Automating Knowledge Acquisition for Machine Translation

Kevin Knight

1 Introduction

How can we write a computer program to translate an English sentence into Japanese? Anyone who has taken a graduate-level course in Artificial Intelligence knows the answer. First, compute the meaning of the English sentence. That is, convert it into logic or your favorite knowledge representation language. This conversion process will appeal to a dictionary, which maps words (like "canyon") onto concepts (like CANYON), and to a world model that contains facts about reality (like "canyons don't fly"). In this way, an ambiguous sentence like "John saw the Grand Canyon flying to New York" gets the right interpretation. Finally, turn the conceptual structure into Japanese (or whatever), using further grammatical and lexical knowledge bases.

Along the way, there will be many fascinating problems to solve. Like: canyons don't "fly", but do people "fly"? Only in the sense of RIDE-IN-AIRPLANE, with the caveat that the WHEELS of the AIRPLANE must at some point leave the GROUND—otherwise, we're just taxiing. How about "John flew me to New York"? That's another meaning of "fly," involving DRIVE-AIRPLANE as well as RIDE-IN-AIRPLANE. And if "United flew me to New York," I may say that the AIRPLANE that I rode in was driven by an EMPLOYEE of the AIRLINE that OWNS the AIRPLANE. And while we're at it, why *don't* canyons fly? AIRPLANEs and CANYONs are both inanimate, but a CANYON seems too big to fly, or anyway not aerodynamic enough ... We seem to be on the right track, but considering the vastness of human language and the intricacies of meaning, we're in for a very long journey.

Meanwhile, in the real world (not the formal model), people are buying shrink-wrapped machine translation (MT) software for fifty dollars. Email programs ship with language translation capacity (optional). Companies use MT to translate manuals and track revisions. MT products help governments to translate web pages and other net-traffic.

What's happening here? Is AI irrelevant? No, but there are many approaches to MT, and not all of them use formal semantic representations. (I'll describe some in this article.) This should come as no surprise, because MT pre-dates AI, as a field. An AI scientist could easily spend two months representing "John saw the Grand Canyon flying to New York," while anybody with a bilingual dictionary can build a general-purpose word-for-word translator in a day. With the right language pair, and no small amount of luck, word-for-word results may be intelligible—"John vi el Grand Canyon volando a New York." That's okay Spanish. But most of the time the translations will be terrible, which is why MT researchers are very busy:

- Building high-quality semantics-based MT systems in circumscribed domains, like weather reports [Chandioux and Grimaila, 1996] and heavy equipment manuals [Nyberg and Mitamura, 1992].

- Abandoning automatic MT, and building software to assist human translators instead [Isabelle *et al.*, 1993; Dagan and Church, 1994; Macklovitch, 1994].

- Developing automatic knowledge acquisition techniques for improving general-purpose MT [Brown *et al.*, 1993b; Yamron *et al.*, 1994; Knight *et al.*, 1995].

There have been exciting recent developments along all these lines. I will concentrate on the third thrust—improving MT quality through automatic knowledge acquisition.

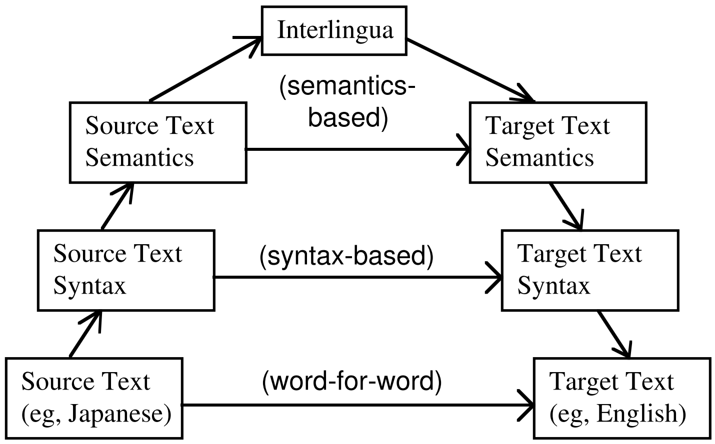

Figure 1: Different strategies for Machine Translation (MT).

If you take a poll of general-purpose MT users, you will find that they want many improvements: speed, compatibility with their word processor, customizable dictionaries, translation memory, revision tracking, etc. At the top of everyone's list, however: better output quality. Unfortunately, the MT companies are busy supplying all those other things, because they know how. Commercial translation quality has reached something of a plateau, as it is difficult to enter so much linguistic knowledge by hand. So there's a great payoff for successful research in automatic, corpus-based knowledge acquisition. Recent corpus-based techniques (parsing, word sense disambiguation, bilingual text analysis, etc.) have yet to show up in commercial MT, and it looks like there are plenty more results to come.

From a scientific point of view, MT remains the classic acid test of how much we understand about human language. If we pour in lots of theories from computer science, linguistics, statistics and AI—and still get wrong translations—then we know we need better theories. Broadly speaking, theories of MT fall into the categories shown in Figure 1. The simplest method, at the bottom of the triangle, is word-for-word substitution. Words are ambiguous, so selecting which substitution to make is not easy. Word-substitution programs often wind up doing a limited amount of re-ordering also, e.g., flipping adjectives and nouns. Word order differences can be handled more elegantly if we do a syntactic analysis of the source text, then transfer this analysis into a corresponding target language structure. In that case, word translations can be sensitive to syntactic relations—e.g., we can translate a verb differently depending on its direct object. Still, the target text syntax will likely mirror that of the source text. We can therefore do a semantic analysis that abstracts away syntactic details (moving on up the triangle in Figure 1).

Ultimately, we arrive at an all-encompassing meaning representation, called Interlingua. You may wonder why semantics and Interlingua are not the same thing—here is an illustration from a Japanese/English MT system I have worked on. It once translated a Japanese sentence as "There is a plan that a baby is happening in her," a pretty reasonable translation, but with a definite Japanese-semantics feel to it. Semantics is not an all-or-nothing proposition in MT any more than in, say, expert systems.

As you go up the triangle, you encounter more good ideas, linguistic generalizations, and explanatory power. It also becomes more difficult to build large-scale systems, because the knowledge requirements become severe. At the bottom, you need to know things like how to say "real estate" in French. To parse, you need to know parts of speech and grammar. To get meaning, you need to know all the meanings of all the words, including the slippery little ones, and have knowledge for combining word meanings into sentence meanings. It's progressively harder to get the knowledge. Fortunately for MT, recent work in corpus-based learning offers the possibility of reducing the knowledge bottleneck.

Note to reader: You will see that I devote more pages to word-for-word MT than to semantic MT. In part, this is to present the statistical word-for-word work a bit more simply and accessibly; the standard literature hits you with terms like "Kronecker delta function" and "Lagrange multiplier" on page two. Furthermore, word-for-word MT comprises a fairly self-contained set of techniques, while semantic MT benefits from the full range of corpus-based language research, most of which I will not review.

2 Word-for-Word Translation

Word-for-word translation was first proposed in the 1950s. Protocomputers had just broken German military codes, successfully transforming encrypted-German into real German by identifying letter shifts and substitutions. Cryptographers and information-theory scientists wondered if Russian couldn't be usefully viewed as encrypted-English—and MT as a kind of decipherment.

As a cipher, Russian looked to be quite complex. Sometimes a word would be encrypted one way, and sometimes another (what we now call lexical ambiguity). Words also changed their order—*transposition* in the cryptographic jargon. Now, to crack complex ciphers, it was always very useful to intercept messages in both their normal and encrypted forms (also known as "plaintext" and "ciphertext"). Fortunately, there were many such messages in both Russian and English available: translations of Tolstoy, for instance. But the cryptographers soon gave up this whole approach, because of the sheer size of the problem. German encryption had been performed on rotor machines in the field, while MT was something else, with complex grammar and hundred-thousand-word substitution "alphabets."

This line of attack was resumed in the 1990s, however, when computers grew more powerful. I will reconstruct the basic approach with an example.

Suppose I give you the translated document shown in Figure 2. Sentences appear in both "Centauri" and "Arcturan" translations. If you aren't fluent in extraterrestrial languages, don't despair—the nonsense words will actually help you to see the text from a computer's point of view. Aware that you may soon be abducted by aliens and put to work in an Interstellar Translation Bureau, you are eager to analyze the data.

You first notice that corresponding sentences have the same number of words, except for (11). You conjecture that the two languages are very close to one another, and perhaps simple word-for-word substitution will suffice for translation. To test this hypothesis, you look at the Centauri word "ghirok," which appears in sentence pairs 3 and 10. It sits directly above "hilat" and "bat" in the two respective Arcturan translations. So perhaps the word "ghirok" is ambiguous, like the English word "bank". On the other hand, the Arcturan word "hilat" appears in both sentence pairs—in fact, "hilat" appears in Arcturan if and only if "ghirok" appears in Centauri. So you might instead assume that while "ghirok" always means "hilat", Centauri and Arcturan employ different word order schemes.

Next, you decide to fry some easy fish. The words "ok-voon" and "at-voon" (1) look suspiciously familiar, so you link them. You do the same for "at-drubel/ok-drubel" (2), "ok-yurp/at-yurp" (9), and "zanzanok/zanzanat" (11). The pair "enemok/eneat" (7) also looks promising, but you decide to wait for more evidence.

Sentence pair (1) is now partially explained, leaving two obvious alternatives; either

(A) "ororok" means "bichat" (and "sprok" means "dat"), or

(B) "ororok" means "dat" (and "sprok" means "bichat").

```
--------------------------------------------------
1a.  ok-voon ororok sprok .

1b.  at-voon bichat dat .
--------------------------------------------------
2a.  ok-drubel ok-voon anok plok sprok .

2b.  at-drubel at-voon pippat rrat dat .
--------------------------------------------------
3a.  erok sprok izok hihok ghirok .

3b.  totat dat arrat vat hilat .
--------------------------------------------------
4a.  ok-voon anok drok brok jok .

4b.  at-voon krat pippat sat lat .
--------------------------------------------------
5a.  wiwok farok izok stok .

5b.  totat jjat quat cat .
--------------------------------------------------
6a.  lalok sprok izok jok stok .

6b.  wat dat krat quat cat .
--------------------------------------------------
7a.  lalok farok ororok lalok sprok izok enemok .

7b.  wat jjat bichat wat dat vat eneat .
--------------------------------------------------
8a.  lalok brok anok plok nok .

8b.  iat lat pippat rrat nnat .
--------------------------------------------------
9a.  wiwok nok izok kantok ok-yurp .

9b.  totat nnat quat oloat at-yurp .
--------------------------------------------------
10a. lalok mok nok yorok ghirok clok .

10b. wat nnat gat mat bat hilat .
--------------------------------------------------
11a. lalok nok crrrok hihok yorok zanzanok .

11b. wat nnat arrat mat zanzanat .
--------------------------------------------------
12a. lalok rarok nok izok hihok mok .

12b. wat nnat forat arrat vat gat .
--------------------------------------------------

Translation dictionary:

ghirok - hilat              ok-yurp - at-yurp
ok-drubel - at-drubel       zanzanok - zanzanat
ok-voon - at-voon
```

Figure 2: Twelve pairs of sentences written in imaginary Centauri and Arcturan languages.

Of course, it could be the case that "ororok" is an (untranslated) auxiliary verb, and that "sprok" has a phrasal translation "bichat dat." But you ignore that possibility for now. So, which of the two alternatives is more likely? To find out, you look for a sentence that contains "sprok" but not "ororok," such as (2a). Its translation (2b) has "dat," lending support to hypothesis (A) above. You can now add two more entries to your translation dictionary and link their occurrences throughout the corpus (1,2,3,6,7).

Sentence pair (2) is a logical place to continue, because you only need to consider how to map "anok plok" onto "pippat rrat." Again, two possibilities suggest themselves, but sentence pair (4) pushes you toward "anok/pippat" and therefore "plok/rrat."

Sentence pair (3) is much more challenging. So far, we have:

```
erok sprok izok hihok ghirok
       |              /
totat dat arrat vat hilat
```

The Centauri word "izok" would seen to translate as either "totat," "asrat," or "vat," and yet when you look at "izok" in (6), none of those three words appear in the Arcturan. Therefore, "izok" looks to be ambiguous. The word "hihok," on the other hand, is fixed by (11) as "arrat." Both (3) and (12) have "izok hihok" sitting directly on top of "arrat vat," so in all possibility, "vat" seems a reasonable translation for (ambiguous) "izok." Sentences (5,6,9) suggest that "quat" is its other translation. By process of elimination, you connect the words "erok" and "totat," finishing off the analysis:

```
erok sprok izok hihok ghirok
  |    |      \ /       /
  |    |       X       /
  |    |      / \     /
totat dat arrat vat hilat
```

Notice that aligning the sentence pairs helps you to build the translation dictionary, and that building the translation dictionary also helps you decide on correct alignments. You might call this the "decipherment method."

Figure 3 shows the progress so far. With a ball-point pen and some patience, you can carry this reasoning to its logical end, leading to the following translation dictionary:

```
anok - pippat            mok - gat
brok - lat               nok - nnat
clok - bat               ok-drubel - at-drubel
crrrok - (none?)         ok-voon - at-voon
drok - sat               ok-yurp - at-yurp
enemok - eneat           ororok - bichat
erok - totat             plok - rrat
farok - jjat             rarok - forat
ghirok - hilat           sprok - dat
hihok - arrat            stok - cat
izok - vat/quat          wiwok - totat
jok - krat               yorok - mat
kantok - oloat           zanzanok - zanzanat
lalok - wat/iat
```

```
-------------------------------------------------
1a.  ok-voon ororok sprok .

1b.  at-voon bichat dat .
-------------------------------------------------
2a.  ok-drubel ok-voon anok plok sprok .

2b.  at-drubel at-voon pippat rrat dat .
-------------------------------------------------
3a.  erok sprok izok hihok ghirok .

3b.  totat dat arrat vat hilat .
-------------------------------------------------
4a.  ok-voon anok drok brok jok .

4b.  at-voon krat pippat sat lat .
-------------------------------------------------
5a.  wiwok farok izok stok .

5b.  totat jjat quat cat .
-------------------------------------------------
6a.  lalok sprok izok jok stok .

6b.  wat dat krat quat cat .
-------------------------------------------------
7a.  lalok farok ororok lalok sprok izok enemok .

7b.  wat jjat bichat wat dat vat eneat .
-------------------------------------------------
8a.  lalok brok anok plok nok .

8b.  iat lat pippat rrat nnat .
-------------------------------------------------
9a.  wiwok nok izok kantok ok-yurp .

9b.  totat nnat quat oloat at-yurp .
-------------------------------------------------
10a. lalok mok nok yorok ghirok clok .

10b. wat nnat gat mat bat hilat .
-------------------------------------------------
11a. lalok nok crrrok hihok yorok zanzanok .

11b. wat nnat arrat mat zanzanat .
-------------------------------------------------
12a. lalok rarok nok izok hihok mok .

12b. wat nnat forat arrat vat gat .
-------------------------------------------------

Translation dictionary:

anok - pippat                ok-yurp - at-yurp
erok - total                 ok-voon - at-voon
ghirok - hilat               ororok - bichat
hihok - arrat                plok - rrat
izok - vat                   sprok - dat
ok-drubel - at-drubel        zanzanok - zanzanat
```

Figure 3: The progress of building a translation dictionary from pairs of sentences, using a decipherment method.

```
ok-drubel anok ghirok farok .  wiwok rarok nok zerok ghirok enemok .
ok-drubel ziplok stok vok erok enemok kantok ok-yurp zinok jok yorok clok .
lalok clok izok vok ok-drubel .  ok-voon ororok sprok .  ok-drubel ok-voon
anok plok sprok .  erok sprok izok hihok ghirok .  ok-voon anok drok brok
jok .  wiwok farok izok stok .  lalok sprok izok jok stok .  lalok brok
anok plok nok .  lalok farok ororok lalok sprok izok enemok .  wiwok nok
izok kantok ok-yurp .  lalok mok nok yorok ghirok clok .  lalok nok crrrok
hihok yorok zanzanok .  lalok rarok nok izok hihok mok .
```

Word pair counts:

1 . erok	1 hihok yorok	1 ok-drubel ok-voon
7 . lalok	1 izok enemok	1 ok-drubel ziplok
2 . ok-drubel	2 izok hihok	2 ok-voon anok
2 . ok-voon	1 izok jok	1 ok-voon ororok
3 . wiwok	1 izok kantok	1 ok-yurp .
1 anok drok	1 izok stok	1 ok-yurp zinok
1 anok ghirok	1 izok vok	1 ororok lalok
2 anok plok	1 jok .	1 ororok sprok
1 brok anok	1 jok stok	1 plok nok
1 brok jok	1 jok yorok	1 plok sprok
2 clok .	2 kantok ok-yurp	2 rarok nok
1 clok izok	1 lalok brok	2 sprok .
1 crrrok hihok	1 lalok clok	3 sprok izok
1 drok brok	1 lalok farok	2 stok .
2 enemok .	1 lalok mok	1 stok vok
1 enemok kantok	1 lalok nok	1 vok erok
1 erok enemok	1 lalok rarok	1 vok ok-drubel
1 erok sprok	2 lalok sprok	1 wiwok farok
1 farok .	1 mok .	1 wiwok nok
1 farok izok	1 mok nok	1 wiwok rarok
1 farok ororok	1 nok .	1 yorok clok
1 ghirok .	1 nok crrrok	1 yorok ghirok
1 ghirok clok	2 nok izok	1 yorok zanzanok
1 ghirok enemok	1 nok yorok	1 zanzanok .
1 ghirok farok	1 nok zerok	1 zerok ghirok
1 hihok ghirok	1 ok-drubel .	1 zinok jok
1 hihok mok	1 ok-drubel anok	1 ziplok stok

Figure 4: Monolingual Centauri text with associated word-pair (bigram) counts.

The dictionary shows ambiguous Centauri words (like "izok") and ambiguous Arcturan words (like "totat"). It also contains a curious Centauri word ("crrrok") that has no translation—after the alignment of (11), this word was somehow left over:

```
lalok nok crrrok hihok yorok zanzanok
 |     |           /     /      /
wat   nnat      arrat  mat zanzanat
```

You begin to speculate whether "crrrok" is some kind of affix, or "crrrok hihok" is a polite form of "hihok"—but you are suddenly whisked away by an alien spacecraft and put to work in the Interstellar Translation Bureau, where you are immediately tasked with translating the following Arcturan dispatch into Centauri:

```
13b.  iat lat pippat eneat hilat oloat at-yurp .
14b.  totat nnat forat arrat mat bat .
15b.  wat dat quat cat uskrat at-drubel .
```

You have never seen these sentences before, so you cannot look up the answers. More reasoning is called for.

The first sentence contains seven Arcturan words. You consult your dictionary to construct a list of seven corresponding Centauri words: "lalok," "brok," "anok," "enemok," "ghirok," "kantok,"

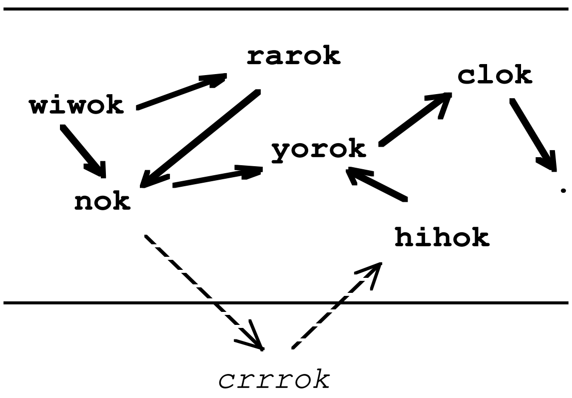

Figure 5: An attempt to put a group of Centauri words in the right order. Arrows represent previously observed word pairs from Figure 4.

and "ok-yurp." You consider writing them down in that order (a simple word-for-word translation), but as you want to make a good first impression in the Bureau, you also consider shifting the words around. There are 5040 (7!) possible word orders to choose from. Centauri text may provide useful data—there you can see that word A follows word B more or less frequently. Your request for more Centauri text is granted (Figure 4). With relish, you set about tabulating word pair frequencies, noting in passing new words like "vok," "zerok," "zinok," and "ziplok."

You are now in a position to evaluate your 5040 alternative word orders. As a shortcut, you may ask: which word is most likely to start a sentence? (Or: which word usually follows a period?). Surely, it is "lalok." Of the remaining six words, which best follows "lalok"? It is "brok." Then "anok". But after "anok," "ghirok" is more suitable than "enemok." Fortunately, "enemok" itself is a good follow-on to "ghirok." So you decide to flip the words "enemok" and "ghirok." Your final translation is

`13a. lalok brok anok ghirok enemok kantok ok-yurp .`

You move to the next sentence (14b). Immediately, you are faced with a lexical ambiguity. Should you translate "totat" as "erok" or "wiwok"? Because "wiwok" occurs more frequently, and because you've never seen "erok" followed by any of the other words you're considering, you decide on "wiwok." But admittedly, this is only a best guess. Next, you consider various word orders. The arrows in Figure 5 represent word pairs you have seen in Centauri text. There appears to be no fluent (grammatical?) path through these words. Suddenly you remember that curious Centauri word "crrrok," which had no translation—which turns out to be a natural bridge between "nok" and "hihok," giving you the seemingly fluent, possibly correct translation:

`14a. wiwok rarok nok crrrok hihok yorok clok .`

The last sentence (15b) is straightforward except that one of the Arcturan words ("uskrat") is new; it does not appear in the bilingual dictionary you built. (You imagine "uskrat" to be some type of animal). You translate the third sentence as

`15a. lalok sprok izok stok ? ok-drubel .`

where the question mark stands for the Centauri equivalent of "uskrat." You decide to consult your Centauri text to find a word that is likely to appear between "stok" and "ok-drubel." Before you can finish, however, you and your translations are rushed before the Arcturan Rewrite Perspicuity Authority.

Although you cannot understand Arcturan, you get the feeling that the Authority is pleased with your work. You are hired and tasked with translating new sentences like "brizat minat stat vat borat" that are full of words you've never seen before. To improve your correspondence tables, you seek out more documents, both bilingual (Arcturan/Centauri) and monolingual (Centauri). You are soon overwhelmed with documents. Perhaps a computer would help ...

* * *

Was this a realistic foray into language translation, or just inspired nonsense? Actual translation is, of course, more complicated:

- Only two of the 27 Centauri words were ambiguous, whereas in natural languages like English, almost all words are ambiguous.
- Sentence length was unchanged in all but one of the translations; in real translation, this is rare.
- The extraterrestrial sentences were much shorter than typical natural language sentences.
- Words are translated differently depending on context. Our translation method only used Centauri word-pair counts for context, preferring "wiwok rarok ..." over "erok rarok ..." But resolving lexical ambiguity in general requires a much wider context, and often sophisticated reasoning as well.
- Output word order should be sensitive to input word order. Our method could not decide between outputs "John loves Mary" and "Mary loves John," even though one of the two is likely to be a terrible translation.
- The data seemed to be cooked—drop out sentence pairs (8) and (9), for example, and we would not be able to settle on alignments for the remaining sentences. Many such alignments would be possible, complicating our translation dictionary.
- Our method does not allow for any phrasal dictionary entries (e.g., "anok plok" = "pippat rrat"), although human translators make extensive use of such dictionaries.

And the list goes on: What about pronouns? What about inflectional morphology? What about structural ambiguity? What about domain knowledge? What about the scope of negation?

On the other hand, our extraterrestrial example was realistic in one respect: it was actually an exercise in Spanish/English translation! Centauri is merely English in light disguise—for "erok," read "his"; for "sprok," read "associates"; et cetera. Spanish and Arcturan are also the same. Here is the real bilingual training corpus:

1a. Garcia and associates.
1b. Garcia y asociados.

2a. Carlos Garcia has three associates.
2b. Carlos Garcia tiene tres asociados.

3a. his associates are not strong.
3b. sus asociados no son fuertes.

4a. Garcia has a company also.
4b. Garcia tambien tiene una empresa.

5a. its clients are angry.
5b. sus clientes están enfadados.

6a. the associates are also angry.
6b. los asociados tambien están enfadados.

7a. the clients and the associates are enemies.
7b. los clientes y los asociados son enemigos.

8a. the company has three groups.
8b. la empresa tiene tres grupos.

9a. its groups are in Europe.
9b. sus grupos están en Europa.

10a. the modern groups sell strong pharmaceuticals.
10b. los grupos modernos venden medicinas fuertes.

11a. the groups do not sell zanzanine.
11b. los grupos no venden zanzanina.

12a. the small groups are not modern.
12b. los grupos pequeños no son modernos.

If you don't know Spanish (even if you do), you can congratulate yourself on having translated the novel sentence "la empresa tiene enemigos fuertes en Europa" (13b) as "the company has strong enemies in Europe" (13a). Had you not flipped the order of "ghirok" and "enemok," your translation would have been worse: "the company has enemies strong in Europe." Likewise, you translated "sus grupos pequeños no venden medicinas" (14b) as "its small groups do not sell pharmaceuticals" (14a). The curiously untranslatable Centauri word "crrrok" was actually the English word "do," as in "do not sell" = "no venden."

Without relying on linguistic phrase structure and real-world knowledge, you were able to learn enough about English and Spanish to translate a few sentences correctly. If you had more training text, you might have learned more. Could such a method be scaled to general-purpose MT? Several questions arise:

- Is there a large bilingual corpus for some pair of natural languages?
- Can the corpus be easily converted to sentence-pair format?
- Can the decipherment method be automated? What does the algorithm look like?
- Can the translation method be automated?

and perhaps most importantly:

- Are the translations good?

2.1 Bilingual Text Alignment

These questions were first posed and studied by a research team at IBM [Brown *et al.*, 1990]. This group pioneered the use of text corpora in MT. IBM used the Hansard corpus, a proceedings of the Canadian Parliament written in French and English (each language on a separate tape). This corpus contains millions of sentences. Of course, corresponding sentence pairs are not marked in the text, and worse, whole paragraphs on one tape are sometimes missing from the other. (A severe case of information getting lost in translation!). Also, one French sentence may get translated as two English ones, or vice versa. So it takes work to produce a database like Figure 1. Here is a small version of the problem [Church, 1993]:

English:

...

The higher turnover was largely due to an increase in the sales volume.

Employment and investment levels also climbed.

Following a two-year transitional period, the new Foodstuffs Ordinance for Mineral Water came into effect on April 1, 1988.

Specifically, it contains more stringent requirements regarding quality consistency and purity guarantees.

...

French:

...

La progression des chiffres d'affaires résulte en grande partie de l'accroissement du volume des ventes.

L'emploi et les investissements ont également augmenté.

La nouvelle ordonnance fédérale sur les denrées alimentaires concernant entre autres les eaux minérales, entrée en vigueur le ler avril 1988 après une période transitoire de deux ans, exige surtout une plus grande constance dans la qualité et une garantie de la pureté.

...

There are multiple ways of matching up the four English sentences with the three French sentences, to say nothing of the million-sentence problem. Manually editing is out of the question, so we must seek automatic solutions. You may imagine an algorithm along the lines of the "decipherment method" itself: if I know that "house/maison" form a word pair, then I could guess that corpus sentences "the house is blue" and "la maison est bleue" may form a pair, in which case "blue/bleue" may form another word pair, in which case ...this would work, although such decipherment is computationally very expensive. More practical methods rely on rougher clues like:

- French sentences are usually in the same order as the English sentences (even though within-sentence word order can be quite different).
- Short French sentences usually correspond to short English sentences, and long to long.

- Corresponding French and English sentences often contain many of the same character sequences, due to proper names, numbers, and cognates.

For example, we can transform the above sentence-alignment problem into one where sentences are replaced by their word counts:

```
English:   ... 13  6  19  12 ...
French:    ... 15  7  43 ...
```

Clearly, the 43-word French sentence is a good candidate to match the two English sentences of 19 and 12 words each. Other alignments, such as one matching the 7 with both the 6 and 19, seems less likely.

By now, many researchers have worked with many sorts of bilingual texts, and all have faced the problem of creating a sentence-aligned corpus. Whenever many researchers face the same problem, competition ensues—in this case, for the most accurate, speedy, noise-robust, language-independent algorithms. These methods are quite successful, and (surprisingly) you can find more recent papers on bilingual text alignment than on machine translation itself. See [Catizone *et al.*, 1989; Brown *et al.*, 1991; Gale and Church, 1991; Kay and Roscheisen, 1993; Chen, 1993; Simard and Plamondon, 1996; Macklovitch and Hannan, 1996]. Alignment problems become more severe when sentence boundaries are hard to find, as is the case with web documents, imperfectly scanned documents, and distant language pairs (e.g., Chinese/English). These problems have led to methods such as [Church, 1993; Fung and McKeown, 1994; Melamed, 1997].

Using the Hansard corpus, [Brown *et al.*, 1990; Brown *et al.*, 1993b] present an MT system that works somewhat like the one we used for Centauri—translate the words, and get them in the right order. However, it deals explicitly with uncertainty and ambiguity: How to translate word x? Should word y go before or after word z? In a given sentence, some decisions will go well together, and others will not. Probability theory helps the machine make the best overall sequence of decisions it can, given what it knows.

2.2 Language Model

First let's look at word order. In our Centauri translation, we had a bag of words and we wanted to get them in the right order. But suppose we had several different bags, corresponding to different possible collections of word translations. We could find the best word order for each bag, but how could we choose between the resulting sentences? The answer is to assign a probability to any conceivable sequence of words. We then pick the most probable sequence (from any bag).

Sequences like "John saw Mary" and "that's enough already" should be probable, while "John Mary saw" and "radiate grouper engines" should be improbable. Linguistics has traditionally divided sequences into grammatical and ungrammatical, but in MT we are constantly forced to choose between two grammatical sentences. For example, which is a better translation, (A) or (B)?

```
(A) John viewed Mary in the television.
(B) John saw Mary on TV.
```

On the other hand, the speech recognition community has plenty of experience assigning probabilities to word sequences—e.g., preferring "bears hibernate" over "bare cyber Nate." Typical methods use word-pair or word-triple counts, which are converted into probabilistic quantities, e.g.,

```
P(oil | Arabian)
```

which is the chance that, given the word "Arabian," the next word will be "oil." The nice thing about these quantities is that they can be directly and automatically estimated from a large English corpus. In my corpus, "Arabian" occurred 471 times and was followed by "oil" 62 times, so P(oil | Arabian) = 62/471, or 13%. This is called a *conditional bigram probability.* A conditional *trigram* probability looks like this:

```
P(minister | Arabian oil)
```

That is, given the words "Arabian oil," what is the chance that the next word is "minister"? My corpus gives 8/25, or 32%.

To assign a probability to a whole sentence, we multiply the conditional probabilities of the n-grams it contains. So, a good sentence will be one with a lot of common subsequences. In the bigram case:

```
P(I found riches in my backyard) ~
   P(I | start-of-sentence) x
   P(found | I) x
   P(riches | found) x
   P(in | riches) x
   P(my | in) x
   P(backyard | my) x
   P(end-of-sentence| backyard)
```

It's easy to see how this is useful for word ordering—there is a strong preference for "I found riches in my backyard" over "My I in riches backyard found." In fact, [Brown *et al.*, 1990] describe a small experiment in restoring order to scrambled English sentences ("bag generation"). For sentences of fewer than ten words, a probabilistic program was able to restore the original word order 63% of the time. Under a looser meaning-preserving metric, the program scored 84%. Longer sentences were significantly tougher to reconstruct, however.

A technical point arises when P(y | x) is zero, i.e, when the word-pair "x y" has never been observed in training. Any zero-probability subsequence will make the whole sentence's product go to zero. This problem is particularly acute for word-triples—a phrase like "found riches in" may never appear in a training corpus, but that doesn't mean it's not a decent trigram. There is now a large literature on how to best assign non-zero probabilities to previously unseen n-grams. This is called *smoothing.* See [Chen, 1996] for a comparison of several methods. The overall topic of assigning probabilities to sentences is called *language modeling.*

Language modeling is useful not only for word ordering, but also for choosing between alternative translations like

```
(A) I found riches in my backyard.
(B) I found riches on my backyard.
```

This decision comes up in Spanish-English MT, where both "in" and "on" correspond to "en." In my corpus, the trigram "in my backyard" appears seven times, while "on my backyard" never occurs—so (A) is preferred. That shows you can attack some disambiguation problems by looking *only* at the target language. But not all! Consider two possible translations:

```
(A) Underline it.
(B) Emphasize it.
```

English bigram frequencies may slightly prefer (B), but the only way to really decide correctly is to look at the original Spanish sentence. The Spanish verb "subrayar" translates either as "underline" or as "emphasize," but mostly as "underline." In fact, to say "emphasize" in Spanish, you usually say "acentuar." Now we are talking about probabilistic quantities that connect Spanish words to English words, rather than English words to each other. These cross-language quantities make up a *translation model* that complements the language model. We can combine the two models by multiplying their scores.

2.3 Translation Model

In our extraterrestrial example, the translation model was simply a bilingual dictionary that supplied possible word translations for the language models. As the "emphasize/underline" example shows, we must also build probabilities into that dictionary. There is one tricky decision to make. Should the translation model contain quantities like P(emphasize | subrayar), or P(subrayar | emphasize)? Using P(english | spanish) seems more intuitive, because we are translating Spanish to English. For a given Spanish sentence S, we would find the English sentence E that maximizes $\mathrm{P(E) \cdot P(E \mid S)}$. Mathematically, however, it is more accurate to maximize $\mathrm{P(E) \cdot P(S \mid E)}$. This is because of Bayes' Rule:

$$\mathrm{P(E \mid S)} = \frac{\mathrm{P(E) \cdot P(S \mid E)}}{\mathrm{P(S)}}$$

Because P(S) is fixed for a given Spanish sentence, we can ignore it while trying to maximize $\mathrm{P(E \mid S)}$:

$$\mathrm{P(E \mid S)} \sim \mathrm{P(E) \cdot P(S \mid E)}$$

We therefore divide the responsibility between English probabilities and Spanish-given-English probabilities. Here are scores for (A) and (B) above (given "subrayar" as input):

```
(A) Underline it.
      P(underline) x
      P(it | underline) x
      P(subrayar | underline)

(B) Emphasize it.
      P(emphasize) x
      P(it | emphasize) x
      P(subrayar | emphasize)
```

Option (A) is good because "underline" is a common word *and* it usually translates as "subrayar." Option (B) is worse because when you translate "emphasize" to Spanish, it usually comes out as "acentuar," leaving very little probability in P(subrayar | emphasize).

If it seems backwards, it is. You have to imagine you are building an English-to-Spanish translator, but when you actually go to run it, you feed in Spanish and ask, "What English input would have caused this Spanish sentence to pop out?" The right answer will be a fluent English sentence (language model) that means what you think it means (translation model).

You may wonder why solving $\mathrm{P(S \mid E)}$ instead of $\mathrm{P(E \mid S)}$ makes life any easier. The answer is that $\mathrm{P(S \mid E)}$ doesn't have to give good Spanish translations. In fact, $\mathrm{P(S \mid E)}$ can assign lots of probability to bad Spanish sentences, as long as they contain the right words. Any of the following might be reasonably probable under the type of $\mathrm{P(S| E)}$ we are considering:

```
P(Yo no comprendo | I don't understand)
P(Comprendo yo no | Don't understand I)
P(No yo comprendo | I don't understand)
P(Comprendo yo no | I don't understand)
P(Yo no comprendo | I understand don't)
P(Yo no comprendo | Understand I don't)
```

P(S | E) can be sloppy because P(E) will worry about word order. This sloppiness actually gives some measure of robustness in translating ungrammatical Spanish input. It is also nice for estimating the translation model probabilities. Suppose we assume that for a given sentence pair S/E, P(S | E) is simply the product of word translation probabilities between them, irrespective of word order:

```
P(Yo no comprendo | I don't understand) ~
   P(Yo | I) x
   P(Yo | don't) x
   P(Yo | understand)
   P(no | I) x
   P(no | don't) x
   P(no | understand)
   P(comprendo | I) x
   P(comprendo | don't) x
   P(comprendo | understand)
```

We could then estimate word translation probabilities from a bilingual corpus. To estimate P(comprendo | understand), we could retrieve all sentence pairs containing the English word "understand," count how many times "comprendo" co-occurred, and divide by the total number of words in the Spanish half of this sub-corpus.

This is a reasonable first cut, but it has problems. For one, P(comprendo | understand) will come out too low in absolute terms. Even if "comprendo" appears every time "understand" appears, P(comprendo | understand) may still be only 0.05. Worse, other probabilities like P(la | understand) will come out too high: when you see "understand" in English, you very often see "la" in Spanish. But that's only because "la" is an extremely frequent word.

The right idea is to use a decipherment method, like the one we used for Centauri and Arcturan. "Understand" might co-occur with both "la" and "comprendo," but if we've *previously established* a strong link between "the" and "la," then we should lean strongly toward "comprendo." Doing that should reduce the chance that the English word "don't" translates as "comprendo," because "don't/comprendo" only co-occur when "understand" is already in the neighborhood. After such decipherment, P(comprendo | understand) should be close to one. P(la | the) might be 0.4, with the rest going to P(el | the), etc.

This whole method needs to be bootstrapped—we can't keep assuming previously established links. Fortunately, there is an automatic bootstrapping algorithm, called estimation-maximization (EM) [Baum, 1972]. The key to applying EM is the idea of word alignments. A word alignment connects words in a sentence pair such that each English word produces zero or more Spanish words, and each Spanish word is connected to exactly one English word. The longer a sentence pair is, the more alignments are possible. For a given sentence pair, some alignments are more reasonable than others, because they contain more reasonable word translations. Now we can revise our approximation of P(S | E):

```
P(Yo no comprendo | I don't understand) ~
     P(Alignment1) x P(Yo | I) x
                     P(no | don't) x
                     P(comprendo | understand)
   + P(Alignment2) x P(Yo | don't) x
                     P(no | I) x
                     P(comprendo | understand)
   + P(Alignment3) x P(Yo | understand) x
                     P(no | I) x
                     P(comprendo | don't)
   + P(Alignment4) x P(Yo | I) x
                     P(no | understand) x
                     P(comprendo | don't)
   + P(Alignment5) x P(Yo | don't) x
                     P(no | understand) x
                     P(comprendo | I)
   + P(Alignment6) x P(Yo | understand) x
                     P(no | don't) x
                     P(comprendo | I)
```

(I have left out alignments where English words produce multiple or zero Spanish words.)

EM training is quite powerful, but difficult to master. At an abstract level, it is simply a way to mechanize the trial-and-error decipherment we used for Centauri/Arcturan. At a deeper level, EM training tries to find the word translation probabilities which maximize the probability of one half the corpus (say, Spanish) given the other half (say, English). Understanding how it really works requires a bit of calculus. Neural networks require a similar bit of calculus. Of course, it is possible to implement both EM and neural networks without precisely understanding their convergence proofs. I'll give a brief description of EM training here.

We first assume all alignments for a given sentence pair are equally likely. One sentence pair might have 256 alignments, each with $p = 1/256$, while another sentence pair might have 10^{31} alignments, each with very small p. Next we count up the word pair connections in all alignments of all sentence pairs. Each connection instance is weighted by the p of the alignment in which it occurs. That way, short (less ambiguous) sentences have more weight to throw around. Now we consider each English word in turn (e.g., "understand"). It has weighted connections to many Spanish words, which we normalize to sum to one. This gives the first cut at word translation probabilities. We then notice that these new probabilities make some alignments look better than others. So we use them to re-score alignments so that they are no longer equally likely. Each alignment is scored as the product of its word translation probabilities, then normalized so that alignments' probabilities for a given sentence pair still sum to one.

Then we repeat. Newer alignment probabilities will yield newer, more accurate word translation probabilities, which will in turn lead to better alignments. Usually one alignment will beat out all of the others in each sentence pair. At that point we stop, and we have our word translation probabilities. Given a new sentence pair S/E, we can estimate P(S | E) by using those probabilities. (See [Dagan and Church, 1994; Smadja *et al.*, 1996; Ker and Chang, 1997] for further discussion of this and other methods for word- and phrase-alignment.)

2.4 Translation Method

So much for decipherment. The last thing we need is a translation algorithm. I mentioned Bayes' Rule earlier—given a Spanish sentence S, we want to find the English sentence E that maximizes $P(E) \cdot P(S \mid E)$. We could try all conceivable E's, but that would take too long. There are techniques with which to direct such a search, sacrificing optimality for efficiency. [Brown *et al.*, 1990] briefly sketches an A*-based stack search, while more detailed discussions can be found in [Wang and Waibel, 1997; Wu, 1996; Tillmann *et al.*, 1997]. A translation method must also deal with unknown words, e.g., names and technical terms. When languages use different alphabets and sound patterns, these terms must often be translated phonetically [Knight and Graehl, 1997].

2.5 Results

Initial results in statistical word-for-word MT were mixed. Computational limitations restricted experiments to short sentences and a 3000-word vocabulary. While good with individual words, this system did not cope well with simple linguistic/structural issues, preferring, for example, "people with luggage is here" over "people with luggage are here." It used very little context for sense disambiguation, and it failed to take source language word order into account. You might imagine that these shortcomings would lead naturally to parsing and semantic analysis, but [Brown *et al.*, 1993b] iconoclastically continued to push the word-for-word paradigm, adding "distortion" probabilities (for keeping French and English words in roughly the same order), context-sensitive word translation probabilities, and long-distance language modeling. Bilingual dictionaries were used to supplement corpus knowledge [Brown *et al.*, 1993a]. These improvements, combined with more efficient decipherment and translation algorithms, led to a full-scale French-English MT system called CANDIDE. This system performs as well as the best commercial systems, with no hand-built knowledge bases! That's the good news. Where to go from here? It is unclear whether the outstanding problems can be addressed within the word-for-word framework, via better statistical modeling or more training data. It is also unclear how this method would perform on language pairs like Vietnamese/English, with radically different linguistic structure and less bilingual data on line.

It is interesting to note that the statistical method will always work hard to find a translation, even if the input sentence happens to appear verbatim in the training corpus. In this case, a good translation can be retrieved by simple lookup. This idea is the basis of another corpus-based MT approach, called *example-based* MT [Nagao, 1984; Sato, 1992]. When exact lookup fails, an example-based system will look for a close match and attempt to modify the corpus translation to fit the new sentence. This type of "retrieve-and-tweak" strategy has strengths and weaknesses similar to those of case-based reasoning in AI.

3 Syntax-Based Translation

Knowing the syntactic structure of a source text—where phrase boundaries are, and which phrases modify which—can be very useful in translation. Most hand-crafted commercial systems do a syntactic analysis followed by transfer, in which phrases are translated and re-ordered. There are many opportunities for empirical methods in such a framework. The most obvious is trainable parsing [Magerman, 1995; Bod, 1996; Charniak, 1996; Collins, 1997]. Unfortunately, such parsers often require a large treebank (collection of manually parsed sentences), and treebanks are not yet available in most languages. Any advances in grammar induction from raw text will therefore have a big impact on MT. Some MT systems use hand-crafted grammars with a word-skipping parser

[Lavie, 1994; Yamada, 1996] that tries to find a maximal parsable set of words.

Given reasonably accurate parsing systems (trained or handcrafted) it is possible to write transfer rules by hand and use a language model to do lexical and structural disambiguation [Yamron *et al.*, 1994; Hatzivassiloglou and Knight, 1995]. It is also possible to learn transfer rules from bilingual corpora automatically: both halves of the corpus are parsed, and learning operates over tree pairs rather than sentence pairs.

A more ambitious, potentially powerful idea is to train directly on sentence pairs, learning both phrase structure and translation rules at the same time. While a treebank tells you a lot about phrase structure in a given language, translations may also tell you something—serving as a sort of poor man's treebank. Research in this vein includes [Wu, 1995] and [Alshawi *et al.*, 1997]. The basic idea is to replace the word-for-word scheme, in which words fly around willy-nilly, with a tighter syntax-based MT model; probabilities are then still selected to best fit the sentence-pair corpus. While it is clear that fairly good word-for-word alignments are recoverable from bilingual text, it remains to be seen whether accurate syntactic alignments are similarly recoverable, and whether those alignments yield reasonable translations.

Yet another possibility is to bring a human linguist back into the loop [Hermjakob and Mooney, 1997] as a source of correct parse and transfer decisions. The linguist also supplies general features that are useful for learning to make good decisions in new contexts.

4 Semantics-Based Translation

Semantics-based MT has already produced high-quality translations in circumscribed domains. Its output is fluent because it employs meaning-to-text language generation instead of gluing phrases together and hoping the result is grammatical. Its output is accurate because it reasons with a world model. However, this strategy has not yet scaled up to general-purpose translation.

Semantics-based MT needs parsing plus a whole lot more. Fuel for the analysis side includes a semantic lexicon (for mapping words onto concept and roles), semantic rules (for combing word meanings into sentence meanings), and world knowledge (for preferring one reading over another). The language generation phase also needs a lexicon and rules, and some way of preferring one rendering over another. There are many opportunities for empirical techniques. A language model may be used to resolve any ambiguities percolated from morphology, parsing, semantics, and generation. In general, statistical knowledge can usefully plug gaps in all incomplete knowledge bases [Knight *et al.*, 1995], letting designers and linguists focus on deeper problems that elude automatic training. Semi-automated knowledge acquisition plays an important role in creating large-scale resources like conceptual models and lexicons [Knight and Luk, 1994; Viegas *et al.*, 1996].

For the statistically oriented, Bayes' rule is still usefully applied—let E be an English sentence, S be Spanish, and M be a representation of a sentence meaning. This M may be a deep Interlingua or a shallow case frame. Then we can break translation down into two phases:

$$\begin{array}{ll} P(M \mid S) \sim P(M) \cdot P(S \mid M) & \text{analysis} \\ P(E \mid M) \sim P(E) \cdot P(M \mid E) & \text{generation} \end{array}$$

P(M) is essentially a world model. It should, for instance, assign low probability to FLY(CANYON). P(S | M) and P(M | E) are like translation models from Section 2. P(E) is our old friend the language model. There are many open problems: Can these distributions be estimated from existing resources? Can a system learn to distinguish sensible meanings from nonsense ones by bootstrapping off its own (ambiguous) analyses? Can translation models be learned, or can they be supplanted with easy-to-build handcrafted systems?

The language generation phase provides a good case study. Although there are many applications for language generation technology, MT is a particularly interesting one, because it forces issues of scale and robustness. [Knight and Hatzivassiloglou, 1995] describe a hybrid generator called NITROGEN, which uses a large but simple dictionary of nouns, verbs, adjectives, and adverbs, plus a hand-built grammar. This grammar produces alternative renderings, which are then ranked by a statistical language model. Consider a meaning like this one, computed from a Japanese sentence:

```
(A / ACCUSATION
   :agent SHE
   :patient (T / THEFT
               :agent HE
               :patient (M / MOTORCAR)))
```

(Roughly: there is an accusation of theft, the accuser is "she", the thiever is "he", and the thieved-object is a motorcar).

This is a bare-bones representation. There are events and objects, but no features for singular/plural, definiteness, or time—because many of these are not overtly marked in the Japanese source. NITROGEN's grammar offers 381,440 English renderings, including:

```
Her incriminates for him to thieve an automobiles.

There is the accusation of theft of the car by him by her.

She impeaches that he thieve that there was the auto.
```

It is extremely time-consuming to add formal rules describing why each of these thousands of sentences is suboptimal, but a statistical language model fills in nicely, ranking as its top five choices:

```
1  She charged that he stole the car.
2  She charged that he stole the cars.
3  She charged that he stole cars.
4  She charged that he stole car.
5  She charges that he stole the car.
```

Comparable scale-ups—particularly in syntactic grammar, semantic lexicons, and semantic combination rules—will be necessary before semantics-based MT can realize its promise.

5 Evaluation

Evaluating MT is a tricky business. It's not like speech recognition, where you can count the number of wrong words. Two translations may be equally good without having a single word in common. Omitting a small word like "the" may not be bad, while omitting a small word like "not" may spell disaster.

The military routinely evaluates *human* translators, but machine translators fall off the low end of that scale. Many specialized methods for evaluating machines have been proposed and implemented. Here are a few:

- Compare human and machine translations. Categorize each machine-generated sentence as (1) same as human, (2) equally good, (3) different meaning, (4) wrong, or (5) ungrammatical [Brown *et al.*, 1990].
- Build a multiple-choice comprehension test based some newspaper article, but force the test takers to work from a translation instead of the original article [White and O'Connell, 1994]. If the translation is too garbled, the test takers won't score very high.
- Develop error categories (pronoun error, word selection error, etc.) and divide them according to improvability and effect on intelligibility [Flanagan, 1994]. Tabulate errors in text.

These methods can be quite expensive. More automatic methods can be envisioned—a common idea is to translate English into Spanish and back into English, all by machine, and see if the English comes back out the same or not. Even if it does, that's no guarantee. I have a translator on my Macintosh that turns the phrase "why in the world" into "porqué en el mundo," then nicely back into "why in the world." Great, except "porqué en el mundo," doesn't mean anything in Spanish! A more useful automatic evaluation [Gdaniec, 1994] correlates human quality judgments with gross properties of text, such as sentence length, clauses per sentence, not-found words, etc. While this correlation won't let you compare systems, it will tell you whether or not a new document is suitable for MT.

There are also metrics for human-machine collaboration. Such collaboration usually takes the form of human pre-editing, MT, and human-postediting. A lot of translation is now done this way, but the savings over human-alone-translation vary quite a bit depending on the type of document.

What can we conclude from this work on evaluation?

First, MT evaluation will continue to be an interesting topic and an active field in its own right, no matter what happens in MT proper. Second, formal MT evaluation is still too expensive for individual researchers. They will continue to use the "eyeball" method, rarely publishing learning curves or comparative studies. Third, general-purpose MT output is nowhere near publication quality (this requires human postediting). Of course, many applications do not require publication quality. For people who use web- and email-MT, the choice is not between machine translation and human translation—it is between machine translation and no translation. And fourth, general-purpose MT is more accurate for closer language pairs (like Spanish/English) than more distant ones (like Japanese/English).

6 Conclusion

I have described several directions in empirical MT research. There is as yet no consensus on what the right direction is. (In other words, things are exciting.) Word-for-word proponents look to semantics as a dubious, mostly uninterpretable source of training features, while semantics-proponents view statistics as a useful but temporary crutch. Knowledge-bottlenecks, data-bottlenecks, and efficiency-bottlenecks pose interesting challenges.

I expect that in the near future, we will be able to extract more useful MT knowledge from bilingual texts, by applying more linguistically plausible models. I also expect to see knowledge being gleaned from monolingual (non-parallel) corpora, which exist in much larger quantities. Semantic dictionaries and world models, driven by AI applications mostly outside MT, will continue to scale up.

Will general-purpose MT quality see big improvements soon? In this difficult field, it is useful to remember the maxim: "Never be more predictive than 'watch this!'" I am optimistic, though, because the supply of corpus-based results is increasing, as is the demand for MT products. I see MT following a path somewhat like that of computer chess. Brute force brought the computer to the table, but it took carefully formalized chess knowledge to finally beat the human champion. A similar combination of brute-force statistics and linguistic knowledge makes up the current attack on MT. The main thing is to keep building and testing MT systems, the essence of the empirical approach.

References

Alshawi, Hiyan, Adam Buchsbaum, and Fei Xia. 1997. A comparison of head transducers and transfer for a limited domain translation applications. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL).*

Baum, L. E. 1972. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities* 3.

Bod, Rens. 1996. *Enriching Linguistics with Statistics: Performance Models of Natural Language.* PhD thesis, University of Amsterdam.

Brown, Peter, John Cocke, Stephen Della Pietra, Vincent Della Pietra, Fredrick Jelinek, John Lafferty, Robert Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics* 16(2).

Brown, Peter, Jennifer Lai, and Robert Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL).*

Brown, Peter, Stephen Della Pietra, Vincent Della Pietra, Meredith Goldsmith, Jan Hajic, Robert Mercer, and Surya Mohanty. 1993. But dictionaries are data too. In *Proceedings of the ARPA Human Language Technology Workshop.*

Brown, Peter, Stephen Della Pietra, Vincent Della Pietra, and Robert Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics* 19(2).

Catizone, Roberta, Graham Russell, and Susan Warwick. 1989. Deriving translation data from bilingual texts. In *Proceedings of the First International Lexical Acquisition Workshop, IJCAI.*

Chandioux, John and Annette Grimaila. 1996. Specialized machine translation. In *Proc. Conference of the Association for Machine Translation in the Americas (AMTA).*

Charniak, Eugene. 1996. Tree-bank grammars. In *Proceedings of the National Conference on Artificial Intelligence (AAAI).*

Chen, Stanley. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL).*

Chen, Stanley. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL).*

Church, Ken. 1993. Char-align: A program for alinging parallel texts at the character level. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL).*

Collins, Michael. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL).*

Dagan, Ido and Ken Church. 1994. Termight: Identifying and translating technical terminology. In *Proceedings of EACL.*

Flanagan, Mary. 1994. Error classification for MT evaluation. In *Proc. Conference of the Association for Machine Translation in the Americas (AMTA).*

Fung, Pascale and Kathy McKeown. 1994. Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping. In *Proc. Conference of the Association for Machine Translation in the Americas (AMTA).*

Gale, William and Ken Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL).*

Gdaniec, Claudia. 1994. The Logos translatibility index. In *Proc. Conference of the Association for Machine Translation in the Americas (AMTA).*

Hatzivassiloglou, Vasileios and Kevin Knight. 1995. Unification-based glossing. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).*

Hermjakob, Ulf and Raymond Mooney. 1997. Learning parse and translation decisions from examples with rich context. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL).*

Isabelle, Pierre, M. Dymetman, G. Foster, J-M. Jutras, Eliott Macklovitch, F. Perrault, X. Ren, and Michel Simard. 1993. Translation analysis and translation automation. In *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI).*

Kay, Martin and Martin Roscheisen. 1993. Text-translation alignment. *Computational Linguistics* 19(1).

Ker, Sue and Jason Chang. 1997. A class-based approach to word alignment. *Computational Linguistics* 23(2).

Knight, Kevin, Ishwar Chander, Matthew Haines, Vasileios Hatzivassiloglou, Eduard Hovy, Masayo Iida, Steve Luk, Richard Whitney, and Kenji Yamada. 1995. Filling knowledge gaps in a broad-coverage MT system. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI).*

Knight, Kevin and Jonathan Graehl. 1997. Machine transliteration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL).*

Knight, Kevin and Vasileios Hatzivassiloglou. 1995. Two-level, many-paths generation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL).*

Knight, Kevin and Steve K. Luk. 1994. Building a large-scale knowledge base for machine translation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI).*

Lavie, Alon. 1994. An integrated heuristic scheme for partial parse evaluation. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL), Student Session.*

Macklovitch, Elliott. 1994. Using bi-textual alignment for translation validation: The TransCheck system. In *Proc. Conference of the Association for Machine Translation in the Americas (AMTA)*.

Macklovitch, Elliott and M. L. Hannan. 1996. Line 'em up: Advances in alignment technology and their impact on translation support tools. In *Proc. Conference of the Association for Machine Translation in the Americas (AMTA)*.

Magerman, David. 1995. Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Melamed, I. Dan. 1997. A portable algorithm for mapping bitext correspondence. In *Proceedings of the 35th Conference of the Association for Computational Linguistics*.

Nagao, M. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In *Artificial and Human Intelligence*, ed. A. Elithorn and R. Bernerji, 173–180. North-Holland.

Nyberg, Eric and Teruko Mitamura. 1992. The KANT system: Fast, accurate, high-quality translation in practical domains. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*.

Sato, S. 1992. CTM: An example-based translation aid system. In *Proceedings of the 14th International Conference on Computational Linguistics*.

Simard, Michel and Pierre Plamondon. 1996. Bilingual sentence alignment: Balancing robustness and accuracy. In *Proc. Conference of the Association for Machine Translation in the Americas (AMTA)*.

Smadja, Frank, Kathleen McKeown, and Vasileios Hatzivassiloglou. 1996. Translating collocations for bilingual lexicons: A statistical approach. *Computational Linguistics* 22(1).

Tillmann, Christoph, Stephan Vogel, Hermann Ney, and Alex Zubiaga. 1997. A DP-based search using monotone alignments in statistical translation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Viegas, Evelyne, Boyan Onyshkevych, Victor Raskin, and Sergei Nirenburg. 1996. From submit to submitted via submission: On lexical rules in large-scale lexicon acquisition. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Wang, Ye-Yi and Alex Waibel. 1997. Decoding algorithm in statistical machine translation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*.

White, John and Teri O'Connell. 1994. Evaluation in the ARPA machine translation program: 1993 methodology. In *Proceedings of the ARPA Human Language Technology Workshop*.

Wu, Dekai. 1995. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Wu, Dekai. 1996. A polynomial-time algorithm for statistical machine translation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yamada, Kenji. 1996. A controlled skip parser. In *Proc. Conference of the Association for Machine Translation in the Americas (AMTA).*

Yamron, Jonathan, James Cant, Anne Demedts, Taiko Dietzel, and Yoshiko Ito. 1994. The automatic component of the LINGSTAT machine-aided translation system. In *Proceedings of the ARPA Human Language Technology Workshop.*