AI Magazine Volume 18 Number 4 (1997) (© AAAI)

Automating Knowledge Acquisition for Machine Translation

Kevin Knight

■ Machine translation of human languages (for example, Japanese, English, Spanish) was one of the earliest goals of computer science research, and it remains an elusive one. Like many AI tasks, translation requires an immense amount of knowledge about language and the world. Recent approaches to machine translation frequently make use of text-based learning algorithms to fully or partially automate the acquisition of knowledge. This article illustrates these approaches.

How can we write a computer program to translate an English sentence into Japanese? Anyone who has taken a graduate-level course in AI 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 (such as *canyon*) onto concepts (such as canyon) and to a world model that contains facts about reality (such as canyons don't fly). In this way, an ambiguous sentence such as "John saw the Grand Canyon flying to New York" gets the correct 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, such as 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, do we fly; otherwise, we're just taxiing. How about "John flew me to New York"? This is another meaning of *fly*, involving drive-airplane as well as ride-in-airplane. In addition, if I state "United flew me to New York," I might say that the airplane that I rode in was driven by an employee of the airline that owns the airplane. 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 long journey.

Meanwhile, in the real world (not the formal model), people are buying shrink-wrapped machine-translation software for $50. E-mail programs ship with optional language-translation capacity. Companies use machine translation to translate manuals and track revisions. Machine-translation 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 machine translation, and not all of them use formal semantic representations. (I'll describe some in this article.) This should come as no surprise because machine translation predates AI as a field. An AI scientist could easily spend two months representing "John saw the Grand Canyon flying to New York," but anybody with a bilingual dictionary can build a general-purpose, word-for-word translator in a day. With the correct language pair, and no small amount of luck, word-for-word results might be intelligible: "John vi el Grand Canyon volando a New York." This is okay Spanish. However, most of the time, the translations will be terrible, which is why machine-translation researchers are busy building high-quality semantics-based machine-translation systems in circumscribed domains, such as weather reports (Chandioux and Grimaila 1996) and heavy-equipment manuals (Nyberg and Mitamura 1992); abandoning automatic machine-translation and building software to assist human translators instead (Dagan and Church 1997; Macklovitch 1994; Isabelle et al. 1993); and developing automatic knowledge-acquisi-

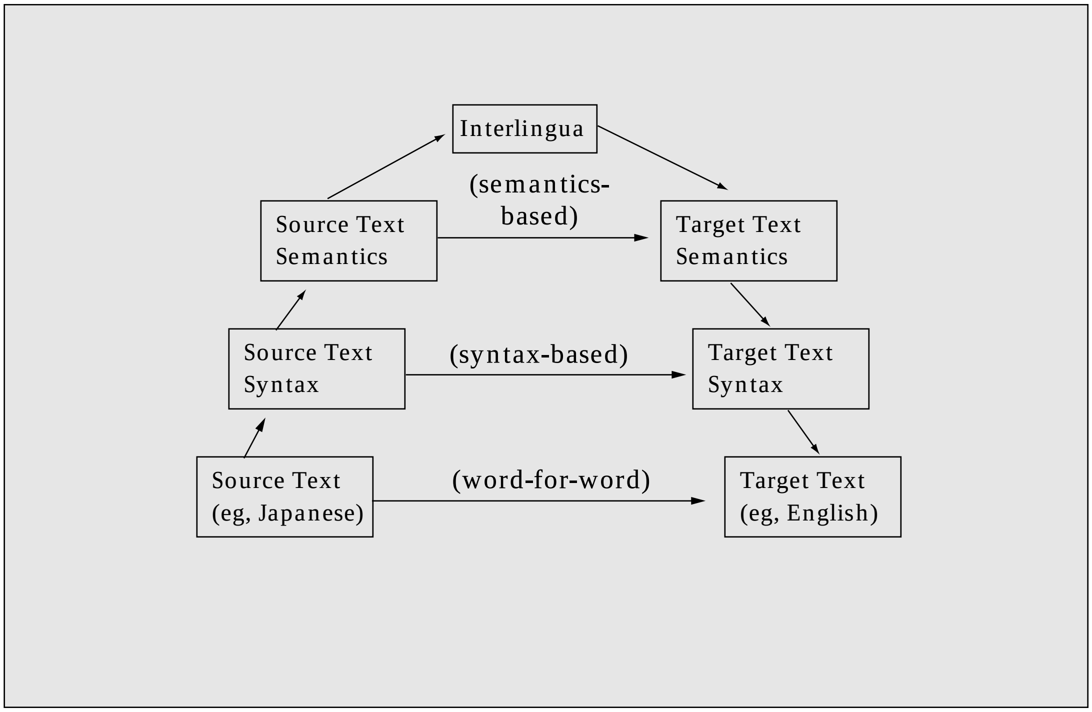


Figure 1. Different Strategies for Machine Translation.

tion techniques for improving general-purpose machine translation (Knight et al. 1995; Yamron et al. 1994; Brown et al. 1993b).

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

If you take a poll of general-purpose machine-translation users, you will find that they want many improvements: speed, compatibility with their word processor, customizable dictionaries, translation memory, revision tracking, and so on. At the top of everyone's list, however, is better output quality. Unfortunately, the machine-translation companies are busy supplying all these other things because they know how. Commercial translation quality has reached something of a plateau because 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, and so on) have yet to show up in commercial machine translation, and it looks like there are plenty more results to come.

From a scientific point of view, machine translation 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 machine translation 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 also wind up doing a limited amount of reordering, for example, 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 this case, word translations can be sensitive to syntactic relations; for example, we can translate a verb differently depending on its direct object. Still, the target text syntax will likely mirror that of the source text. Therefore, we can do a semantic analysis that abstracts away syntactic details (moving up the triangle in figure 1).

Ultimately, we arrive at an all-encompassing meaning representation called *interlingua*. You might wonder why semantics and interlingua are not the same thing: Here is an illustration from a Japanese-English machine-translation system I have worked on. It once translated a

Japanese sentence as "there is a plan that a baby is happening in her"—a reasonable translation but with a definite Japanese-semantics feel to it. Semantics is not an all-or-nothing proposition in machine translation 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 such as 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 machine translation, recent work in corpus-based learning offers the possibility of reducing the knowledge bottleneck.[1]

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 usefully be viewed as encrypted English—and machine translation as a kind of decipherment.

As a cipher, Russian looked to be complex. Sometimes a word would be encrypted one way and sometimes in 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 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 example. However, the cryptographers soon gave up this approach because of the sheer size of the problem. German encryption had been performed on rotor machines in the field, but machine translation 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 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 might soon be abducted by aliens and put to work in the Interstellar Translation Bureau, you are eager to analyze the data.

You first notice that corresponding sentences have the same number of words, except for sentence pair 11. You conjecture that the two languages are 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*. However, 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 although *ghirok* always means *hilat*, Centauri and Arcturan use different word-order schemes.

From a scientific point of view, machine translation remains the classic acid test of how much we understand about human language.

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

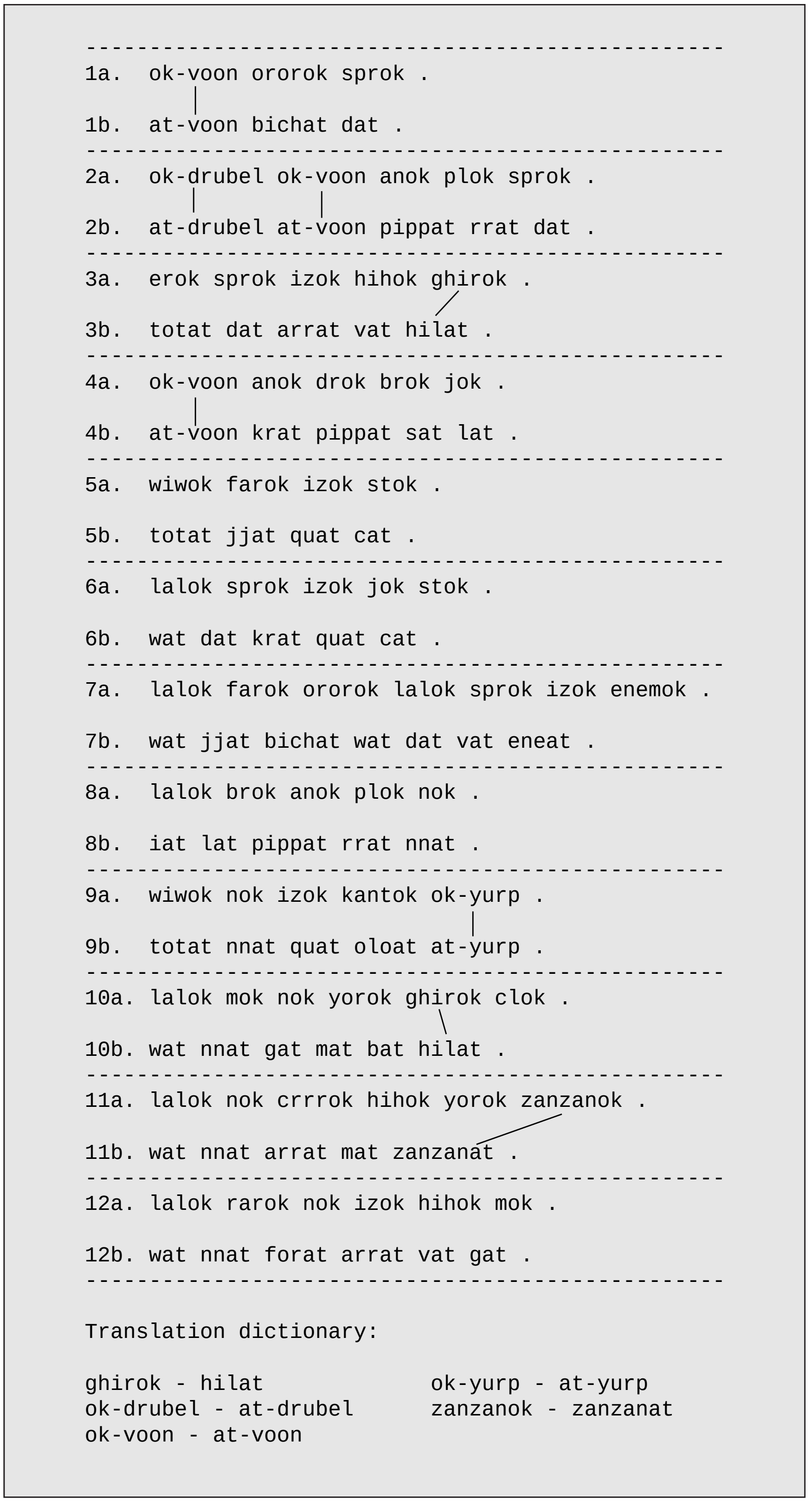
Sentence pair 1 is now partially explained, leaving two obvious hypotheses:

1. *ororok* means *bichat*
   (and *sprok* means *dat*).

2. *ororok* means *dat*
   (and *sprok* means *bichat*).

Of course, it could be the case that *ororok* is an (untranslated) auxiliary verb and that *sprok* has a phrasal translation *bichat dat*. However, 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 sentence 2a. Its translation (sentence 2b) has *dat*, lending support to the first hypothesis. You can now add two more entries to your translation dictionary and link their occurrences throughout the corpus (sentence pairs 1, 2, 3, 6, and 7).

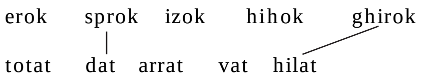
Sentence pair 2 is a logical place to continue because you only need to consider how to map *anok plok* onto *pippat rrat*. Again, two possibilities suggest themselves, but sentence pair 4 pushes you toward *anok-pippat* and, therefore, *plok-rrat*.



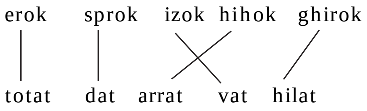
```
------------------------------------------------
1a.  ok-voon ororok sprok .

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.

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



The Centauri word *izok* would be translated as either *totat*, *arrat*, or *vat*, yet when you look at *izok* in sentence pair 6, none of those three words appear in the Arcturan. Therefore, *izok* appears to be ambiguous. The word *hihok*, however, is fixed in sentence pair 11 as *arrat*. Both sentence pairs 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*. Sentence pairs 5, 6, and 9 suggest that *quat* is its other translation. Through process of elimination, you connect the words *erok* and *totat*, finishing off the analysis:

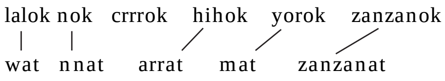


Notice that aligning the sentence pairs helps you to build the translation dictionary and that building the translation dictionary also helps you decide on correct alignments. You might call this the *decipherment method.*

Figure 3 shows the progress so far. With a ballpoint 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	

The dictionary shows ambiguous Centauri words (such as *izok*) and ambiguous Arcturan words (such as *totat*). It also contains a curious Centauri word (*crrrok*) that has no translation—after the alignment of sentence pair 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 an 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: (1) *lalok*, (2) *brok*, (3) *anok*, (4) *enemok*, (5) *ghirok*, (6) *kantok*, and (7) *ok-yurp*. You consider writing them down in this order (a simple word-for-word translation), but because you want to make a good first impression at the bureau, you also consider shifting the words around. There are 5040 (7!) possible word orders to choose from. Centauri text can 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 such as *vok, zerok, zinok,* and *ziplok*.

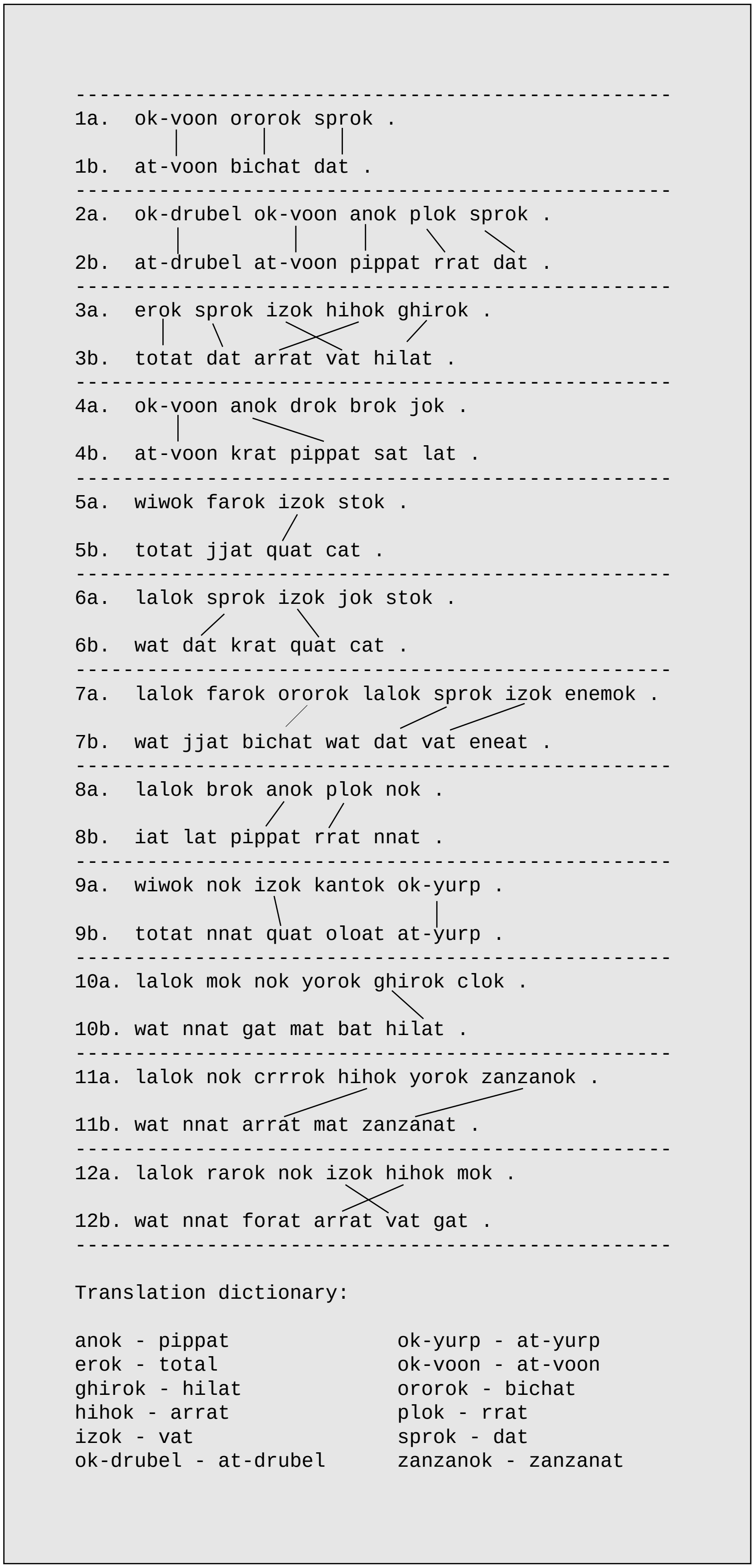
You are now in a position to evaluate your 5040 alternative word orders. As a shortcut, you might ask which word is most likely to start a sentence (or which word usually follows a period). Surely, it is *lalok*. Of the remaining six words, which best follows *lalok?* It is *brok*, then *anok*. However, 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*. However, 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 but 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



```
---------------------------------------------------
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.

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 such as “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:

First, only 2 of the 27 Centauri words were ambiguous, whereas in natural languages such as English, almost all words are ambiguous.

Second, sentence length was unchanged in all but one of the translations; in real translation, this is rare.

Third, the extraterrestrial sentences were much shorter than typical natural language sentences.

Fourth, words are translated differently depending on context. The translation method only used Centauri word-pair counts for context, preferring "wiwok rarok..." over "erok rarok...." However, resolving lexical ambiguity in general requires a much wider context and, often, sophisticated reasoning as well.

Fifth, output word order should be sensitive to input word order. Our method could not decide between output "John loves Mary" and "Mary loves John," even though one of the two is likely to be a terrible translation.

Sixth, 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.

Seventh, our method does not allow for any phrasal dictionary entries (for example, *anok plok* = *pippat rrat*), although human translators make extensive use of such dictionaries.

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?

However, 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;* and so on. 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.

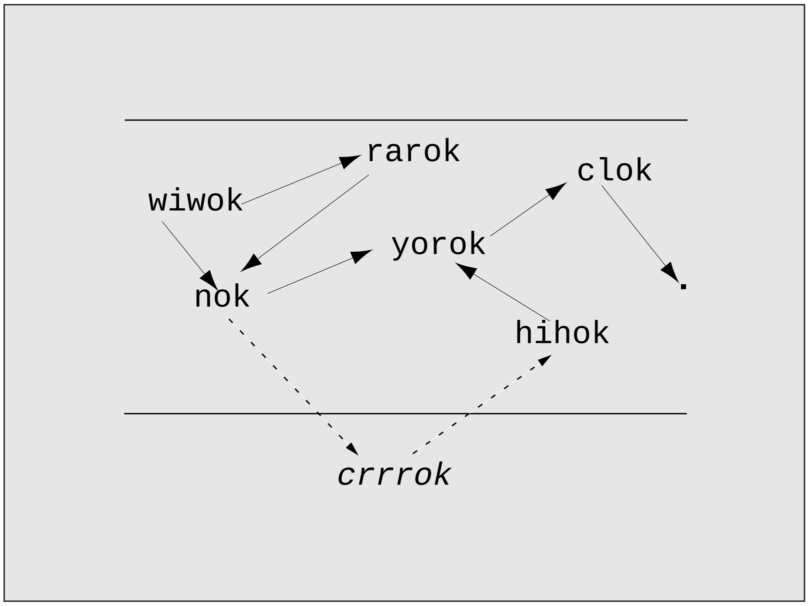


Figure 5. An Attempt to Put a Group of Centauri Words in the Right Order.

Arrows represent previously observed word pairs from figure 4.

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*; "do not sell" translates to "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 machine translation? Several questions arise: Is there a large bilingual corpus for some pair of natural languages? Can the corpus easily be converted to sentence-pair format? Can the decipherment method be automated? What does the algorithm look like? Can the translation method be automated? Perhaps most importantly, are the translations good?

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 machine translation. 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 can get translated as two English ones, or vice versa. 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. Manual editing is out of the question; so, we must seek automatic solutions. You can imagine an algorithm along the lines of the decipherment method itself: If I know that *house* and *maison* form a word pair, then I could guess that corpus sentences *the house is blue* and *la maison est bleue* might form a pair, in which case, *blue* and *bleue* might form another word pair, and so on. This method would work, although such decipherment is computationally expensive. More practical methods rely on rougher clues such as (1) French sentences are usually in the same order as the English sentences (even though within-sentence word order can be different); (2) short French sentences usually correspond to short English sentences; and (3) corresponding French and English sentences often contain many of the same character sequences because of proper names, numbers, and cognates.

For example, we can transform the previous 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 the 19, seem less likely.

By now, many researchers have worked with many sorts of bilingual text, 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 successful, and (surprisingly) you can find more recent papers on bilingual text alignment than on machine translation itself. See Macklovitch and Hannan (1996), Simard and Plamondon (1996), Chen (1993), Kay and Röscheisen (1993), Brown et al. (1991), Gale and Church (1991), and Catizone et al. (1989). 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 (for example, Chinese-English). These problems have led to the methods discussed by Melamed (1997), Fung and McKeown (1994), and Church (1993).

Using the Hansard corpus, Brown et al. (1993b, 1990) present a machine-translation 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.

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. However, 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 such as "John saw Mary" and "that's enough already" should be probable, but "John Mary saw" and "radiate grouper engines" should be improbable.

Linguistics has traditionally divided sequences into grammatical and ungrammatical, but in machine translation, 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.

The speech-recognition community has plenty of experience assigning probabilities to word sequences, for example, preferring "bears hibernate" over "bare cyber Nate." Typical methods use word-pair or word-triple counts, which are converted into probabilistic quantities, for example,

$$P(oil \mid Arabian) \; ,$$

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 directly and automatically be estimated from a large English corpus. In my corpus, *Arabian* occurred 471 times and was followed by *oil* 62 times; so, $P(oil \mid Arabian)$ = 62/471, or 13 percent. This is called a *conditional bigram probability.* A *conditional trigram probability* looks like the following:

$$P(minister \mid Arabian\ oil) \; .$$

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

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,

$$\begin{aligned} &P(I\ found\ riches\ in\ my\ backyard) \sim \\ &\quad P(I \mid start\text{-}of\text{-}sentence) \times \\ &\quad P(found \mid I) \times \\ &\quad P(riches \mid found) \times \\ &\quad P(in \mid riches) \times \\ &\quad P(my \mid in) \times \\ &\quad P(backyard \mid my) \times \\ &\quad P(end\text{-}of\text{-}sentence \mid backyard) \; . \end{aligned}$$

It's easy to see how simple probabilities are 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 10 words, a probabilistic program was able to restore the original word order 63 percent of the time. Under a looser meaning-preserving metric, the program scored 84 percent. Longer sentences were significantly tougher to reconstruct however.

A technical point arises when $P(y \mid x)$ is zero, that is, 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" might 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 nonzero probabilities to previously unseen *n*-grams, a process 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 such as

A. I found riches in my backyard.
B. I found riches on my backyard.

This decision comes up in Spanish-English machine translation, where both *in* and *on* correspond to *en*. In my corpus, the trigram "in my backyard" appears seven times, but "on my backyard" never occurs; so, A is preferred. Thus, 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 might 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.

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 the dictionary. There is one tricky decision to make. Should the translation model contain quantities such as *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 $P(E) \cdot P(E \mid S)$. Mathematically, however, it is more accurate to maximize $P(E) \cdot P(S \mid E)$ because of Bayes's rule:

$$P(E \mid S) = P(E) \cdot P(S \mid E) / P(S) \ .$$

Because *P*(*S*) is fixed for a given Spanish sentence, we can ignore it while we try to maximize *P*(*E* | *S*):

$$P(E \mid S) \sim P(E) \cdot P(S \mid E) \ .$$

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

A. Underline it.
P(*underline*) ×
P(*it* | *underline*) ×
P(*subrayar* | *underline*) .

B. Emphasize it.
P(*emphasize*) ×
P(*it* | *emphasize*) ×
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* into Spanish, it usually comes out as *acentuar*, leaving 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 correct answer will be a fluent English sentence (language model) that means what you think it means (translation model).

You might wonder why solving *P*(*S* | *E*) instead of *P*(*E* | *S*) makes life any easier. The answer is that *P*(*S* | *E*) doesn't have to give good Spanish translations. In fact, *P*(*S* | *E*) can assign lots of probability to bad Spanish sentences, as long as they contain the correct words. Any of the following might be reasonably probable under the type of *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*) ×
P(*Yo* | *don't*) ×
P(*Yo* | *understand*)
P(*no* | *I*) ×
P(*no* | *don't*) ×
P(*no* | *understand*)
P(*comprendo* | *I*) ×
P(*comprendo* | *don't*) ×
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* cooccurred, and divide by the total number of words in the Spanish half of this subcorpus.

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*) might still be only 0.05. Worse, other probabilities such as *P*(*la* | *understand*) will come out too high; when you see *understand* in English, you often see *la* in Spanish, but that's only because *la* appears frequently. The right idea is to use a decipherment method, such as the one we used for Centauri and Arcturan. *Understand* might cooccur 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*. Furthermore, the English word *don't* will not translate as *comprendo* because *don't* and *comprendo* only cooccur when *understand* is already in the neighborhood. After such decipherment, *P*(*comprendo* | *understand*) might be close to one. *P*(*la* | *the*) might be 0.4, with the rest going to *P*(*el* | *the*), and so on.

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* (Baum 1972). The key to applying estimation maximization 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 $P(S \mid E)$:

$$\begin{aligned} &P(\mathit{Yo\ no\ comprendo} \mid \mathit{I\ don't\ understand}) \sim \\ &\quad P(\mathit{Alignment1}) \times P(\mathit{Yo} \mid \mathit{I}) \times P(\mathit{no} \mid \mathit{don't}) \times P(\mathit{comprendo} \mid \mathit{understand}) \\ &\quad + P(\mathit{Alignment2}) \times P(\mathit{Yo} \mid \mathit{don't}) \times P(\mathit{no} \mid \mathit{I}) \times P(\mathit{comprendo} \mid \mathit{understand}) \\ &\quad + P(\mathit{Alignment3}) \times P(\mathit{Yo} \mid \mathit{understand}) \times P(\mathit{no} \mid \mathit{I}) \times P(\mathit{comprendo} \mid \mathit{don't}) \\ &\quad + P(\mathit{Alignment4}) \times P(\mathit{Yo} \mid \mathit{I}) \times P(\mathit{no} \mid \mathit{understand}) \times P(\mathit{comprendo} \mid \mathit{don't}) \\ &\quad + P(\mathit{Alignment5}) \times P(\mathit{Yo} \mid \mathit{don't}) \times P(\mathit{no} \mid \mathit{understand}) \times P(\mathit{comprendo} \mid \mathit{I}) \\ &\quad + P(\mathit{Alignment6}) \times P(\mathit{Yo} \mid \mathit{understand}) \times P(\mathit{no} \mid \mathit{don't}) \times P(\mathit{comprendo} \mid \mathit{I}) \ . \end{aligned}$$

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

Estimation-maximization training is 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 and Arcturan. At a deeper level, estimation-maximization training tries to find the word-translation probabilities that 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 estimation maximization and neural networks without precisely understanding their convergence proofs. I give a brief description of estimation-maximization 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, but another sentence pair might have 10^{31} alignments, each with a 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. Thus, short (less ambiguous) sentences have more weight to throw around. Now we consider each English word in turn, for example, *understand.* It has weighted connections to many Spanish words, which we normalize to sum to one, giving 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 rescore 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 alignment 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 the others in each sentence pair. At this point, we stop, and we have our word-translation probabilities. Given a new sentence pair *S-E*, we can estimate $P(S \mid E)$ by using these probabilities. (See Ker and Chang [1997]; Smadja, McKeown, and Hatzivassiloglou [1996]; and Dagan and Church [1997] for further discussion of this and other methods for word and phrase alignment.)

Translation Method

That's it for decipherment. The last thing we need is a translation algorithm. I mentioned Bayes's 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 Es, but it 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, but more detailed discussions can be found in Wang and Waibel (1997), Wu (1996), and Tillmann et al. (1997). A translation method must also deal with unknown words, for example, names and technical terms. When languages use different alphabets and sound patterns, these terms must be translated phonetically (Knight and Graehl 1997).

Results

Initial results in statistical word-for-word machine translation were mixed. Computational limitations restricted experiments to short sentences and a 3000-word vocabulary. Although 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 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 machine-translation system called CANDIDE. This system performs as well as the commercial systems, with no hand-built knowledge bases! That's the good news. Where does word-for-word translation go from here? It is unclear whether the outstanding problems can be addressed within the word-for-word framework, using better statistical modeling or more training data. It is also unclear how this method would perform on language pairs such as Vietnamese-English, with radically different linguistic structure and less bilingual data online.

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 machine-translation approach, called *example-based machine translation* (Sato 1992; Nagao 1984). 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.

Knowing the syntactic structure of a source text—where phrase boundaries are and which phrases modify which—can be useful in translation.

Syntax-Based Translation

Knowing the syntactic structure of a source text—where phrase boundaries are and which phrases modify which—can be useful in translation. Most handcrafted commercial systems do a syntactic analysis followed by *transfer*, in which phrases are translated and reordered. There are many opportunities for empirical methods in such a framework. The most obvious is trainable parsing (Collins 1997; Hermjakob and Mooney 1997; Bod 1996; Charniak 1996; Magerman 1995). Unfortunately, such parsers often require a *tree bank* (a collection of manually parsed sentences), and tree banks are not yet available in most languages. Any advances in grammar induction from raw text will therefore have a big impact on machine translation. Some machine-translation systems use handcrafted 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 (Hatzivassiloglou and Knight 1995; Yamron et al. 1994). 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 (Matsumoto, Ishimoto, and Utsuro 1993).

A more ambitious, potentially powerful idea is to train directly on sentence pairs, learning both phrase structure and translation rules at the same time. Although a tree bank tells you a lot about phrase structure in a given language, translations can also tell you something, serving as a sort of poor man's tree bank. Research in this vein includes Wu (1995) and Alshawi, Buchsbaum, and Xia (1997). The basic idea is to replace the word-for-word scheme in which words fly around willy-nilly with a tighter syntax-based machine-translation model; probabilities are then still selected to best fit the sentence-pair corpus. Although 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 these alignments yield reasonable translations.

Semantics-Based Translation

Semantics-based machine translation has already produced high-quality translations in circumscribed domains. Its output is fluent because it uses meaning-to-text language generation instead of the gluing together of phrases 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 machine translation 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 can 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. Semiautomated knowledge acquisition plays an important role in creating large-scale resources such as conceptual models and lexicons (Viegas et al. 1996; Knight and Luk 1994). For the statistically oriented, Bayes's 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 can be a deep interlingua or a shallow case frame. Then, we can break translation down into two phases:

$$P(M \mid S) \sim P(M) \cdot P(S \mid M) \quad \text{Analysis}$$
$$P(E \mid M) \sim P(E) \cdot P(M \mid E) \quad \text{Generation} \quad .$$

$P(M)$ is essentially a world model. It should, for example, assign low probability to *fly*(*canyon*). $P(S \mid M)$ and $P(M \mid E)$ are like translation models we saw earlier. $P(E)$ 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, machine translation 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 such as 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 thief is *he*, and the stolen object is a *motorcar*).

This representation is bare bones. 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 the following 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 machine translation can realize its promise.

Evaluation

Evaluating machine translation is a tricky business. It's not like speech recognition, where you can count the number of wrong words. Two translations can equally be good without having a single word in common. Omitting a small word such as *the* might not be bad, but omitting a small word such as *not* might spell disaster.

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

First, 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).

Second, build a multiple-choice comprehension test based on 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.

Third, develop error categories (pronoun error, word-selection error, and so on), and divide them according to improvability and effect on intelligibility (Flanagan 1994). Tabulate errors in text.

These methods can be 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. Even if it does, it is no guarantee. I have a translator on my personal computer 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 in a sentence, and not-found words. Although this correlation won't let you compare systems, it will tell you whether a new document is suitable for machine translation.

There are also metrics for human-machine collaboration. Such collaboration usually takes the form of human preediting, machine translation, 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, machine-translation evaluation will continue to be an interesting topic and an active field in its own right, no matter what happens in machine translation proper. Second, formal machine-translation 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 machine-translation 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 e-mail machine translation, the choice is not between machine translation and human translation; it is between machine translation and no translation. Fourth, general-purpose machine translation is more accurate for closer language pairs (such as Spanish-English) than more distant ones (such as Japanese-English).

I see machine translation 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.

Conclusion

I have described several directions in empirical machine-translation research. As yet, there is 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, but 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 machine-translation knowledge from bilingual texts by applying more linguistically plausible models. I also expect to see knowledge being gleaned from monolingual (nonparallel) corpora, which exist in much larger quantities. Semantic dictionaries and world models, driven by AI applications mostly outside machine translation, will continue to scale up.

Will general-purpose machine-translation 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 machine-translation products. I see machine translation 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 machine translation. The main thing is to keep building and testing machine-translation systems, the essence of the empirical approach.

Note

[1] You will see that I devote more pages to word-for-word machine translation than to semantic machine translation. In part, it is to present the statistical word-for-word work a bit more simply and accessibly. Furthermore, word-for-word machine translation comprises a fairly self-contained set of techniques, but semantic machine translation benefits from the full range of corpus-based–language research, most of which I do not review.

References

Alshawi, H.; Buchsbaum, A.; and Xia, F. 1997. A Comparison of Head Transducers and Transfer for Limited-Domain Translation Applications. In Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics, 360–365. Somerset, N.J.: Association for Computational Linguistics.

Baum, L. E. 1972. An Inequality and Associated Maximization Technique in Statistical Estimation of

Probabilistic Functions of a Markov Process. *Inequalities* 3(1): 1–8.

Bod, R. 1996. Enriching Linguistics with Statistics: Performance Models of Natural Language. Ph.D. thesis, Department of Linguistics, University of Amsterdam.

Brown, P.; Lai, J.; and Mercer, R. 1991. Aligning Sentences in Parallel Corpora. In Proceedings of the Twenty-Ninth Annual Meeting of the Association for Computational Linguistics. Somerset, N.J.: Association for Computational Linguistics.

Brown, P.; Della Pietra, S.; Della Pietra, V.; and Mercer, R. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19(2): 263–312.

Brown, P.; Della Pietra, S.; Della Pietra, V.; Goldsmith, M.; Hajic, J.; Mercer, R.; and Mohanty, S. 1993. But Dictionaries Are Data Too. In Proceedings of the ARPA Human Language Technology Workshop, 202–205. Washington, D.C.: Advanced Research Projects Agency.

Brown, P.; Cocke, J.; Della Pietra, S.; Della Pietra, V.; Jelinek, F.; Lafferty, J.; Mercer, R.; and Roossin, P. S. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics* 16(2): 79–85.

Catizone, R.; Russell, G.; and Warwick, S. 1989. Deriving Translation Data from Bilingual Texts. Presented at the First International Lexical Acquisition Workshop, the Eleventh International Joint Conference on Artificial Intelligence, 19 August, Detroit, Michigan.

Chandioux, J., and Grimaila, A. 1996. Specialized Machine Translation. In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), 206–211. Washington, D.C.: Association for Machine Translation in the Americas.

Charniak, E. 1996. Tree-Bank Grammars. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, 1031–1036. Menlo Park, Calif.: American Association for Artificial Intelligence.

Chen, S. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. In Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, 310–318. Somerset, N.J.: Association for Computational Linguistics.

Chen, S. 1993. Aligning Sentences in Bilingual Corpora Using Lexical Information. In Proceedings of the Thirty-First Annual Meeting of the Association for Computational Linguistics, 9–16. Somerset, N.J.: Association for Computational Linguistics.

Church, K. 1993. CHARALIGN: A Program for Aligning Parallel Texts at the Character Level. In Proceedings of the Thirty-First Annual Meeting of the Association for Computational Linguistics, 1–8. Somerset, N.J.: Association for Computational Linguistics.

Collins, M. 1997. Three Generative, Lexicalized Models for Statistical Parsing. In Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics, 16–23. Somerset, N.J.: Association for Computational Linguistics.

Dagan, I., and Church, K. 1997. TERMIGHT: Coordinating Humans and Machines in Bilingual Terminology Acquisition. *Machine Translation* 12(1–2): 89–107.

Flanagan, M. 1994. Error Classification for Machine-Translation Evaluation. In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), 65–72. Washington, D.C.: Association for Machine Translation in the Americas.

Fung, P., and McKeown, K. 1994. Aligning Noisy Parallel Corpora across Language Groups: Word-Pair Feature Matching by Dynamic Time Warping. In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), 81–88. Washington, D.C.: Association for Machine Translation in the Americas.

Gale, W., and Church, K. 1991. A Program for Aligning Sentences in Bilingual Corpora. In Proceedings of the Twenty-Ninth Annual Meeting of the Association for Computational Linguistics, 177–184. Somerset, N.J.: Association for Computational Linguistics.

Gdaniec, C. 1994. The LOGOS Translatibility Index. In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), 97–105. Washington, D.C.: Association for Machine Translation in the Americas.

Hatzivassiloglou, V., and Knight, K. 1995. Unification-Based Glossing. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1382–1389. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Hermjakob, U., and Mooney, R. 1997. Learning Parse and Translation Decisions from Examples with Rich Context. In Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics, 482–489. Somerset, N.J.: Association for Computational Linguistics.

Isabelle, P.; Dymetman, M.; Foster, G.; Jutras, J.-M.; Macklovitch, E.; Perrault, F.; Ren, S.; and Simard, M. 1993. Translation Analysis and Translation Automation. Paper presented at the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation, Kyoto, Japan.

Kay, M., and Röscheisen, M. 1993. Text-Translation Alignment. *Computational Linguistics* 19(1): 121–145.

Ker, S., and Chang, J. 1997. A Class-Based Approach to Word Alignment. *Computational Linguistics* 23(2): 313–344.

Knight, K., and Graehl, J. 1997. Machine Transliteration. In Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics, 128–135. Somerset, N.J.: Association for Computational Linguistics.

Knight, K., and Hatzivassiloglou, V. 1995. Two-Level, Many-Paths Generation. In Proceedings of the Thirty-Third Annual Meeting of the Association for Computational Linguistics, 252–260. Somerset, N.J.: Association for Computational Linguistics.

Knight, K., and Luk, S. K. 1994. Building a Large-Scale Knowledge Base for Machine Translation. In Proceedings of the Twelfth National Conference on Artificial Intelligence, 773–778. Menlo Park, Calif.: American Association for Artificial Intelligence.

Knight, K.; Chander, I.; Haines, M.; Hatzivassiloglou, V.; Hovy, E.; Iida, M.; Luk, S.; Whitney, R.; and Yamada, K. 1995. Filling Knowledge Gaps in a Broad-Coverage Machine-Translation System. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1390–1397. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Lavie, A. 1994. An Integrated Heuristic Scheme for Partial Parse Evaluation. In Proceedings of the Thirty-Second Annual Meeting of the Association for Computational Linguistics (Student Session), 316–318. Somerset, N.J.: Association for Computational Linguistics.

Macklovitch, E. 1994. Using Bi-Textual Alignment for Translation Validation: The TRANSCHECK system. In Proceedings of the Conference of the Association for Machine Translation in the Americas, 157–168. Washington, D.C.: Association for Machine Translation in the Americas.

Macklovitch, E., and Hannan, M. L. 1996. Line ‘Em Up: Advances in Alignment Technology and Their Impact on Translation Support Tools. In Proceedings of the Conference of the Association for Machine Translation in the Americas, 145–156. Washington, D.C.: Association for Machine Translation in the Americas.

Magerman, D. 1995. Statistical Decision Tree Models for Parsing. In Proceedings of the Thirty-Third Annual Meeting of the Association for Computational Linguistics, 276–283. Somerset, N.J.: Association for Computational Linguistics.

Matsumoto, Y.; Ishimoto, H.; and Utsuro, T. 1993. Structural Matching of Parallel Texts. In Proceedings of the Thirty-First Annual Meeting of the Association for Computational Linguistics. Somerset, N.J.: Association for Computational Linguistics.

Melamed, I. D. 1997. A Portable Algorithm for Mapping Bitext Correspondence. In Proceedings of the Thirty-Fifth Conference of the Association for Computational Linguistics, 305–312. Somerset, N.J.: 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,* eds. A. Elithorn and R. Bernerji, 173–180. New York: North-Holland.

Nyberg, E., and Mitamura, T. 1992. The KANT System: Fast, Accurate, High-Quality Translation in Practical Domains. In Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING), 1069–1073. Nantes, France: GETA.

Sato, S. 1992. CTM: An Example-Based Translation Aid System. In Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING), 1259–1263. Nantes, France: GETA.

Simard, M., and Plamondon, P. 1996. Bilingual Sentence Alignment: Balancing Robustness and Accuracy. In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), 135–144. Washington, D.C.: Association for Machine Translation in the Americas.

Smadja, F., McKeown, K.; and Hatzivassiloglou, V. 1996. Translating Collocations for Bilingual Lexicons: A Statistical Approach. *Computational Linguistics* 22(1): 1–38.

Tillmann, C.; Vogel, S.; Ney, H.; and Zubiaga, A. 1997. A DP-Based Search Using Monotone Alignments in Statistical Translation. In Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics (ACL), 289–296. Somerset, N.J.: Association for Computational Linguistics.

Viegas, E.; Onyshkevych, B.; Raskin, V.; and Nirenburg, S. 1996. From Submit to Submitted via Submission: On Lexical Rules in Large-Scale Lexicon Acquisition. In Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, 32–39. Somerset, N.J.: Association for Computational Linguistics.

Wang, Y.-Y., and Waibel, A. 1997. Decoding Algorithm in Statistical Machine Translation. In Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics, 366–372. Somerset, N.J.: Association for Computational Linguistics.

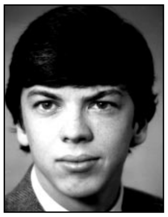
White, J., and O’Connell, T. 1994. Evaluation in the ARPA Machine-Translation Program: 1993 Methodology. In Proceedings of the ARPA Human Language Technology Workshop, 135–140. Washington, D.C.: Advanced Research Projects Agency.

Wu, D. 1996. A Polynomial-Time Algorithm for Statistical Machine Translation. In Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, 152–158. Somerset, N.J.: Association for Computational Linguistics.

Wu, D. 1995. Stochastic Inversion Transduction Grammars, with Application to Segmentation, Bracketing, and Alignment of Parallel Corpora. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1328–1335. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Yamada, K. 1996. A Controlled Skip Parser. In Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA), 14–23. Washington, D.C.: Association for Computational Linguistics.

Yamron, J.; Cant, J.; Demedts, A.; Dietzel, T.; and Ito, Y. 1994. The Automatic Component of the LINGSTAT Machine-Aided Translation System. In Proceedings of the ARPA Human-Language Technology Workshop, 163–168. Washington, D.C.: Advanced Research Projects Agency.



Kevin Knight is a senior research scientist at the University of Southern California (USC) Information Sciences Institute and a research assistant professor in computer science at USC. He received a B.A. from Harvard University and a Ph.D. from Carnegie Mellon University. Knight’s interests are in large-scale AI, knowledge acquisition, and natural language semantics. He cowrote (with Elaine Rich) the textbook *Artificial Intelligence* (McGraw-Hill, 1991).