberia
8	8	VIASA	Viasa
11	12	ocho aviones	eight airplane
15	15	tenían	do.
16	21	en promedio 13 años de vuelo	"thirteen in"

Figure 2: Knowledge-Based MT (KBMT) candidates

Position		Input (Spanish)	Output (English)
Left	Right		
19	21	años de vuelo	"flight activities" "of years"
19	21	años de vuelo	"years of experience with space flight"

Figure 3: Example-Based MT (EBMT) candidates

Position		Input (Spanish)	Output (English)
Left	Right		
1	1	momento	time moment hour momentum
3	3	su	his her your their its
4	4	venta	sale
6	6	Iberia	Iberia
9	10	contaba con	"count on" have
12	12	aviones	airplane
18	18	13	13

Figure 4: Transfer-Based MT (lexicon candidates)

## 2.1. Scoring the outputs of MT engines

The scores in the chart are normalized to reflect the empirically derived expectation of the relative quality of output produced by a particular engine. In the case of KBMT and EBMT, the pre-existing scores are modified, while edges from other engines receive scores determined by a constant for each engine.

These modifications can include any calculation which can be made with information available from the edge. For example, currently the KBMT scores are reduced by a constant, except for known erroneous output, which has its score set to zero. The EBMT scores initially range from 0 being perfect to 10,000 being totally bad; but the quality is nonlinear. So a region selected by two cutoff constants is converted by a simple linear equation into scores ranging from zero

Position		Input (Spanish)	Output (English)
Left	Right		
0	0	Al	Al
0	1	Al momento	"In a minute" "At once"
3	3	su	his her its one's your their
4	4	venta	inn sale selling marketing
6	6	Iberia	Iberia
7	7	,	NIL
9	9	contaba	"was count" count
9	10	contaba con	"was rely on" "rely on"
			"was count on" "count on" "was depending on" "depended on"
13	13	,	NIL
15	15	tenían	"were have" have "were hold" hold "were thinking" thought "were considering" considered "were deeming" deemed "were coming" came
17	17	promedio	average mean middle midpoint
19	19	años	year
21	21	vuelo	flight

Figure 5: Transfer-Based MT (glossary candidates)

to a normalized maximum EBMT score. Lexical transfer results are scored based on the reliability of individual glossaries.

In every case, the base score produced by the scoring functions is multiplied by the length of the candidate in words, on the assumption that longer items are better. This may be producing too large an effect on the chart-walk. We intend to test functions other than multiplication in order to find the right level of influence for length.

The scoring functions represent all of the chart manager's knowledge about relative quality of edges. Once the edges are scored, the cover is produced using a simple dynamic programming algorithm, described below.

## 2.2. The chart-walk algorithm

Figure 7 presents the chart-walk algorithm used to produce a single, best, non-overlapping, contiguous combination of the available component translations. The algorithm uses dynamic programming to find the optimal cover (a cover with the best cumulative score), assuming correct component quality scores. The code is organized as a recursive divide-and-conquer procedure: for each position within a segment, the sentence is split into two parts, the best possible cover for each part is recursively found, and the two scores are combined to give a score for the chart-walk containing the two best subwalks. This primitive step is repeated for each possible top-level split of the input sentence, compared with each other and with any simple edges (from the chart) spanning the segment, and the overall best result is used.

Without dynamic programming, this would have a combinatorial time complexity. Dynamic programming utilizes a large array to store partial results, so that the best cover of any given subsequence is only computed once; the second time that a recursive call would compute the same result, it is retrieved from the array instead. This reduces the time complexity to polynomial, and in practice it uses an

Position		Input	Output
Left	Right	(Spanish)	(English)
0	0	Al	"To the" "To it" "To him" "To you"
1	1	momento	moment instant time "just a moment!" "in due time" "in due course" "when the time is right" momentum consequence importance
2	2	de	of from about for by
3	3	su	its his her one's your their
4	4	venta	sale selling marketing country inn small shop stall booth
5	5	a	to a of
6	6	Iberia	NIL
7	7	,	,
8	8	VIASA	VIASA
9	9	contaba	"was count" count "number off" "was include" include "count in" "was reckon" reckon "was consider" consider
10	10	con	with by although in toward
11	11	ocho	eight eighth
12	12	aviones	aeroplanes planes aircrafts airplanes martins hopscotch
13	13	,	,
14	14	que	who that whom which
15	15	tenían	"were have" have "have got" "were possess" possess "were hold" hold "hold on to" "hold up" "were grasp"
16	16	en	in on onto at by
17	17	promedio	average middle mid-point
18	18	13	NIL
19	19	años	years
20	20	de	of from about for by
21	21	vuelo	flight "to dash off" "to clear off" "to leave the parental nest" "spread one's wings" "to overhear sth in passing" "to catch on immediately" "get it at once" "to be pretty smart" 'flight feathers"
22	22	.	.

Figure 6: Transfer-Based MT (MRD candidates)

insignificant part of total processing time.

The combined score for a sequence of edges is the weighted average of their individual scores. Weighting by length is necessary so that the same edges, when combined in a different order, produce the same combined scores. In other words, whether edges a, b, and c are combined as ((a b) c) or (a (b c)), the combined edge must have the same score, or the algorithm can produce inconsistent results.

The chart-walk algorithm can also be visualized as a task of filling a two-dimensional array. The array for our example sentence is shown in Figure 8. Element  $(i, j)$  of the array is the best score for any set of edges covering the input from word  $i$  to word  $j$ . (The associated list of edges is not shown, for readability.) For any position, the score is

To find best walk on a segment:

```

if there is a stored result for this segment
  then return it
  else
    begin
      get all primitive edges for this segment
      for each position p within this segment
        begin
          split segment into two parts at p
          find best walk for first part
          find best walk for second part
          combine into an edge
        end
      find maximum score over all primitive
      and combined edges
    store and return it
  end

```

Figure 7: Chart-walk algorithm

calculated as a weighted average of the scores in the row to its left, in the column below it and the previous contents of the array cell for its position. So to calculate element (1,4), we compare the combined scores of the best walks over (1,1) and (2,4), (1,2) and (3,4), and (1,3) and (4,4) with the scores of any chart edges going from 1 to 4, and take the maximum. When the score in the top-right corner is produced, the algorithm is finished, and the associated set of edges is the final chart-walk result.

It may seem that the scores should increase towards the top-right corner. In our experiment, however, this has not generally been the case. Indeed, the system suggested a number of high-scoring short edges, but many low-scoring edges had to be included to span the entire input. Since the score is a weighted *average*, these low-scoring edges pull it down. A clear example can be seen at position (18,18), which has a score of 15. The scores above and to its right each average this 15 with a 5, for total values of 10.0, and the score continues to decrease with distance from this point as one moves towards the final score, which does include (18,18) in the cover.

### 2.3. Reordering components

The chart-oriented integration of MT engines does not easily support deviations from the linear order of the source text elements, as when discontinuous constituents translate contiguous strings or in the case of cross-segmental substring order differences. Following a venerable tradition in MT, we used a target language-dependent set of postprocessing rules to alleviate this problem (e.g., by switching the order of adjectives and nouns in a noun phrase if it was produced by the word-for-word engine).

## 3. TRANSLATION DELIVERY SYSTEM

Results of multi-engine MT were fed in our experiment into a translator's workstation (TWS)[5], through which a translator either approved the system's output or modified it. The main option for human interaction in TWS currently is the Component Machine-Aided Translation (CMAT) editor[6]. A view of this editor is presented in Figure 9. (The user can see the original source language text in another editor window.) The user can use menus, function keys and mouse clicks to change the system's initially chosen candidate trans-

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	5	10	7.3	6.75	6.4	5.6	5.57	5.5	5.1	5.1	6.0	5.66	5.42	5.39	5.16	5.15	4.97	4.97	5.5	5.47	5.31	5.96	5.78
1		2.5	2.25	3.16	3.62	3.3	3.58	3.78	3.56	3.72	4.85	4.59	4.42	4.46	4.29	4.33	4.19	4.24	4.83	4.84	4.7	5.41	5.25
2			2	3.5	4.0	3.5	3.8	4.0	3.71	3.87	5.11	4.8	4.59	4.63	4.42	4.46	4.3	4.34	4.97	4.97	4.82	5.55	5.38
3				5	5.0	4.0	4.25	4.4	4.0	4.14	5.5	5.11	4.85	4.86	4.63	4.65	4.46	4.5	5.16	5.15	4.97	5.74	5.55
4					5	3.5	4.0	4.25	3.8	4.0	5.57	5.13	4.83	4.85	4.59	4.63	4.42	4.46	5.16	5.16	4.97	5.78	5.58
5						2	3.5	4.0	3.5	3.8	5.66	5.14	4.81	4.83	4.55	4.59	4.38	4.42	5.18	5.16	4.97	5.83	5.61
6							5	5.0	4.0	4.25	6.4	5.66	5.21	5.18	4.83	4.85	4.59	4.62	5.42	5.39	5.16	6.06	5.82
7								5	3.5	4.0	6.75	5.8	5.25	5.21	4.81	4.83	4.55	4.59	5.45	5.42	5.17	6.13	5.87
8									2	3.5	7.33	6.0	5.3	5.25	4.78	4.81	4.5	4.55	5.5	5.45	5.19	6.21	5.93
9										5	10	7.33	6.12	5.9	5.25	5.21	4.81	4.83	5.85	5.77	5.45	6.54	6.21
10											2	2.0	2.16	2.87	2.7	3.08	2.92	3.18	4.5	4.55	4.31	5.58	5.31
11												2	2.25	3.16	2.87	3.3	3.08	3.35	4.81	4.83	4.55	5.91	5.58
12													2.5	3.75	3.16	3.62	3.3	3.58	5.21	5.18	4.83	6.30	5.91
13														5	3.5	4.0	3.5	3.8	5.66	5.57	5.12	6.72	6.25
14															2	3.5	3.0	3.5	5.8	5.66	5.14	6.94	6.39
15																5	3.5	4.0	6.75	6.4	5.66	7.64	6.94
16																	2	3.5	7.33	6.75	5.8	8.09	7.22
17																		5	10.0	8.33	6.75	9.30	8.09
18																			15	10.0	7.33	10.3	8.70
19																				5	3.5	8.84	7.13
20																					2	3.5	3.0
21																						5	3.5
22																							2

Figure 8: Triangular array produced by chart-walk

lation string, as well as perform both regular and enhanced editing actions.

The phrases marked by double angle brackets are "components", each of which is the first translation from a candidate chosen by the chart-walk. In the typical editing action shown, the user has clicked on a component to get the main CMAT menu. This menu shows the corresponding source text, and provides several functions (such as moving or deleting the whole constituent) and alternate translations, followed by the original source text as an option. If the user selects an alternate translation, it instantly replaces the component in the editor window, which becomes the first alternative in this menu if it is used again. The alternate translations are the other translations from the chosen edge<sup>1</sup>.

Figure 10 presents the sets of candidates in the best chart-walk that are presented as choices to the human user through the CMAT editor in our example. It also shows their individual engine-level quality scores.

#### 4. TESTING AND EVALUATING MULTI-ENGINE PERFORMANCE

As a development tool, it is useful to have an automatic testing procedure that would assess the utility of the multi-engine system relative to the engines taken separately. The best method we could come up with was counting the number of keystrokes, in an advanced text processor, such as the TWS, necessary to convert the outputs of individual engines and the multi-engine configuration to a "canonical" human translation. A sample test on a passage of 2060 characters from the June 1993 evaluation of Pangloss is shown in figure 11.

The difference in keystrokes was calculated as follows: one keystroke for deleting a character; two keystrokes for inserting a character; three keystrokes for deleting a word (in an editor with

<sup>1</sup>The CMAT editor may also include translations from other candidates, lower in the menu, if they have the same boundaries as the chosen candidate and the menu is not too long.

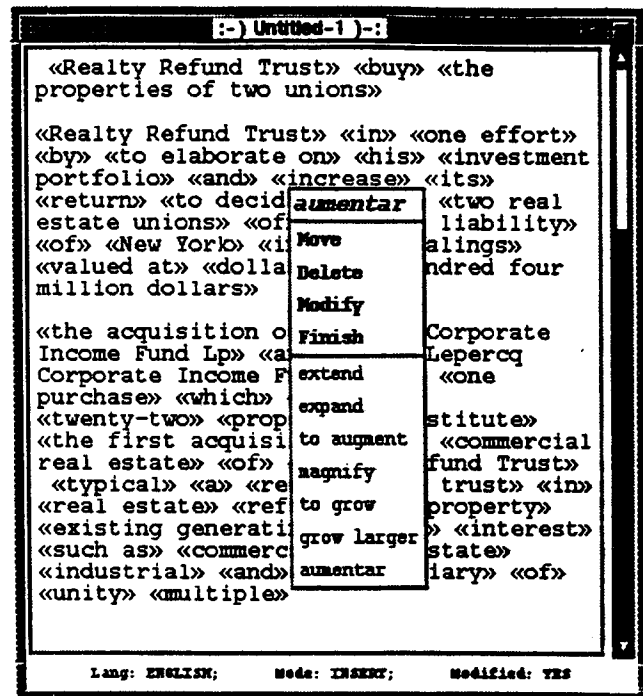


Figure 9: The TWS CMAT editor (main menu)

mouse action); three keystrokes plus the number of characters in the word being inserted for inserting a word. It is clear from the above table that the multi-engine configuration works better than any of our available individual engines, though it still does not reach the quality of a Level 2 translator. It is clear that using keystrokes as a measure is not completely satisfactory under the given conditions. It would be much better to make the comparison not against a single

Position		Input (Spanish)	Output (English)	Engine	Score
Left	Right				
0	1	Al momento	"In a minute" "At once" "A moment"	GLOSS	10
2	2	de	of from about for by	MRD	2
3	3	su	his her its	GLOSS	5
4	4	venta	one's your their inn sale selling marketing "country inn" "small shop" stall booth	GLOSS	5
5	5	a	to a of	MRD	2
6	6	Iberia	Iberia	GLOSS	5
7	7	,	,	GLOSS	5
8	8	VIASA	VIASA	MRD	2
9	10	contaba con	"was rely on" "rely on" "was count on" "count on" "was depending on" "depended on" have	GLOSS	10
11	11	ocho	eight eighth	MRD	2
12	12	aviones	airplane aeroplanes planes aircrafts airplanes martins	MTLEX	2.5
13	13	,	hopscotches	GLOSS	5
14	14	que	who that whom which	MRD	2
15	15	tenían	"were have" "have" "were hold" hold "were thinking" thought "were considering" considered "were deeming" deemed "were coming" came	GLOSS	5
16	16	en	in on onto at by	MRD	2
17	17	promedio	average mean middle midpoint mid-point	GLOSS	5
18	18	13	13	MTLEX	15
19	21	años de vuelo	"years of experience with space flight" "flight activities" "of years"	EBMT	8.85
22	22	.	.	MRD	2

Figure 10: Chart-walk results

"canonical" translation but against a set of equivalent paraphrastic translations, the reason being that, as all translators know, there are many "correct" ways of translating a given input, so that a more appropriate test would be counting the number of keystrokes of difference between the system output and the *closest* member of the set of correct translation paraphrases. However, this is predicated on the availability of a "paraphraser" system, developing which is not a trivial task.

Type of translation	Number of keystrokes to convert to canonical translation
human tester (US Government Level 2 translator)	1542
word-for-word lookup in MRDs	1829
lookup in phrasal glossaries	1973
KBMT	1883
Example-Based MT	1876
Multi-engine configuration	1716

Figure 11: Results of keystroke test

## 5. FUTURE WORK

Ultimately, a multi-engine system depends on the basic quality of each particular engine. We expect the performance of some of the individual engines (especially, KBMT and EBMT) to grow. Consequently, the multi-engine environment will improve, as larger static knowledge sources are added and the scoring mechanism is further adjusted. We expect to gain insight into how to improve the scoring mechanism: we plan a battery of tests to help adjust the coefficients on the function which combines the individual scores in the final score. We plan to use a standard regression mechanism to modify these scores based on feedback from having humans select the best covers for test texts. We expect such calibration further to optimize the system to produce the best possible output from the set of available candidate translations produced by the multiple engines. We also intend to develop a method of how to empirically assess the expected output quality of each translation engine based on its available resources, such as dictionaries, glossaries, grammars, parallel corpora, etc.

## References

- Frederking, R., A. Cohen, P. Cousseau, D. Grannes and S. Nirenburg. "The Pangloss Mark I MAT System." Proceedings of EACL-93, Utrecht, The Netherlands, 1993.
- Nirenburg, S. C. Domashnev and D.J. Grannes. "Two Approaches to Matching in Example-Based Machine Translation." Proceedings of TMI-93, Kyoto, 1993.
- Nirenburg, S., S. Beale, C. Domashnev and P. Sheridan. "Example-Based Machine Translation of Running Text." *In preparation*.
- Nagao, M. "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, 1984.
- Cohen, A., Cousseau, P., Frederking, R., Grannes, D., Khanna, S., McNeilly, C., Nirenburg, S., Shell, P., Waeltermann, D. *Translator's WorkStation User Document*, Center for Machine Translation, Carnegie Mellon University, 1993.
- Frederking, R., Grannes, D., Cousseau, P., and Nirenburg, S. "An MAT Tool and Its Effectiveness." In Proceedings of the DARPA Human Language Technology Workshop, Princeton, NJ, 1993.