Applying Statistical English Language Modeling to Symbolic Machine Translation

Ralf Brown and Robert Frederking
ralf+@cs.cmu.edu, ref+@cs.cmu.edu

Center for Machine Translation
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890 USA

Abstract

The PANGLOSS Mark III system [Frederking et al. 94] was from the outset designed to be a symbolic, human-aided machine translation (MT) system. The need arose to rapidly adapt it for use as a fully-automated MT system. Our solution to this problem was to add a statistical English language model (ELM) to replace the most significant user activity, selecting between alternate translations produced by the system. The language model used is a trigram model with backoff to bigram and unigram probabilities. The language modeling and search procedure are described in detail, and comparison is made to other trigram-based statistical MT work.

1 Introduction

Machine translation (MT) systems may be divided into those that are designed to work interactively with a user and those that are designed to be fully automatic. In the current state of the art, only human-aided machine translation (HAMT) can produce high-quality output on unrestricted texts. Yet there are times when fully-automated machine translation (FAMT) may be preferred; for example, if inexpensive but low-quality MT output is required, or the context in which the system is used rules out human interaction.

Another dimension in which MT systems differ is the degree to which they use symbolic versus statistical information. Systems may be purely statistical, purely symbolic, or attempt to merge the two somehow. And of course there are a wide variety of methods within each category; symbolic methods range from simple transfer systems to complex knowledge-based systems involving numerous levels of analysis and generation, while statistical schemes include methods as diverse as probabilistic grammars and purely statistical translation models.

The work presented here adapts the PANGLOSS Mark III system [Frederking et al. 94, Nirenburg et al. 95a], which is fundamentally designed as a symbolic HAMT system, to work as a FAMT system. It does this through the addition of a statistical English language model (ELM), thus demonstrating one way to combine statistical and symbolic techniques.

The version of PANGLOSS described here translates from Spanish to English. The PANGLOSS project is distributed between three sites: the Computing Research Laboratory of New Mexico State University, the Information Sciences Institute of the University of Southern California, and the Center for Machine Translation of Carnegie Mellon University.

2 Multi-engine MT and statistical models

The PANGLOSS Mark III system (see Figure 1) is based on the multi-engine approach to MT [Frederking & Nirenburg 94]: several MT engines, each employing a different MT technology, are applied in parallel to each input text. Each engine attempts to translate the entire input text, segmenting each sentence in whatever manner is most appropriate for its technology, and putting the resulting output segments into a shared chart data structure after giving each segment a score indicating the engine's internal assessment of the quality of the output segment. The output segments are indexed in the chart based on the positions of the corresponding input segments[1].

In normal HAMT mode, a dynamic programming algorithm [Frederking & Nirenburg 94] determines the best cover of the input based on the scores of the available edges, after normalizing between engines. The user is then presented with this best cover as a set of component phrases in a specialized posteditor window, where a menu-based interface embedded in a normal text editor is used to select between different alternate translations. Using this system, a high-quality translation can be produced with a 40% reduction in human time compared to manual translation using a text editor [White & O'Connell 94].

In order to produce FAMT output, at a minimum the user selection of alternatives must somehow be replaced. Of course, if the reliability of the scores were very good, the first alternative would generally be correct, and the selection step could be skipped. Unfortunately, in the current state of the art, perfect scores are unavailable. What is worse, a major component of the current system is a simple transfer-based engine, with 176,000 glossary entries and a machine-readable dictionary as knowledge-bases; this system does not provide a different score for each different output, and it was difficult for us to see how to reasonably make it do so.

Fortunately we can make an analogy at this point to speech recognition work. There, acoustic recognizers produce many hypotheses for each word, with scores that are not always very accurate. The standard approach in the speech community is to apply a statistical

[1]While all the MT engines in the current system are symbolic, the architecture is fully general, and a statistical translation system could be included as another engine.

language model to such results [Katz 87]. Intuitively, what one does is to analyze large amounts of, for example, English text to determine what the statistically most probable sequences of words are in English. This produces a statistical model. One uses the model to select among the available choices for each word by finding the set of choices that produces the sequence most likely to be an English sentence, taking into account the (acoustic) scores of the component words. Normally one looks at the statistics of all sequences of 3 words (trigrams), because models based on longer sequences are intractable and require excessive amounts of training data.

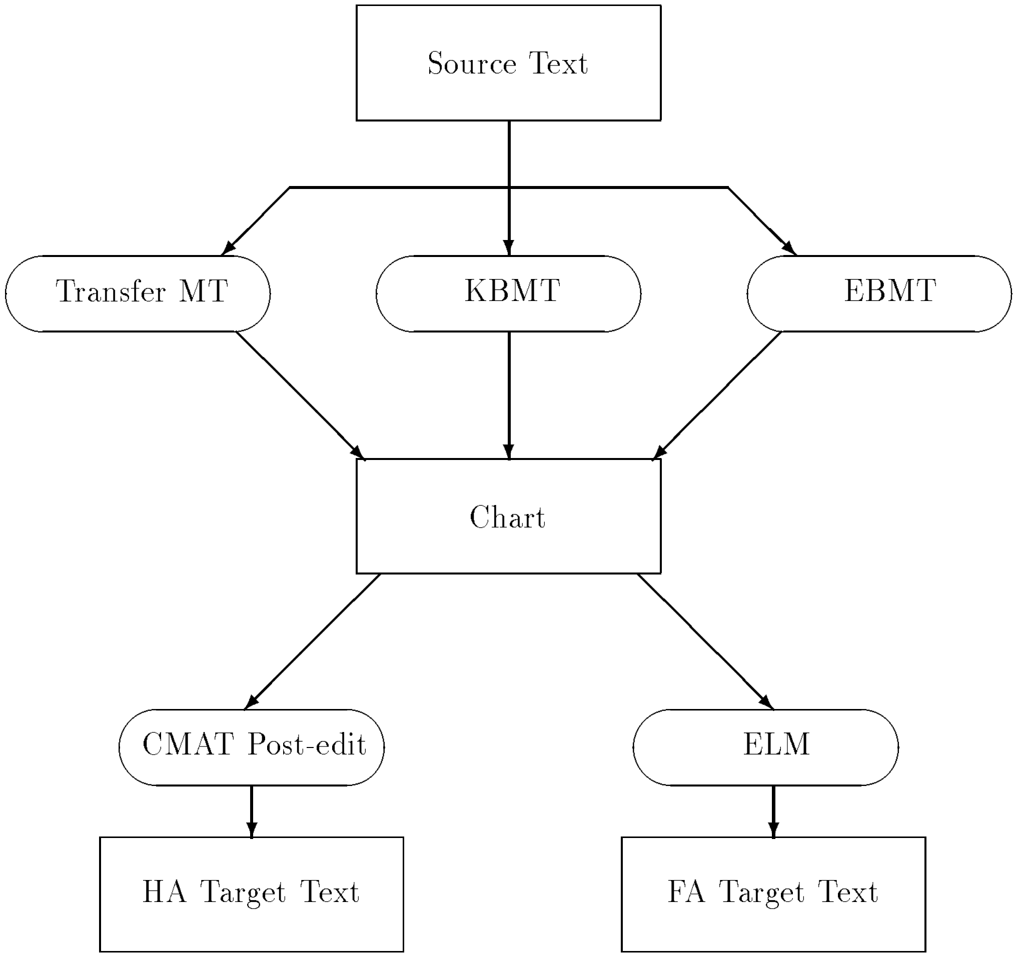

Figure 1: Pangloss System Architecture

We have transferred this idea more or less directly to MT. We use a trigram model, with backoff to bigrams and unigrams. That is to say, we use the probabilities of word triples when we have these available. However, it is often the case that a particular triple has never been seen before by the model. In this case, we use the probability of word pairs, or if that is unavailable, the simple probability of occurence of a word. Because of the extremely large number of combinations of segment hypotheses, search through the lattice of possible sentences becomes necessary, as described below[2].

3 The actual system

While the rest of the Pangloss system is written in Common Lisp, Prolog and C, the English Language Model (ELM) program is written in C^{++}. It makes use of Rosenfeld's CMU Statistical Language Modeling Toolkit (CMU-SLM-Toolkit) [Rosenfeld 94] for statistical model processing and the FramepaC C^{++} frame-representation package [Brown 95] (originally designed for the Mikrokosmos project [Nirenburg et al. 95b]) for basic Lisp-like data structures. The ELM program consists of an input stage, the search engine that attempts to determine the best path through the chart, and an output post-processor (see Figure 2). To support experimentation, nearly all values used internally may be specified on the commandline at run-time, and many options may be switched on or off.

As mentioned above, the language modeler's search engine contains at its heart a trigram probability model with backoff to bigram and unigram probabilities. This model is created with and processed by CMU-SLM-Toolkit programs and function libraries. The particular trigram model that was used was created from approximately 450 million bytes of newswire text from the Associated Press and Reuters, consisting of the entire 1990 AP text (300 million bytes) and much of the January 1994 to August 1994 Clarinet newsfeed (150 million bytes, fairly evenly split between AP and Reuters texts). A small amount of preprocessing was performed on the text to remove certain obvious common typographical errors and to tokenize elements such as numbers, ordinals, monetary amounts, times, telephone numbers, and possessives.

The trigram model was created with a 65,533-word vocabulary (the maximum supported by CMU-SLM-Toolkit)[3]; since the text was case-folded before processing, the effective vocabulary compared to a case-sensitive model is somewhat larger. For this vocabulary and our training texts, the out-of-vocabulary rate was about 1.2 percent of all words. During the final stage of model creation, the CMU-SLM-Toolkit program **vtric2bbo** was told to omit all bigrams and trigrams that only occurred once in the input texts, which reduced the size of the model by more than half and left a 57 megabyte trigram model file. This model file contains 2,307,329 bigrams (of 4,643,174 distinct bigrams in the input) and 6,276,520 trigrams (of 18,637,479 encountered).

Along with this trigram model, the ELM program takes as input a file containing the combined outputs of Pangloss' translation engines. This file contains a Lisp structure for

[2]The dynamic programming algorithm used for the HAMT system [Frederking & Nirenburg 94] is not applicable in this situation, because it assumes the choice for each segment is independent of other segments, which is clearly no longer true.

[3]The toolkit uses 16-bit integers to represent words, from which three values are used for out-of-vocabulary, begin-sentence, and end-sentence markers

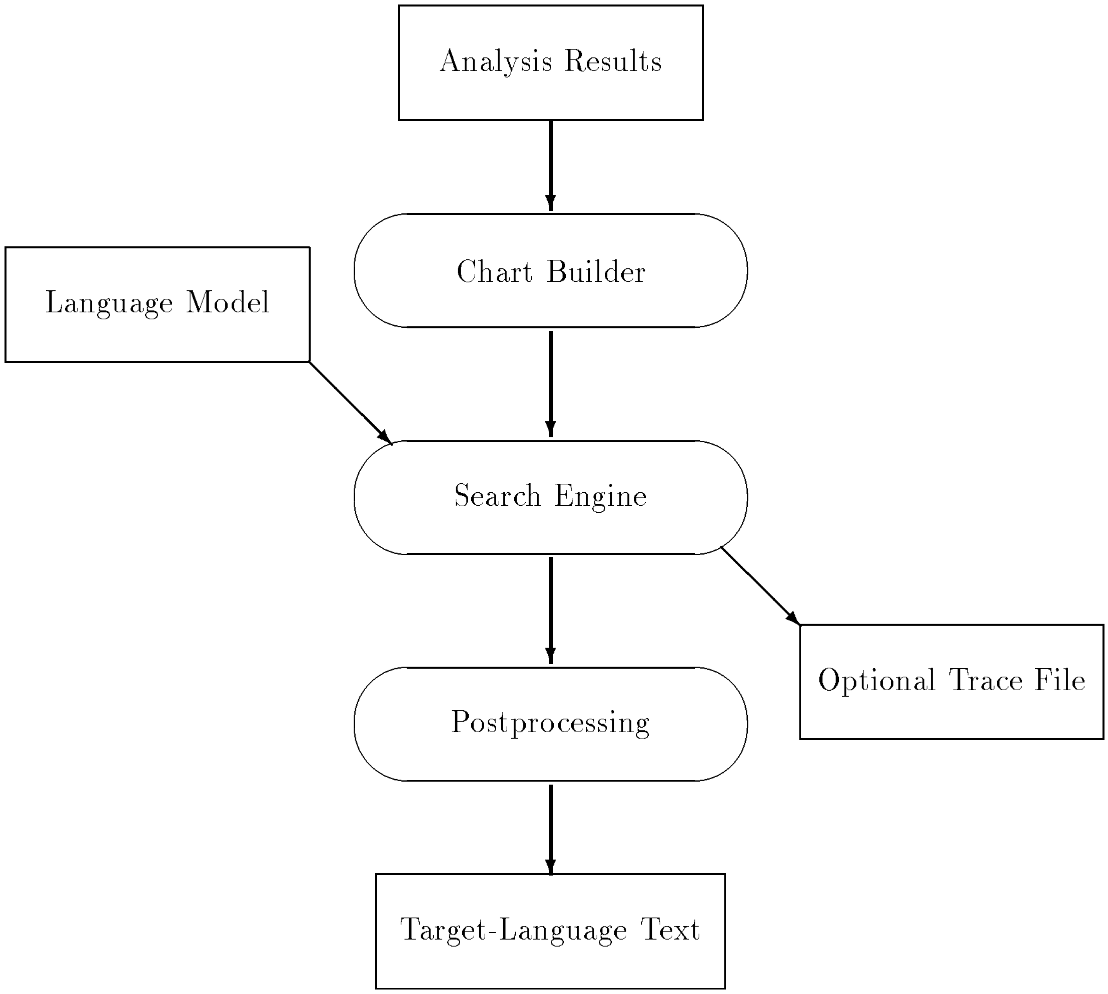

Figure 2: Language Modeler Flow Diagram

each sentence (see Figure 3), one of whose elements is the chart, represented as an array of lists of translated segments. ELM uses FramepaC to read this file, since one of FramepaC's features is the ability to read and manipulate Lisp data.

After reading the Pangloss file, the Lisp-style chart structure is converted into a C^{++} data structure representing the chart in an internal form more conducive to the operations performed on it during language modeling, which have the goal of determining the best path through the chart. As part of the conversion process, identical arcs produced by different engines are combined (with an appropriate adjustment of the resultant arc's weight) to reduce the amount of searching later. If requested by a program option, arcs from certain engines may be omitted from this internal chart entirely.

Figures 4 through 7 shows a readable dump of the internal chart generated for the Lisp structure in Figure 3. Each line represents one arc; the numbers in the left margin indicate

```
(#S(MTRANS-SENTENCE :POSITION 16 :STATUS :OUTPUT-DONE
   :INPUT-STRING "La Corte examinara ese recurso el miercoles, pero no
  fallara sobre el caso hasta agosto proximo."
   :OUTPUT-STRING
     (#S(CHART :SOURCE
                 #(#S(WORD :STRING "La" :ROOT "la"
   :MORPH (ARTICLE FEMININE DEFINITE SINGULAR))
                   #S(WORD :STRING "Corte" :ROOT "corte"
   :MORPH (PROPER_NOUN NOUN MASCULINE SINGULAR))
   ...)
           :SOURCE-LENGTH 18
           :STRUCT
             #((#S(CHART-ENTRY :TYPE :MTPHRASE :START 0 :END 1 :LENGTH 2
       :INFO ("the court" ((1 . 2)) 0.8))
                #S(CHART-ENTRY :TYPE :GLOSS :START 0 :END 0 :LENGTH 1
                               :INFO #S(GLOSS-ENTRY :KEY ("la")
                 :TRANSLATIONS ("the" "it" "her" "<ignore>")
                         :SOURCE ("dictionary-repair-3q94")))
                #S(CHART-ENTRY :TYPE :DICT :START 0 :END 0 :LENGTH 1
       :INFO ("La" ("the" "it" "her" "what" "that which"))))
               (#S(CHART-ENTRY :TYPE :GLOSS :START 1 :END 1 :LENGTH 1
                               :INFO #S(GLOSS-ENTRY :KEY ("corte")
                  :TRANSLATIONS ("court")
                :SOURCE ("procedure")))
                #S(CHART-ENTRY :TYPE :DICT :START 1 :END 1 :LENGTH 1
                               :INFO ("Corte"
                                       ("cut" "cutting" "cutting out"
"felling" "deletion" "failure"
"block" "section" "stint" "piece"))))
               (#S(CHART-ENTRY :TYPE :MTPHRASE :START 2 :END 4 :LENGTH 3
       :INFO ("will study that expedient" ((3 . 5)) 1))
                #S(CHART-ENTRY :TYPE :MTPHRASE :START 2 :END 4 :LENGTH 3
       :INFO ("will study that remedy" ((3 . 5)) 1))
                #S(CHART-ENTRY :TYPE :MTPHRASE :START 2 :END 4 :LENGTH 3
       :INFO ("will study that resort" ((3 . 5)) 1))
                #S(CHART-ENTRY :TYPE :MTPHRASE :START 2 :END 4 :LENGTH 3
       :INFO ("will study that resource" ((3 . 5)) 1))
...)))
```

Figure 3: Lisp Input File (fragment)

the starting positions of the arcs. For each arc, the dump shows

- the translation engine producing the arc (multiple sources are indicated, such as "DICT+GLOSS")[4]
- the source text covered by the arc
- the arc's length
- the proposed translation for the source text
- the adjusted score, based on the engine's score
- the arc's weight, based on the engine's weight and other factors

Once the chart has been read into memory, some further preprocessing is performed in order to precompute frequently referenced values. Each arc is linked to all possible predecessors, and trigram probability scores are computed for the individual arcs to the extent possible.

The best path through the chart, and thus the "best" translation of the original input, is determined through a prioritized search (see Figure 8). Each node in the search tree represents a partial path, and nodes are expanded by adding one additional arc beginning at the point in the chart where the partial path for the node ends. The scores used during the search are biased so that shorter partial paths will normally be processed before longer ones, in order to avoid proceeding directly to a complete path without first ensuring that there are no better alternatives beginning with some other initial partial path. The combination of prioritization and biasing makes the search method essentially equivalent to beam search. The exact method used to produce the bias may be selected by a commandline switch, and may involve the absolute amount or proportion of the input covered by the partial path and/or the number of arcs contained in the partial path.

Because the list of partial paths can become so enormous (queue lengths of more than 50,000 are not uncommon), the best-first priority queue is normally pruned. As a consequence of using a trigram model, only the final two words of the partial translation have any impact on the score after a node is expanded[5]. Thus, any node in the search queue with the same two final words as another but having a lower score may be removed (or may be skipped before being added). This pruning reduces the maximum queue length to less than 300 in the majority of cases, with the average queue lengths and total number of nodes expanded during a search correspondingly lower. It also causes 80% or more of the possible expansions to be skipped without being added to the search queue.

[4]The current engines include a machine-readable dictionary (DICT), the large set of glossaries previously mentioned (GLOSS) and a knowledge-based MT engine (MTPHRASE); the special marker MERGED is used when an arc covering multiple words of the input can be composed from a number of shorter arcs.

[5]Since new words are added to a path based on the probability of the resulting sequence in English, and the probabilities only span triples of words, words before the last two cannot affect the outcome.

```
0:  DICT: "La" (len 1) --> "what" (sc 2 wt 0.5)
    DICT: "La" (len 1) --> "that" "which" (sc 2 wt 0.5)
    DICT+GLOSS: "la" (len 1) --> "the" (sc 1.95 wt 2.828)
    DICT+GLOSS: "la" (len 1) --> "it" (sc 1.625 wt 2.828)
    DICT+GLOSS: "la" (len 1) --> "her" (sc 1.625 wt 2.828)
    GLOSS: "la" (len 1) --> (sc 1.5 wt 1.5)
    MTPHRASE+MERGED: "La ..." (len 2) --> "the" "court"
                                                  (sc 1.61446 wt 4.828)
1:  DICT: "Corte" (len 1) --> "cut" (sc 2.4 wt 0.5)
    DICT: "Corte" (len 1) --> "cutting" (sc 2 wt 0.5)
    DICT: "Corte" (len 1) --> "cutting" "out" (sc 2 wt 0.5)
    DICT: "Corte" (len 1) --> "felling" (sc 2 wt 0.5)
    DICT: "Corte" (len 1) --> "deletion" (sc 2 wt 0.5)
    DICT: "Corte" (len 1) --> "failure" (sc 2 wt 0.5)
    DICT: "Corte" (len 1) --> "block" (sc 2 wt 0.5)
    DICT: "Corte" (len 1) --> "section" (sc 2 wt 0.5)
    DICT: "Corte" (len 1) --> "stint" (sc 2 wt 0.5)
    DICT: "Corte" (len 1) --> "piece" (sc 2 wt 0.5)
    GLOSS: "corte" (len 1) --> "court" (sc 1.2 wt 1.5)
2:  DICT: "examinara" (len 1) --> "look" "through" (sc 2 wt 0.5)
    DICT: "examinara" (len 1) --> "go" "over" (sc 2 wt 0.5)
    DICT: "examinara" (len 1) --> "inquire" "into" (sc 2 wt 0.5)
    DICT: "examinara" (len 1) --> "will" "investigate" (sc 2 wt 0.5)
    DICT: "examinara" (len 1) --> "look" "into" (sc 2 wt 0.5)
    DICT: "examinara" (len 1) --> "will" "consider" (sc 2 wt 0.5)
    DICT: "examinara" (len 1) --> "take" "an" "examination" (sc 2 wt 0.5)
    DICT+GLOSS: "examinar" (len 1) --> "will" "test" (sc 1.4 wt 2.828)
    GLOSS: "examinar" (len 1) --> "will" "check" (sc 1 wt 1.5)
    GLOSS: "examinar" (len 1) --> "will" "review" (sc 1 wt 1.5)
    DICT+GLOSS: "examinar" (len 1) --> "will" "examine" (sc 1.35 wt 2.828)
    DICT+GLOSS: "examinar" (len 1) --> "will" "inspect" (sc 1.25 wt 2.828)
    GLOSS: "examinar" (len 1) --> "will" "scrutinize" (sc 1 wt 1.5)
    GLOSS: "examinar" (len 1) --> "will" "sit" "for" (sc 1 wt 1.5)
    GLOSS: "examinar" (len 1) --> "will" "sit" (sc 1 wt 1.5)
    GLOSS: "examinar" (len 1) --> "will" "enter" "for" (sc 1 wt 1.5)
```

Figure 4: Internal Chart (readable display), 1st part

```
    MTPHRASE: "examinara" (len 1) --> "study" (sc 0.6 wt 0.5)
    MTPHRASE: "examinara ..." (len 3) --> "will" "study" "that" "recourse"
                                          (sc 1.2 wt 0.5)
    MTPHRASE: "examinara ..." (len 3) --> "will" "study" "that" "resource"
                                          (sc 1.2 wt 0.5)
    MTPHRASE: "examinara ..." (len 3) --> "will" "study" "that" "resort"
                                          (sc 1.2 wt 1.414)
    MTPHRASE: "examinara ..." (len 3) --> "will" "study" "that" "remedy"
                                          (sc 1.2 wt 0.5)
    MTPHRASE: "examinara ..." (len 3) --> "will" "study" "that" "expedient"
                                          (sc 1.2 wt 0.5)
3:  DICT: "ese" (len 1) --> "that" (sc 2.4 wt 0.5)
    MTPHRASE+MERGED: "ese ..." (len 2) --> "that" "expedient"
                                                       (sc 1.74667 wt 1.5)
    MTPHRASE+MERGED: "ese ..." (len 2) --> "that" "remedy" (sc 1.248 wt 2.5)
    MTPHRASE: "ese ..." (len 2) --> "that" "resource" (sc 0.84 wt 0.5)
    MTPHRASE+MERGED: "ese ..." (len 2) --> "that" "recourse"
                                                       (sc 1.53135 wt 3.828)
    MTPHRASE+MERGED: "ese ..." (len 2) --> "that" "resort"
                                                       (sc 1.40338 wt 2.414)
4:  DICT: "recurso" (len 1) --> "resort" (sc 2 wt 0.5)
    DICT: "recurso" (len 1) --> "means" (sc 2 wt 0.5)
    DICT: "recurso" (len 1) --> "expedient" (sc 2 wt 0.5)
    DICT: "recurso" (len 1) --> "resources" (sc 2 wt 0.5)
    DICT: "recurso" (len 1) --> "borrowed" "capital" (sc 2 wt 0.5)
    DICT+GLOSS: "recurso" (len 1) --> "recourse" (sc 1.5 wt 2.828)
    GLOSS: "recurso" (len 1) --> "remedy" (sc 1 wt 1.5)
    DICT+GLOSS: "recurso" (len 1) --> "appeal" (sc 1.25 wt 2.828)
    GLOSS: "recurso" (len 1) --> "motion" (sc 1 wt 1.5)
    GLOSS: "recurso" (len 1) --> "petition" (sc 1 wt 1.5)
    GLOSS: "recurso" (len 1) --> "device" (sc 1 wt 1.5)
5:  DICT: "el" (len 1) --> "the" (sc 2.4 wt 0.5)
    GLOSS: "el" (len 1) --> (sc 1.8 wt 1.5)
    MTPHRASE: "el ..." (len 2) --> "the" "wednesday" (sc 0.672 wt 0.5)
6:  DICT+GLOSS: "miercoles" (len 1) --> "Wednesday" (sc 1.5 wt 2.828)
    GLOSS: "miercoles" (len 1) --> "Weds" (sc 1 wt 1.5)
    GLOSS: "miercoles" (len 1) --> "Wed" (sc 1 wt 1.5)
7:  DICT: "," (len 1) --> "," (sc 2.4 wt 0.5)
```

Figure 5: Internal Chart (readable display), 2nd part

```
8:  DICT: "pero" (len 1) --> "but" (sc 2.4 wt 0.5)
    DICT: "pero" (len 1) --> "yet" (sc 2 wt 0.5)
9:  DICT: "no" (len 1) --> "not" (sc 2.4 wt 0.5)
    DICT: "no" (len 1) --> "did" "not" (sc 2 wt 0.5)
    DICT: "no" (len 1) --> "do" "not" (sc 2 wt 0.5)
    DICT: "no" (len 1) --> "does" "not" (sc 2 wt 0.5)
    DICT: "no" (len 1) --> "no" (sc 2 wt 0.5)
    MTPHRASE: "no ..." (len 2) --> "will" "not" "fail" (sc 1.2 wt 0.5)
    MTPHRASE: "no ..." (len 2) --> "fail" (sc 0.6 wt 0.5)
10: DICT: "fallara" (len 1) --> "will" "ruff" (sc 2.4 wt 0.5)
    DICT: "fallara" (len 1) --> "will" "trump" (sc 2 wt 0.5)
    DICT: "fallara" (len 1) --> "pronounce" "sentence" "on" (sc 2 wt 0.5)
    DICT: "fallara" (len 1) --> "will" "award" (sc 2 wt 0.5)
    DICT: "fallara" (len 1) --> "will" "decide" (sc 2 wt 0.5)
    DICT: "fallara" (len 1) --> "will" "miss" (sc 2 wt 0.5)
    DICT: "fallara" (len 1) --> "go" "wrong" (sc 2 wt 0.5)
    DICT: "fallara" (len 1) --> "will" "miscarry" (sc 2 wt 0.5)
    DICT: "fallara" (len 1) --> "go" "astray" (sc 2 wt 0.5)
    DICT+GLOSS: "fallar" (len 1) --> "will" "fail" (sc 1.4 wt 2.828)
    GLOSS: "fallar" (len 1) --> "will" "adjudge" (sc 1 wt 1.5)
    GLOSS: "fallar" (len 1) --> "will" "find" (sc 1 wt 1.5)
    GLOSS: "fallar" (len 1) --> "will" "judge" (sc 1 wt 1.5)
    GLOSS: "fallar" (len 1) --> "will" "rule" (sc 1 wt 1.5)
11: DICT+GLOSS: "sobre" (len 1) --> "on" (sc 1.5 wt 2.828)
    DICT+GLOSS: "sobre" (len 1) --> "upon" (sc 1.25 wt 2.828)
    DICT+GLOSS: "sobre" (len 1) --> "on" "top" "of" (sc 1.25 wt 2.828)
    DICT+GLOSS: "sobre" (len 1) --> "over" (sc 1.25 wt 2.828)
    DICT+GLOSS: "sobre" (len 1) --> "above" (sc 1.25 wt 2.828)
    DICT+GLOSS: "sobre" (len 1) --> "envelope" (sc 1.25 wt 2.828)
12: DICT: "el" (len 1) --> "the" (sc 2.4 wt 0.5)
    GLOSS: "el" (len 1) --> (sc 1.8 wt 1.5)
    MTPHRASE+MERGED: "el ..." (len 2) --> "the" "case" (sc 1.5627 wt 3.828)
```

Figure 6: Internal Chart (readable display), 3rd part

```
13: DICT: "caso" (len 1) --> "subject" (sc 2 wt 0.5)
    DICT: "caso" (len 1) --> "instance" (sc 2 wt 0.5)
    DICT: "caso" (len 1) --> "event" (sc 2 wt 0.5)
    DICT: "caso" (len 1) --> "happening" (sc 2 wt 0.5)
    DICT: "caso" (len 1) --> "circumstances" (sc 2 wt 0.5)
    DICT: "caso" (len 1) --> "act" "of" "God" (sc 2 wt 0.5)
    DICT: "caso" (len 1) --> "unforeseen" "circumstance" (sc 2 wt 0.5)
    DICT: "caso" (len 1) --> "notice" (sc 2 wt 0.5)
    DICT+GLOSS: "caso" (len 1) --> "case" (sc 1.5 wt 2.828)
14: GLOSS: "hasta" (len 1) --> "to" (sc 1.2 wt 1.5)
    DICT+GLOSS: "hasta" (len 1) --> "even" (sc 1.35 wt 2.828)
    DICT+GLOSS: "hasta" (len 1) --> "as" "far" "as" (sc 1.25 wt 2.828)
    DICT+GLOSS: "hasta" (len 1) --> "till" (sc 1.25 wt 2.828)
    DICT+GLOSS: "hasta" (len 1) --> "until" (sc 1.25 wt 2.828)
    DICT+GLOSS: "hasta" (len 1) --> "up" "to" (sc 1.25 wt 2.828)
    DICT+GLOSS: "hasta" (len 1) --> "down" "to" (sc 1.25 wt 2.828)
15: DICT: "agosto" (len 1) --> "harvest" (sc 2 wt 0.5)
    DICT: "agosto" (len 1) --> "harvest-time" (sc 2 wt 0.5)
    DICT: "agosto" (len 1) --> "boom" "period" (sc 2 wt 0.5)
    DICT+GLOSS: "agosto" (len 1) --> "August" (sc 1.5 wt 2.828)
    MTPHRASE: "agosto ..." (len 2) --> "august" (sc 1.2 wt 0.5)
16: DICT: "proximo" (len 1) --> "near" (sc 2.4 wt 0.5)
    DICT: "proximo" (len 1) --> "close" (sc 2 wt 0.5)
    DICT: "proximo" (len 1) --> "neighboring" (sc 2 wt 0.5)
    DICT: "proximo" (len 1) --> "next" (sc 2 wt 0.5)
17: DICT: "." (len 1) --> "." (sc 2.4 wt 0.5)
```

Figure 7: Internal Chart (readable display), final part

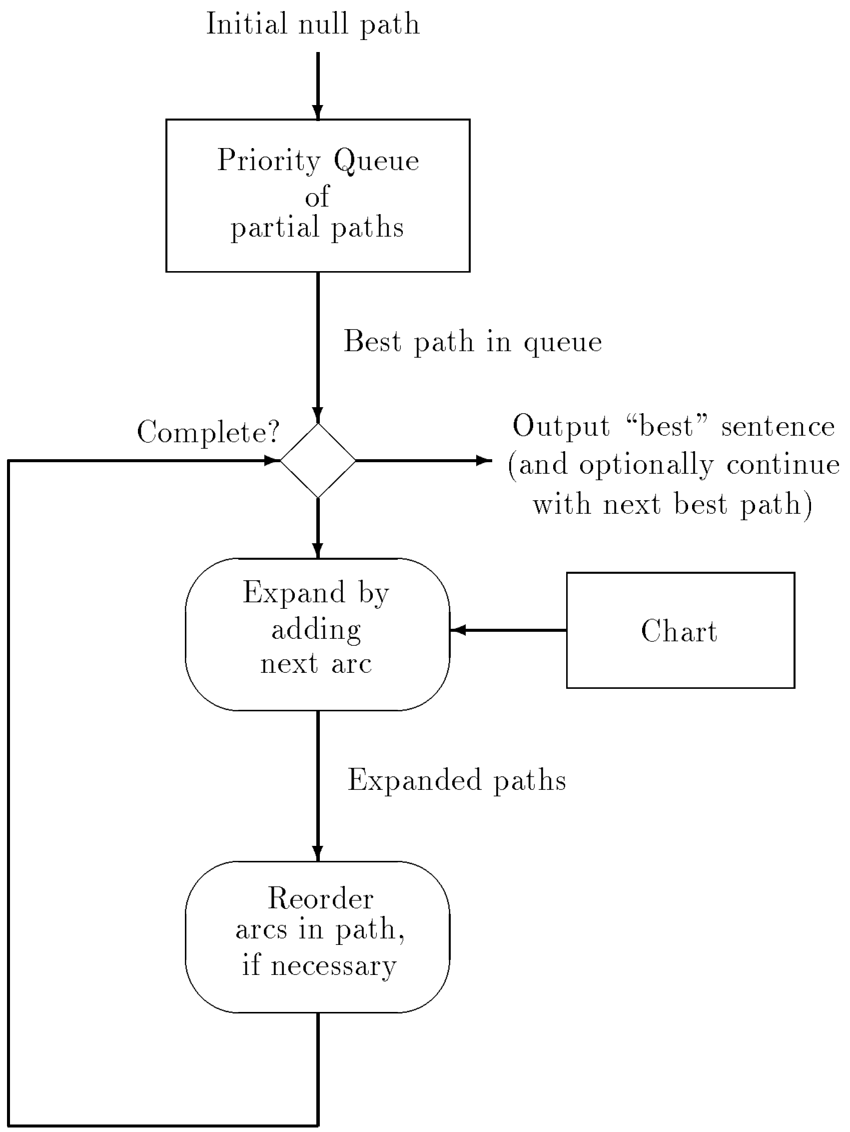

Figure 8: Search Process

As mentioned in section 2, the language modeler program does not use the raw trigram probabilities in determining the best chart walk, but rather a combination of the probabilities and several other factors. Each Pangloss translation engine is given a weight based on its perceived accuracy, which is multiplied with the score the engine itself assigns to each proposed translation. There are further bonuses for longer input coverage, agreement between engines, the first alternative proposed by an engine, and (for the glossary engine) use of certain preferred glossaries. Penalties are applied for excessively verbose translations, out-of-vocabulary words, and untranslated strings. Most of the bonuses and penalties can be adjusted and/or eliminated via commandline switches in order to experiment with their effects.

The score for a partial path is a weighted average of the scores for the individual arcs spanned by the path. Normally, the average is computed as the arithmetic mean of the

Original Ordering	Adjusted Order
Noun Adj Adj	Adj Adj Noun
Noun Adj Conj Adj	Adj Conj Adj Noun
Clitic Verb	Verb Clitic
"not" AuxVerb	AuxVerb "not"
quote period	period quote

Figure 9: Arc Reordering Rules

partial scores based on the absolute trigram probabilities, but a geometric mean may be used instead. The geometric mean is computed as the arithmetic mean of the logarithms of the trigram probabilities (which produces the same relative results as an actual geometric mean computation but required fewer changes to the program's code). As discussed later, the initial test results with the geometric mean method were very disappointing.

One trick that considerably improves the quality of output is to reorder the arcs in certain circumstances by applying some simple transfer rules (see Figure 9). The three cases in which arc reordering is performed all involve single-word arcs, and thus apply mainly to the dictionary and glossary engines, which often perform word-for-word translations. The first case is placing a noun after any immediately following adjectives, the second case is placing a verb before an immediately preceding clitic pronoun, and the third case is placing a period prior to an immediately preceding closing quotation mark. These rearrangements not only generate proper English word orderings, they also improve the performance of the trigram model; with the original Spanish word order, many would-be trigrams are split, which has a negative impact on the quality of scores for the surrounding text in addition to the words actually affected by the different word ordering. The rearrangement of quotation mark and period matches the word-for-word translations with the training texts, since the former normally place the period following a closing quotation mark rather than inside the quotation.

An additional word reordering was recently added which ignores the segmentation into arcs. For this transfer rule, the word "not" and an immediately following auxiliary verb (such as "have", "is", "would", etc.) are exchanged. Constructions such as "not would give" occur fairly frequently as the result of two separate dictionary translations producing "not" and "would give"; splitting the second arc to insert the "not" in the proper position for English improves the output and ensures the applicability of the trigram model.

In the process of determining where the ELM performed poorly in order to tune its parameters, it was discovered that the program showed a very strong tendency to select null translations (various words have the empty string as one possible dictionary translation, i.e. the Spanish "se" may be entirely omitted from the English translation). This erroneous preference for null translations caused a significant proportion of articles and pronouns to be omitted from the English output. Selecting the null translation for a particular input is now treated as though there were a fractional word in the English output which received a zero score from the language model, i.e. if the output consists of ten words and one null translation, the average per-word score (which is the metric used to determine the best sentence) is computed as though there were, for example, 10.7 words in the translation, without changing the total score for the sentence. This modification restores most of the articles which had previously been dropped, because non-empty translations now have a slight advantage; although it also adds a few extraneous pronouns to the output, the overall quality of the final translation is noticeably higher.

After the optimal path through the chart has been found, the words represented by the arcs on the path are strung together into a sentence, massaged slightly, and then output as the translation. The postprocessing consists of capitalizing the first word, removing extra whitespace (since punctuation marks are treated as separate words in analysis), and optionally converting accented characters into unaccented characters (most useful for Spanish names). If the proper options have been specified, the program also outputs the actual path through the chart (see Figure 10). The arcs in the path are listed as in Figures 4-7, with the addition of the raw per-source-word score for each arc, and the final overall score for the entire sentence.

The example used in these figures was chosen because it was one of the shorter sentences and illustrates both some features and some problems. One can clearly see the "not" reordering–the "will fail" arc is listed twice, once before the "not" arc and once after. The extra weight given to arcs in the internal chart corresponding to multiple arcs in the Pangloss output is obvious from the weights listed after "wt": only the DICT+GLOSS and MTPHRASE+MERGED arcs have weights greater than 2.0.

The problems exposed by this example are largely in the input the language model receives from Pangloss, which have been or are being addressed (the example uses Pangloss output from the September 1994 external evaluation). First, the dictionary and glossaries contain numerous errors, and often provide an excessive number of alternative translations. Thus, a bad (or even just an inappropriate) translation often makes it to the output simply because the trigram model happens to produce a higher score for the bad translation than for a good translation. A related problem, which is much harder to address, is the lack of inflections in many of the glossary entries, which affects the language model's scoring. The other main problem with the Pangloss output which the language modeler receives is the poor results produced by the September 1994 version of the KBMT engine, which frequently outputs untranslated strings or omits modifiers. In this example, KBMT produced the translation "august" instead of "next august"; since the arc covers two words of input, it received a much higher weight and was thus selected.[6]

Although it does not affect the statistical model proper, the copy of the CMU-SLM-

[6]As can be seen from the reported arc weights in Figures 4-7, we had given the KBMT engine a rather low weight compared to the other knowledge sources, due to its poor performance. The bonus for longer arcs often tips the balance in favor of KBMT anyway.

```
Translating "La Corte examinara ese recurso el miercoles, pero no fallara
             sobre el caso hasta agosto proximo."
The best translation is:
0.47683811 "The court inquire into that borrowed capital Wednesday, but will
                 not fail over the case to august."
   0.00502752   MTPHRASE+MERGED: "La ..." (len 2) --> "the" "court"
                                                     (sc 1.61446 wt 4.828)
   0.08510639 DICT: "examinara" (len 1) --> "inquire" "into" (sc 2 wt 0.5)
   0.00236343 DICT: "ese" (len 1) --> "that" (sc 2.4 wt 0.5)
   0.00003297 DICT: "recurso" (len 1) --> "borrowed" "capital" (sc 2 wt 0.5)
   0.00000000 DICT+GLOSS: "miercoles" (len 1) --> "Wednesday"
                                                     (sc 1.5 wt 2.828)
   0.38461538 DICT: "," (len 1) --> "," (sc 2.4 wt 0.5)
   0.03924622 DICT: "pero" (len 1) --> "but" (sc 2.4 wt 0.5)
   0.00000000 DICT+GLOSS: "fallar" (len 1) --> "will" "fail"
                                                     (sc 1.4 wt 2.828)
   0.01894564 DICT: "no" (len 1) --> "not" (sc 2.4 wt 0.5)
   0.00000000 DICT+GLOSS: "fallar" (len 1) --> "will" "fail"
                                                     (sc 1.4 wt 2.828)
   0.00940505 DICT+GLOSS: "sobre" (len 1) --> "over" (sc 1.25 wt 2.828)
   0.44575716 MTPHRASE+MERGED: "el ..." (len 2) --> "the" "case"
                                                     (sc 1.5627 wt 3.828)
   0.03110823 GLOSS: "hasta" (len 1) --> "to" (sc 1.2 wt 1.5)
   0.00154996 MTPHRASE: "agosto ..." (len 2) --> "august" (sc 1.2 wt 0.5)
   0.03039713 DICT: "." (len 1) --> "." (sc 2.4 wt 0.5)
```

Figure 10: Resultant Chart Walk

Toolkit used by the ELM has recently been modified to load the language model into memory incrementally on demand rather than loading the entire model at startup. This significantly speeds up small test runs–instead of loading all 57 megabytes, it is only necessary to load 6 megabytes plus data for the bigrams (and trigrams beginning with those bigrams) actually referenced, typically 6-8 megabytes for the first sentence and progressively less for additional sentences). Memory requirements are reduced correspondingly; after processing 100 sentences, the original implementation used 62 megabytes of main memory allocations, while the modified incrementally-loaded version stabilizes at about 38 megabytes of main memory. This will permit larger trigrams models to be used in the future, or running the same model on machines with less available virtual memory (62 megabytes approaches the amount of virtual memory available on the majority of the workstations used by the Pangloss project).

4 Discussion

One serious problem with a purely statistical MT approach such as that of Brown et al. [Brown et al. 90] is that it is very difficult to obtain the large amounts of alignable bilingual text necessary for producing the bilingual statistical models they require. Indeed, even the monolingual models used in speech recognition suffer from a lack of sufficient monolingual data, which is the motivation for the backoff technique. The main strength of our approach compared to such fully statistical MT is that our statistical model is only monolingual, modeling the target language. In the case of English, there are very large amounts of training data available. With some target languages, even monolingual online texts may be in short supply.

This strength is also its main weakness. Because the model is monolingual, the question it can answer is "How close is this set of selections to an English sentence?" The model contains no information about Spanish or translation from Spanish to English. Thus alternative translations of a segment that produce equally good English sentences cannot, even in principle, be distinguished by this technique.

As alluded to above, the first tests with the geometric mean averaging method produced output that was much less coherent than the corresponding output from the arithmetic mean averaging method. This was initially thought to be due to the enormous range of possible logarithmic probability values compared to the absolute probabilities used for the arithmetic mean, but later tests with an improved bias function which takes the actual values assigned to arcs into account yielded similar results. Whether this effect is the result of an implementation error or some other undetermined cause has not yet been determined.

5 Conclusion and future work

The use of an English language model produced a definite improvement in the quality of our FAMT output, although it is clearly not at the same quality level as HAMT output (which generally is slightly better than that produced by the same translator without assistance). As in the example in Figure 11, the output generally reads like English, although with awkward and occasionally bizarre word choices.

It is difficult and expensive to rigorously evaluate FAMT output quality, so we do not have quantitative comparisons between our current FAMT system with and without the ELM. However, our overall FAMT system showed distinct improvement between its last two external evaluations [White & O'Connell 94], most of which we believe is due to the use of the ELM–even in its at the time relatively untuned state[7].

As for future improvements, we intend to continue improving the performance of the individual translation engines in our system, since the quality of the proposed translations presented to the ELM places a limit on the quality of the final output; the dictionaries and glossaries in particular require cleaning to remove the bizarre or inappropriate translations they often produce. We also envision a number of improvements to the ELM system. In addition to further fine-tuning of the language model's parameters, we are seeking ways of training a model based on correct system output. That is, statistical language models are extremely sensitive to the closeness of the match between training texts and test texts; even if the model is trained on one newswire and tested on another, the style differences between different news organizations produce a distinct degradation in the model's effectiveness. If we can find a way of generating significant amounts of text consisting of correctly-selected system output segments, a model trained on such a well-matched corpus should perform significantly better.

We would like to thank Sergei Nirenburg for major contributions to the early design of this system, and Roni Rosenfeld for significant help in our statistical modeling efforts.

[7]At the time of the September 1994 external evaluation, the initial version of the ELM had just been completed, and did not yet have a number of the features described in this paper, nor had there been enough time to carefully optimize the parameters' settings. In particular, since the evaluation, a number of bugs affecting the quality of the output have been corrected, and the null-translation penalty and additional reordering rules have been added; the subjective impression of the ELM's output quality is now much higher than it was in September, given identical input.

```
La Comision Europea acepto examinar los problemas planteados por la
nueva reglamentacion sobre las importaciones de bananas latinoamericanas,
anuncio el jueves uno de sus voceros.

La nueva Organizacion Comun del Mercado de la Banana (OCMB), que entro
en vigor en julio de 1993, despues de ser adoptada por los Doce en
febrero, contra la opinion de Alemania, Belgica y Holanda, limita el
volumen e impone un arancel a las cuotas de bananas latinoamericanas
que se pueden importar todos los anos en la Union Europea.

Un acuerdo con cuatro paises latinoamericanos --Colombia, Costa Rica,
Nicaragua y Venezuela--aumenta gradualmente esa cuota de importacion
de 2 millones de toneladas a 2,2 millones de toneladas en 1995,
disminuye de 100 a 75 ecus (1 ecu = 1,3 dolar) por tonelada el arancel
y permite que esos paises otorguen licencias de exportacion.

Estas disposiciones hicieron que esos paises abandonasen su querella
contra la OCMB en el GATT (Acuerdo General sobre Aranceles y Comercio).


The European Commission approved look into the problems created across
the new rules over the importations of Latin-American bananas, a had
announced the thursday one of his spokesman.

The new organization general of the staple of the banana (ocmb), which
took effect in July from 1993, after borrowing across the twelve in
February, opposite the feelings of germany, belgium and holland,
limits the volume and imposes a duty at the quotas for Latin-American
bananas that authority themselves involve every year in the union
europea.

An agreement with four Latin American countries -- colombia, costa
rica, nicaragua and venezuela -- extend that import quota of two
million tons two million two hundred thousand tons in 1995, decrease
by 100 to 75 ecu (1 ecu = nine dollars) for ton the tariff and
provides that these countries make export license.

These steps did that these regions forewent their lawsuit against the
OCMB onto the gatt (General Agreement on Tariffs and Trade).
```

Figure 11: Example ELM-enhanced Translation

References

[Brown et al. 90] Brown, P., Cocke, J., DellaPietra, S., DellaPietra, V., Jelinek, F., Lafferty, J., Mercer, R., and Roossin, P. "A Statistical Approach to Machine Translation." *Computational Linguistics* 16(2), 1990.

[Brown 95] Brown, R., "FramepaC User's Reference," CMU CMT Tech-Memo, in preparation. Current draft available on WWW at http://www.cs.cmu.edu/afs/cs.cmu.edu/user/ralf/pub/WWW/papers.html

[Frederking et al. 94] Frederking, R., Nirenburg, S., Farwell, D., Helmreich, S., Hovy, E., Knight, K., Beale, S., Domashnev, C., Attardo, D., Grannes, D., Brown, R. "Integrating Translations from Multiple Sources within the Pangloss Mark III Machine Translation." Proceedings of the first conference of the Association for Machine Translation in the Americas, AMTA-94, Columbia, MD, 1994.

[Frederking & Nirenburg 94] Frederking, R. and Nirenburg, S. "Three Heads are Better than One." Proceedings of the fourth Conference on Applied Natural Language Processing, ANLP-94, Stuttgart, Germany, 1994.

[Katz 87] Katz, S., "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", in "IEEE Transactions on Acoustics, Speech and Signal Processing", volume ASSP-35, pages 400-401, March 1987.

[Nirenburg et al. 95a] Nirenburg, S., (ed.). "The Pangloss Mark III Machine Translation System." Joint Technical Report, Computing Research Laboratory (New Mexico State University), Center for Machine Translation (Carnegie Mellon University), Information Sciences Institute (University of Southern California). Issued as CMU technical report CMU-CMT-95-145. 1995.

[Nirenburg et al. 95b] Nirenburg, S., et al. Unpublished project overview available via Mikrokosmos home page on WWW at http://crl.nmsu.edu/users/mikro/Home.html

[Rosenfeld 94] Rosenfeld, R., "The CMU Statistical Language Modeling Toolkit and its use in the 1994 ARPA CSR Evaluation", in Proceedings of the ARPA Spoken Language Technology Workshop, Austin, TX, January 1995. Official CMU-SLM Toolkit code distribution available at ftp://ftp.cs.cmu.edu/project/fgdata/CMU_SLM_Toolkit_V1.0_release.tar.Z

[White & O'Connell 94] White, J.S. and T. O'Connell. "Evaluation in the ARPA Machine Translation Program: 1993 Methodology." Proceedings of the ARPA HLT Workshop. Plainsboro, NJ. 1994.