CLAIRE Makes Machine Translation BLEU No More

by

Ali Mohammad

B.Sc. Mathematics and Physics, Kansas State University (2003)
B.Sc. Computer Engineering, Kansas State University (2003)
B.Sc. Computer Science, Kansas State University (2003)
S.M., Massachusetts Institute of Technology (2006)

Submitted to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctorate of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

Author .....
Department of Electrical Engineering and Computer Science
February 14, 2012

Certified by .....
Boris Katz
Principal Research Scientist in Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Chair of the Committee on Graduate Students
Department of Electrical Engineering and Computer Science

CLAIRE Makes Machine Translation BLEU No More

by

Ali Mohammad

Submitted to the Department of Electrical Engineering and Computer Science
on February 14, 2012, in partial fulfillment of the
requirements for the degree of
Doctorate of Science

Abstract

We introduce CLAIRE, a mathematically principled model for inferring ranks and scores for arbitrary items based on forced-choice binary comparisons, and show how to apply this technique to statistical models to take advantage of problem-specific assistance from non-experts. We apply this technique to two language processing problems: parsing and machine translation. This leads to an analysis which casts doubts on modern evaluation methods for machine translation systems, and an application of CLAIRE as a new technique for evaluating machine translation systems which is inexpensive, has theoretical guarantees, and correlates strongly in practice with more expensive human judgments of system quality. Our analysis reverses several major tenants of the mainstream machine translation research agenda, suggesting in particular that the use of linguistic models should be reexamined.

Thesis Supervisor: Boris Katz
Title: Principal Research Scientist in Computer Science

Acknowledgments

MIT is an amazing place, and I probably would have chosen to never graduate if that were an option. So many people were involved in making it amazing that it would take another thesis to thank them all.

Boris Katz was a wonderful advisor. He gave consistently good advice, but allowed me to make the mistakes I was determined to make. I grew as a scientist under his tutelage, and I am incredibly grateful for the guidance he gave me.

I would like to thank my committee members Bob Berwick and Tommi Jaakkola, who bore with my meandering progress with patience and gave nuanced comments on several drafts of this thesis, and mentored me for the many years that I was at MIT.

Thanks to my master's thesis advisor Michael Collins and his research group for giving me some excellent early guidance and an introduction to language processing.

I owe a great deal to Leslie Pack Kaelbling, Tomas Lozano-Perez, Denny Freeman, Eric Grimson, and the rest of the 6.01 crew for providing a tranquil oasis right when I really needed it.

I am deeply indebted to Andrew Correa, Yuan Shen, and Gabriel Zaccak who provided feedback, encouragement, and helpful discussions, especially as I was wrapping up. This thanks for discussions extends to anyone that was ever my office-mate or sat anywhere near me.

I have to thank the CSAIL Olympics (which was the AI Olympics when I first came

to MIT!) for introducing me to so many other students and getting me into student activities right when I started and led me into the CSAIL Student Committee.

Thanks to the good folks at Serious Business and danger!awesome who have given me a much needed creative/artistic outlet, and to Ron Wiken and the folks at TIG for tolerating my presence at the machine shop and for preventing me from slowly demolishing the lab.

Thanks to my work-out buddies who sometimes got me out of the building and kept me from getting too frustrated at lab.

Thanks to all my new friends at Google that showed me that there are amazing places to be outside of MIT.

Most of all, I am grateful to my family which suffered burdens beyond description with fortitude and dignity. My parents and my sisters have been profoundly supportive and patient in spite of everything from start to finish, and my extended family has always been there to encourage me. You are my inspiration.

وَقُل رَبِّ زِدنِي علمًا

And say, O My Lord! Increase me in Knowledge.

Contents

List of Figures

List of Tables

Introduction

I.1 Prologue

This thesis reflects an interest in the pursuit of machine translation in particular, although the methods and findings have bearing on the natural language processing enterprise in general.

We might define the natural language processing enterprise as comprising of methods engaged in *engineering systems that produce or consume natural language text in some way that corresponds to aspects inherent to that modality.*

Why the interest in this field? Putting aside the scientific interest an anthropologist might have in the "language phenomenon", there has been an absolute explosion in the amount of natural language text that is available.

Natural language is sought after as a desirable interface between man and machine more and more as we expect our devices to perform increasingly complex tasks.

Even before the information explosion, a globalization phenomenon has required us to communicate with others across a language barrier.

Even when communicating with others in the same language, we are each personally required to produce more prodigious quantities of text than ever before.

What are the greatest desires of the natural language practitioner? What is the most that can reasonably be expected? Let's for a moment imagine a world in which natural language were solved—it becomes easy to convert back and forth

between natural language and various logical forms. Precise queries could be executed to extract specific information from large bodies of natural language text. Large documents could be summarized and brief summaries could be expanded into large documents! Documents written in other languages could be translated accurately and quickly, preserving meaning, tone, meter, etc.

The possibilities are staggering and indeed have bewitched researchers since the early days of computation. Warren Weaver[47], an early cryptanalyst and cryptographer renowned for his work decrypting the ENIGMA code in World War 2, famously compared translation to codebreaking in the 1940's. Early natural language practitioners achieved small victories in the 1950's, and with exciting advancements in linguistics (particularly by Noam Chomsky), they promised the dreams outlined above to funding agencies in the US and abroad. Anyone familiar with the modern fruits of machine translation, taking into consideration the great advancements in the theory of computation, learning theory, linguistics, as well as the massive improvements in supporting infrastructure (microprocessors and datasets), would hardly be surprised by the demoralizing failures that were to come.

After pouring funding into one promising project after another for decades, the NSF, DOD, and CIA commissioned a report from the National Academy of Sciences to advise them on sensible next steps in the field. The National Academy of Sciences formed an Automatic Language Processing Advisory Committee (ALPAC) in 1964 to answer this need. ALPAC shared their findings in 1965 [42]. As a result of the report, support of machine translation projects was suspended in the United States for twenty years.

This report and its findings have a negative reputation in the statistical language community as an unjustified attack on a nascent field and which subsequently held the artificial intelligence community in general, and the natural language processing community in particular, back for decades.

ALPAC was chaired by John R. Pierce, a seasoned electrical engineer and physicist from Bell Labs who supervised the team that invented the transistor. ALPAC's

other members were the American psychologist John B. Carroll, known for his contributions to educational linguistics and psychometrics, then at Harvard; the American linguist Eric P. Hamp, who remains a professor of linguistics at the University of Chicago; David G. Hays of RAND, an important pioneer of computational and corpus linguistics (and the inventor of the term); American linguist Charles F. Hockett; American mathematician Anthony G. Oettinger, who is currently the chair of the Center for Information Policy Research at Harvard; and American computer scientist Alan Perlis, a pioneer in programming languages and the recipient of the first Turing Award. In short, the committee consisted of academic heavyweights of the highest caliber and included members who were interested in the continued support of fields relating to computation and linguistics.

The report is supported by extensive research, not only of the progress of the natural language processing enterprise itself, but also to establish the needs that the projects could reasonably be expected to satisfy.

The committee concluded that:

- Machine-aided translation tools (such as user-friendly technical dictionaries) improved both the speed of a translator's work and the quality of his output. This is an area of research worth supporting.

- The quality of automatic machine translation was too poor to be used without extensive post-editing by a translator, and this was more costly than simply having the translator translate from the source text directly. This result rested on extensive comprehension tests on scientific documents translated by then state-of-the-art systems.

- Furthermore, there was no prospect of improved translation systems under current research agendas.

- The only real, immediate need that could not be met by the available human translators is for rapid, accurate translation for a small circulation (too few to justify the cost of expedited human translation), and the research directions

that were being pursued were too superficial to achieve it. They went so far as to suggest that document typesetting was a bottleneck that could be more readily alleviated by development to improve the throughput of the conventional translation teams. Also, recurring needs of individual researchers for translations from a particular language could be alleviated with basic language training for that researcher in the language in question.

- Research had resulted in a number of useful results in related fields, including stimulating energetic research in corpus linguistics. This research should be supported with funding as a science with expectations of long-term improvement to automatic systems.

- There was an immediate need for improved evaluation methodologies.

The funding agencies responded by following the recommendations to eliminate funding to fully automatic systems, but did not follow recommendations to provide funding to supporting fields.

Perhaps the funding agencies deserve some derision for not taking the long view on translation. At the same time, it is clear that the popular view at the time the funding proposals were written was that translation would be solved with minimal government expenditure and the benefits were manifest; this was certainly reflected in the report. Series of failed projects could not continue to receive funding; if administrators in the funding agencies felt inclined to be sympathetic, they would certainly have been replaced by more skeptical colleagues at this point.

Nearly twenty years after the ALPAC report, the British computational linguist and computer scientist Martin Kay wrote a brief statement [27] expressing his view on the future of automatic machine translation. The statement is brief enough to include here in its entirety:

> Large expenditures on fundamental scientific research are usually limited to the hard sciences. It is therefore entirely reasonable to suppose that, if

large sums of money are spent on machine translations, it will be with the clear expectation that what is being purchased is principally development and engineering, and that the result will contribute substantially to the solution of some pressing problem.

Anyone who accepts large (or small) sums on this understanding is either technically naive or dangerously cynical. It may certainly be that

1. machine translation could provide a valuable framework for fundamental research;
2. texts in highly restricted subsets of natural language could be devised for particular purposes and texts in [*sic*] translated automatically;
3. computers have an important role to fill in making translations;
4. translations of extremely low quality may be acceptible [*sic*] on occasions.

However,

1. the fundamental research is so far from applicability,
2. the language subsets are so restricted,
3. the useful computer technologies are so different from machine translation,
4. the quality of the translations that can be produced of natural texts by automatic means is so low, and
5. the occasions on which those translations could be useful are so rare,

that the use of the term in these cases can only result in confusion if not deception.

> A determined attempt was made to bring machine translation to the point of usability in the sixties. It has become fashionable to deride these as "first generation" systems and to refer to what is being done now as belonging to the second or third generation. It should surely be possible for those who think that the newer systems can succeed where the earlier ones failed, to point to problems that have been solved since the sixties that are so crucial as substantially to change our assessment of what can be achieved. We know a good deal more about programming techniques and have larger machines to work with; we have more elegant theories of syntax and what modern linguists are pleased to call semantics; and there has been some exploratory work on anaphora. But, we still have little idea how to translate into a closely related language like French or German, English sentences containing such words as "he", "she", "it", "not", "and", and "of". Furthermore, such work as has been done on these problems has been studiously ignored by all those currently involved in developing systems.
>
> Unfortunately, the sums that are being spent on MT in Europe and Japan are large enough to make virtually inevitable the production of a second ALPAC report sometime in the next few years. This will inevitably have a devastating effect on the whole field of computational linguistics, everywhere in the world. The report will be the more devastating for the fact that much of the money has in fact been spent frivolously, and much of the work has been incompetent, even by today's limited standards.

Fortunately for us, Kay's predictions on funding have not come true. The needs that ALPAC reported have grown by leaps and bounds, and machine translation systems have found use in niche applications. For example, the Canadian weather service has famously used machine translation (with a human post-editing phase) to translate weather bulletins from English to French for 30 years (including using the same system, METEO, for two decades 1981–2001) [45]. In 2009, President

Obama released a "Strategy for American Innovation" which named "automatic, highly accurate and real-time translation between the major languages of the world" an ambitious goal that will "improve our quality of life and establish the foundation for the industries and jobs of the future." [36]

How can we distinguish our work from the pseudo-science that ALPAC and Kay described? It is difficult to declare natural language processing to be a science since our goal is not to learn about an existing system, but is instead to build useful systems of our own. I would argue that it is still possible to do science in this arena, but that it requires care to understand the limitations of our results. It's difficult to make broad statements about the value of a particular approach when so much is still unknown about language in general and considering how far state-of-the-art systems fall short of the dream. It is not at all inconceivable that the best research systems extant would bear little resemblance to a future system that fulfills the promises of our predecessors.

In spite of this uncertainty, there is a definitive mainstream thrust of research in the natural language community: **Quantity Leads to Quality**. State-of-the-art performance is more surely and readily achieved by an appeal to a massive dataset than an appeal to linguistic theory. An extension of this is that improvements in quality are achieved by more advanced statistical models that are capable of modelling more exotic relationships between input and output, with careful regularization to account for the sparsity of data.

Indeed, researchers entering natural language processing are often warned of the consequences of expecting and promising too much, and veteran researchers are careful to establish metrics that can illustrate sustained, gradual improvement and to point out the immediate uses of the state-of-the-art translation systems, notwithstanding their manifestly poor quality. Church and Hovy go as far as stating that it may be more important to improve performance by seeking an appropriate application than by improving the system in question [15].

I.2 Background

I.2.1 Parsing

Our baseline parsing model is Collins Model 2 parser. We describe it here briefly; please see [16] for more details.

Just as a statistical speech recognition system aims to maximize $\Pr(\mathbf{e} \mid \mathbf{a})$, given a sentence S, the statistical parser aims to find the parse tree T that maximizes $\Pr(T \mid S)$. The Collins parsers are generative models based on the probabilistic context free grammar formalism (PCFG), except that they are lexicalized. An unlexicalized PCFG would be broken down as follows:

$$\begin{aligned} \arg\max_T \Pr(T \mid S) &= \arg\max_T \Pr(T, S) \\ &= \arg\max_T \prod_i \Pr(RHS_i \mid LHS_i), \end{aligned}$$

where RHS_i and LHS_i denote the left- and right-hand sides of the context-free grammar rule that is used at the ith step in the derivation of the tree; the probabilities of the rules make up the parameters of the model, and maximum likelihood estimates are easy to obtain from a corpus of trees.

Collins lexicalizes the PCFG by adding every possible word and part-of-speech to each non-terminal in the grammar, greatly increasing the number of parameters. The maximum-likelihood estimates can be obtained as before (technically, this is still a PCFG), but data sparsity quickly becomes an issue, so the generative story is broken down further. Each LHS now generates a head[1], and the head generates *subcategorization frames* for the left and the right (just a set of required complements on each side of the head phrase); each terminal of the rule to the left and right of the head is then generated depending only on the head and the constraint that the final rule math the subcategorization frame. Model 2 also adds a flag which indicates

[1]The head itself is not easy to define and is the subject of some debate. Collins provides hand-designed head rules in his thesis which seem to work well for the Wall Street Journal. In [4], Bikel provides similar rules for use in other languages.

whether each nonterminal is a complement or an adjunct.

I.2.2 Machine Translation

Let us state the machine translation problem: our goal is to translate French[2] sentences to English[3] sentences. For simplicity, we will consider sentences in isolation, ignoring the impact of context. We will always denote sentences in the source language by $\mathbf{f}$ and sentences in the target language by $\mathbf{e}$. m is the number of words in the source sentence $\mathbf{f}$ and ℓ is the number of words in the target sentence $\mathbf{e}$. Now, in Bayesian terms, given a French sentence $\mathbf{f}$, we wish to find the English sentence $\mathbf{e}$ that maximizes $\Pr(\mathbf{e} \mid \mathbf{f})$, in effect imagining that French sentences are generated by some unknown transformation on English sentences. Using Bayes' Law, we write:

$$\begin{aligned} \arg\max_{\mathbf{e}} \Pr(\mathbf{e} \mid \mathbf{f}) &= \arg\max_{\mathbf{e}} \frac{\Pr(\mathbf{f} \mid \mathbf{e}) \Pr(\mathbf{e})}{\Pr(\mathbf{f})} \\ &= \arg\max_{\mathbf{e}} \Pr(\mathbf{f} \mid \mathbf{e}) \Pr(\mathbf{e}) . \end{aligned}$$

The $\Pr(\mathbf{f})$ term can be ignored since $\mathbf{f}$ is constant. The first term, $\Pr(\mathbf{f} \mid \mathbf{e})$, is called the "translation model" and the second, $\Pr(\mathbf{e})$, is called the "language model." It is just good to know that our translation model does not have to worry about assigning low probabilities to English sentences that look like they could be translations of $\mathbf{f}$ but don't really look like they could be English sentences; a good language model can make up for some deficiencies in the translation model. One other advantage of factoring the model in this way is that the language model can be trained on very large unlabeled (i.e., untranslated) data sets in the target language.

In this work, we use a trigram language model; that is, we assume that the procedure that produces English sentences is a Markov process with a history of two

[2]or in the general case, Foreign. The original papers translated from French to English, and it has become a tradition.

[3]or in the general case, Native.

words: the probability of an English sentence $\mathbf{e}$ is broken down like this:

$$\Pr(\mathbf{e}) = \Pr(e_1, e_2, \cdots, e_\ell) = \prod_{i=1}^{\ell+1} \Pr(e_i \mid e_{i-1}, e_{i-2}),$$

where $e_{\ell+1}$ is a stop symbol implicitly included at the end of every English sentence. This is a raw trigram model. To learn the parameters, we can simply count the number of times each trigram appears in a corpus:

$$\Pr(e \mid e_2, e_1) = \frac{\text{count}(e_1, e_2, e)}{\text{count}(e_1, e_2)}$$

where $\text{count}(\cdots)$ denotes the number of times the words $\cdots$ appear together in the corpus in the given order. One problem with this model is that it will assign a zero probability to any sentence that has a trigram that was never seen in the corpus. To fix this, one uses a *smoothed* trigram model [22]:

$$\Pr(e \mid e_2, e_1) = \alpha_t \Pr_t(e \mid e_2, e_1) + \alpha_b \Pr_b(e \mid e_2) + \alpha_m \Pr_m(e)$$

where $\alpha_t + \alpha_b + \alpha_m = 1$, the αs are nonnegative, and $\Pr_t(\cdot)$, $\Pr_b(\cdot)$, and $\Pr_m(\cdot)$ denote trigram, bigram, and unigram probabilities, respectively. This is called an *affine* combination of the three distributions.

I.2.3 Five IBM Models

In a seminal 1993 paper, Brown *et al* introduced a set of five machine translation systems based on fairly simple statistical models (under the "noisy-channel" framework described above) and large parallel corpora. The later models are significantly more complex than the earlier models, with each subsequent model corresponding to an increase in complexity and improved accuracy (but with diminishing returns in the later models) [7].

We assume that we have at our disposal a corpus of N pairs of sentences $(\mathbf{e}^{(1)}, \mathbf{f}^{(1)})$,

$(\mathbf{e}^{(2)}, \mathbf{f}^{(2)})$, ..., $(\mathbf{e}^{(N)}, \mathbf{f}^{(N)})$.

In this document, our analysis will not extend beyond the first and second IBM Models, so we will limit our discussion of the later models to a brief overview.

Model 1

We begin by describing an idea fundamental to both Model 1 and Model 2: an *alignment* between a pair of sentences $\mathbf{f}$ and $\mathbf{e}$ is an ordered set $a_1, a_2, \cdots, a_m \in \{0, 1, \ldots, \ell\}$. The French word f_j is said to be *aligned* to the English word e_{a_j} (where e_0 denotes a fake word "NULL" that is used to explain function words that may not have an obvious analog in the English sentence). Notice that we don't demand that English words are aligned to French words; a single English word could be used to explain an entire French sentence (a developed model would declare such an alignment as very improbable, however).

Model 1 makes the following assumptions/approximations:

- All alignments are equally likely.
- All French sentence lengths m are equally likely (we will ignore the obvious problem that there are infinitely many French sentence lengths[4]; if it bothers you, you can assume that someone gives you the length or that there are only finitely many possible French sentence lengths, which is true in practice, anyway). We will generally omit this term.
- Each word is translated independently of the other words.

These assumptions are outrageously simplifying, but it is important to start with a tractable model. It is also important to remember that our language model will clean up output problems: we can expect short-range alignment problems and, to some extent, poor grammar to be dealt with there.

[4]There really are. As proof, I present a regular expression that matches an infinite number of grammatical French sentences: *Je suis un très* grand singe.* Obvious analogs exist in other languages.

Ultimately, we obtain the following formulation[5]:

$$\begin{aligned}
\Pr(\mathbf{f} \mid \mathbf{e}) &= \frac{1}{(\ell+1)^m} \sum_{\mathbf{a}} \prod_{j=1}^{m} \Pr(f_j \mid e_{a_j}) \\
&= \frac{1}{(\ell+1)^m} \sum_{a_1=0}^{\ell} \sum_{a_2=0}^{\ell} \cdots \sum_{a_m=0}^{\ell} \prod_{j=1}^{m} \Pr(f_j \mid e_{a_j}) \\
&= \frac{1}{(\ell+1)^m} \prod_{j=1}^{m} \sum_{a_j=0}^{\ell} \Pr(f_j \mid e_{a_j}) .
\end{aligned}$$

Here, a_j is the index of the English word that is *aligned* to the jth French word; Model 1's parameters are the translation probabilities $\Pr(f \mid e)$ (the probability that the French word f was generated by the English word e). Model 1 is an excellent candidate for optimization by EM; it is convex and has only one local maximum (outside of saddle points due to symmetry) so given a random starting point, it will always converge to the same, optimum translation table. Some more math gives us the following update rules:

$$T'(f \mid e) = \frac{1}{Z_T(e)} \sum_{\substack{i,j,k \\ \mathbf{e}_i^{(k)}=e, \mathbf{f}_j^{(k)}=f}} \frac{T(\mathbf{f}_j^{(k)} \mid \mathbf{e}_i^{(k)})}{\sum_{i'=0}^{\ell} T(\mathbf{f}_j^{(k)}, \mathbf{e}_{i'}^{(k)})},$$

where Z_T is a normalization constant.

Model 2

In Model 2, we wish to relax Model 1's assumption that all alignments are equally likely. However, we will assume for simplicity that the words all "move" independently; that is, which English word a French word is aligned to is independent of the

[5]The reversal of the product and sum is an important trick, since it makes it easy to optimize the terms independently.

alignment of the remaining words. Here is the formulation of Model 2:

$$\begin{aligned}\Pr(\mathbf{f} \mid \mathbf{e}) &= \sum_{\mathbf{a}} \prod_{j=1}^{m} \Pr\left(f_j \mid e_{a_j}\right) \Pr\left(a_j \mid j, \ell, m\right) \\ &= \prod_{j=1}^{m} \sum_{a_j=0}^{\ell} \Pr\left(f_j \mid e_{a_j}\right) \Pr\left(a_j \mid j, \ell, m\right).\end{aligned}$$

Here, in addition to the translation probabilities $\Pr(f \mid e)$ Model 2 inherits from Model 1, we find alignment probabilities, $\Pr(a_j \mid j, \ell, m)$ (the probability that, given a French sentence of length m that is the translation of an English sentence of length ℓ, the jth French word was generated by the a_jth English word). Model 2 is not as good a candidate for EM as Model 1 was; it is riddled with saddle points and local maxima. Typically, one initializes the translation parameters by training Model 1 before training Model 2, whence we use the following update rules:

$$T'(f \mid e) = \frac{1}{Z_T(e)} \sum_{\substack{i,j,k \\ \mathbf{e}_i^{(k)}=e,\, \mathbf{f}_j^{(k)}=f}} \frac{D(a_j = i \mid j, \ell, m) T(\mathbf{f}_j^{(k)} \mid \mathbf{e}_i^{(k)})}{\sum_{i'=0}^{\ell} D(a_j = i' \mid j, \ell, m) T(\mathbf{f}_j^{(k)}, \mathbf{e}_{i'}^{(k)})}$$

$$D'(a_j = i \mid j, \ell, m) = \frac{1}{Z_D(j, \ell, m)} \sum_{\substack{k \\ |\mathbf{e}^{(k)}|=\ell,\, |\mathbf{f}^{(k)}|=m}} \frac{D(a_j = i \mid j, \ell, m) T(\mathbf{f}_j^{(k)} \mid \mathbf{e}_i^{(k)})}{\sum_{i'=0}^{\ell} D(a_j = i' \mid j, \ell, m) T(\mathbf{f}_j^{(k)}, \mathbf{e}_{i'}^{(k)})},$$

where Z_T and Z_D are normalization constants.

The pseudocode to train IBM Models 1 and 2 is given in figure I-1.

Models 3, 4, and 5

IBM Model 3 introduces fertility parameters, modeling the number of French words a single English word generates. IBM Model 4 introduces distortion parameters to the alignment models as a way of encouraging words to move in groups. Both of these models are formulated "deficiently": that is, they assign probability mass to impossible French sentences (four-word French sentences without a third word, for instance); Model 5 is the non-deficient version of Model 4. Since this makes little

Initialize $t(f \mid e)$ and $D(i \mid j, \ell, m)$

do:

- zero $t'(f \mid e)$ and $D'(i \mid j, \ell, m)$
- **for** $(\mathbf{e}, \mathbf{f})$ **in** corpus:
  - $m = |\mathbf{f}|$, $\ell = |\mathbf{e}|$
  - **for** j=1...m:
    - **for** i=0...ℓ:
      - $a_i = t(\mathbf{f}_j \mid \mathbf{e}_i) \cdot D(i \mid j, \ell, m)$
    - $a_i = a_i / (\sum_{i'} a_{i'})$
    - **for** i=0...ℓ:
      - $t'(\mathbf{f}_j \mid \mathbf{e}_i) = t'(\mathbf{f}_j \mid \mathbf{e}_i) + a_i$
      - $D'(i \mid j, \ell, m) = D'(i \mid j, \ell, m) + a_i$
- $t'(f \mid e) = t'(f \mid e) \Big/ \left(\sum_{f'} t'(f' \mid e) \right)$
- $D'(i \mid j, \ell, m) = D'(i \mid j, \ell, m) \Big/ \left(\sum_{i'} D'(i' \mid j, \ell, m) \right)$ (Set to 1 for Model 1)
- $t = t', D = D'$

until convergence

Figure I-1: IBM Models 1 and 2 pseudocode.

empirical difference and is a great computational burden, Model 5 is rarely used in practice [38].

I.2.4 Phrase-Based Models

The primary unit of information in all of the systems we have described up to this point is the word; in phrase-based systems, the primary unit of information is the phrase, a collection of (lexically) consecutive words and the lexical entry in a phrase-based system is a triple containing a source phrase $f_1 \ldots f_n$, a target phrase $e_1 \cdots e_m$, and a score $s \in [0, 1]$. That is, instead of considering probabilites of word-to-word translations and word-movement, a phrase-based system will deal with probabilities of phrase-to-phrase translations and phrase-movement. There is a great deal of evidence to suggest that machine translation systems generally experience a performance boost by making this change.

Some phrase-based models, such as Marcu's [33], simply introduce mechanisms for phrase-to-phrase translations and invent policies to assign probability mass to phrase-to-phrase translations. Others, such as Köhn's [31], build a dictionary of phrases from other information sources. Our experiments are centered on the Köhn system, as it achieves state-of-the-art performance.

There are a number of ways one can build phrase dictionaries depending on the data available. Phrases can be built from word-based alignments (such as those generated by the IBM Models). If syntactic information is available, it can be used to restrict our attention to syntactic phrases; although it seems that syntactic phrases may be more useful, experiment suggests that phrases that are not syntactically motivated are, generally, just as useful[6]. Furthermore, even weighting syntactic phrases produces virtually no improvement at best and is sometimes harmful. Phrase dictionaries can also be built from phrase-aligned data generated from phrase-based systems. Again, we can place more confidence in these phrases if we wish, but gen-

[6]For example, the German phrase "es gibt" corresponds nicely to the English phrase "there is", even though they cross a syntax boundary in both languages.

erally, the lesson from the experiments with syntactic phrases is applicable: it is better in practice simply to consider as many phrases as possible than to restrict our knowledge to satisfy any bias we may have [29].

Experiments by Köhn *et al* show that simple heuristic methods based on word-based alignments from the IBM models generate state-of-the-art translations. To generate a phrase dictionary, he begins by observing that the IBM models are not symmetric; the alignments generated by a model trained to translate from French to English can be different from alignments generated by a model trained to translate from English to French (in fact, it is often impossible for alignments generated in one direction to match those generated in the other direction, due to inherent restrictions of the IBM models). Köhn's method begins by considering the intersection of the two alignments as a starting point for the phrases it must generate; that is, it begins by suggesting that words that are aligned in both models are probably related. Next, Köhn uses a growing technique to induce the phrase dictionary; phrase dictionaries generated in this fashion tend to be very large because of the generality of this technique; however, the method naturally also generates scores and generates many very low-scoring phrase pairs.

Decoding is done using an algorithm described in [24]. The output sentence is generated left-to-right in the form of partial translations which are formed using a beam-search algorithm including scores for the phrase translations and for the language model, rewards for increased sentence length, and penalties for phrase reordering. Typical states are depicted in Figure I-2.

The two factors that govern the quality of translations generated using this technique are the quantity and quality of the alignments that are given to the system during training. In practice, translation quality from this method is significantly better than any of the IBM Models. Surprisingly, it does almost as well with IBM Model 2 alignments as with IBM Model 4 alignments.

Translation Hypothesis: $e_1\ e_2$
Source: $f_1\ f_2\ \cancel{f_3}\ \cancel{f_4}\ f_5\ f_6$
Score: S

$$\Downarrow$$

Translation Hypothesis: $e_1\ e_2\ e_3\ e_4$
Source: $f_1\ f_2\ \cancel{f_3}\ \cancel{f_4}\ \cancel{f_5}\ \cancel{f_6}$
Score: $S + [\mathrm{Score}(e_3, e_4; f_5, f_6) + \log \Pr(e_3 \mid e_1, e_2)$
$\quad + \log \Pr(e_4 \mid e_2, e_3) + \mathrm{Distortion}(5\text{–}6,\ 3\text{–}4)]$

Figure I-2: Typical states in the Köhn decoder. In the original state, depicted above, the decoder has hypothesised that the French phrase $f_3\ f_4$ corresponds with the English phrase $e_1\ e_2$ with score S. This state is updated by adding that the phrase $f_5\ f_6$ corresponds to the English phrase $e_3\ e_4$ with score terms corresponding to this assignment ($\mathrm{Score}(e_3, e_4; f_5, f_6)$), log likelihood terms corresponding to the language model, and a score for the phrase reordering.

I.3 Answering Kay

It is now, armed with a metric and a modern system design substantially different from those that Kay described in his statement [27] above, that we can respond to his demands to point to the lessons that have been learned and which summarize the mainstream agenda of the field.

1. Noisy-channel methods, like those described above, are vastly superior to rule-based systems, and require substantially less effort to design.

2. Linguistic formalisms can be incorporated into statistical translation models, but the effort required to do so is substantial, and the effect on the scores does not justify the effort.

3. n-gram language models are easy to train and tough to beat.

This second point is now a byword in the field; the following quote, by Fred Jelinek, famously conveys this:

> Every time I fire a linguist, the performance goes up!

We will review these and other lessons in Chapter 3.

I.4 Outline

Chapter 1 presents details on our work in using human assistance for parsing, with an eye toward but without application to machine translation. Chapter 2 presents our work on word-sense disambiguation for machine translation using source-language human assistance. Chapter 3 revisits Kay's question, presents a critique of automatic metrics, and presents CLAIRE, our alternative. We close with conclusions and a few useful appendices.

Chapter 1

Human Assisted Parsing

What makes language processing tasks difficult? Early practitioners believed that better-than-human performance was just around the corner for many problems that remain enticingly unsolved, despite decades of active inquiry.

Perhaps the difficulties can be summarized as two broad issues: linguistic formalism and language ambiguity.

The former corresponds roughly to choice of representation, which is indeed a major concern. We do not wish to restrict ourselves to a synthetic or controlled language; we wish to permit all of the ambiguity and flexibility of open-domain natural language. Logical forms are too fragile and disconnected from the text, and traditional parse trees do not supply much of the information needed to make confident use of the text, and yet often contain a great deal of information that corresponds to linguistic hair-splitting and not to any valuable distinction in emphasis, structure, or meaning. This problem corresponds to a major share of the scope of the field of linguistics, so it comes as no surprise that we find it challenging.

However, in fully automatic systems, genuine ambiguity is such a great concern that the linguistic formalism is often regarded as a minor issue with little impact on performance. This is exemplified by the fact that the techniques that are known to consistently improve the quality of modern systems are those that reduce ambiguity

in the problem (most notably, constraining the domain of the text in question).

Consider the classic example:

I saw the man with the telescope.

There are several ambiguities in this relatively short, simple sentence, including:

- the intended sense of the word *saw* (either past tense of the verb *to see* or present tense of the verb *to saw*);
- the attachment of the prepositional phrase *with the telescope*, either to the verb *saw* or the object *the man*.
- the many possible meanings of the preposition *with* (which may be reduced given a particular attachment choice):
  - I used the telescope to see the man.
  - I saw the man when I saw the telescope.
  - I saw the man adjacent to the telescope.
  - I saw the man who owns a telescope.

and so on. Clearly there is ambiguity at many levels (lexical, syntactic, semantic), resulting in a combinatorial explosion in the number of readings of a sentence. In spite of this, language users generally have such little difficulty winnowing them that the ambiguities are often not even evident without explanation. Most English speakers would never consider the possibility that the author of this sentence wished to convey that he was using a telescope to cut a man in two, yet it is not difficult to imagine a context (perhaps a crime novel) where this *would* be the intended meaning of the sentence. In any case, two of the readings (that he used the telescope to see the man and that the man had the telescope when he saw him) are entirely plausible, and, without context, no system should entirely eliminate either of these readings.

Our goal is a system that enables a non-expert user to perform expert parsing with performance that exceeds that of fully automatic systems. We propose a transformation-based approach: users input a sentence for processing, and candidate readings of the sentence are transformed and presented to the user for correction.

Why would such a system be valuable? Some processing tasks currently must be performed by experts because the quality of fully automatic systems is still too poor for many uses. Researchers have developed tools to assist the experts in tasks such as computer assisted translation, but expert labeling is still prohibitively expensive. On the other hand, non-experts are in bountiful supply. Furthermore, as long as the performance of human-assisted systems surpasses that of fully automatic systems, their outputs could be used to augment existing corpora.

It may even be argued that, in lieu of strong artificial intelligence, fully-automatic systems will never be able to resolve world-knowledge or common-sense ambiguities. Certainly, without some sort of reasoning subsystem, it is inconceivable that we will be able to properly respond to context, even if we were to solve syntax and reverse-engineer the language center. Without the mind's superior pattern-matching machinery and world model, language understanding would (probably) still be unsolved.

What difficulties might we face? First of all, an improvement in quality may require prohibitively great effort on the part of the user. This is particularly true of a system that cannot generate a reasonable partial understanding automatically, in which case it may be difficult to form meaningful questions.

In spite of these possibilities, there may yet be hope. Ambiguities that cannot be expressed with transformation (i.e., by *rewriting the sentence*) do not exist for all practical purposes, so an ideal system should be able to communicate any difficulty it has. Anyone is qualified to respond: virtually every human being has a first-class world-model and pattern-recognizer waiting to be used. The labels he gives for one sentence, or even for part of one sentence, may be useful for subsequent ambiguities, so we can hope that the number of questions we have to ask will increase more slowly

than the amount of information we are interested in (that is, we have to know a great deal to ask an intelligent question, but much of that knowledge will be helpful for the following question). Finally, we can hope that the number of questions we need to ask overall will gradually diminish as we collect enough data to better estimate the answers.

1.1 Previous Work

Many recent papers have targeted specific attachment types directly with automatic methods, by reranking or self-training. These have achieved significant gains, but the authors are quick to note that these gains are specific to the corpus being used (i.e., the Wall Street Journal corpus), and clearly do not represent gains in broad-domain text.

Disambiguating by transformation is not a radical thing; the Treebank parsing guidelines include criteria for especially tricky examples by syntactically transforming difficult sentences.[3]

However, perhaps the earliest paper to suggest disambiguation by asking in any formal context is Kay's description of the MIND system [26], a multilingual translation engine. Later proposals for the use of monolingual consultants in translation are described in Chapter 2.

In parsing, Carter [13] presents TreeBanker, a system similar to an earlier system by Tomita for translation [46], but targeted at expert consultants and with an eye toward corpus-building.

1.2 Our Approach

To reiterate, our goal is to improve quality of parsing using transformations, with an eye toward applications that would make use of argument structure.

Our approach differs primarily from the previous approaches in three respects: first, we incorporate interaction in a statistical (not rule-based) parser; second, our mode of interaction with the user is by way of transformations of the original text using a probabilistic tree transducer; third, instead of trying to design tests for ambiguity patterns, we propose using linguistically meaningful transformations *without* an eye toward particular ambiguity classes. The advantage of favoring statistical techniques is that they are less labor-intensive and less fragile than their rule-based counterparts, at the cost of being mechanistically opaque to system designers. Considering that we are expecting human attention, using probabilistic techniques confers another serious advantage: the system can be designed to continuously adapt its behavior to new data.

Ideally, the machine learning practitioner would have sentences in some canonical form, along with corresponding versions authoritatively transformed in some way. In lieu of such a corpus, we are consigned to manual design methods. We therefore produced a list of transformation types and designed the corresponding transduction rules. We envision that statistics would play its part by incorporating failure probabilities and in lexicalizing the rules.

Machine learning concepts also motivate how we question the user. We envision a system in which the user may choose to stop answering questions at any time, so we assume every question we ask may be the last. Consequently, we will always ask the question that gives us the most information in expectation. The amount of information a question and answer pair gives us is given by the change in the entropy of the belief distribution, which is based on probabilities given by our baseline parser.

1.2.1 Collins Parser

Our baseline parser is the Model 2 Collins parser (described briefly in the introduction and in full in [16]) modified to recover the full Penn extended tag-set ([20] and Appendix B). This parser is fully automatic and is a standard baseline, because of

its relative simplicity and high performance. It recovers labeled constituents with ~88% precision/recall. Certain recurring attachment decisions are highly ambiguous because constituents can feasibly be attached to many previous points in a sentence; consequently, these correspond to a substantial portion of the error that Collins reports. For instance, in the sentence:

She announced a program to promote safety in trucks and vans.

the prepositional phrase *in trucks and vans* can attach to several points in the sentence:

- Safety in trucks and vans is what the program promotes.
- The program promotes safety, and it does so in trucks and vans.
- The announcement was made in trucks and vans.

and the conjunct *and vans* can attach to several points in the sentence as well:

- Safety in trucks and vans is what the program promotes.
- The program promotes safety and also promotes vans.
- She announced a program and she announced vans.

So it comes as no surprise that the scores for these attachment types are lower: prepositional phrase attachments are recovered with ~82% precision/recall, and coordinating conjunction attachments with ~62% precision/recall.

1.2.2 Algorithm

We make use of any available syntactic transformations that are meaningful linguistically (ones we can expect a non-expert consultant to understand). We select which transformation to use based on our belief, which is initially just the probability scores

given by the parser, and which is updated by any information we obtain from the user(s).

Our intended algorithm, then, is as follows:

Given a sentence s, obtain top N output from the Collins parser
Transform this sentence using the available rules for each candidate parse
Group the outputs of the transformation by the word span of the constituents
do

- For each collection of word spans, compute the entropy of the partition
- Ask the user to label the outputs corresponding to the maximum entropy
- Adjust the scores of the parses

until the user quits or we run out of questions
return the highest scoring parse

Now we approach the issue of the transformations themselves.

1.2.3 Tree Transduction Grammar

For the purposes of this chapter, we will only be transforming clauses from active to passive voice or clefting sentences. However, since we expect to use other transformations in later work, and since changing voice in particular is a fairly complex procedure, we will sketch the formal machinery and attach our grammars for both transformations (see the end of chapter).

A transducer is a finite state automaton with two tapes. We use a transducer that operates on the internal nodes of a tree, and call this machine a tree transducer. The match or input portion of the transducer amounts to little more than executing an exhaustive regular expression matcher at each level of the tree for each rule, and then making the results of the match available to parent nodes for recursive matching. The output portion is slightly more complicated: since changing voice can involve long-range movement of constituents, output rules have to be able to "pass" trees as

Original Sentence	John saw the man with the telescope.
Passive	The man with the telescope was seen by John. The man was seen with the telescope by John.
Cleft	It is John that saw the man with the telescope. It is the man with the telescope that John saw. It is the man that John saw with the telescope. It is with the telescope that John saw the man.

Table 1.1: Examples of voice-changing and clefting.

arguments to children outputs. To leave this stage as flexible as possible, the output rules are simply lambda expressions. Stylized outputs from the active-to-passive rule and the cleft rule are given in table 1.2.3, illustrating how they may be used to disambiguate prepositional phrase attachments.

Once again, the input to the transducer is a *parse forest*—the dynamic programming table that is generated when the parser analyzes a sentence—and the output is a collection of all of the possible outputs of the original sentence and the amount of information we would gain by knowing whether or not each output is correct (e.g., if it is semantically equivalent).

We define a tree transduction grammar to be a collection of tree transduction rules, and we define a tree transduction rule to be a triple consisting of a label, an input rule, and an output rule. Input rules map a particular node in a candidate tree to a set (possibly empty) of "matches" (generally, input rules will correspond to regular expressions) and output rules will map a match to an output for that node. A particular node label is designated to be the root of the grammar; the output of the transducer is the set all of the possible outputs labeled with the root of the grammar. The input rules can require that a child match another input rule in the grammar, so the transduction grammar is a graph as well. We have constructed two tree transduction grammars: one to change sentences from active to passive voice,

and another to cleft a sentence's arguments.

The transducer does the work of applying every rule to every node in the parse forest without repeating work; in general, neither the transduction grammar nor the parse forest will be acyclic (a parse tree containing a "cycle" is possible because the Collins grammar permits unary productions; such a tree will never be the maximum likelihood tree, however).

An early version of the transducer operated on single parse trees and was applied to top N output of the Collins parser, but the transducer was modified after our early experiments to operate on parse forests (i.e., on the parser's dynamic programming chart) instead. This posed a significant technical challenge, but ultimately (somewhat surprisingly) resulted in a much more efficient system (since there are generally only minor differences between the top N trees, much of the work the early system was doing was repeated). The modifications to the code are described in appendix C, and we have made an implementation of the transducer (in python) available online for future work.

The transducer is a very powerful tool, but because it's so flexible, it's difficult to give performance guarantees. In practice, even for the types of grammars that we gave, the performance is decent; on a modestly powered computer, it can produce outputs for a 40-word sentence in about two seconds, comparable to the time it takes to parse the sentence.

The grammars for the active-to-passive and cleft transformations are given at the end of the chapter.

For each output that is generated by the system, we can compute the probability that the nodes that produced the output would appear in the correct parse tree. If we were completely confident that each grammar produced perfect output, we simply choose the output that partitions the space of possibilities most evenly. This does not change if we don't completely trust the grammars, as long as we trust them all equally, which we do for simplicity.

FOREST	:=	List of NODEs	
NODE	:=	label	
		headLabel	
		headWord	
		spanStart, spanEnd	(sentence span)
		List of CHILDRENs	
CHILDREN	:=	score	
		List of NODEs	
MATCH	:=	content	
		NODE	
		CHILDREN	
		spanStart, spanEnd	(subspan of the CHILDREN, for partial matches)
TGRAMMAR	:=	List of TRULEs	
		rootLabel	
TRULE	:=	label	
		INPUTRULE	
		OUTPUTRULE	
INPUTRULE	:=	RERULE	
	|	NAMEDRULE	
	|	NOTRULE	
	|	KLEENERULE	
	|	ANDRULE	
	|	ORRULE	
	|	CONCATRULE	
	|	mapping from (NODE, CHILDREN) $\mapsto$ MATCH	
RERULE	:=	regularExpression	denoted by `"<regexp>"`
NAMEDRULE	:=	label	denoted by `#<label>`
NOTRULE	:=	INPUTRULE	denoted by `!<rule>`
KLEENERULE	:=	INPUTRULE	denoted by `<rule>*`
ANDRULE	:=	List of INPUTRULEs	denoted by `<rule> & <rule> & ... & <rule>`
ORRULE	:=	List of INPUTRULEs	denoted by `<rule> | <rule> | ... | <rule>`
CONCATRULE	:=	List of INPUTRULEs	denoted by `<rule> @ <rule> @ ... @ <rule>`
OUTPUTRULE	:=	mapping which accepts MATCH	

Figure 1-1: Types used in the Transducer

```
toExamine = {Leaf Rules} × {(Nodes, Children)}
toExamine′ = {}
partialMatches = empty hash with default value of {}
matches = empty hash with default value of {}
while |toExamine| > 0:
·   for each rule, node, children in toExamine:
·   ·   n = |children|
·   ·   preMatchCount = |matches[rule, node, children]|
·   ·                       + |partialMatches[rule, node, children]|
·   ·   for each subspan of children:
·   ·   ·   update partialMatches[rule, node, children, subspan]
·   ·   update matches[rule, node, children, subspan]
·   ·   postMatchCount = |matches[rule, node, children]|
·   ·                       + |partialMatches[rule, node, children]|
·   ·   if preMatchCount ≥ postMatchCount:
·   ·   ·   continue
·   ·   for each parent rule pRule of rule:
·   ·   ·   for each parent node pNode of node:
·   ·   ·   ·   for each pChildren in of pNode.children:
·   ·   ·   ·   ·   if node ∈ pChildren:
·   ·   ·   ·   ·   ·   toExamine′.add((pRule, pNode, pChildren))
·   toExamine = toExamine′
·   toExamine′ = {}
return matches
```

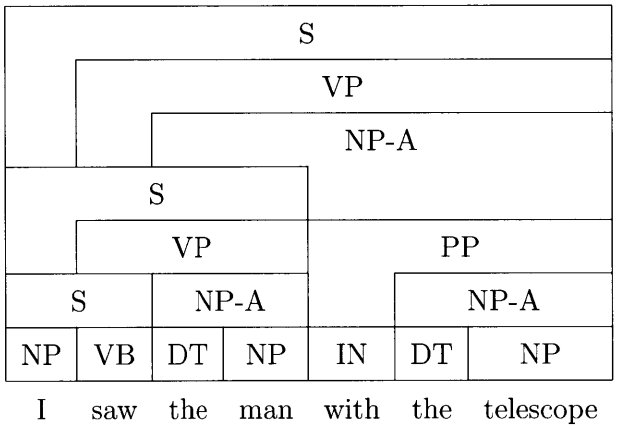
Figure 1-2: Transducer Pseudocode

1.2.4 Sample Output

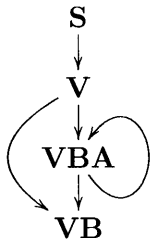
We will briefly run our sample sentence "I saw the man with the telescope" through the active-to-passive transducer to make its operation more clear. Let us suppose that our parser produced the following set of possible subtrees:

{(NP, 0–1), (VB, 1–2), (DT, 2–3), (NP, 3–4), (IN, 4–5), (DT, 5–6), (NP, 6–7), (S, 0–2), (NP-A, 2–4), (NP-A, 5–7), (VP, 1–4), (PP, 4–7), (S, 0–4), (NP-A, 2–7), (VP, 1–7), (S, 0–7)}

This corresponds to the following chart:



The active-to-passive transduction grammar (given at the end of this chapter) has the following dependency structure:



We begin by applying the input rule of the leaves of this dependency structure on the nodes in the parse forest in reverse tree-order (so that a node will only be processed after the children are processed). The algorithm beings by setting the

variable toExamine accordingly (we exclude the children of each node):

$$\text{toExamine} \leftarrow \{(\mathbf{VB}, (\text{NP}, 0\text{–}1)), (\mathbf{VB}, (\text{VB}, 1\text{–}2)), ..., (\mathbf{VB}, (\text{S}, 0\text{–}7))\}$$

After the first iteration of the transducer, **VB** will match the node (VB, 1–2), so the parents of **VB** must be examined:

$$\begin{aligned}\text{toExamine} \leftarrow \{&(\mathbf{V}, (\text{S}, 0\text{–}2)), (\mathbf{V}, (\text{VP}, 1\text{–}4)), (\mathbf{V}, (\text{VP}, 1\text{–}7)),\\ &(\mathbf{VBA}, (\text{S}, 0\text{–}2)), (\mathbf{VBA}, (\text{VP}, 1\text{–}4)), (\mathbf{VBA}, (\text{VP}, 1\text{–}7))\}\end{aligned}$$

After the second iteration of the transducer, **V** will match the nodes (VP, 1–4) and (VP, 1–7) with outputs († $\mapsto$ was seen by †, † $\mapsto$ the man) and († $\mapsto$ was seen by †, † $\mapsto$ the man with the telescope), respectively. Now the parents of **V** must be examined (this time, we include the children):

$$\begin{aligned}\text{toExamine} \leftarrow \{&(\mathbf{S}, \text{S} \rightarrow \text{NP}(0\text{–}1)\ \text{VP}(1\text{–}7)),\\ &(\mathbf{S}, \text{S} \rightarrow \text{NP}(0\text{–}1)\ \text{VP}(1\text{–}4)\ \text{PP}(4\text{–}7))\}\end{aligned}$$

After the third and final iteration of the transducer, **S** matches both nodes and yields the following outputs:

The man was seen by me with the telescope.

The man with the telescope was seen by me.

1.3 Further Augmenting the Tree

The trees in the Penn treebank have a great deal of information beyond flat sentences in the Wall Street Journal; they include basic part-of-speech information at the word level, constituent dependency information, and information about the argument structure. This information is so rich that the vanilla Collins parser does

not attempt to recover it. However, there is an absence of many syntactic features on internal nodes even on these trees. This is true even of agreement features, such as number and gender, which we find ourselves in need of to confidently write new sentences with the noun phrases in the corpus. For a moment, let us attempt to recover these features given the information that is present in the corpus.

Why are these features absent in the first place? Perhaps it is imagined that number, for example, can be reliably extracted deterministically, particularly considering that number is given at the leaves. An algorithm that immediately comes to mind is to simply use the head of a tree to recover these properties at each internal node.

The first difficulty we encounter with the "head" approach is that there is some disagreement about what exactly the head should be! Conjunction phrases, for example, are notorious sticky points: should the head of the first conjoined phrase be the head of the whole phrase? Perhaps the conjunction itself should be the head? When we consider these options with our problem, the answer becomes even less clear: if we form a phrase from two singular phrases using *and*, certainly the entire phrase should not be singular. Yet, we have some well-founded discomfort with the notion that *and* is plural. (Clearly there is a problem with the representation that is used in the treebank; this is addressed in [19].)

We would run into even more trouble with other languages, like Arabic, where verbs are conjugated based on gender as well. If we merge (via *and*) a masculine noun phrase and a feminine noun phrase, the resulting noun phrase is considered masculine; however, even if we get over our discomfort at assigning *and* noun-like features, there is no correct answer here: no fixed decision on constituent head assignment works for any combination of masculine and feminine subphrases. Other conjunctions, like *or*, are even more difficult; it is not clear whether a phrase like *a man or the boys* should be singular or plural, so we should not be surprised that a general rule is elusive.

Even if we adopt some set of rules, say the hand-written head rules used to parametrize the Collins parser, we run into difficulty with fairly common phrases like *a number of men*, where *number*, a singular noun, is the head. Perhaps the rea-

son that such rules would fail is that number is really a semantic thing with some syntactic manifestations, and not simply an arbitrary syntactic property. If English speakers were to begin using a new word, say *joople*, they may assume that it refers to some singular thing based on its ending and would conjugate verbs accordingly; however, if they were told that *joople* referred to a plural thing, they would at the very least consider treating it as such (analogous phenomena were described in [32]).

Furthermore, when we are considering two such features simultaneously, it is not clear that both properties for an entire noun phrase would even come from a single head. These difficulties pushed the NLP community to use statistical methods in the first place.

1.3.1 Method

Statistical parsers are built using hand annotated parse trees, such as the treebank. Unfortunately, the treebank does not label entire noun phrases as singular or plural, without which we cannot build a supervised model.

However, the treebank does have rich argument labels, so we can extract subject-verb pairs from sentences in the treebank. Since the verbs and their subjects must agree in number in English, wherever we can identify the number of the verb, we can identify the number its subject. Hence, the same agreement property of the language that motivated this problem in the first place is the source of the data we will use to solve it. This method can be used to recover any syntactic property that is needed for agreement in any language. We apply our approach to number in English and gender in Arabic.

1.3.2 Data Collection

In both treebanks, we make use of the extended set of internal node tags; in particular, the `NP-SBJ` tag helps us handle sentence constructions that do not have ordinary agreement, such as expletive and cleft constructions. In English, we search for the

first verb leaf of the predicate; if it is labeled VBZ, we take the subject to be singular; if it is labeled VBP, we take the subject to be plural; otherwise, we discard the subject. Note that this heuristic will work for sentences with auxilliary verbs as well, since these verbs carry the number agreement, and its verbal arguments do not carry number (e.g., *John (and Bob) eats (eat) the apple* become *John (and Bob) has (have) eaten/been eating/etc. the apple*).

In Arabic, circumstances are slightly more complicated. Formally, Arabic has strict gender agreement. However, there are two major word orders in Arabic with differing number agreement rules. In particular, sentences with the VSO (Verb Subject Object) ordering (جُمْلَةٌ فِعْلِيَّةٌ or verbal sentences) have gender agreement between verb and subject, but do not have number agreement (the verb is usually conjugated for the singular). On the other hand, sentences with the SVO order (جُمْلَةٌ إِسْمِيَّةٌ or nominal sentences) have agreement in both gender and number. In Classical Arabic, the VSO order is dominant; however, in Modern Standard Arabic (MSA) the SVO order is quite common.[1]

In Arabic, we collect sentences with the same criterion as in English and search for the verb with the same heuristic (the modal verb certainly must agree with the subject; in Arabic, the verbal arguments will also agree, but this information is not used in our experiments). Also, since Arabic is morphologically rich, the leaf tags take on a more "factored" appearance; on the other hand, the internal nodes are more impoverished (there are only 17 distinct internal node labels, compared with 240 distinct internal node labels for English, but there are 284 distinct leaf labels in Arabic, compared with only 43 for English). The verb tags that are taken as singular/dual/plural and masculine/feminine are given in table 1.4.

sentence type	verb gender	verb number
VSO جُمْلَةٌ فِعْلِيَّةٌ	agreement	singular
SVO جُمْلَةٌ إِسْمِيَّةٌ	agreement	agreement

[1]It may seem surprising that Arabic would preserve any verb-subject number agreement considering that the classically dominant order lacks this type of agreement; however, Arabic is pro-drop and sentences without explicit subjects have subject-verb agreement in both gender and number.

	VSO — (Verbal Sentence / جُمْلَةٌ فِعْلِيَّةٌ)					
	Masculine / مُذَكَّر			Feminine / مُؤَنَّث		
Singular / مُفْرَد	أَكَلَ	اَلوَلَدُ	اَلتُّفَّاحَةَ	أَكَلَتْ	اَلبِنتُ	اَلتُّفَّاحَةَ
	ate(MS)	*the boy*	*the apple*	*ate*(FS)	*the girl*	*the apple*
Dual / مُثَنَّى	أَكَلَ	اَلوَلَدَانِ	اَلتُّفَّاحَةَ	أَكَلَتْ	اَلبِنتَانِ	اَلتُّفَّاحَةَ
	ate(MS)	*the two boys*	*the apple*	*ate*(FS)	*the two girls*	*the apple*
Plural / جَمع	أَكَلَ	اَلأَولَادُ	اَلتُّفَّاحَةَ	أَكَلَتْ	اَلبَنَاتُ	اَلتُّفَّاحَةَ
	ate(MS)	*the boys*	*the apple*	*ate*(FS)	*the girls*	*the apple*
	SVO — (Nominal Sentence / جُمْلَةٌ إِسْمِيَّةٌ)					
	Masculine / مُذَكَّر			Feminine / مُؤَنَّث		
Singular / مُفْرَد	اَلوَلَدُ	أَكَلَ	اَلتُّفَّاحَةَ	اَلبِنتُ	أَكَلَتْ	اَلتُّفَّاحَةَ
	the boy	*ate*(MS)	*the apple*	*the girl*	*ate*(FS)	*the apple*
Dual / مُثَنَّى	اَلوَلَدَانِ	أَكَلَا	اَلتُّفَّاحَةَ	اَلبِنتَانِ	أَكَلَتَا	اَلتُّفَّاحَةَ
	the two boys	*ate*(MD)	*the apple*	*the two girls*	*ate*(FD)	*the apple*
Plural / جَمع	اَلأَولَادُ	أَكَلُوا	اَلتُّفَّاحَةَ	اَلبَنَاتُ	أَكَلْنَ	اَلتُّفَّاحَةَ
	the boys	*ate*(MP)	*the apple*	*the girls*	*ate*(MP)	*the apple*

Table 1.2: To distinguish between the different conjugated forms in Arabic, we append labels to the English verb in the gloss; M and F denote masculine and feminine, respectively, and S, D, and P denote singular, dual, and plural, respectively.

1.3.3 The Model

Our setting is a standard supervised classification task: the input is a noun phrase tree and the output is the number/gender of the noun phrase. Our model is that the number/gender is propagated upward from the leaves to the root according to feature-specific head rules; the parameters of our model dictate the probability at each stage that the feature-head is a particular child. For each noun phrase, we are given the feature value for the root (by virtue of subject-verb agreement) and for all of the leaves (which are hand labeled). We treat the labels of the remaining nodes as hidden variables and use expectation-maximization (EM) to optimize the model.

With some notation, we can make this more explicit: let T denote a parse tree, T_ϕ the root node of T, T_i the ith subtree of the root, and T_* the root rule (i.e., $T_\phi \rightarrow T_{1\phi}\ T_{2\phi}\ \cdots\ T_{N\phi}$). Our goal is to model the propagation of some feature F; we do so as follows:

$$\Pr\left(F(T_\phi) = f \mid T\right) = \sum_i \Pr\left(F(T_{i\phi}) = f \mid T_i\right) \Pr\left(F\text{-head} = i \mid T_*\right)$$

Hence, the parameters of the model are the probabilities $\Pr\left(F\text{-head} = i \mid T_*\right)$ for each rule in the grammar. The distributions of the feature-values for the leaf nodes is fixed

	Masculine	Feminine
Singular	PVSUFF_SUBJ:3MS	PVSUFF_SUBJ:3FS
	PVSUFF_SUBJ:2MS	PVSUFF_SUBJ:2FS
	IV3MS	IV3FS
	IV2MS	IV2FS
	CVSUFF_SUBJ:2MS	CVSUFF_SUBJ:2FS
Dual	PVSUFF_SUBJ:3MD	PVSUFF_SUBJ:3FD
	IV3MD	IV3FD
Plural	PVSUFF_SUBJ:3MP	PVSUFF_SUBJ:3FP
	PVSUFF_SUBJ:2MP	PVSUFF_SUBJ:2FP
	IV3MP	IV3FP
	IV2MP	IV2FP
	CVSUFF_SUBJ:2MP	CVSUFF_SUBJ:2FP

Table 1.3: The agreement tags of conjugated verbs in the Arabic treebank. PV, IV, and CV are perfect, imperfect, and imperative (command) aspects, respectively.

for each language and is given in table 3.

We optimize the parameters of the model using EM; the resulting algorithm is given in table 1.5.

After the model has been optimized, we can evaluate new trees; in our algorithm, in case of a tie, we back-off to a simple majority model. That is, for example, if singular noun phrases are more common and the probability that a given noun phrase is plural according to our model is exactly 0.5, we will return singular.

We compare our method to a simple majority model in Arabic and to the one suggested in the introduction (drawing number from the head of the noun phrase) in English, using the head rules given in Collins 1999. We trained our models on a random selection of 80% of the noun phrases obtained from each corpus; we ran the algorithm for 10 iterations (we found that increasing the number of iterations had little effect on the performance).

English	
Singular	NN, NNP
Plural	NNPS, NNPS

	Arabic	
	Masculine	Feminine
Singular	CVSUFF_SUBJ:2MS DEM_PRON_MS IVSUFF_DO:3MS PRON_2MS PRON_3MS PVSUFF_DO:3MS IV1S PRON_1S PVSUFF_DO:1S	NSUFF_FEM_SG CVSUFF_SUBJ:2FS DEM_PRON_FS IVSUFF_DO:3FS PRON_3FS DEM_PRON_F IV1S PRON_1S PVSUFF_DO:1S PVSUFF_DO:3FS
Dual	NSUFF_MASC_DU_ACC NSUFF_MASC_DU_ACC_POSS NSUFF_MASC_DU_GEN NSUFF_MASC_DU_GEN_POSS NSUFF_MASC_DU_NOM NSUFF_MASC_DU_NOM_POSS CVSUFF_SUBJ:2MP DEM_PRON_MP IV1P PRON_1P PRON_2MP IV2MP IVSUFF_DO:3MP PRON_3D PVSUFF_DO:1P PVSUFF_DO:3D	NSUFF_FEM_DU_ACC NSUFF_FEM_DU_ACC_POSS NSUFF_FEM_DU_GEN NSUFF_FEM_DU_GEN_POSS NSUFF_FEM_DU_NOM NSUFF_FEM_DU_NOM_POSS DEM_PRON_FD IV1P PRON_1P DEM_PRON_F PRON_3D PVSUFF_DO:1P PVSUFF_DO:3D
Plural	NSUFF_MASC_PL_ACC NSUFF_MASC_PL_ACC_POSS NSUFF_MASC_PL_GEN NSUFF_MASC_PL_GEN_POSS NSUFF_MASC_PL_NOM NSUFF_MASC_PL_NOM_POSS PRON_3MP CVSUFF_SUBJ:2MP DEM_PRON_MP IV1P PRON_1P PRON_2MP IV2MP IVSUFF_DO:3MP PVSUFF_DO:1P PVSUFF_DO:3MP	NSUFF_FEM_PL IV1P PRON_1P DEM_PRON_F PVSUFF_DO:1P

Table 1.4: Leaf feature values in Arabic and English. Tags not given here are evenly distributed between all possible labels. Tags placed in multiple categories are evenly distributed between those categories. Also, note that we have included verbal categories for Arabic; this is in anticipation of future work with statistically inducing these features on verb phrases.

Given:

- ℓ possible feature values
- An array of N trees, T
- An array of feature values, F_T, for the roots
- An array of feature values, f, for the leaves

Initialize Θ*:*

- **foreach** rule T_*:
  - $n \leftarrow$ the number of children in rule T_*
  - $\Theta[T_*] \leftarrow n$ random values from $[0, 1]$
  - normalize $\Theta[T_*]$

do

- *E-Step*:
  - **foreach** tree $T[i]$:
    - estep($T[i]$, Θ)
    - $T[i].F \leftarrow F_T[i]$
- *M-Step*:
  - zero Θ
  - **foreach** tree $T[i]$:
    - mstep($T[i]$, Θ)
  - normalize Θ

until convergence

function estep(tree τ, parameters Θ)

- **if** τ is a leaf:
  - $\tau.F \leftarrow f(\tau.\text{tag})$
- **else**:
  - $p \leftarrow$ array of ℓ zeros
  - **foreach** subtree τ_i:
    - estep(τ_i, Θ)
    - $p \leftarrow p + \tau_{i\phi}.F * \Theta[\tau_*][i]$
  - $\tau.F \leftarrow p$

function mstep(tree τ, parameters Θ)

- **if** τ is not a leaf:
  - **foreach** subtree τ_i:
    - mstep(τ_i, Θ)
    - $\Theta[\tau_*][i] \leftarrow \Theta[\tau_*][i] + \tau.F \cdot \tau_i.F$

Table 1.5: The model optimization algorithm.

1.4 Results

Let's start by looking at the results of the tree-augmenter. We performed experiments on the English and Arabic Penn Treebank corpora. For English, we start with 35,752 sentences with a total of 910K tokens which yielded 28K subject noun phrases for which numbers could be extracted. For Arabic, we start with 12,412 sentences with a total of 430K tokens which yielded 24K subject noun phrases for which gender could be extracted. Unfortunately, the SVO order is relatively rare in the Arabic corpus; only 30 noun phrases could be marked for number. Hence the results for Arabic number were not statistically significant.

Language	Feature	Baseline	Head	EM
Arabic	Gender	66%	–	**69%**
English	Number	61%	75%	**82%**

We ran our transformation experiments on the standard evaluation section (section 0) of the Penn Treebank II (the Wall Street Journal corpus). We ran the modified Collins parser on this data. We ran the active-to-passive and cleft tree-transducers on this data to produce questionnaires for each sentence; 14% of the sentences processed produced multiple outputs; another 4% produced a single output. We gave the questionnaires for 300 randomly selected sentences to two users for labeling. For each user, the scores for the parses were adjusted for the labels (parses corresponding to images that were marked as incorrect were simply penalized by 10 log probability points). The resulting top-scoring parses were compared to the gold-standard parses using the evalb metric described in [16]; the results are as follows:

Baseline (Collins Parser)	0.871	
Collins + Oracle	0.893	+17.0%
Collins + User 1	0.877	+4.7%
Collins + User 2	0.874	+2.2%
Collins + Users 1 and 2	0.875	+2.4%

User 2 reported that he labelled the data very quickly and may have made some mistakes, possibly explaining his diminished performance. Below are the agreement statistics between the two users.

		User 1		
		Correct	Incorrect	Total
User 2	Correct	65%	5%	70%
	Incorrect	12%	18%	30%
Total		77%	23%	100%

We expect that the reason for the high (~20%) false image rate is due to the rough implementation of the transformations. Despite this, the method shows significant improvement to a baseline parser with relatively little effort on the part of the user (one or two questions per sentence). We expect that better statistics would be achieved once the transformation was improved to incorporate lexical information. The labels from both users correspond to a significant improvement in ambiguous, semantically meaningful attachment types (particularly PP).

1.5 Future Work

The first and most obvious step is to further refine the transduction grammar by hand. Improvements could be made simply by modifying the parser to output traces. Several transformations can be added to help other attachment problems, and also to attempt anaphora resolution and other language understanding tasks.

Naturally, more data would improve our performance and allow us to advance our model. In particular, an attractive next step is to obtain enough data to lexicalize the rule scores, so that we can estimate when a rule will fail. An even more interesting idea (and therefore one which requires yet more data) would be to predict correlated failures; that is, we may find that the sentence will break when one constituent is placed before the other for some linguistic reason (e.g., one refers to the other), so all transformations that rearrange these constituents will fail.

Beyond a specialized corpus for this task, we are interested in using this system to develop treebanks in new domains and languages and to refine existing treebanks, as in [13].

In lieu of more labeled data, we can make use of unlabeled (i.e., parsed, but not annotated) data to estimate the lexicalized rule scores via self-training. For instance, if we find that a verb does not appear in passive voice in our data set, we can confidently hypothesize that it cannot be passivized.

The Transduction Grammars

Bold-faced capital letters denote internal nodes of the transduction grammar, normal capital letters denote internal nodes of the parse trees (often accompanied with regular expression markers), the asterisk denotes any number of children of any type, subscripts are indices for output rules that return multiple items (tuples), † and ‡ when present denote the arguments to output rules, and *c* and *pp* are auxiliary functions which output the copula and the passive participle, respectively. Patterns that appear in the output rules are intended to refer to the region in the input that was matched; regions matched by the asterisks are intended to be replaced in the order they appeared in the input.

S	$\Leftarrow$	S(-A)?	$\rightarrow$	* NPB?-A * **V** *
	$\Rightarrow$	"	$\rightarrow$	* $\mathbf{V}_2$(NPB?-A) * $\mathbf{V}_1$(NPB?-A) *
V	$\Leftarrow$	VP	$\rightarrow$	**VB** * **VBA** *
	$\Rightarrow$		$\rightarrow_1$ (†)	**VB** * $\mathbf{VBA}_1$(†,**VB**) *
			$\rightarrow_2$ (†)	$\mathbf{VBA}_2$(†,**VB**)
V	$\Leftarrow$	VP	$\rightarrow$	**VB** * NPB?-A *
	$\Rightarrow$		$\rightarrow_1$ (†)	c(†) pp(**VB**) * by † *
			$\rightarrow_2$ (†)	NPB?-A
VBA	$\Leftarrow$	VP-A	$\rightarrow$	VB * **VBA** *
	$\Rightarrow$		$\rightarrow_1$ (†, ‡)	VB * $\mathbf{VBA}_1$(†,VB) *
			$\rightarrow_2$ (†, ‡)	$\mathbf{VBA}_2$(†,VB)
VBA	$\Leftarrow$	VP-A	$\rightarrow$	**VB** * NPB?-A *
	$\Rightarrow$		$\rightarrow_1$ (†, ‡)	c(‡,NPB?-A) p(VB) * by † *
			$\rightarrow_2$ (†, ‡)	NPB?-A
VB	$\Leftarrow$	VB[^N]	$\rightarrow$	*not a form of to be or to have*
	$\Rightarrow$		$\rightarrow$	itself

Table 1.6: The active-to-passive transduction grammar

CS	$\Leftarrow$	S(-A)?	$\rightarrow$	(CC|INTJ)* (**PREARG**)* **CSUB** * **CVB** *
	$\Rightarrow$		$\rightarrow$	It c(**CVB**) **CSUB** that * **CVB** *
CS	$\Leftarrow$	S(-A)?	$\rightarrow$	(CC|INTJ)* (**PREARG**)* **CSUB** * **CVBARG** *
	$\Rightarrow$		$\rightarrow$	It c(**CVBARG**) **CSUB** that * **CVBARG** *
CVBARG	$\Leftarrow$	VP	$\rightarrow$	* **CSUB** *
	$\Rightarrow$		$\rightarrow_1$	**CSUB**
			$\rightarrow_2$	* *
CVBARG	$\Leftarrow$	VP	$\rightarrow$	* **CVBARG** *
	$\Rightarrow$		$\rightarrow_1$	$\mathbf{CVBARG}_1$
			$\rightarrow_2$	* $\mathbf{CVBARG}_2$ *
CVB	$\Leftarrow$	VP	$\rightarrow$	*head is not a form of to, said, add, contend*
	$\Rightarrow$		$\rightarrow$	itself
PREARG	$\Leftarrow$	!(S.*|CC|INTJ)	$\rightarrow$	*
	$\Rightarrow$		$\rightarrow$	itself
CSUB	$\Leftarrow$	[^VCMSI].*	$\rightarrow$	*
	$\Rightarrow$		$\rightarrow$	itself

Table 1.7: The cleft transduction grammar

Chapter 2

Human Assisted Word-Sense Disambiguation for Translation

The goal of word-sense disambiguation systems is to identify the meaning of a word from context. Word-sense disambiguation, like parsing, is considered a fundamental problem in natural language processing, which is broadly applicable and, like parsing, word-sense disambiguation systems are rarely used outside of natural language research labs. In spite of this, word-sense disambiguation remains an active area of research.

Formally, for a given word or phrase in context, word-sense disambiguators are asked to identify which *synset* the word or phrase belongs to. The difficulty depends on the granularity of distinctions that are being made; in a recent evaluation which used synsets from WordNet, interannotator agreement on word senses was as low as 85%: this is a practical upper bound on an automatic systems performance [1].

Shallow methods yield surprisingly good results on word-sense disambiguation; the simplest method that achieves state-of-the-art performance is one which incorporates a handful of shallow methods by voting.

One of the most-cited potential uses of word-sense disambiguation is automatic machine translation. In [11], Carpuat and Wu incorporated a state-of-the-art Chinese

word sense disambiguation model into a state-of-the-art statistical machine translation system and found that their system performed worse than an uninformed system! This was (to the best of our knowledge) the only negative result published in ACL that year, which suggests just how unexpected the result is. Indeed, several later papers (notably [12] and [14]) seek to reverse it, with some success. Nevertheless, the fact that the original paper found that it did not help and that the later papers (despite careful design surrounding the inclusion of the system) found limited improvement is suggestive.

We will modify a current state-of-the-art system to obtain word-senses from a monolingual consultant instead of from a word-sense disambiguation system to more directly evaluate whether eliminating this type of ambiguity improves translation. Notably, our system is the first that uses source-language consultants to improve a statistical translation system.

2.1 Previous Work

The idea of disambiguating by asking is an old one; perhaps the earliest paper to suggest disambiguation by asking is Kay's description of the MIND system [26], a multilingual translation engine. Kay points out that use of the term "fully automatic" when describing translation systems is misleading, because users of fully automatic translation systems will edit the output if they are familiar with the target language. He therefore suggests the use of a monolingual consultant to resolve ambiguities in the source language, but he does not propose algorithms for producing or processing the interaction, nor does he indicate what form the interaction should take.

Tomita [46] later refines the problem definition by disallowing assumptions about the consultant—in particular, that the consultant has any specialized background (in linguistics or computer science)—so the interaction cannot include parse trees or phrase structure rules. Tomita presents a rule-based system for parsing by augmenting a context-free grammar with annotation rules to explain each attachment and solicit

correction. For the sentence:

Mary saw a man in the park with the telescope.

Tomita's system generates the following questions:

1) (a man) is (in the park)
2) The action (Mary saw a man) takes place (in the park)

1) (the park) is (with a telescope)
2) (a man) is (with a telescope)
3) The action (Mary saw a man) takes place (with a telescope)

Ben-Ari *et al* propose embedding a similarly designed parsing engine in a rule-based translation system [2], with an emphasis on transfer (i.e., preferring to preserve ambiguity in translation whenever the source and target languages make that possible). They also point out that more sophisticated queries than those of Tomita may be used for certain types of ambiguity.

Maruyama [35] presents a system for ambiguity resolution in a Japanese-to-English translation system by interactively displaying dependencies. When the user selects a phrase (underlined), the system displays the phrases that the chosen phrase might modify (the automatic choice is in reverse video; all other choices are highlighted in red). In the screenshots below (adapted from [35]), the user first selects the first phrase, then the second phrase.

A recent paper [10] presents an elegant variation of the target language rewriting technique: display a chart of possible phrasal translations and permit the user to select a path through the phrase chart, but which reports negative results in spite of

the human assistance.

2.2 Methods

`moses`[30] is a free (both *gratis* and *libre*) state-of-the-art statistical translation system. The `moses` decoder can be asked to produce the set of all possible outputs (in its search space) for a given input as a directed acyclic graph. Each edge corresponds to a phrase in the output, and each path from the start node to an end node corresponds to a candidate translation.

Formally, `moses` generates a collection of nodes $\mathcal{V}$ and edges $\mathcal{E}$. Each node represents a possible state of the decoder (as described in the introduction), and each edge consists of a source phrase that is removed from the source, a target phrase that is generated and added to the candidate translation, and a score that is added to the total score at that point. Each path from the empty start node to a final node (one for which the entire source sentence has been absorbed) corresponds to a candidate translation.

$$\mathcal{V} \subseteq \bigcup_{\ell \in \mathbb{N} \cup \{0\}} E^{\ell} \times \mathcal{P}(\{1, ..., m\}) \times [0, 1]$$

where E denotes the set of all words in English and m is the length of the input sentence. Members of $\mathcal{V}$ are triples:

1. an English hypothesis prefix,
2. the indices of the French words that have already been translated, and
3. the score so far.

$$\mathcal{E} \subseteq \bigcup_{\ell \in \mathbb{N} \cup \{0\}} E^{\ell} \times \bigcup_{m \in \mathbb{N} \cup \{0\}} F^{m} \times [0, 1]$$

Members of $\mathcal{E}$ are triples:

1. an English hypothesis fragment,

2. a French hypothesis fragment, and

3. the partial score.

If we had access to a bilingual consultant, we could ask them whether or not a particular edge in the graph would be visited in an ideal translation. Which nodes in a given graph would we ask about? As always, we would ask the consultant the question that would give us the most information. Assuming (for simplicity) that there are no redundant paths in the search graph (i.e., that every possible translation corresponds to at most one path), the edge that would give us the most information is the one which we are most uncertain about; that is, we are interested in edges that we will traverse with as close to a 50% probability as possible (or more generally, the edges that have the highest *entropy*).

for each node n **in** G (traverse in topological order):

- $P[n] := \sum_{n'} P[n']P[n' \to n]$

Asking about particular edges is effectively asking a bilingual consultant whether or not a phrase in the target language is an appropriate translation of a particular phrase in the source sentence. Unfortunately, we only have access to consultants that speak the source language; hence, instead of asking questions involving both languages, we can only ask questions involving the source language. Our model provides translation tables of the form $\Pr(f \mid e)$, so given a particular phrase in the target language, we can produce a number of candidate phrases in the source language (including the original source phrase) as well as scores for each of these phrases. Notice that the flipped translation model that came out of the noisy channel approach is in exactly the right form for our use!

Now our questions are of the form: which of the following two phrases is the more suitable synonym of this phrase in the original sentence? How much information is

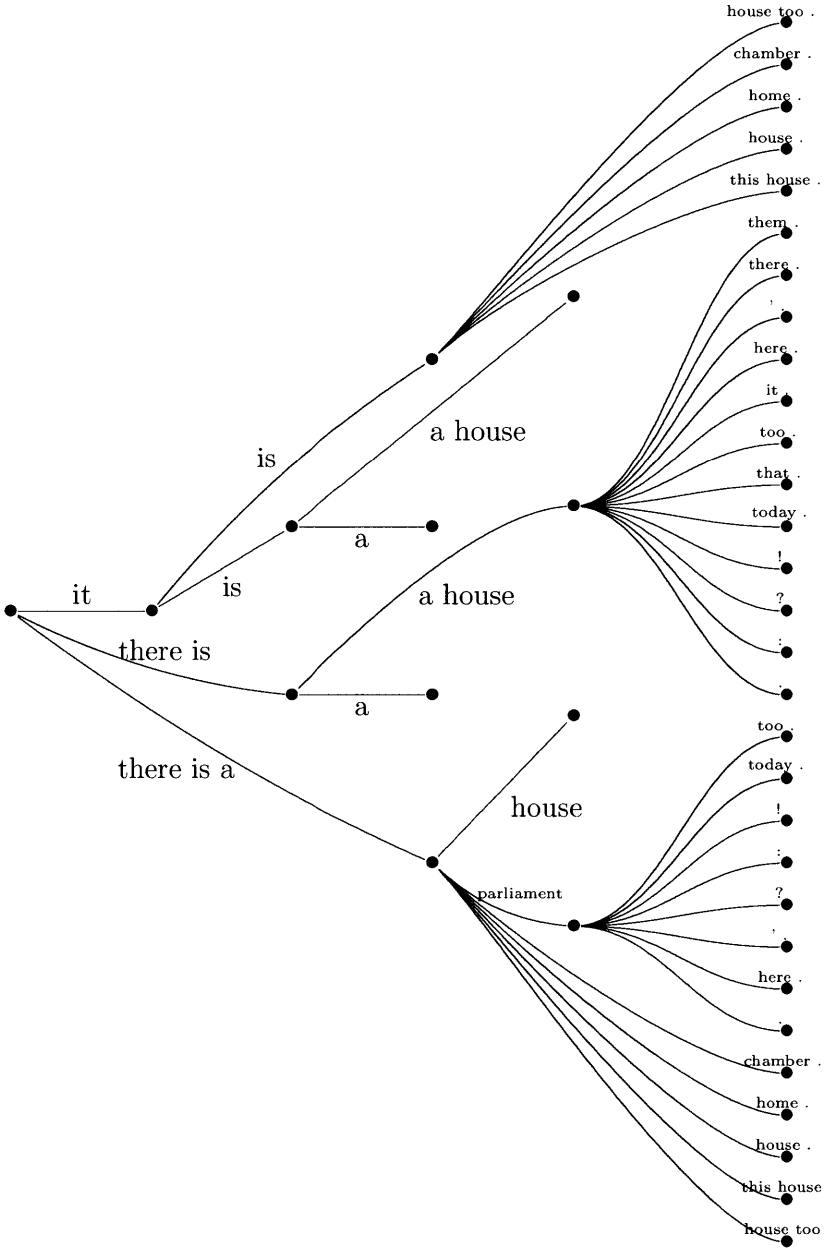


Figure 2-1: A search graph produced by `moses` for the German sentence *Es gibt ein Haus.*

gained by answering such a question?

$$\begin{aligned}
\Pr(\text{translation} \mid \text{answer}) &= \prod \Pr(\text{node} \mid \text{answer}) \\
&= \prod \Pr(\text{answer} \mid \text{node}) \cdot \frac{\Pr(\text{node})}{\Pr(\text{answer})} \\
&= \prod \Pr(f_{\text{answer}} \mid e_{\text{node}}) \cdot \frac{\Pr(\text{node})}{\Pr(\text{answer})}.
\end{aligned}$$

Without knowing the answer, we can tell what information will be gained in expectation by asking this question. We simply ask the question that will maximize this quantity.

In practice, this idea is complicated by the fact that $\Pr(f \mid e)$ is not given and is very often overtrained. `moses` will, by default, learn five models for $\Pr(f \mid e)$; the final model is optimized with a held-out subset of the corpus as an affine combination of the five models. These other models can be used to estimate the model's uncertainty without going back to training. We simply assume that the underlying distribution of $\Pr(f \mid e)$ is drawn uniformly from the simplex of all affine combinations of the five models.

We use a Monte Carlo algorithm to estimate the entropy of the resulting model.

$$\begin{aligned}
& \arg\max_q E_{\text{ans}}\left[H\left[\mathbf{e} \mid \text{ans}\right] \mid q\right] \\
= \; & \arg\max_q \sum_{\text{ans}} \Pr(\text{ans}) H\left[\mathbf{e} \mid \text{ans}\right] \\
= \; & \arg\max_q \sum_{\text{ans}} \sum_{\mathbf{e}} \Pr(\text{ans} \mid \mathbf{e}) \Pr(\mathbf{e}) \log\left(\frac{\Pr(\text{ans} \mid \mathbf{e}) \Pr(\mathbf{e})}{\Pr(\text{ans})}\right) \\
= \; & \arg\max_q \sum_{\text{ans}} \sum_{\mathbf{e}} \Pr(\mathbf{e}) \Pr(\text{ans} \mid \mathbf{e}) \log\left(\Pr(\text{ans} \mid \mathbf{e})\right) \\
& \quad - \sum_{\text{ans}} \Pr(\text{ans}) \log \Pr(\text{ans})
\end{aligned}$$

To incorporate uncertainty in the values of $\Pr(\mathbf{e})$ into the model, we will sample from the space of possible models which `moses`'s training script generates. The

complete algorithm is in figure 2-2.

2.3 Mechanical Turk

The source for the human judgments needed by our algorithm is Mechanical Turk. Mechanical Turk is a service offered by Amazon for outsourcing tasks to be completed by human agents, and for doing so at a large scale. Amazon calls these tasks "Human Intelligence Tasks" (HITs), corresponding to our use-case for crowdsourcing AI complete tasks. We offer $.10 per task. The tasks are easy to complete so this corresponds to a reasonable hourly wage. Unfortunately, a fair percentage of Mechanical Turk Workers simply spam the questions, so it is important to include a test question with a known answer. An example question including the test question is shown in Figure 2.3.

2.4 Results

We performed experiments on the same corpus used for the 2007 NIST evaluation. Our results show a significant improvement on the sentences for which the decisions had an effect. However, the system only had an impact on a small portion (~4%) of sentences; hence, the overall improvement is slight (from 27.7 to 28.1) and is just below the threshold of statistical significance.

for each phrase f_p **in f**:
- **for each** pair of english translations e_1 and e_2 of f_p:
  - $Z' = 0$
  - **for** node n **in** $\mathcal{G}$ corresponding to f_p:
    - $Z' = Z' + n.p$
  - $s = 0$
  - **repeat** N times:
    - draw $\vec{\alpha}$ from the five-dimensional simplex
    - $p'_1 = 0$
    - $p'_2 = 0$
    - **for** node n **in** $\mathcal{G}$ corresponding to f_p:
      - $p = n.p/Z'$
      - $p_1 = \sum_i \alpha_i \Pr(e_1 \mid f_p)$
      - $p_2 = \sum_i \alpha_i \Pr(e_2 \mid f_p)$
      - $Z = p_1 + p_2$
      - $p_1 = p_1/Z$
      - $p_2 = p_2/Z$
      - $s = s + Z'p_1 \log(-logAnsGi) + Z'p_1 \log(-logAnsGi)$
      - $p'_1 = p'_1 + p_1$
      - $p'_2 = p'_2 + p_2$
    - $H = Z_1 \log Z_1 + Z_2 \log Z_2$
    - $S = S + s - H$
  - $S = S/N$

Figure 2-2: Algorithm for incorporating human assistance in phrase-based translation system.

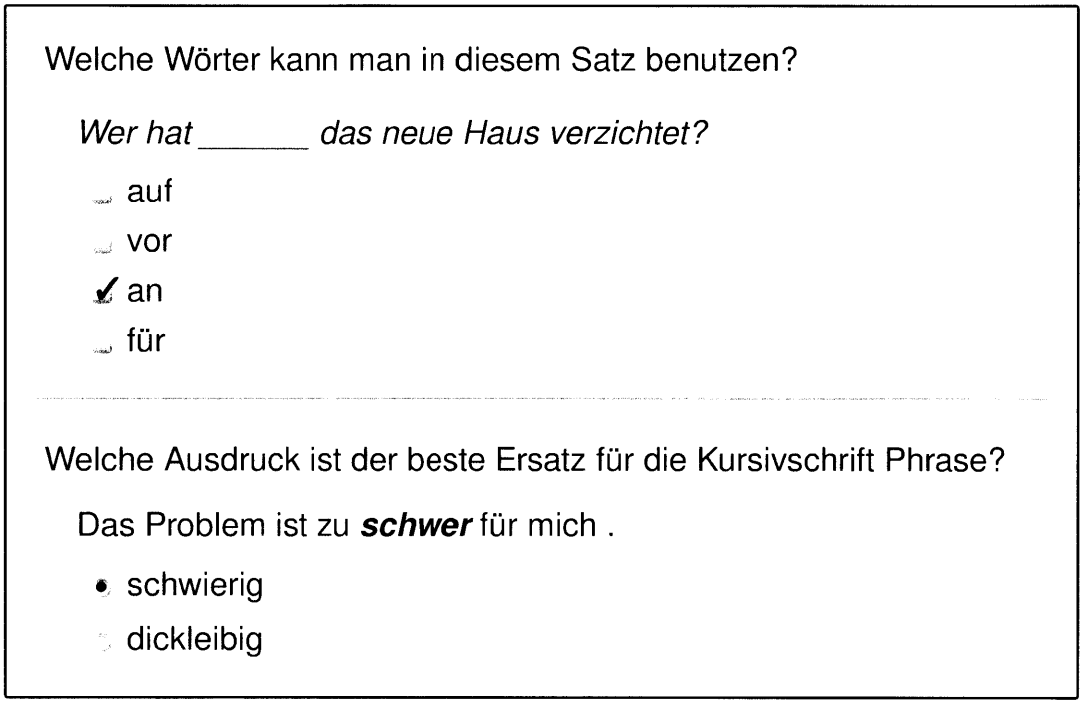


Figure 2-3: Sample Mechanical Turk Questionnaire including a test question.

Chapter 3

Human Assisted Translation Evaluation

We return to the lessons of machine translation (in response to Kay's question) that we touched on in the introduction. The lessons are established as the modern conventional wisdom of the statistical machine translation practitioner and are intended as a review of the major results of the field since Kay first posed this question (roughly the past twenty years); after the review, we will discuss the methodology that was used to obtain these results.

3.1 Answering Kay

3.1.1 Rule-Based MT is Bad

Noisy-channel methods, those described in the introduction, are vastly superior to rule-based systems, and require substantially less effort to design. In our own experiments in 2006 with German to English translation on the EUROPARL corpus [29], Systran (a commercial rule-based translation system) achieved a BLEU score of 0.11, whereas Pharaoh [28] (the predecessor of `moses`[30]) scored a BLEU score of 0.20, a

massive difference.

3.1.2 Hybrid Systems Are OK

In the same series of experiments in 2006, we created a hybrid model where a statistical system was trained on Systran's output of 100,000 EUROPARL sentences. The resulting BLEU score was 0.20, suggesting that statistical systems are able to fix the errors of the rule-based system. This result was reproduced by [17] and is the basis of Systran's current hybrid systems.

3.1.3 Minimum Error-Rate Training

In [37], Och suggested adding a number of high-level parameters to statistical language models and optimizing the BLEU score over these parameters as a meta-step in training a translation system. For example, he proposed optimizing:

$$\arg\max_{\mathbf{e}} \Pr(\mathbf{e} \mid \mathbf{f}) \Pr(\mathbf{e})^{\alpha}$$

to generalize the original:

$$\arg\max_{\mathbf{e}} \Pr(\mathbf{e} \mid \mathbf{f}) \Pr(\mathbf{e}) .$$

This has nothing to do with the Bayesian formalism that we started with for any value of α other than 1; however, minimum error-rate training typically finds $\alpha \in [2, 2.5]$ to be optimal.

3.1.4 Linguistics Doesn't Work

It is appealing to rest an engineering effort on a scientific one; the scientific counterpart of natural language processing is linguistics (and the closely related field of psycholinguistics). A number of intricate linguistic models have been in development for decades. These formalisms can be incorporated into statistical translation models,

Average information content per word	BLEU score
3.5	0.28
5.5	0.52
7.6	0.82
10.0	1.00

Table 3.1: Amount of information content per word in nits (according to a trigram model) needed to obtain a certain BLEU score on the EUROPARL corpus. A BLEU score of 0.28 is state-of-the-art on this corpus, and corresponds to the translation model providing 3.5 nits of information per word, compared to the 10.0 nits required to obtain a perfect BLEU score.

but incorporating anything more than the most coarse, general observations (such as the idea that sentences have hierarchical structure) into the design calls for substantial effort which are not rewarded by increased BLEU scores. Hence, in spite of their appeal, formal linguistics does not often find its way into modern machine translation.

3.1.5 n-gram Language Models Are Best

In particular, the idea that language is a simple Markov chain is frustrating; [6] describes this model as "almost moronic... [capturing] local tactic constraints by sheer force of numbers, but the more well-protected bastions of semantic, pragmatic and discourse constraint and even morphological and global syntactic constraint remain unscathed, in fact unnoticed". In spite of this, n-gram language models are used universally in speech recognition and in machine translation. In fact, n-gram models are easy to train and hard to beat[23, 44].

In our own experiments, we found that it is possible to remove a substantial amount of information-heavy content (as judged by a trigram model) and to obtain a high BLEU score, showing that n-gram models are closely tied to the BLEU score (see Table 3.1).

Perhaps it is because data remains sparse and more complex language models would be successful if only more data were available. On the other hand, more (albeit unlabeled) data is readily available for training English language models than

for almost any other artificial intelligence task.

3.1.6 More n-grams Help... Forever

No group is better placed to take advantage of this data than Google, and the Google translation systems are all based on massive language models harvested from websites and newspapers, and have achieved record performance on open-domain machine translation [5]. Beginning with a state-of-the-art Arabic to English translation system, they found that simply increasing the amount of data available to their n-gram model, even data from sources other than the original target text, consistently increases the BLEU score. This is true in spite of simplifications that had to be made to the back-off model to deal with such massive datasets. They report in their conclusions:

> Significantly, we found that translation quality as indicated by BLEU continues to improve with increasing language model size, at even the largest sizes considered. This finding underscores the value of being able to train and apply very large language models, and suggests that further performance gains may be had by pursuing this direction further.

Their results are shown in figure 3-1.

Google provides researchers with an excellent opportunity for exploration: they offer the use of their translation system *gratis* on their website[1]. The effect that the emphasis on the language model (based on Och's work, described above) combined with the largest language model in the world offers some interesting examples in table 3.2 to reflect on; in order to optimize the language model term $\Pr(\mathbf{e})$, the system will sacrifice $\Pr(\mathbf{f} \mid \mathbf{e})$.

[1] `http://translate.google.com/`

fr →en	La pomme mange le garçon.	
⇒	The apple eats the boy.	Reference
⇒	The apple eats the boy.	Systran
⇒	The boy eats the apple.	Google
fr →en	Un hippopotame me veut pour Noël.	
⇒	A hippopotamus wants me for Christmas.	Reference
⇒	An hippopotamus wants me for Christmas.	Systran
⇒	I want a hippopotamus for Christmas.	Google
de→en	Leute stahlen mein Weißes Auto.	
⇒	People stole my white car.	Reference
⇒	People stole my white car.	Systran
⇒	White people stole my car.	Google
it →en	George Bush non è un idiota.	
⇒	George Bush is not an idiot.	Reference
⇒	George Bush is not an idiot.	Systran
⇒	George Bush is an idiot.	Google
it →en	Ali Mohammad non è un idiota.	
⇒	Ali Mohammad is not an idiot.	Reference
⇒	Ali Mohammad is not an idiot.	Systran
⇒	Mohammad Ali is not an idiot.	Google

Table 3.2: Examples from Systran and Google translate (at the time of this writing), elucidating some possible issues with BLEU.

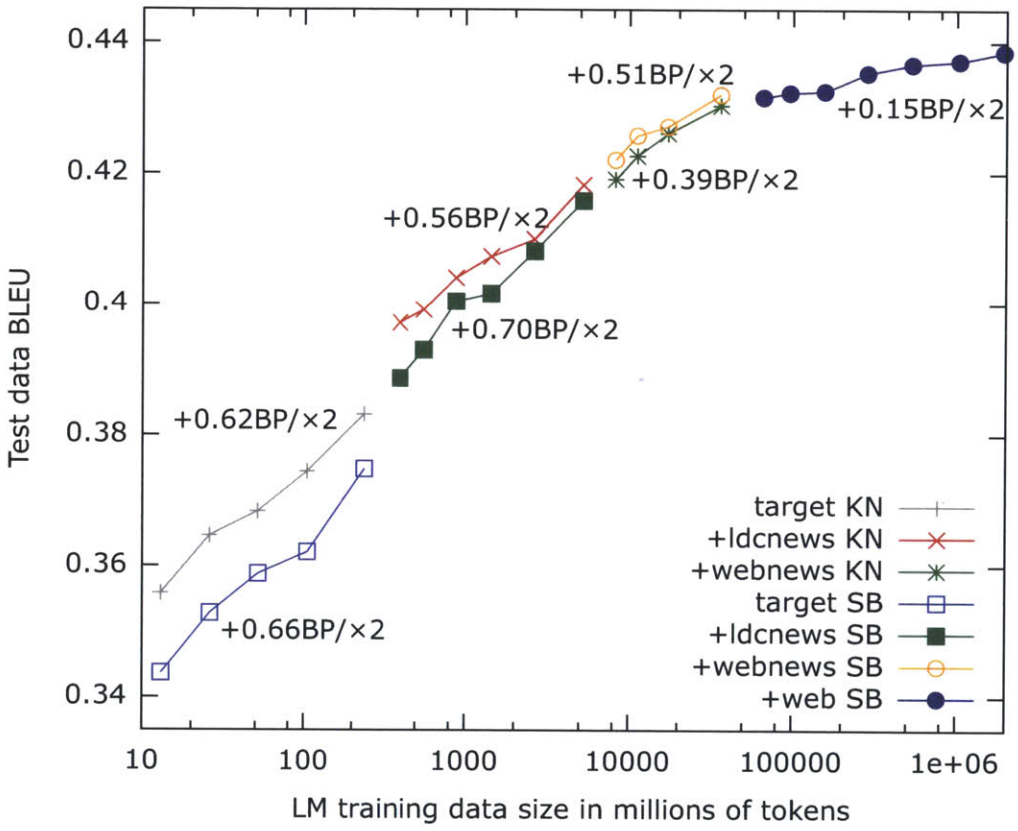


Figure 3-1: BLEU scores for a translation system with varying amounts of data using Kneser-Ney (KN) and Stupid Backoff (SB). BP/×2 indicates the increase in the BLEU score (in 0.01 BLEU points, or BP) for every doubling of the size of the language model (in tokens). Differences greater than 0.51 BP are significant at the 0.05 level. Adapted from [5].

3.1.7 Shrinking the Translation Model is OK

Adding complexity to the language model seems to help quite a bit; equivalently, reducing the complexity of the language model will significantly degrade the quality of the translation system. This does not seem to be the case for the translation model. In fact two experiments on state-of-the-art systems causing a massive reduction in the number of parameters of the translation model have shown no statistically significant effect on the BLEU score (one is our own experiment, described in Appendix D; another is the work of [25]).

3.1.8 Word-Sense Disambiguation Does Not Help Machine Translation

The previous chapter details an attempt by Carpuat and Wu and our own attempt to incorporate a Word-Sense Disambiguation model into machine translation, with disappointing results: even with the assistance of monolingual consultants to perform disambiguation, the scores do not improve significantly. Fully automatic word-sense disambiguation systems attenuate the scores.

3.1.9 Having Monolingual Consultants Revise Translations is Bad

[8] puts forward Kay's idea [26] that monolingual consultants are able to improve the output of a modern translation system by introducing a controlled revision phase. Unfortunately, the results are disappointingly poor: the BLEU score plummeted from 0.51 to 0.43.

3.2 BLEU

Under the force of these results, many of which are surprising but not unbelievable, let's take a moment to understand the methodology that yielded them.

Evaluation is a serious problem for Machine Translation. In other supervised learning tasks, such as hand-written digit recognition, evaluation is a simple matter of comparing the output of a system to the output of a reference (usually a human capable of performing the given task). For hand-written digit recognition, a perfect match is given a score of 1, and a mismatch is given a score of 0: guessing "7" when the reference is "8" is just as bad as guessing "0", even though "7" is closer.

In automatic speech recognition, researchers can't afford to be quite as heavy handed since an all-or-nothing approach is too coarse and (for many applications) a hypothesis which has an error in one word is substanitally better than random text. The typical method of evaluation in this context is Word Error Rate (WER): it is the average number of edit operations (substitutions, deletions, and insertions) that would have to be performed per word to change the output to the reference. WER is closely related to the Levenshtein distance between two strings; fast (linear-time) algorithms exist for computing WER.

WER can be used to evaluate machine translation as well; however, there are some important differences between what is acceptable for a machine translation system and what is acceptable for an automatic speech recognition system that are not captured by WER: in particular, for automatic speech recognition, a single reference is seen as authoritative, whereas in machine translation, two translators are unlikely to ever agree completely.

Indeed, human evaluations consider two aspects of translation quality separately: fluency and adequacy. Fluency is a measure of the quality of text in the target language, whereas adequacy is a measure of the accuracy of the translation. Fluency and adequacy are competing qualities; a translation which is more faithful to the source text, particularly for languages that are unrelated, tends to feel alien in the

target language. For example, the Arabic sentence:

عليٌ مَوجُودٌ

could be translated as "Ali is present" or as "Ali is a findable one." Technically, "a findable one" is a more precise rendering of

مَوجُودٌ

than "present", but clearly "Ali is present" is a much more fluid English sentence.

The BLEU metric[40] is a standard method for evaluating machine translation system performance by comparing translations to one or many human translations. The translations are compared by the precision of n-grams of successively greater length; the BLEU score typically refers to a smoothed 4-gram comparison; mathematically, it can be described by the following formula:

$$\begin{aligned} \text{BLEU} &= e^{I_{c\leq r}\cdot(1-r/c)} \cdot \sqrt[4]{p_1 p_2 p_3 p_4} \\ \log \text{BLEU} &= I_{c\leq r}\left(1-\frac{r}{c}\right) + \sum_{i=1}^{4} \frac{1}{4} p_i \end{aligned}$$

where c is the total length of the candidate translation produced by the system being evaluated, r is sum of the lengths of the reference sentences that most closely match the lengths of the candidate sentences, $I_{c\leq r}$ is 1 if $c \leq r$ and 0 otherwise, and p_j refers to the j-gram precision of the test set.

Evaluating translation systems is a difficult task—BLEU has a number of useful properties that make it a popular choice: it is fast, it is cheap, and it correlates well to judgments of translation fluency and adequacy by bilingual judges. Hence, BLEU is the de facto standard for evaluating automatic machine translation systems and has been for over a decade. It is largely on the authority of BLEU that the conclusions of the previous section rest; over 90% of papers in machine translation published after BLEU use BLEU as their sole method of evaluation. BLEU has had an impressive impact on the agenda of the field.

3.2.1 BLEU Correlates Well With Human Judgments?

BLEU is used as a general metric for translation quality, but considering its definition, it is more precise to call it an n-gram precision model. The papers that introduce BLEU [39, 40] report then that an n-gram precision model correlates strongly with human judgments of translation *quality*. Of course, an improvement in a correlated value does not imply an improvement in the value of interest; however, it is the strength of the correlation that is so promising. In [39], the authors report that the correlation between BLEU and human judgments of adequacy and fluency for French-English translation systems are **0.94** and **1.00**, respectively.

These figures are incredibly impressive. A correlation of 1.0 implies that BLEU is a linear function of the human evaluation of fluency, which implies that BLEU can be used to predict the human evaluation of fluency *perfectly*.

Another interpretation of this result, however, is that humans, when asked to evaluate the fluency of a sentence *against reference sentences*, simply report values that are proportional to the n-gram precision of the sentence versus the references. This interpretation is a statement on the methods that were used to collect the human judgments; that is, under certain conditions, when humans are asked to perform semantic or syntactic comparisons, they will resort to simple string comparisons, and these papers established a set of conditions under which this is true.

It is alarming that this seemingly minor issue of methodology calls into question the research agenda that mainstream researchers have been laboring under for decades.

3.2.2 Re-evaluating BLEU

When we restate the experimental results we cited above as statements about the n-gram precision, the results are far less surprising and the conclusions are less convincing.

The problems with BLEU are becoming better known; Callison-Burch *et al* followed the surprising results of their experiment with human evaluations that showed lower correlations with BLEU than were reported by Papineni *et al* [10]. However, they conclude that, although BLEU clearly cannot be used as a substitute for human evaluations in general, it can be used to compare models within the same class.

Another common belief is that even if BLEU has its problems, it may be possible to account for them with some other automatic metric. It seems that an evaluation method for translation is hardly useful if it cannot be used to compare two arbitrary systems; any performance difference can simply be attributed to the systems being overly different. I propose that any automatic metric can be fooled—building an automatic evaluator is just as hard as building an automatic translation system. Indeed, recent experiments show that, for most language pairs, BLEU correlates more strongly to human judgments than any other available automatic metric does [9].

Reconsidering the results of table 3.1, BLEU is effectively a detector for an n-gram language model. Hence, we propose the use of monolingual consultants for evaluation. With humans in the loop, we have a good chance of mimicing a bilingual evaluator.

According to earlier work [39], monolingual evaluators favor translations that are more fluid to translations that are more adequate. On the other hand, bilingual evaluators tend to be more forgiving to sentences that favor adequacy over fluency. It seems that the gold standard should be the bilingual evaluator; an evaluator that is able to judge the source text as well as the target text will obviously have an advantage. At the same time, there is some reason to pay attention to the monolingual evaluator's perspective, as it is generally people that have little to no familiarity with the source language that will be using a translation system.

3.3 Methods

The psychometrics literature tells us that when faced with difficult judgments, human evaluators experience cognitive fatigue and will turn to heuristics to perform the evaluation. This fatigue can be measured (and lessened) by asking questions which can be answered quickly and consistently [18]. Forced-choice binary comparisons are a premiere method of obtaining information from human evaluators without causing the kind of fatigue that we described.

We will withdraw from the machine translation setting for the remainder of this section to analyze the mathematical aspect of this question. This yields a new evaluation metric which we call CLAIRE[2].

You are the head judge of a baking competition and you are required to announce a full ranking of the cakes that were submitted to you. You are democratic, so you'd like to give a ranking that corresponds to the rankings that would be given by the average cake-taster. Tasters are able to compare exactly two cakes at a time, and will indicate which of the two cakes is better. With an unlimited queue of tasters at your disposal, how can you arrive at the correct ranking of these cakes?

If the tasters were absolutely consistent, this would be a sorting problem, and it could be solved in $N \log N$ comparisons in the worst case. However, we can't guarantee that the tasters are in total agreement. Let us instead model the tasting as a probabilistic process.

Let's assume that each of the cakes $i = 1, 2, \ldots N$ has a secret score $x_i \in \mathbb{R}$, such that when we ask a taster to try cakes i and j, he draws a value X which represents his momentary preference for cake i versus j; if $X + x_i > x_j$, he will report that cake i is better than cake j, which we will denote $i \succ j$; otherwise, he reports $j \succ i$. This will happen with some probability, $F(x_i - x_j)$:

$$F(x_i - x_j) := P\left[i \succ j \mid \vec{x}\right].$$

[2]CLAIRE stands for **C**LAIRE **L**ets **A**nyone **I**nfer **R**anks **E**asily.

There are a few nice properties that will be required of this function for it to be useful; obtaining these properties will help us impose restrictions on the distribution that the preference variable X is drawn from.

1. $F : \mathbb{R} \longrightarrow [0, 1]$, making it produce meaningful probabilities.

2. $F - 1/2$ is odd; also required since the choices are forced. That is, when a taster is given cakes i and j, they must either select $i \succ j$ or $j \succ i$. Consequently, F' is even. This implies that X must be drawn from a distribution that is symmetric about $X = 0$ (i.e., one whose density function is even).

3. $F(x) \to 1$ as $x \to \infty$. That is, our judges will correctly identify differences that are far apart. If we added a random selection phase to our process (i.e., after a decision is made, with some probability, a taster may decide to flip a coin instead of tasting the cakes), it would violate this condition. This will happen if X is finite in expectation.

4. $\log F$ is concave, so when the tasters report $i \succ j$ more often than they report that $\ell \succ m$, it indicates that it is quite likely that $x_i - x_j > x_\ell - x_m$. This will happen if the density function of X decreases somewhat rapidly for positive X.

5. F has a continuous first derivative (that is, F is C^1). This requires X to be drawn from a distribution which has a continuous density.

Now that we've put together a model, given some comparisons C from our tasters (where C_{ij} is the number of times that a taster reported $i \succ j$), we'd like to find the set of scores $\vec{x} \in \mathbb{R}^N$ that is most likely given our data. So we'd like to find the value of $\vec{x}$ that maximizes the following quantity:

$$\begin{aligned} P(\vec{x} \mid C) &\propto P(C \mid \vec{x}) \\ &= \prod_{i=1}^{N} \prod_{j=i+1}^{N} \binom{c_{ij} + c_{ji}}{c_{ij}} F^{c_{ij}}(x_i - x_j) F^{c_{ji}}(x_j - x_i). \end{aligned}$$

It's more convenient to work with the logarithm of this quantity; since log monotonically increases, we can maximize that instead:

$$\begin{aligned}\mathcal{L}(C \mid \vec{x}) &= \log P(C \mid \vec{x}) \\ &= \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left[\log \binom{c_{ij} + c_{ji}}{c_{ij}} + c_{ij} \log F(x_i - x_j) + c_{ji} \log F(x_j - x_i) \right].\end{aligned}$$

Since C is constant with respect to $\vec{x}$, we can ignore the first term and optimize this:

$$\sum_{i=1}^{N} \sum_{j=i+1}^{N} \left[c_{ij} \log F(x_i - x_j) + c_{ji} \log F(x_j - x_i) \right].$$

This is a mapping from $\mathbb{R}^N$ to $\mathbb{R}$, so we'll compute the gradient of this quantity with respect to $\vec{x}$ and find the critical points.

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial x_i} &= \sum_{j \neq i} \left[\frac{c_{ij} F'(x_i - x_j)}{F(x_i - x_j)} - \frac{c_{ji} F'(x_j - x_i)}{F(x_j - x_i)} \right] \\ &= \sum_{j \neq i} F'(x_i - x_j) \left[\frac{c_{ij}}{F(x_i - x_j)} - \frac{c_{ji}}{1 - F(x_i - x_j)} \right].\end{aligned}$$

Assuming the c_{ij} are all nonzero, we can make a few observations about this quantity:

1. If we hold x_j constant for $j \neq i$ and increase i, $\partial \mathcal{L} / \partial x_i$ will decrease (since $\log F$ is concave).

2. If we increase x_j for some $j \neq i$ and hold the remaining components constant, $\partial \mathcal{L} \mid / \partial x_i$ will increase.

3. If we select a subset $A \subset \{1, 2, ..., N\}$ and increase x_i for $i \in A$ such that $x_i - x_j$ remains constant for $i, j \in A$, and the remaining components of $\vec{x}$ also remain constant, $\partial \mathcal{L} / \partial x_i$ will decrease for $i \in A$ and increase for $i \notin A$.

Assume that we have two distinct critical points $\vec{x}$ and $\vec{y}$. Without loss of generality,

order the indices by the difference $y_i - x_i$, so that:

$$y_1 - x_1 \leq y_2 - x_2 \leq \cdots \leq y_N - x_N.$$

We construct a sequence of points $\vec{x}^{(1)}$, $\vec{x}^{(2)}$, ..., $\vec{x}^{(N)}$ in the following fashion:

$$\begin{aligned} \vec{x}^{(0)} &= \vec{x} \\ x_i^{(t+1)} &= \begin{cases} x_i^{(t)} + (y_t - x_t^{(t)}) & \text{for } i \geq t \\ x_i^{(t)} & \text{otherwise} \end{cases} \end{aligned}$$

Hence, $\vec{x}^{(N)} = \vec{y}$. Observe that $\vec{\nabla}\mathcal{L}(\vec{x}^{(1)}) = \vec{\nabla}\mathcal{L}(\vec{x})$ since the two vectors only differ by a constant offset in each coordinate, and that :

$$0 = \frac{\partial \mathcal{L}(\vec{x})}{\partial x_N} = \frac{\partial \mathcal{L}(\vec{x}^{(1)})}{\partial x_N} \leq \frac{\partial \mathcal{L}(\vec{x}^{(2)})}{\partial x_N} \leq \cdots \frac{\partial \mathcal{L}(\vec{x}^{(N)})}{\partial x_N} = \frac{\partial \mathcal{L}(\vec{y})}{\partial x_N} = 0.$$

By the sandwich theorem, all of the quantities are 0; hence, the change from $\vec{x}^{(N-1)}$ to $\vec{x}^{(N)}$ implies that $x_N - y_N = x_{N-1} - y_{N-1}$, etc. That is, $x_i - y_i = x_j - y_j$ for all i, j; thus $\mathcal{L}(\vec{x}) = \mathcal{L}(\vec{y})$. Thus, every critical point is unique up to an offset. Ignoring such offsets, there is at most one global maximum likelihood.

If we close $\mathbb{R}^N$ to include infinite values and extend $\mathcal{L}$ to take its limiting values in that space and remain continuous, it must achieve a maximum value; hence there is a unique global maximum up to offset.

Now, we can optimize $\mathcal{L}$ using a nonlinear optimization method (such as conjugate gradient with Newton-Raphson and Fletcher-Reeves in the case where F is twice differentiable); this is guaranteed to converge rapidly to a local maximum from a random starting point; since every local maximum is a global maximum, **it is easy to optimize $\mathcal{L}$ for nice F's** (those that satisfy the properties we outlined above).

The psychometrics literature suggests the logistic function as a realistic choice for F (obtained by drawing X from a logistic distribution with mean 0, scale 1/2). This

is especially convenient when it comes to computing the gradient:

$$\begin{aligned} F(x_i - x_j) &= \frac{1}{2}(\tanh(x_i - x_j) + 1) \\ &= \frac{e^{x_i - x_j}}{e^{x_i - x_j} + e^{x_j - x_i}}. \end{aligned}$$

This helpfully satisfies the differential equation:

$$F' = F - F^2;$$

hence,

$$\begin{aligned} \frac{\partial \mathcal{L}(\vec{x})}{\partial x_i} &= \sum_{j \neq i} \frac{F'(x_i - x_j)}{F(x_i - x_j)(1 - F(x_i - x_j))} \left[c_{ij}(1 - F(x_i - x_j)) - c_{ji}F(x_i - x_j)\right] \\ &= \sum_{j \neq i} \frac{F'(x_i - x_j)}{F(x_i - x_j)(1 - F(x_i - x_j))} \left[c_{ij} - (c_{ij} + c_{ji})F(x_i - x_j)\right] \\ &= \sum_{j \neq i} \left[c_{ij} - (c_{ij} + c_{ji})F(x_i - x_j)\right]. \end{aligned}$$

We compute the Hessian H for use in nonlinear optimization:

$$\begin{aligned} H_{ij} &= \frac{\partial^2 \mathcal{L}(\vec{x})}{\partial x_i \partial x_j} \\ &= \begin{cases} (c_{ij} + c_{ji})F'(x_i - x_j) & \text{for } i \neq j \\ -\sum_{k \neq i}(c_{ik} + c_{ki})F'(x_i - x_k) & \text{otherwise.} \end{cases} \end{aligned}$$

3.3.1 Active Ranking by Forced Comparisons

Suppose now that having a person taste the cakes is expensive. What can we do to minimize the number of comparisons that must be made? We would like to maximize the amount of information that is gained by each comparison (in expectation), so let's try to minimize the entropy of the distribution $P(\vec{x} \mid C)$.

Let's begin by considering the effect of including an additional comparison $i \succ j$

on the distribution of $\vec{x}$:

$$\begin{aligned} P(\vec{x} \mid C \cup \{i \succ j\}) &= \frac{P(i \succ j \mid \vec{x}, C) P(\vec{x} \mid C)}{P(i \succ j \mid C)} \\ &= \frac{P(i \succ j \mid \vec{x}) P(\vec{x} \mid C)}{P(i \succ j \mid C)}. \end{aligned}$$

We would like to ask the user to compare the items i and j that will have the greatest impact on the entropy in expectation:

$$\begin{aligned} & E_{i \succ j}\left[H(\vec{x} \mid C \cup \{i \succ j\}) \mid C\right] \\ = \; & P(i \succ j \mid C) H(\vec{x} \mid C \cup \{i \succ j\}) + P(j \succ i \mid C) H(\vec{x} \mid C \cup \{j \succ i\}) \\ = \; & P(i \succ j \mid C) \int_{\mathbb{R}^n} P(\vec{x} \mid C \cup \{i \succ j\}) \log P(\vec{x} \mid C \cup \{i \succ j\})\, d\vec{x} \\ & + P(j \succ i \mid C) \int_{\mathbb{R}^n} P(\vec{x} \mid C \cup \{j \succ i\}) \log P(\vec{x} \mid C \cup \{j \succ i\})\, d\vec{x} \\ = \; & \int_{\mathbb{R}^n} P(i \succ j \mid \vec{x}) P(\vec{x} \mid C) \log P(\vec{x} \mid C \cup \{i \succ j\})\, d\vec{x} \\ & + \int_{\mathbb{R}^n} P(j \succ i \mid \vec{x}) P(\vec{x} \mid C) \log P(\vec{x} \mid C \cup \{j \succ i\})\, d\vec{x} \\ = \; & \int_{\mathbb{R}^n} P(i \succ j \mid \vec{x}) P(\vec{x} \mid C) \log P(i \succ j \mid \vec{x})\, d\vec{x} \\ & + \int_{\mathbb{R}^n} P(j \succ i \mid \vec{x}) P(\vec{x} \mid C) \log P(j \succ i \mid \vec{x})\, d\vec{x} \\ & + \int_{\mathbb{R}^n} P(i \succ j \mid \vec{x}) P(\vec{x} \mid C) \log P(\vec{x} \mid C)\, d\vec{x} \\ & + \int_{\mathbb{R}^n} P(j \succ i \mid \vec{x}) P(\vec{x} \mid C) \log P(\vec{x} \mid C)\, d\vec{x} \\ & - \int_{\mathbb{R}^n} P(i \succ j \mid \vec{x}) P(\vec{x} \mid C) \log P(i \succ j \mid C)\, d\vec{x} \\ & - \int_{\mathbb{R}^n} P(j \succ i \mid \vec{x}) P(\vec{x} \mid C) \log P(j \succ i \mid C)\, d\vec{x} \\ = \; & E_{\vec{x}}\left[H(i \succ j \mid \vec{x}) \mid C\right] + H(\vec{x} \mid C) - H(i \succ j \mid C). \end{aligned}$$

Since $H(\vec{x} \mid C)$ is independent of i and j, we simply choose the pair which minimizes

$$E_{\vec{x}}\left[H(i \succ j \mid \vec{x}) \mid C\right] - H(i \succ j \mid C).$$

This is difficult to compute analytically; however, it can be estimated to an arbitrary degree of accuracy via importance sampling, using a multivariate Gaussian centered on the maximum likelihood value of $\vec{x}$ and with a covariance matrix such that the Gaussian has the same Hessian at the mean as $\mathcal{L}$ (i.e., with $\Sigma = H^{-1}$—since the nullity of H is one, eliminating any row will make H invertible). The full algorithm is given in figures 3-2 and 3-3.

3.3.2 Previous Work

Forced choice methods were introduced in the late 20s by L. L. Thurstone, who used them to measure social attitudes, and have been extensively studied in the psychometric community, where they are used to measure human sensitivity to various stimuli. They have also been used to develop guided decision-making processes based on both objective and subjective criteria. Unexpectedly, the type of analysis that we have gone about here does not (to the best of our knowledge) exist in the literature, perhaps because the prevailing scenarios in these fields are directed at minimizing computation instead of minimizing the effort of those surveyed.

Analogous models are also used to establish rankings for games, notably ELO for chess and (much more recently) TrueSkill for multiplayer games on Xbox Live [21]. It is rare to be able to select pairs for comparison in these scenarios, and these models are generally restricted in that only the scores of the items compared should be affected by a comparison, whereas in our case, a single comparison can affect all of the scores.

We believe our model has wide application. A python version of library that uses Amazon Mechanical Turk for data collection is available.

3.3.3 Consequences of Choosing a Bad Activation Function

Although scaling X does not functionally change F, changing $G(p) = F(2F^{-1}(p))$ does. Given a Lipschitz continuous bijection $G : [0, 1] \to [0, 1]$, there exists a unique F

function FINDOPTIMALWEIGHTS(c):

- $\vec{x}$ = random vector in $[0,1]^N$
- $i = 0$
- $k = 0$
- $\vec{r} = \nabla\mathcal{L}(\vec{x})$
- $\vec{d} = \vec{r}$
- $\delta' = \|r\|^2$
- $\delta_0 = \delta'$
- **while** $i < i_{max}$ **and** $\delta' > \varepsilon^2\delta_0$:
  - $j = 0$
  - $\delta_D = \|d\|^2$
  - $\alpha = (\varepsilon^2/\delta_D)/2 + 1$
  - **while** $j < j_{max}$ **and** $\alpha^2\delta_D > \varepsilon^2$:
    - $\alpha = (\nabla\mathcal{L}(\vec{x}) \cdot d)/(\vec{d} \cdot (\mathcal{H}(\vec{x})\vec{d}))$
    - $\vec{x} = \vec{x} + \alpha\vec{d}$
    - $j = j + 1$
  - $\vec{r} = \nabla\mathcal{L}(\vec{x})$
  - $\delta_o = \delta'$
  - $\delta' = \|r\|^2$
  - $\beta = \delta'/\delta_o$
  - $\vec{d} = \vec{r} + \beta\vec{d}$
  - $k = k + 1$
  - **if** $k == N$ **or** $\vec{r} \cdot \vec{d} < 0$:
    - $\vec{d} = \vec{r}$
    - $k = 0$
  - $i = i + 1$
- **return** $\vec{x}$

Figure 3-2: Find the optimal scores based on counts; this is a nonlinear gradient descent algorithm. $\mathcal{H}$ denotes the Hessian of $\mathcal{L}$.

do:

- $\vec{x}$ = FindOptimalWeights(c)
- $\vec{x}_0 = \vec{x}$ with the first element removed
- $\mathcal{H}$ = Hessian of $\mathcal{L}$ at $\vec{x}$ with the first row and column removed
- Diagonalize $\mathcal{H}$, yielding $\mathcal{H} = QDQ'$
- $A = QD^{-1/2}$ (so that $\Sigma = AA' = \mathcal{H}^{-1}$)
- initialize samples, an empty list of (sample, weight) tuples
- **repeat** many **times**:
  - $\vec{z}_0$ = an $(N-1)$-dimensional vector of unit normal samples
  - $\vec{z}_0 = A\vec{z}_0 + \vec{x}_0$
  - $\vec{z} = [\|\vec{x}\|_1 - \|\vec{x}_0\|_1 \mid \vec{z}_0]$ (augment the vector)
  - weight $= \exp(\mathcal{L}(\vec{z})) - \exp(-((\vec{z}_0 - \vec{x}_0) \cdot A(\vec{z}_0 - \vec{x}_0))/2) \Big/ \sqrt{(2\pi)^{(N-1)}|\mathcal{H}|}$
  - samples.append(($\vec{z}$, weight))
- bestScore $= -\infty$; bestPair = None
- **for** i **in** $[0, ..., N-2]$:
  - **for** j **in** $[i+1, ..., N-1]$:
    - $Z = 0$; $\hat{h} = 0$; $\hat{f} = 0$
    - **for** $\vec{z}$, weight **in** samples:
      - $s_{ij} = F(z_i - z_j)$
      - $\hat{h} = \hat{h} + (s_{ij} \log s_{ij} + (1 - s_{ij}) \log(1 - s_{ij}))\cdot$ weight
      - $\hat{f} = \hat{f} + s_{ij}\cdot$ weight
      - $Z = Z +$ weight
    - $\hat{h} = \hat{h}/Z$; $\hat{f} = \hat{f}/Z$
    - score $= \hat{h} - (\hat{f} \log \hat{f} + (1 - \hat{f}) \log(1 - \hat{f}))$
    - **if** bestScore $<$ score:
      - bestScore = score ; bestPair $= (i, j)$
- ask the user to compare the items in bestPair and update c

until convergence

Figure 3-3: Active ranker pseudocode to rank N items.

(up to scaling) that satisfies this relation; this gives a way that one could (in practice) argue for a particular choice of F.

Suppose that for some set of items, there is a true activation function G with true scores $\hat{x}_1, ..., \hat{x}_N$ and we are training against an activation function F. Since $\log F$ is concave, there is a path from the true values $\hat{x}_1, ..., \hat{x}_N$ to the values optimized for F $\vec{x}_1, ..., \vec{x}_N$. If the order of these values differs, then there will be some cross-over point where all of the items are ordered properly except for one pair. However, in the limit of infinitely many observations, this will make the likelihood decrease. Hence, in the limit, the ranking of the items will be correct regardless of the choice of activation function.

3.4 Mechanical Turk

As in the previous chapter, we make use of Mechanical Turk for our judgments. Once again, the tasks are easy to complete (average time is ~30s), showing that they do not impart a significant psychometric load suggesting that the results are reliable. Once again we rely on a preposition test as proof that the workers have reasonable English fluency and are giving the questionnaire some attention. An example question including the test question is shown in Figure 3.4.

3.5 Results

When applying this technique to the machine translation task, we compared our results to those obtained in the 2007 Meta-evaluation task, which obtained results from human evaluators and a number of automatic metrics [9]. See table 3.3 for the results.

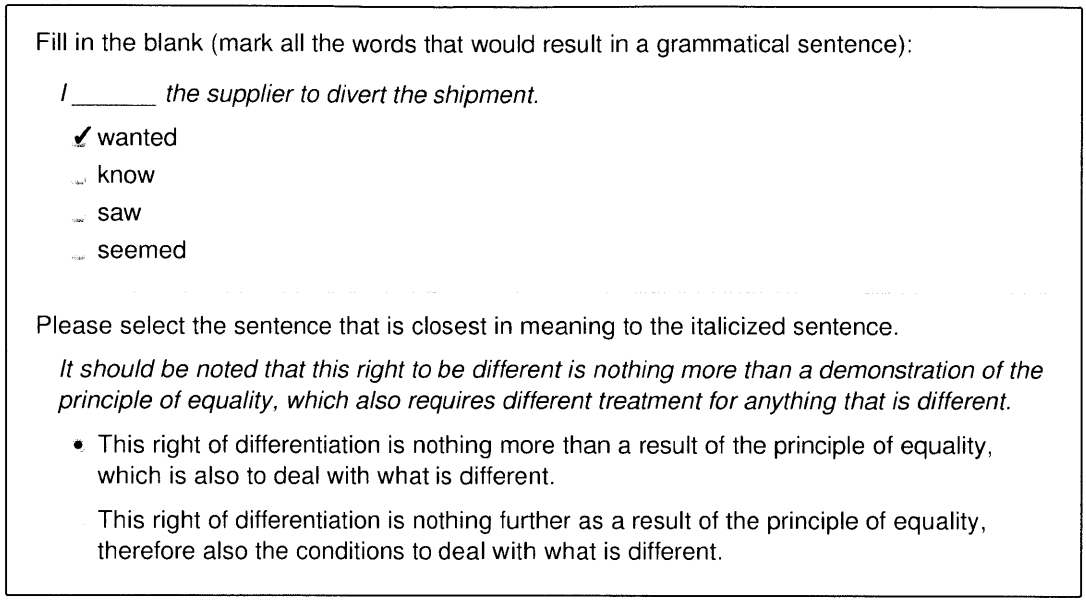


Figure 3-4: Sample Mechanical Turk Questionnaire including a test question.

	CLAIRE	(sigma)	BLEU	Human Evaluation
systran	0.23015	0.027	0.154	1
uedin	0.21232	0.026	0.277	2
liu	0.11797	0.024	0.263	4
nrc	0.08926	0.024	0.254	5
saar	0.06191	0.024	0.198	6
cmu-uka	0.01335	0.024	0.247	7
upc	0.00000	0.021	0.250	3

Table 3.3: Our proposed metric, CLAIRE, captures the results of the shared evaluation and correlates strongly across different types of translation systems.

3.6 Conclusions

We propose an alternative to BLEU and other automatic metrics, which we call CLAIRE. CLAIRE is a method for performing evaluations of translation systems which judges them without undue preference toward a single architecture. CLAIRE scores can be obtained quickly, they are comparable against each other, and correlate strongly with more traditional full-scale human judgments of quality. The benefits of a clear and consistent design carry over from the automatic metrics, and the costs of performing incremental evaluations are minimal ($< \$20$ for a full evaluation).

Chapter 4

Conclusions and Future Work

The question answered by a typical publication in NLP is: "does modelling ⟨*phenomenon*⟩ improve ⟨*metric*⟩?" Consequently, we have learned how to improve certain popular metrics. Over time, differences between the metric and genuine measures of performance yield designs that optimize the former at the expense of the latter.

In particular, we analyze BLEU, the dominant metric in statistical machine translation. We show differences between BLEU and human measures of performance, issues with the papers that originally presented BLEU, and show how systems have optimized BLEU scores to eventually yield diminished performance in human evaluations. We argue that automatic metrics are just as difficult to design as automatic systems. Consequently, we contribute a new metric based on inexpensive human evaluations in the cloud guided by the psychometrics literature; our metric correlates strongly with more expensive conventional human evaluations and is sensitive to minor differences in performance.

We proposed a promising research agenda that can gradually draw us closer to this goal, putting us hand-in-hand with modern linguistics efforts, and more realistically promising systems that are broadly usable. We show ways of applying this technique to parsing, word-sense disambiguation, and machine translation.

Invoking Fred Jelinek's quote from the introduction yields a second interpretation:

Every time I fire a linguist, the performance goes up!

The fruits of linguistics as a field extend beyond linguistic theories for the language process: the many researchers devoted to evaluating the theories scientifically by providing counterexamples that falsify them. Language systems are a marriage between linguistic theory and statistical model: these are scientific elements that warrant scientific rigor. It is difficult to imagine an established scientific enterprise directed at optimizing an automatic metric such as BLEU: physical models so motivated would ignore bodies moving near the speed of light, and the strange behavior of very small particles passing through narrow slits as rare events that are infrequently experienced. How could we have progressed beyond Newtonian mechanics by ignoring the unusual phenomena that lie at the fringe of our models?

As Fred Jelinek himself says, researchers have a right to devote themselves to the solution of intrinsically interesting questions even during an era of senseless product competition. We should be willing to accept an immediate reduction in performance (even in our improved metrics) if we are building a principled system that can reasonably offer enhanced performance in the long-run, and funding agencies should be willing to accept this, too.

The members of ALPAC did not reserve all of their criticism for machine translation. ALPAC's chairman, John R. Pierce, wrote of speech recognition in 1969[41]:

> Most recognizers behave, not like scientists, but like mad inventors or untrustworthy engineers. The typical recognizer gets it into his head that he can solve "the problem." The basis for this is either individual inspiration (the "mad inventor" source of knowledge) or acceptance of untested rules, schemes, or information (the untrustworthy engineer approach)... The typical recognizer... builds or programs an elaborate system that either does very little or flops in an obscure way. A lot of money and time are spent. No simple, clear, sure knowledge is gained. The work has been an experience, not an experiment.

It took decades for speech recognition to plateau, to exhaust the immediate gratification that can be had by doubling the clock-speed of a server or the size of a data-set; recent research in speech recognition begins to look like real science, including explorations on the benefits of linguistic representation, a willingness to sacrifice performance in the short-term for nuanced models that can capture the rare events as well as the common ones. It has become a respectable scientific enterprise with many applications and (with the pervasiveness of mobile phones) ubiquity. Research agenda in hand, I have high hopes that machine translation can match and surpass that success.

Appendix A

EM Algorithm Reference

A.1 Definitions

$$
\begin{aligned}
\mathcal{L}(\Theta) &= \sum_X \log P(X;\Theta) && \text{Likelihood} \\
Q(\Theta',\Theta) &= \sum_X E_Y\left(\log P(X,Y;\Theta') \mid X;\Theta\right) \\
H(\Theta',\Theta) &= -\sum_X E_Y\left(\log P(Y \mid X;\Theta') \mid X;\Theta\right) && \text{Cross-Entropy} \\
KL(\Theta',\Theta) &= H(\Theta',\Theta) - H(\Theta,\Theta) && \text{KL-Divergence} \\
\Theta^{(i)} &= \arg\max_{\Theta} Q\left(\Theta,\Theta^{(i-1)}\right)
\end{aligned}
$$

A.2 Lemmas

Jensen's Inequality:

$$E(\log(X)) \leq \log(E(X))$$

An Obvious Identity:

$$E_X\left(\frac{f(X)}{P(X)}\right) = \int_X f(X)\, dX$$

Lemma:

$$H(\Theta, \Theta) \leq H(\Theta', \Theta) \qquad \forall\ \Theta, \Theta'$$

Equivalently:

$$KL(\Theta', \Theta) \geq 0 \qquad \forall\ \Theta, \Theta'$$

Proof:

$$\begin{aligned}
\forall\ X \qquad 0 &= \log(1) = \log\left(\int_Y P\left(Y \mid X; \Theta'\right)\ dY\right) \\
&= \log\left(\int_Y P\left(Y \mid X; \Theta\right) \cdot \frac{P\left(Y \mid X; \Theta'\right)}{P\left(Y \mid X; \Theta\right)}\ dY\right) \\
&= \log E_Y\left(\frac{P\left(Y \mid X; \Theta'\right)}{P\left(Y \mid X; \Theta\right)} \,\middle|\, X; \Theta\right) \\
&\geq E_Y\left(\log\left[\frac{P\left(Y \mid X; \Theta'\right)}{P\left(Y \mid X; \Theta\right)}\right] \,\middle|\, X; \Theta\right) \\
&= E_Y\left(\log P\left(Y \mid X; \Theta'\right) \mid X; \Theta\right) - E_Y\left(\log P\left(Y \mid X; \Theta\right) \mid X; \Theta\right) \\
E_X(0) = 0 \quad &\geq E_X\left(E_Y\left(\log P\left(Y \mid X; \Theta'\right) \mid X; \Theta\right)\right) - E_X\left(E_Y\left(\log P\left(Y \mid X; \Theta\right) \mid X; \Theta\right)\right) \\
&= H(\Theta, \Theta) - H(\Theta', \Theta).
\end{aligned}$$

$$\Rightarrow \qquad H(\Theta, \Theta) \leq H(\Theta', \Theta)$$

A.3 EM is Nondecreasing

Theorem:

$$\boxed{\mathcal{L}\left(\Theta^{(i)}\right) \leq \mathcal{L}\left(\Theta^{(i+1)}\right)}$$

Proof:

$$\begin{aligned}
\mathcal{L}\left(\Theta\right) &= \sum_X \log P\left(X; \Theta\right) \\
&= \sum_X E_Y\left(\log P\left(X; \Theta\right) \mid X; \Theta^{(i)}\right) \\
&= \sum_X E_Y\left(\log P\left(X; \Theta\right) \mid X; \Theta^{(i)}\right) - Q\left(\Theta, \Theta^{(i)}\right) + Q\left(\Theta, \Theta^{(i)}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_X \left(E_Y \left(\log P\left(X;\Theta\right) \mid X;\Theta^{(i)} \right) - E_Y \left(\log P\left(X,Y;\Theta\right) \mid X;\Theta^{(i)} \right) \right) \\
&\qquad + Q\left(\Theta,\Theta^{(i)}\right) \\
&= -\sum_X E_Y \left(\log \left(\frac{P\left(X,Y;\Theta\right)}{P\left(X;\Theta\right)} \right) \,\middle|\, X;\Theta^{(i)} \right) + Q\left(\Theta,\Theta^{(i)}\right) \\
&= -\sum_X E_Y \left(\log P\left(Y \mid X;\Theta\right) \mid X;\Theta^{(i)} \right) + Q\left(\Theta,\Theta^{(i)}\right) \\
&= H\left(\Theta,\Theta^{(i)}\right) + Q\left(\Theta,\Theta^{(i)}\right). \\
\mathcal{L}\left(\Theta^{(i+1)}\right) - \mathcal{L}\left(\Theta^{(i)}\right) &= \left[H\left(\Theta^{(i+1)},\Theta^{(i)}\right) - H\left(\Theta^{(i)},\Theta^{(i)}\right) \right] \\
&\qquad + \left[Q\left(\Theta^{(i+1)},\Theta^{(i)}\right) - Q\left(\Theta^{(i+1)},\Theta^{(i)}\right) \right] \\
&\geq 0.
\end{aligned}
$$

Thus, the likelihood of successive EM parameter vectors is non-decreasing. (This is a long way from convergence proof...)

Incidentally, we have also shown that, $\forall\ \Theta,\ \Theta'$:

$$\boxed{\mathcal{L}\left(\Theta\right) = H\left(\Theta,\Theta'\right) + Q\left(\Theta,\Theta'\right)}$$

A.4 EM on Multinomials

EM is easy in the special case when $P(X,Y \mid \Theta)$ is a multinomial distribution; that is, it can be written in the form:

$$P(X,Y;\Theta) = \prod_{r=1}^{N} \Theta_r^{\mathrm{Count}_r(X,Y)}.$$

This is a form that occurs very often in language processing tasks.

Let's go!

$$
\begin{aligned}
Q(\Theta',\Theta) &= \sum_X E_Y(\log P(X,Y;\Theta') \mid X;\Theta) \\
&= \sum_X \sum_{r=1}^{N} E_Y\left(\mathrm{Count}_r(X,Y) \mid X;\Theta\right) \log \Theta'_r
\end{aligned}
$$

$$= \sum_{r=1}^{N} \left(\sum_X E_Y \left(\text{Count}_r(X,Y) \mid X; \Theta \right) \right) \cdot \log \Theta'_r$$

It's easy to write an algorithm to maximize Q; we just have to set each Θ'_r to its coefficient and normalize (in ways corresponding to inherent constraints of the parameters):

$$\Theta'_r \propto \sum_X E_Y \left(\text{Count}_r(X,Y) \mid X; \Theta \right).$$

Let's formalize the normalization; the indices $r \in \{1, 2, \ldots, N\}$ are partitioned into disjoint subsets R_1, R_2,..., R_m such that $\sum_{r \in R_i} \Theta_r = 1$. Now we set:

$$\Theta'_r = \frac{\sum_X E_Y \left(\text{Count}_r(X,Y) \mid X; \Theta \right)}{\sum_{r' \in R_i} \sum_X E_Y \left(\text{Count}_{r'}(X,Y) \mid X; \Theta \right)} \qquad \forall\, r \in R_i$$

A.5 Parametrizing Multinomials

Suppose we wish to further parametrize the parameters Θ by another set of parameters α; that is, we define a set of events E_r and set $\Theta_r = P(E_r; \alpha)$ whilst preserving the normalization conditions on Θ_r (i.e., that $\sum_{r \in R_i} P(E_r; \alpha) = 1$, $\forall\ R_i$); thus,

$$P(X, Y; \alpha) = \prod_{r=1}^{N} P(E_r; \alpha)^{\text{Count}_r(X,Y)}.$$

We assume that we can easily find maximum-likelihood α given E_r data (that is, the number of occurrences of each event E_r—not necessarily integral). We proceed:

$$\begin{aligned} Q(\alpha', \alpha) &= \sum_X E_Y (\log P(X, Y; \alpha') \mid X; \alpha) \\ &= \sum_{r=1}^{N} \left(\sum_X E_Y (\text{Count}_r(X,Y) \mid X; \alpha) \right) \log P(E_r; \alpha') \end{aligned}$$

Then, the EM update of α is simply the maximum-likelihood α' where event E_r has

occurred this many times:

$$\sum_X E_Y(\text{Count}_r(X,Y) \mid X, \alpha).$$

Note that these counts are pre-normalization! When the α_i are "grouped" the same way as the Θ_r, the normalization does not enter into the picture (that is, when

$$P(E_r; \alpha') = P(E_r; \alpha'_i) \qquad \forall r \in R_i,$$

where the R_i are defined as above to be sets of Θ parameters that must be normalized); consequently, the Q-maximizing Θ_r can themselves be used as counts to maximize the likelihood of α. Again, if, for any i, we were to multiply the coefficients of Θ_r for $r \in R_i$ by a constant, the maximum likelihood values do not change; thus, no normalization is necessary.

[It is easy to see that normalization can be harmful; consider, for instance, the following experiment: we repeatedly select one of two biased coins to flip and record which coin we flipped and the outcome. Then, the maximum-likelihood probability that the first coin will flip heads, for instance, is the number of heads we got from the first coin divided by the number of times we flipped the first coin. Suppose, however, that we add the constraint that the coins are identically biased. Then the number of times we flipped each coin is important; we cannot correctly estimate the probability of heads with the unconstrained maximum-likelihood probability of heads for each coin alone.]

A.6 EM on IBM2+1dG

Here we give the full derivation of the EM updates for the one-dimensional gaussian framework described in Appendix D for the sake of the mathematically skeptical.

Let's begin by defining the model:

- Training Data

  The training data consists of several triplets $(\mathbf{e}, \mathbf{f}, \vec{a})$.

  1. $\mathbf{e}$ and $\mathbf{f}$ are English and French sentences, respectively, that are translations of each other (observed in training).

  2. $\vec{a} \in \{0, \ldots, \ell\}^m$ represents an alignment between the sentences, where $\ell = |\mathbf{e}|$ and $m = |\mathbf{f}|$ and $a_j = i$ implies that the ith English word corresponds to the jth French word (hidden in training).

- Parameters

  1. $T(f \mid e)$ for all French words f and English words e (the NULL word is added to the English vocabulary).

  2. $\mu_{j,\ell,m}$, $\sigma_{j,\ell,m}$ for all French sentence lengths m, English sentence lengths ℓ, and French word indices $j \in \{1, \ldots, m\}$, respectively the mean and standard deviation of the index of the corresponding English word, given that it is not the NULL word.

  3. $N(j, \ell, m)$ for all French sentence lengths m, English sentence lengths ℓ, and French word indices $j \in \{1, \ldots, m\}$, the probability that that French word is aligned to the English NULL word.

- Model

$$P(\mathbf{f}, \vec{a} \mid \mathbf{e}; T, \mu, \sigma) = \prod_{j=1}^{m} T(\mathbf{f}_j \mid \mathbf{e}_{a_j}) D(a_j \mid j, \ell, m)$$

  where

$$D(a_j \mid j, \ell, m) = \begin{cases} (1 - N(j, \ell, m)) \cdot f_{\mathcal{N}}(a_j \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m}) & j \neq 0 \\ N(j, \ell, m) & j = 0 \end{cases}$$

  Note that this model is deficient; that is, we are not enforcing the proper normalization constraints on D.

- Normalization conditions

$$\sum_f T(f \mid e) = 1 \qquad \forall\, e$$

- Parametrized Multinomial

$$P(\mathbf{f}, \vec{a} \mid \mathbf{e}; T, \mu, \sigma) = \prod_{f,e} T(f \mid e)^{\text{Count}_{f,e}(\mathbf{e},\mathbf{f},\vec{a})} \times \prod_{i,j,\ell,m} D(i \mid j, \ell, m)^{\text{Count}_{i,j,\ell,m}(\mathbf{e},\mathbf{f},\vec{a})}$$

where

$$\begin{aligned} \text{Count}_{f,e}(\mathbf{e}, \mathbf{f}, \vec{a}) &= \sum_{j=1}^{m} \delta(f, \mathbf{f}_j)\delta(e, \mathbf{e}_{a_j}) \\ \text{Count}_{i,j,\ell,m}(\mathbf{e}, \mathbf{f}, \vec{a}) &= \delta(\ell, |\mathbf{e}|)\delta(m, |\mathbf{f}|)\delta(a_j, i), \end{aligned}$$

where $\delta(\cdot, \cdot)$ denotes the Kronecker delta function.

We parametrize T and D by T, μ, σ, and N. Thus, T is trivially parametrized by itself, whereas D is parametrized as described above by μ, σ, and N.

We wish to maximize the function $Q((T, \mu, \sigma, N), (T^i, \mu^i, \sigma^i, N^i))$ given T^i, μ^i, σ^i, and N^i. Since the model can neatly be factored into a term that depends on T alone and a term that depends on D alone, we can optimize these parameters independently. Clearly, the EM procedure for finding the optimal T is unchanged from Model 2; thus, we need only focus on finding the optimal values for μ, σ, and N. We compute:

$$\begin{aligned} &Q((\mu, \sigma, N), (T', \mu', \sigma', N')) \\ &\quad = \sum_{\mathbf{e},\mathbf{f}} \sum_{\vec{a}} P(\vec{a} \mid \mathbf{e}, \mathbf{f}, T', N', \mu', \sigma') \log P(\mathbf{f}, \vec{a} \mid \mathbf{e}; N, \mu, \sigma) \\ &\quad = \sum_{i,j,\ell,m} \Bigg[\underbrace{\left(\sum_{\mathbf{e},\mathbf{f}} E_{\vec{a}}(\text{Count}_{i,j,\ell,m}(\mathbf{e}, \mathbf{f}, \vec{a}) \mid \mathbf{e}, \mathbf{f}; T', \mu', \sigma', N') \right)}_{C(i,j,\ell,m)} \log D(i \mid j, \ell, m) \Bigg] \\ &\quad = \sum_{j,\ell,m} C(0, j, \ell, m) \log N(j, \ell, m) \end{aligned}$$

$$+ \sum_{j,\ell,m} \sum_{i=1}^{\ell} C(i,j,\ell,m) \log[(1 - N(j,\ell,m)) f_{\mathcal{N}}(i \mid \mu_{i,j,\ell,m}, \sigma_{i,j,\ell,m})]$$

$$= \sum_{j,\ell,m} C(0,j,\ell,m) \log N(j,\ell,m) + \sum_{j,\ell,m} \sum_{i=1}^{\ell} C(i,j,\ell,m) \log(1 - N(j,\ell,m))$$

$$+ \sum_{j,\ell,m} \sum_{i=1}^{\ell} C(i,j,\ell,m) \log f_{\mathcal{N}}(i \mid \mu_{i,j,\ell,m}, \sigma_{i,j,\ell,m}).$$

Thus, the optimal value of N is given by:

$$N(j,\ell,m) = \frac{C(0,j,\ell,m)}{\sum_{i=0}^{\ell} C(i,j,\ell,m)}$$

and the μ and σ are optimized by their usual maximum-likelihood estimators:

$$\mu_{j,\ell,m} = \sum_{i=1}^{\ell} i \; C(i,j,\ell,m) \Big/ \sum_{i=1}^{\ell} C(i,j,\ell,m)$$

$$\sigma^2_{j,\ell,m} = \sum_{i=1}^{\ell} (i - \mu_{j,\ell,m})^2 \; C(i,j,\ell,m) \Big/ \sum_{i=1}^{\ell} C(i,j,\ell,m)$$

Appendix B

Information on the WSJ Corpus Tagset

The annotations in the Penn treebank in general are quite difficult to interpret; the manual that is distributed with it is a mere 300 pages or so. The early chapters are extremely formal and handle "simple" sentences, but the last half of the book is special cases and unusual structures that are difficult to analyze.

We'll begin with the nominal terminal tags, a list of all of the members of each of the closed-class parts-of-speech, a list of the nonterminal tags, a list of the extended tags and their definitions, and a list of all of the combinations of extended tags with representative examples that appear in the corpus. This final table suffers contradictions and redundancies; clearly the order of the extended tags is important in some situations, unimportant in others; the inconsistencies in the table reflect corresponding inconsistencies in the corpus and are not errors.

B.1 Terminal Tags (Parts of Speech)

CC	Coordinating Conjunction
CD	Cardinal Number

DT		Determiner
EX		Existential *there*
FW		Foreign Word
IN		Preposition/Subordinating Conjunction
JJ[RS]?		Adjective
	R	Comparative
	S	Superlative
LS		List
MD		Modal
NNP?S?		Noun (singular or mass)
	P	Proper
	S	Plural
PDT		Predeterminer
POS		Possessive Ending
PRP$?		Personal Pronoun
	$	Possessive Pronoun
RB[RS]?		Adverb
	R	Comparative
	S	Superlative
RP		Particle
SYM		Symbol
TO		*to*
UH		Interjection
VB[DGNPZ]?		Verb (base form)
	D	past tense
	G	gerund
	N	past participle
	P	present, not 3rd-person singular
	Z	present, 3rd-person singular
WDT		Wh-determiner

WP$?		Wh-pronoun
	$	Wh-possessive Pronoun
WRB		Wh-adverb

B.2 Members of Closed-Class Tags

Punctuation

``	‘ “
''	’ ”
,	,
:	- – ; : ...
.	! ? .
$	$ A$ C$ HK$ FFr M$ NZ$ S$ US$
#	#
-LRB-	-LCB- -LRB-
-RRB-	-RCB- -RRB-
SYM	= @ * ** & a b c d e f r x z

Closed Class

CC	& and and/or both but either et less minus 'n 'n' neither nor or plus so times versus vs. whether yet
DT	a all an another any both each either every half many neither no some that the them these this those
EX	there

IN	aboard about above across against ago a/k/a albeit along alongside amid among amongst around astride at atop becase behind below beneath beside besides between beyond but by despite down during except expect fiscal for from in inside into lest like minus near nearer nearest neither next notwithstanding of off on onto opposite out outside over par past pending per plus post save so than that then though through throughout 'til till toward towards under underneath unless unlike until up upon v. versus via vs. whereas whether while with within without worth after although as because before how if once since than that though 'til till until when where whether while[1] both... and, either... or, neither... nor, not only... but also, so... as whether... or
LS	1 2 3 4 a b c d e f r x first second third
MD	ca can could 'd dare 'll may might mighta must need ought shall should will wo would
PDT	all both half many nary quite such
POS	' 's
PRP	'em he her him his I it me mine one ours s 's she 't- t' theirs them they us we ya y'all you herself himself itself myself ourselves themselves thyself yourself
PRP$	her his its my our their your
RBR	about down less less-perfectly more than worse
RBS	best hardest highest least most worst
RP	about across ahead along apart around aside at away back before behind by down even for forth forward in of off on open out over through together up upon with yet
TO	to na[2]
WDT	that what whatever which whichever

[1] subordinating conjunctions
[2] as in *gonna*

WP	what who whoever whom
WP$	whose
WRB	how however when whenever where whereby wherever why

Other Things

UH	ah alas amen aw bam boy damn egad heck hello howdy indeed man no nope oh oink ok okay please quack say true uh uh-uh welcome well wham whoopee wow yeah yes zounds

B.3 Nonterminals

ADJP	Adjective Phrase
ADVP	Adverb Phrase
CONJP	Conjunction Phrase
FRAG	Fragment
INTJ	Interjection
LST	List Marker
NAC	Not A Constituent
NP	Noun Phrase
NX	NP head marker
PP	Prepositional Phrase
PRN	Parenthetical
PRT	Particle
QP	Quantifier Phrase
RRC	Reduced Relative Clause
SBARQ	Direct question
SBAR	Subordinate clause
SINV	Sentence, Inverted
SQ	Sentence, Question (inverted yes/no, or the argument of a Wh)
S	Sentence, Declarative
UCP	Unlike Coordinated Phrase
VP	Verb Phrase

WHADJP	Wh-adjective Phrase
WHADVP	Wh-adverb Phrase
WHNP	Wh-noun Phrase
WHPP	Wh-prepositional Phrase
X	Unknown, Uncertain, or Unbracketable

B.4 Extended Tags

I took the next two tables from "The Penn Treebank: Annotating Predicate Argument Structure" by Mitch Marcus *et al* [34].

Text Categories	
-HLN	headlines and datelines
-LST	list markers
-TTL	titles
Grammatical Functions	
-CLF	true clefts
-NOM	non NPs that function as NPs
-ADV	clausal and NP adverbials
-LGS	logical subjects in passives
-PRD	non VP predicates
-SBJ	surface subject
-TPC	topicalized and fronted constituents
-CLR	closely related; should be part of the VP
Semantic Roles	
-PUT	where something is put? (not in the Marcus paper)
-VOC	vocatives
-DIR	direction and trajectory
-LOC	location
-MNR	manner
-PRP	purpose and reason
-TMP	temporal phrases

B.5 Pseudo-Attachment/Null Element Markers

ICH	Interpret Constituent Here
PPA	Permanent Predictable Ambiguity
RNR	Right Node Raising
EXP	Expletive

B.6 Examples of the Full Extended Tags

ADJP	Adjective Phrase
ADJP	This is an *enviably low* level.
ADJP-ADV	...Sarah Lee closed *unchanged* at 60 1/8.
ADJP-CLR	falls *flat*, let *loose*, sat *idle*
	ended *mixed*, closed *higher*
	rated *triple-A*, increased *nearly fivefold*
ADJP-LOC	the site *adjacent to the refinery*
ADJP-MNR	It's better to sell *private.*
ADJP-PRD	Growth is *relatively small.*
ADJP-PRD-TPC	*Conspicuous by its absence* is California.
ADJP-SBJ	*Bigger* is better.
ADJP-TPC	*Not likely*, I think.
	Typical is this response:...
ADVP	Adverb Phrase
ADVP	*Again*, I think *so*, *too.*
	The dollar finished *lower.*
	No one *else* does.
ADVP-CLR	goes *further*, leave them *alone*, broke *loose*
ADVP-CLR-MNR	live *together* cooperatively
ADVP-CLR-TPC	*So* says John, *Along* comes Bobby
ADVP-DIR	shot *up*, go *out*
ADVP-DIR-CLR	bring my prices *down*, trickle *down*, head *off* in some direction
ADVP-DIR-TPC	*here* comes the sun, *along* came Bob
ADVP-EXT	It rose *slightly*, prolong *somewhat*,
ADVP-LOC	Stop *there*, He bought *elsewhere*
ADVP-LOC-CLR	It won't get *there*, He laid *low*, want *out*
ADVP-LOC-CLR-TPC	*therein* lies the draw

ADVP-LOC-PRD	It was *all over the place.*
ADVP-LOC-PRD-TPC	*Here* is an example.
ADVP-LOC-TMP	Where and when it will occur *
ADVP-LOC-TPC	*Here* is an example.
ADVP-LOC-TPC-PRD	*Here* is an example.
ADVP-MNR	Speak *sincerely*, quit *suddenly*
ADVP-MNR-CLR	doing *very well*, do *better*
ADVP-MNR-TMP	how and when the goals would be achieved * *
ADVP-MNR-TPC	But as stock prices recovered, *so* did the US currency
ADVP-PRD	Is it *over*? It was *down a little*, Give it *back*
ADVP-PRD-LOC	get *down in the dumps*,
ADVP-PRD-LOC-TPC	*Here*'s a look at some of the alternatives:
ADVP-PRD-TMP	It only pays when there's a catastrophe *
ADVP-PRD-TPC	*so* did he, *here*'s an idea, *first* on the list of ideas is...
	Soon to feel the glare of attention are lawyers...
ADVP-PRP	Why is this happening *?
	Climate varies *due to natural causes.*
ADVP-PUT	That put's us *back in the soup.*
ADVP-PUT-TPC	*Next door* she put a glass house.
ADVP-TMP	I want it *now.*
	Finally, he got help.
ADVP-TMP-CLR	It starts *as soon as tomorrow*, it didn't last *long*, why it
	took *so long*, it begins *soon*
ADVP-TMP-PRD	Why must it be *so soon*? That was *early in our history*,
	The time is *now.*
ADVP-TMP-TPC	*Initially*, the company said "hello."
ADVP-TPC	He was hungry, or *so* it seemed.
ADVP-TPC-PRD	John panicked, and *so* did Bob.
CONJP	Conjunction Phrase

CONJP	*as well as, rather than, not only, if not, but also, instead of*
FRAG	Fragment
FRAG	The game hasn't changed, *only the name.*
	While no guarantee, an increased salary might improve performance.
	He yelled out "*dolce! dolce!*"
	As *usual*, I'm going nuts.
FRAG-ADV	*Earthquake or not, If anything, Not only that, Who knows?*
FRAG-HLN	
FRAG-PRD	The answer is, "*Yes, of course.*"
FRAG-TPC	*Not so fast*, said Boris.
FRAG-TTL	His magazine, "*Cornhuskers*," criticizes wheat farmers.
INTJ	Interjection
INTJ	*Yes*, they are.
	He said *no* again.
	It was, *well*, fake.
INTJ-CLR	He's learning to say *no.*
INTJ-HLN	
LST	List Marker
LST	*2.* Provide better toilet paper.
NAC	Not A Constituent
NAC	*MIT* students study the wrong books.
	Former president George W. Bush
NAC-LOC	a *Boston* manufacturer, the *Chicago* office
NAC-TMP	The *Oct 12* editorial
NAC-TTL	He engraved a *"#1 Dad"* plaque
NP	Noun Phrase
NP	

NP-ADV	costing $280 *a share*; a dozen cases *a year*; will be billed *several weeks after the expenditure*; climbed *a solid 47%*
NP-BNF	pour *me* a cup of tea, buy *the cat* a present
NP-CLR	take *heart*; follow *suit*; cost *about $2*; Thank *Goodness!*; take *several steps*; make *it* past sth; etc.
NP-CLR-LOC	the house, located *about 50 yards from here*, was destroyed. (the only example)
NP-CLR-TMP	spend two days working and *two days* in the yard (the only example, and 'two days' from 'two days working' is labeled as NP-TMP-CLR)
NP-DIR	foolish to look *the other way*, sit *here* and wait
NP-EXT	sales grew *16%*; funds increased *$13 billion*
NP-HLN	*DISCIPLINARY PROCEEDINGS* against lawyers open to public in Illinois Dow shot up 23 points, in part due to buy programs generated by *stock-index arbitrage* (*Jacksonville, Florida*) Some Headline Here
NP-LGS	bills recently passed by *the House and Senate*, earnings reduced by *the sale of 4 million shares*
NP-LOC	Petrie stores, of *Secaucus, N.J.*; Brown and Platt, *Chicago*
NP-LOC-CLR	changes sweeping *the East bloc*; i once lived *there*
NP-LOC-HLN	
NP-LOC-PRD	the earthquake was *50 miles to the south*; you're *right there*
NP-LOC-TPC-PRD	*here* are some of the major components
NP-MNR	he tries to have it *both ways* on the abortion issue; the bigotry of seeing things *only the Japanese way* the law should restrict citizens *as little as is consistent with good manners.*

NP-MNR-CLR	
NP-PRD	i swam in the lake – *lake eerie*, that is.
	although it is good, *Hollywood on the Hudson* it isn't.
NP-PRD-TPC	*issues discussed* were a, b, and c.
NP-PRD-TTL	he calls cotton "*the fabric of our lives.*"
	titled "*Comments from Students*," it focuses on the real shame of college athletics
NP-SBJ	*the farmer* leaves.
	both deny wrongdoing.
NP-SBJ-TTL	"*Feelings*" is a good song.
NP-TMP	she gave up running *three times a week* in favor of playing golf.
	company a was acquired *last year* by company b.
	in a meeting *last tuesday*, we discussed stocks
NP-TMP-CLR	expires *November 16th*, ended *yesterday*
NP-TMP-HLN	
NP-TMP-PRD	it was *a long time* coming, it was *5:00pm* when i started working
NP-TPC	"*no wonder the competition's green with envy*," said bob; "*wonderful!*" said jim
NP-TTL	the company stopped selling "*the Big Earl*;" I approached "*Mastergate*" with trepidation
NP-TTL-PRD	a movie called "*Marmelade*"; a speech titled "*Marmelade is Tasty*"
NP-TTL-SBJ	"*Baker's Boys*" is both bluesy and funny
NP-TTL-TPC	"*Tivoli Motel*," I read on the sign.
NP-VOC	no, *darling*; move over, *pornographic phone services*; *Hollywood*, you slay me.
NX	NP head marker
NX	Seems that this is intended to mark the head of a noun phrase; I couldn't find anything terribly meaningful

NX-TTL	
PP	Prepositional Phrase
PP	He lived *with her.*
	I'm out *of bed.*
	He had lots *of them.*
	He became angry *in return.*
	There is hope *of change.*
	He can live *with little pleasures.*
PP-BNF	He prints ads *exclusively for retailers.*
PP-CLR	beware *of my dog*, look *at me*, feed *on*, serve *as emcee*, talked *of the aftermath*
PP-CLR-LOC	used *in these strategies*, based *in Houston*,
PP-CLR-TMP	expires *on Dec 31 1990*
PP-CLR-TPC	*Out of the mouths of revolutionaries* are coming words of moderation.
	As factors contributing to the slowdown, he cited ...
PP-DIR	The yield rose *to 5.38%.*
	Fanuc gained 100 *to 7,580.*
	We're not rushing *into anything.*
	Mr. Bush returned *to Washington* Saturday night.
PP-DIR-CLR	The family moves *to another house* at night.
	China exported 65 million pounds of mushrooms, valued at $47 million, *to the U.S.*
	This will lead *to increased litigation.*
PP-DIR-PRD	People say they swim, and that may mean they've been *to the beach.*
	If he wants $70K *out of me*, they have to take everything I have.
PP-DTV	They lied *to me*, They sent it *to the Senate*
PP-EXT	Kyocera advanced 80 yen *to 5,440.*

PP-HLN	
PP-LGS	Buyouts may be curbed *by two rules pending legislation.*
PP-LOC	Most sleep *on the floor*, Reform starts *in the Pentagon*
PP-LOC-CLR	We took it *on the chin*, Politics got *in the way*, Mr. Bass is based *in Ft. Worth*
PP-LOC-CLR-TPC	*At the core of all this* stands a hotel.
PP-LOC-HLN	
PP-LOC-MNR	It should open up channels of communications with the Tigrean rebels *through neighboring Sudan.*
PP-LOC-PRD	I'm *out of bed.* You're *in good company.* Help is *on the way.*
PP-LOC-PRD-TPC	*Among the leading products* is a flu shot, *At the core* is a love for plants.
PP-LOC-TPC	*Behind the posturing* lies a dispute.
PP-LOC-TPC-PRD	*Among the new issues* was Massachusetts's debt.
PP-MNR	He responded *in kind.* This being typed *in a standing position.* This will reduce spending *in a very effective fashion.*
PP-MNR-PRD	The only way to find out is *by listening.*
PP-NOM	He spent *between $5 and $6.*
PP-PRD	That is *for the future.* It looks *like a holiday.*
PP-PRD-LOC	He remains *on the board* as director. It's ironic that David Boren should be *in the center of this.*
PP-PRD-LOC-TPC	*In the corner of the room* is a desk.
PP-PRD-TPC	*Among the most upbeat* was Bobby.
PP-PRP	The shop is closed *for a holiday.*
PP-PRP-CLR	She was jailed in a child custody case *for refusing to reveal the whereabouts of her daughter.*

PP-PRP-PRD	The rise was *partly because of higher demand.*
PP-PUT	This put Mrs. Thatcher *in a bind.*
PP-SBJ	
PP-TMP	*In 1985*, it was warm; I'm done in *two minutes.*
PP-TMP-CLR	He reset opening arguments *for today.*
	The sentencing is set *for Jan 5.*
PP-TMP-PRD	The change is *since year-end.*
PP-TMP-TPC	*Starting in September*, the index started to slide.
PP-TPC	*Of 1500 people sent a questionnaire*, 951 replied.
PP-TPC-CLR	*With that authority* goes an accountability.
PP-TPC-LOC-PRD	*Among the possible suitors* is Italy's Fiat.
PP-TPC-PRD	*Along with the exodus of shopping* is an exodus of jobs.
PP-TTL	Tomorrow's "*On Sports*" will look at another aspect.
PP-TTL-PRD	The name of this column is "*On Sports*".
PRN	Parenthetical
PRN	The alternative—*Giuliani*—is ghastly.
	The rest, *as they say,* is history.
PRT	Particle
PRT	He cashed *in.* Dream *on.*
PRT|ADVP	Help the U.S. win *back* business.
QP	Quantifier Phrase
QP	He talked *about 20* minutes. *Not a* peso is offered.
RRC	Reduced Relative Clause
RRC	Everyone *at the Stick that day* started out as a spectator and ended up as a participant.
	There are still some uncertainties, *particularly regarding possible side effects.*
SBARQ	Direct question
SBARQ	*But who knows?*
SBARQ-HLN	*WHO'S NEWS??*

SBARQ-NOM	now the interest is in *what else can i do*
SBARQ-PRD	the question is, *what is stock worth*?
SBARQ-TPC	*why be a middleman*? asked joe
SBARQ-TTL	the old refrain, "*Who am I to judge*"
SBAR	Subordinate clause
SBAR	
SBAR-ADV	*as it turns out, John loves Mary*; i can run *if i need to run.*
SBAR-ADV-TPC	*if profits don't improve*, we may need to close the company.
SBAR-CLR	i feel *as though i'm being watched*; the agreement calls *for you to give me a lot of money*
SBAR-DIR	capital flows *where it is needed*; go *where the money is*
SBAR-DIR-TPC	i hold that *wherever Mary goes*, John will follow.
SBAR-HLN	*WHO'S NEWS*:
SBAR-LOC	seems indistinguishable from SBAR-LOC-CLR
SBAR-LOC-CLR	my parents will stay *where they are.*
SBAR-LOC-PRD	that is *where i first met my wife.*
SBAR-MNR	they didn't play the game on saturday *as scheduled.*
SBAR-NOM	he has *what all publishers wish for.*
	i hate answering questions about *what would happen if we went to war.*
SBAR-NOM-LGS	they are put off by *what they consider to be restrictive investment regulations.*
SBAR-NOM-PRD	that is *what we did.*
SBAR-NOM-SBJ	*what is true for sheep* is true for goats as well.
SBAR-NOM-TPC	*whatever peopl want to buy*, i'll sell.
SBAR-PRD	our hope is *that the technique could identify diseased vessels.*
SBAR-PRD-TPC	*What counts* is the bottom line.

SBAR-PRP	the charge didn't affect net for the quarter, *as it was offset by tax benefits.*
	we expect a large market in the future, *so the long term it will be profitable.*
SBAR-PRP-PRD	that is *because John ran up the hill.*
SBAR-PUT	...put our resources *where they could do the most*; put his money *where his mouth is*
SBAR-SBJ	he has made it clear *that the issue is important to him personally.*
	we want to make sure *we hold on to our existing customers.*
	where they lag behind the Japanese is in turning the inventiveness into increased production.
SBAR-TMP	i will be happy *when terms are fixed Oct. 26.*
SBAR-TMP-CLR	it didn't help *when she was charged with public drunkenness.*
SBAR-TMP-PRD	that was *before the tax reform made things more complicated.*
SBAR-TPC	he jailed them for several hours *after they defied his order*;
SBAR-TTL	"*When Harry Met Sally*"; "*When Irish Eyes Are Smiling*"
SINV	Sentence, Inverted
SINV	*"I am hungry," said Bob.*
	Says Joe, "I am hungry, too."
SINV-ADV	protected themselves against squalls in any area, *be it stocks, bonds, or real estate*
SINV-HLN	seems same as SINV; just used when it is a headline instead of a normal sentence
SINV-TPC	*Offsetting the lower stake in Lyondell were high crude oil prices, among other things.*

SINV-TTL	the children sang "*Here Comes Santa Claus*"
SQ	Sentence, Question (inverted yes/no, or the argument of a Wh)
SQ	How the hell *can you live with yourself?*
SQ-PRD	What gets by me every time is *has the milk expired?*
	Jimmy asked, "*Can I go to the store?*"
SQ-TPC	*Is that the forecast? Is the government really not helping anybody? Would I have done all those things?*
SQ-TTL	"*Is Science, Or Private Gain, Driving Ozone Policy?*" (article title)
S	Sentence, Declarative
S	
S-ADV	*A piece down*, the computer resigned.
	Investment bonds ended *1/4 point lower.*
	The company wouldn't elaborate, *citing competitive reasons.*
S-CLF	*It is the total relationship that is important.*
S-CLF-TPC	"*It's not very often something like this comes up,*" said Ron.
S-CLR	It helps *to be male.* The farmer stands *to go.*
S-CLR-ADV	Share prices closed lower in Paris, and *mixed* in Amsterdam.
S-HLN	*JAMAICA FIRES BACK*
S-LOC	At the end of the third quarter McDonald's had 10K units operating *world-wide.*
S-MNR	Bonuses would be paid *based on playing time and performance.*
S-MNR-CLR	He began his career *peddling stock to individual investors.*
S-NOM	He apologizes for *sounding pushy.* They don't flinch at *writing them.*

S-NOM-LGS	The insurance provided by *purchasing puts* is worthwhile. It was followed by *our driving to the nearest watering hole.*
S-NOM-PRD	That is *gilding the lily.*
S-NOM-SBJ	*Avoiding failure* is easy.
S-PRD	That lawsuit is *pending.* There is *more volatility to come.* It is *the New Journalism come to television.*
S-PRD-TPC	*Still to come* are issues by Monsanto.
S-PRP	Mr. Gargan favors simply giving money to the SEC *to hire more staff.*
S-PRP-CLR	Lotus Notes is designed *to sort e-mail sent within work groups.*
S-PRP-PRD	The manufacturers said 14.2% of their spending is designed to improve products, 17.5% is *to cut costs*, etc.
S-PRP-TPC	*To provide for the restructuring's costs*, Trinova took an after-tax charge.
S-SBJ	*To watch your child die* is an inhuman experience. "*Work hard, play hard*" is advice best taken with caution.
S-TMP	*Going into the fourth quarter* the sales comparison will be more difficult.
S-TPC	*The market changed*, he adds.
S-TPC-TMP	*A year ago you'd spend two days working and two days in the yard,* he recalls.
S-TTL	He stressed the "*always working*" theme.
S-TTL-PRD	The theme of the conference will be "*take a pension fund manager to lunch.*"
S-TTL-SBJ	It's possible that "*Look Who's Talking*" isn't as entertaining as it seems.
UCP	Unlike Coordinated Phrase

UCP	This requires *regulatory and shareholder* approval. *consumer and other* goods
UCP-ADV	*Third and most important*
UCP-CLR	Stocks closed *lower but above intraday lows*
UCP-DIR	They moved *away from one thing and toward another.*
UCP-EXT	We will stay *through the skiing season or until the money runs out.*
UCP-LOC	The cuts will be made *half within Germany and half abroad.*
UCP-LOC-PRD	
UCP-MNR	It was mentioned *very briefly and in passing.*
UCP-PRD	Long dollar bonds were *flat to up 3/8 point.*
UCP-PRD-LOC	The SEC is *closer to the markets and in a good position to sing.*
UCP-PRP	Growers bred them *more for looks and to satisfy deamnds of long-term storage.*
UCP-TMP	*The next day or even an hour later, this year and in 1990*
UCP-TPC	
VP	Verb Phrase
VP	He *dies.*
VP-TPC	Also *baking a cake* is his mother.
VP-TTL	This newspaper's *Heard on the Street* column
WHADJP	Wh-adjective Phrase
WHADJP	*How strong* is Mr. Mohammad?
WHADVP	Wh-adverb Phrase
WHADVP	*How, Why, How quickly*, etc.
WHADVP-TMP	*When*
WHNP	Wh-noun Phrase
WHNP	*Who, What, Whom, Which*
WHPP	Wh-prepositional Phrase

WHPP	*by how much, under what weight, for whom*
X	Unknown, Uncertain, or Unbracketable
X	*the closer* they got, *the more* the price rose
	the stock tumbled, to end *at*
	the earthquake *was*
	the crowd shouted, "*viva peace, viva.*"
	c- list item 3
	i struggled *to* to eat my sandwich.
	i am married, *no children.*
	it was a funny time, *what* with the vietnam war and all.
	i was hungry *to begin with*
X-ADV	*the more extensive* the voir dire, the easier you make it.
	the more he muzzles his colleagues, *the more* leaks will pop up.
X-CLF	
X-DIR	earnings declined by $120 million *last year's robust levels.*
X-EXT	exports from canada jumped 11% while imports from canada rose *only 2.7%*
X-HLN	
X-PUT	mr. bush's veto power puts him *a commanding position* in the narrowly divided house
X-TTL	

Appendix C

Modifying the Bikel Parser

The Collins-Bikel parser is a very heavily optimized chart parser. The algorithm for the parser is described in the appendices of Collins' PhD thesis [16], and remains the basis of Bikel's implementation. The main modification that is done is to add an equivalentItems list to each element in the chart and to store every item and link that would have been pruned away either by the search or just by virtue of the dynamic program (which is looking for the top-scoring parse). In the Bikel parser, this change should occur in the `add` method of the `Chart` class.

Following this, any calculations that need to be done (to compute inside and outside probabilities, for example) can be done in `Decoder.parse` once the entire forest has been computed. This is also the appropriate point for the parse forest to be emitted.

Appendix D

A Massively Simpler Alignment Model

D.1 Motivation

Assuming that a random variable is Gaussian is a natural choice when the distribution is unknown, because Gaussians have many nice properties. In particular, they are an especially good choice for use in EM algorithms since the maximum likelihood estimates for a Gaussian can be written in closed form. Furthermore, assuming that a variable is Gaussian is often a good approximation due to the Central Limit Theorem, which states that a sum of independent and identically distributed variables with finite mean and variance tends to be Gaussian as the number of addends approaches infinity, and particularly due to the fuzzy Central Limit Theorem, which states that data influenced by many independent sources of noise are roughly normally distributed [48].

For instance, the number of words in English sentences in the EUROPARL corpus, depicted in Figure D-1, is roughly Gaussian. One could explain this based on that fuzzy Central Limit Theorem by imagining a number of independent sources of noise that would influence the length of an English sentence, including such things as

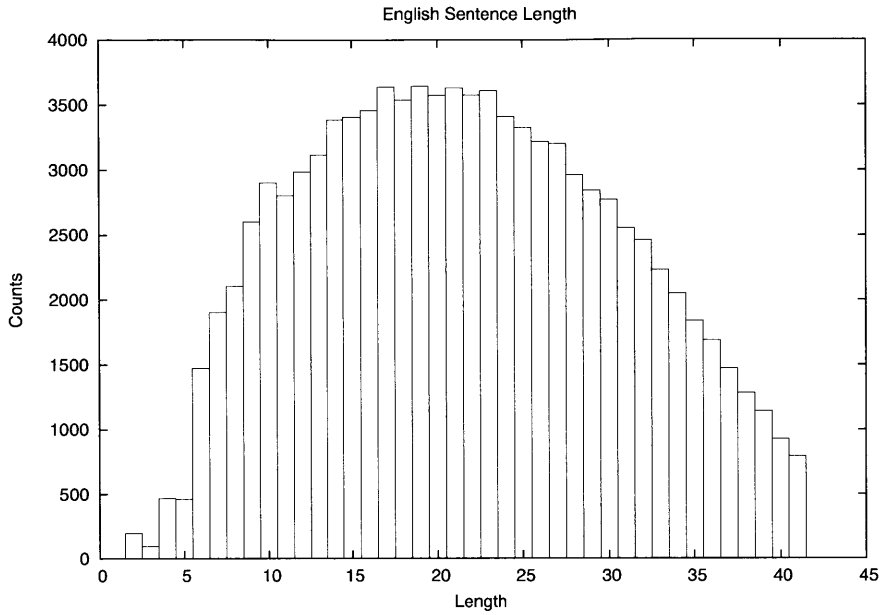
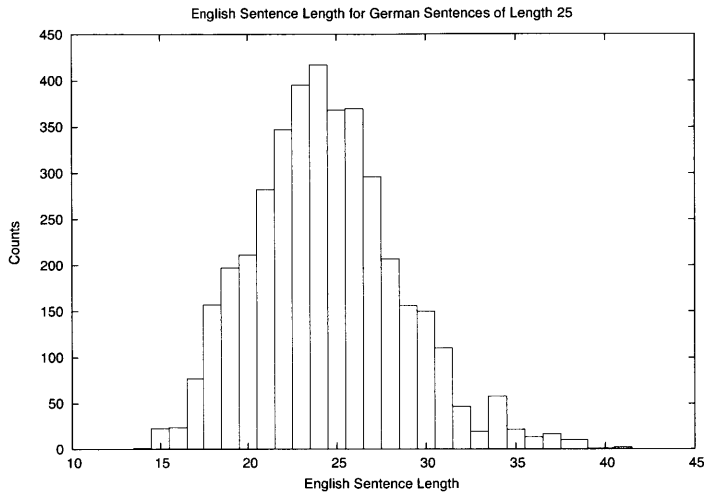
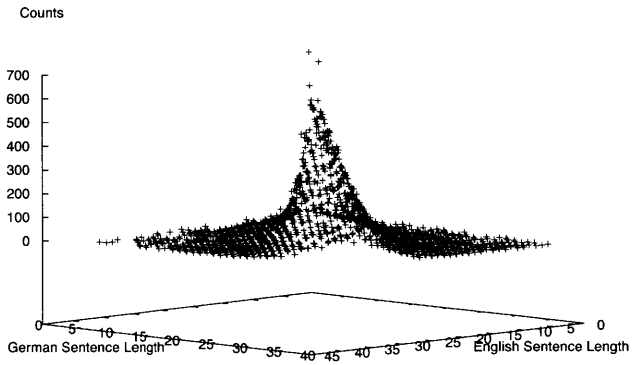


Figure D-1: The length of English sentences drawn from the English translation of the proceedings of the European Parliament appears to have a Gaussian distribution.

the connotation of certain phrases (which influences the author's choice and thereby influences the length of the sentence) and the author's desire to be precise. In fact, the distribution of English sentence lengths for a fixed German sentence length also appears to be Gaussian (see Figure D-2). Most relevant to us, however, is that when one trains an IBM2 model to translate from German to English, the distribution of the index of the word in a, say, 25-word English sentence that the 13th German word of a 25-word German sentence is aligned to also suggests the Gaussian shape, as Figure D-3 shows. This can again be argued using the fuzzy Central Limit Theorem: all else equal, we imagine that a word is most likely to remain in the same relative spot within a sentence; for each transformation that would move it to one side, we imagine there is another transformation that is likely to move it to the other side. Ultimately, the reasons for a translator's choices are innumerable and will be written off as noise here.



(a)



(b)

Figure D-2: (a) The length of English sentences whose German translations are twenty-five words long appears to have a Gaussian distribution. (b) The length of sentences in English and German appears to have a bivariate Gaussian distribution. All things are indeed Gaussian.

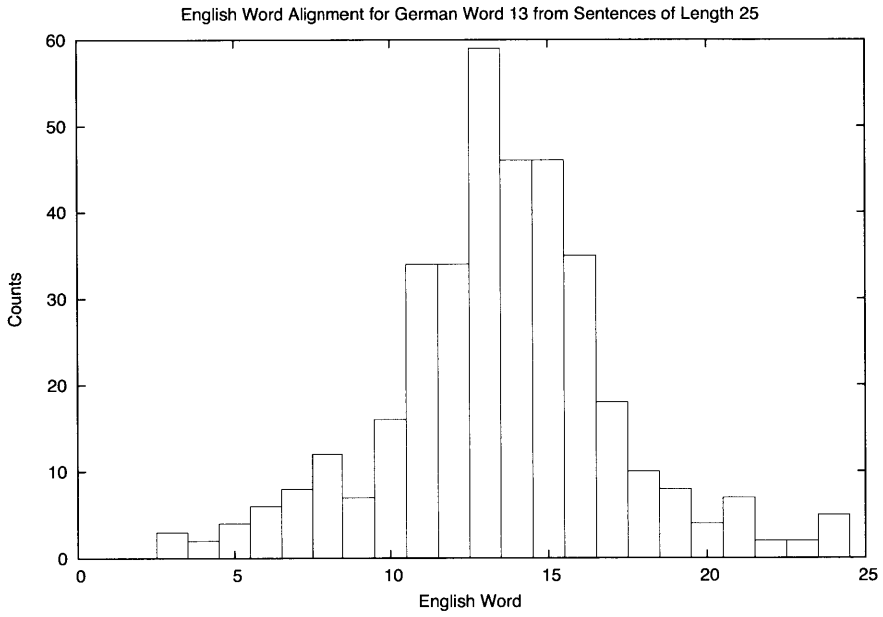


Figure D-3: Our motivation: The indices of aligned words can be approximated by a Gaussian.

One might argue that the best way to model a random variable of unknown distribution is by simply modeling a probability for each of its possible values. This technique is certainly flexible; unfortunately, it is a byword in the machine learning community that one must pay for added flexibility with more training data to avoid over-fitting. Given that even the most sophisticated machine translation models are far from perfect and that they demand massive amounts of training data, our approach is to begin by further constraining existing models instead of creating more flexible ones.

In accordance to this intuition and our guess that alignments look Gaussian, instead of modeling every possible value for each alignment as a separate probability, we will model the alignment probabilities by a single Gaussian. The mathematical formulation for IBM Model 2 is changed by the addition of the second line:

$$\begin{aligned} P(\mathbf{f}, \vec{a} \,|\, \mathbf{e}) &= \prod_{j=1}^{m} T(\mathbf{f}_j \,|\, \mathbf{e}_{a_j}) D(a_j \,|\, j, \ell, m) \\ D(a_j \,|\, j, \ell, m) &= \frac{f_{\mathcal{N}}(a_j \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})}{\sum_{i=0}^{\ell} f_{\mathcal{N}}(i \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})}, \end{aligned}$$

where $f_{\mathcal{N}}(\cdot \,|\, \mu, \sigma)$ denotes the density of the gaussian with mean μ and standard deviation σ, $1/(\sqrt{2\pi}\sigma) \cdot \exp\left((\mu - \cdot)^2/(2\sigma^2)\right)$. That is to say, we replace $D(a_j \,|\, j, \ell, m)$ for $a_j = 1, \ldots, \ell$ with $\mu_{j,\ell,m}$ and $\sigma_{j,\ell,m}$ as our parameters. This typically results in fewer than 10% as many alignment parameters as IBM Model 2.

D.1.1 Algorithm

IBM Model 2 is a multinomial model; consequently, the EM updates are very easy to compute. The type of extension we are discussing ("parametrizing the parameters") corresponds to a simple addition to the algorithm in this case. The full derivation is discussed in the appendix.

Let's begin by casting the model in the standard form: each observation consists of a sentence pair $(\mathbf{e}, \mathbf{f})$ generated from a hidden alignment $\vec{a} \in \{0, \ldots, \ell\}^m$. The IBM

Model 2 probability is:

$$\begin{aligned} P(\mathbf{f}, \vec{a} \mid \mathbf{e}; T, D) &= \prod_{j=1}^{m} T(\mathbf{f}_j \mid \mathbf{e}_{a_j}) D(a_j \mid j, \ell, m) \\ &= \prod_{f,e} T(f \mid e)^{\text{Count}_{f,e}(\mathbf{e},\mathbf{f},\vec{a})} \times \prod_{i,j,\ell,m} D(i \mid j, \ell, m)^{\text{Count}_{i,j,\ell,m}(\mathbf{e},\mathbf{f},\vec{a})} \end{aligned}$$

where

$$\begin{aligned} \text{Count}_{f,e}(\mathbf{e}, \mathbf{f}, \vec{a}) &:= \sum_{j=1}^{m} \delta(f, \mathbf{f}_j) \delta(e, \mathbf{e}_{a_j}) \\ \text{Count}_{i,j,\ell,m}(\mathbf{e}, \mathbf{f}, \vec{a}) &:= \delta(\ell, |\mathbf{e}|) \delta(m, |\mathbf{f}|) \delta(a_j, i), \end{aligned}$$

and $\delta(\cdot, \cdot)$ denotes the Kronecker delta function (1 if the two parameters are equal, and 0 otherwise). The probability model we propose is:

$$\begin{aligned} P(\mathbf{f}, \vec{a} \mid \mathbf{e}; T, \mu, \sigma) &= \prod_{f,e} T(f \mid e)^{\text{Count}_{f,e}(\mathbf{e},\mathbf{f},\vec{a})} \times \prod_{j,\ell,m} N(j, \ell, m)^{\text{Count}_{0,j,\ell,m}(\mathbf{e},\mathbf{f},\vec{a})} \\ &\quad \times \prod_{i,j,\ell,m} \left((1 - N(j, \ell, m)) \cdot f_{\mathcal{N}}(i \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m}) \right)^{\text{Count}_{i,j,\ell,m}(\mathbf{e},\mathbf{f},\vec{a})}. \end{aligned}$$

Here, $N(j, \ell, m)$ is the probability that the jth French word aligns to the NULL English word (we do not wish this probability to be modeled by some slot in the Gaussian, because no position corresponds logically to the NULL word; thus we separate it in this fashion). Note that the $f_{\mathcal{N}}$ term is unnormalized—that is, the model is deficient under this framework for the sake of mathematical convenience. Model 2 clearly falls under the multinomial framework; consequently, this new model falls under the parametrized multinomial framework described in the appendix.

Therefore, the addition we made to the mathematical formulation in the last section results in a single, easy change to the IBM Model 2 algorithm: after each EM iteration, the old D values are replaced by their maximum-likelihood Gaussian

counterparts. That is to say, we transform the D's obtained from Model 2 as follows:

$$\mu_{j,\ell,m} = \frac{1}{m}\sum_{i=1}^{m} i \cdot D(i \mid j,\ell,m)$$

$$\sigma_{j,\ell,m} = \sqrt{\frac{1}{m}\sum_{i=1}^{m} i^2 \cdot D(i \mid j,\ell,m) - \mu_{j,\ell,m}^2}$$

Adding the constraint that the alignment variables are samples of univariate Gaussians corresponds to an altogether simple change to the algorithm; the full algorithm is shown in Figure D-4. (We let the 0th word, e_0, of every English sentence be the NULL word to simplify the notation.)

Our argument for this Gaussian approximation is only based on the idea that the alignment variables look like one-dimensional Gaussian densities and that Gaussians are easy to deal with; this is clearly not the only approximation that satisfies these properties. In fact, there are two other obvious choices based on the Gaussian: "truncated" Gaussians and "integrated" Gaussians. Here they are in math beneath our original formulation:

$$\begin{aligned}
\text{vanilla:} \quad D(a_j \mid j,\ell,m) &= f_{\mathcal{N}}(a_j \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m}), \\
\text{truncated:} \quad D(a_j \mid j,\ell,m) &= \frac{f_{\mathcal{N}}(a_j \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m})}{\sum_i f_{\mathcal{N}}(i \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m})}, \\
\text{and integrated:} \quad D(a_j \mid j,\ell,m) &= \frac{\int_{a_j-1}^{a_j} f_{\mathcal{N}}(i \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m})\, di}{\int_0^{\ell} f_{\mathcal{N}}(i \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m})\, di}.
\end{aligned}$$

In both the vanilla and truncated models, the alignment variable a_j can take on the values 1, ..., ℓ; the difference is that, for mathematical simplicity, the vanilla version is deficient, assigning probabilities to values outside this range (i.e.,that $\sum_{a_j} D(a_j \mid j,\ell,m) \lneq 1$). Although we obviously cannot align words outside of the sentence, we allow the algorithm to assign non-zero probability mass to those alignments; we recover by normalizing over legal alignments afterward. We can instead make this restriction in the algorithm itself. Unfortunately, the M-step of the EM algorithm is no longer a beautiful closed form, but instead requires numerical opti-

Initialize $t(f \mid e)$ and $D(i \mid j, \ell, m)$

do:

· zero $t'(f \mid e)$ and $D'(i \mid j, \ell, m)$

· **for** $(\mathbf{e}, \mathbf{f})$ **in** corpus:

· · $m = |\mathbf{f}|$, $\ell = |\mathbf{e}|$

· · **for** j=1...m:

· · · **for** i=0...ℓ:

· · · · $a_i = t(\mathbf{f}_j \mid \mathbf{e}_i) \cdot D(i \mid j, \ell, m)$

· · · $a_i = a_i / (\sum_{i'} a_{i'})$

· · · **for** i=0...ℓ:

· · · · $t'(\mathbf{f}_j \mid \mathbf{e}_i) = t'(\mathbf{f}_j \mid \mathbf{e}_i) + a_i$

· · · · $D'(i \mid j, \ell, m) = D'(i \mid j, \ell, m) + a_i$

· $t'(f \mid e) = t'(f \mid e) / (\sum_{f'} t'(f' \mid e))$

· $D'(i \mid j, \ell, m) = D'(i \mid j, \ell, m) / (\sum_{i'} D'(i' \mid j, \ell, m))$

· $t = t'$, $D = D'$

· **for** ℓ, m:

· · **for** j=1...m:

· · · $\mu = \frac{1}{\ell} \left(\sum_{i=1}^{\ell} D(i \mid j, \ell, m) \cdot i \right) \Big/ (1 - D(0 \mid j, \ell, m))$

· · · $\sigma = \frac{1}{\ell} \left(\sum_{i=1}^{\ell} D(i \mid j, \ell, m) \cdot (i - \mu)^2 \right) \Big/ (1 - D(0 \mid j, \ell, m))$

· · · **for** i=1...ℓ:

· · · · $D(i \mid j, \ell, m) = \exp(-(i - \mu)^2 / 2\sigma)$

· · · $D(i \mid j, \ell, m) = D(i \mid j, \ell, m) \cdot (1 - D(0 \mid j, \ell, m)) / (\sum_{i'=1}^{\ell} D(i' \mid j, \ell, m))$

until convergence

Figure D-4: The Vanilla Gaussian Algorithm.

mization. Although this is still tractable, it is undesirable and the results are not sufficiently improved to warrant this computational burden (our experiments showed virtually identical results to the vanilla model).

Likewise, the integrated version is very attractive intuitively, but the shape of the density is so close to that of the vanilla edition that any improvements are minute and are outweighed by the added computational complexity, as the Q-function in EM must once again be optimized by a numerical optimization technique. For completeness, we explain how one would perform such optimization; in this case, just as in the vanilla algorithm, the change only amounts to "fitting" (in the maximum-likelihood sense) the appropriate density to the intermediate D-values after each EM step. We can compute the gradient of the likelihood function in both of these cases, so we optimize it using a gradient optimization technique. In our experience, the likelihood functions tend to have long, narrow valleys, so we find that optimizing using conjugate gradient descent is faster than just using steepest descent. (Note again that we do not include the NULL parameter $D(0 \mid \cdot, \cdot, \cdot)$ in the fit.)

We have to fit a curve to each set of D-values, so fix j, ℓ, and m. Define $\overline{D} = D/(1 - D_0)$ to reflect the fact that we are not including the NULL word position in our model and to thus further ease the notational burden. We wish to maximize the log-likelihood, so let's begin by writing it down:

$$\begin{aligned}
\mathcal{L}_{\text{truncated}} &= \sum_{i=1}^{\ell} \overline{D}(i \mid j, \ell, m) \log \frac{f_{\mathcal{N}}(a_j \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m})}{\sum_i f_{\mathcal{N}}(i \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m})} \\
&= \sum_{i=1}^{\ell} \overline{D}(i \mid j, \ell, m) \cdot \frac{-(i-\mu)^2}{2\sigma^2} - \log \left(\sum_{i=1}^{\ell} f_{\mathcal{N}}(i \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m}) \right) \\
\mathcal{L}_{\text{integrated}} &= \sum_{i=1}^{\ell} \overline{D}(i \mid j, \ell, m) \log \frac{\int_{a_{j-1}}^{a_j} f_{\mathcal{N}}(i \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m})\, di}{\int_0^{\ell} f_{\mathcal{N}}(i \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m})\, di} \\
&= \sum_{i=1}^{\ell} \overline{D}(i \mid j, \ell, m) \cdot \log(F_{\mathcal{N}}(i \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m}) - F_{\mathcal{N}}(i-1 \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m})) \\
&\quad - \log(F_{\mathcal{N}}(\ell \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m}) - F_{\mathcal{N}}(0 \mid \mu_{j,\ell,m}, \sigma_{j,\ell,m})),
\end{aligned}$$

where $F_{\mathcal{N}}$ denotes the cumulative distribution function. We evaluate the gradient:

$$\begin{aligned}
\frac{\partial \mathcal{L}_{\text{truncated}}}{\partial \mu} &= \sum_{i=1}^{\ell} \overline{D}(i \,|\, j, \ell, m) \cdot \frac{i-\mu}{\sigma^2} - \frac{\sum_{i=1}^{\ell} f_{\mathcal{N}}(i \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m}) \cdot \frac{i-\mu}{\sigma^2}}{\sum_{i=1}^{\ell} f_{\mathcal{N}}(i \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})} \\
\frac{\partial \mathcal{L}_{\text{truncated}}}{\partial \sigma^2} &= \sum_{i=1}^{\ell} \overline{D}(i \,|\, j, \ell, m) \cdot \frac{-(i-\mu)^2}{2\sigma^4} - \frac{\sum_{i=1}^{\ell} f_{\mathcal{N}}(i \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m}) \cdot \frac{-(i-\mu)^2}{2\sigma^4}}{\sum_{i=1}^{\ell} f_{\mathcal{N}}(i \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})} \\
\frac{\partial \mathcal{L}_{\text{integrated}}}{\partial \mu} &= \sum_{i=1}^{\ell} \overline{D}(i \,|\, j, \ell, m) \cdot \sqrt{2\sigma^2} \cdot \frac{f_{\mathcal{N}}(i-1 \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m}) - f_{\mathcal{N}}(i \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})}{F_{\mathcal{N}}(i \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m}) - F_{\mathcal{N}}(i-1 \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})} \\
&\quad -\sqrt{2\sigma^2} \cdot \frac{f_{\mathcal{N}}(0 \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m}) - f_{\mathcal{N}}(\ell \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})}{F_{\mathcal{N}}(\ell \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m}) - F_{\mathcal{N}}(0 \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})} \\
\frac{\partial \mathcal{L}_{\text{integrated}}}{\partial \sigma^2} &= \sum_{i=1}^{\ell} \overline{D}(i \,|\, j, \ell, m) \cdot \frac{1}{4\sigma^2} \cdot \frac{f_{\mathcal{N}}(i-1 \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m}) - f_{\mathcal{N}}(i \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})}{F_{\mathcal{N}}(i \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m}) - F_{\mathcal{N}}(i-1 \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})} \\
&\quad -\frac{1}{4\sigma^2} \cdot \frac{f_{\mathcal{N}}(0 \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m}) - f_{\mathcal{N}}(\ell \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})}{F_{\mathcal{N}}(\ell \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m}) - F_{\mathcal{N}}(0 \,|\, \mu_{j,\ell,m}, \sigma_{j,\ell,m})}
\end{aligned}$$

We apply the conjugate gradient descent algorithm to the D values [43]:

Given $D \in \mathbb{R}^m$

- Select $\vec{x}_0 = (\mu, \sigma^2) \in \mathbb{R}^2$ at random
- $i = 0$, $\vec{g}_0 = \nabla \mathcal{L}(\vec{x}_0)$, $\vec{h}_0 = -\vec{g}_0$
- **do**:
  - $\lambda_i = \arg\min_{\lambda \geq 0} \mathcal{L}(\vec{x}_i + \lambda_i \vec{h}_i)$
  - $\vec{x}_{i+1} = \vec{x}_i + \lambda_i \vec{h}_i$
  - $\vec{g}_{i+1} = \nabla \mathcal{L}(\vec{x}_{i+1})$
  - $\gamma_i = (\vec{g}_{i+1} - \vec{g}_i) \cdot \vec{g}_{i+1} / \|\vec{g}_i\|^2$
  - $\vec{h}_{i+1} = -\vec{g}_{i+1} + \gamma_i \vec{h}_i$
  - $i = i + 1$
- **until** convergence

return $\vec{x}_i$

D.1.2 Implementation and Evaluation

The dataset used to evaluate our system was the standard EUROPARL set of Köhn *et al.* We used German-English as a representative language pair, as translation is neither especially easy nor especially difficult [29]. The data was aligned at the sentence-level using the standard tools and sentences of vastly differing lengths were removed. Finally, we trained the system on the data and produced Viterbi alignments for each sentence pair. These alignments were output to the Köhn phrase-based system, which itself produced a phrase dictionary. This dictionary was applied to the Pharoah decoder (with a language model trained on the entire available training set). Finally, we applied the system to the standard test set chosen by Köhn in [29]. The resulting translations were compared to the human-translated reference using the BLEU metric of [40].

The BLEU metric is a standard method for evaluating machine translation system performance by comparing translations to one or many human translations. The translations are compared by precision and recall on n-grams of successively greater length; the BLEU score typically refers to a smoothed 4-gram comparison; mathematically, it can be described by the following formula:

$$\text{BLEU} = e^{I_{c \leq r} \cdot (1 - r/c)} \cdot \sqrt[4]{p_1 p_2 p_3 p_4},$$

where r is the length of the reference corpus, c is the total length of the candidate translation produced by the system being evaluated, r is sum of the lengths of the reference sentences that most closely match the lengths of the candidate sentences, $I_{c \leq r}$ is 1 if $c \leq r$ and 0 otherwise, and p_j refers to the j-gram precision of the test set.

We evaluated our technique using the EUROPARL corpus [29] and the applied a BLEU scorer to our model's output on the standard section 24 test set. Our results (when training is done on the full data set) are shown in Table 1. Our results are clearly very competitive with IBM Model 2.

Considering the small number of parameters in the one-dimensional gaussian

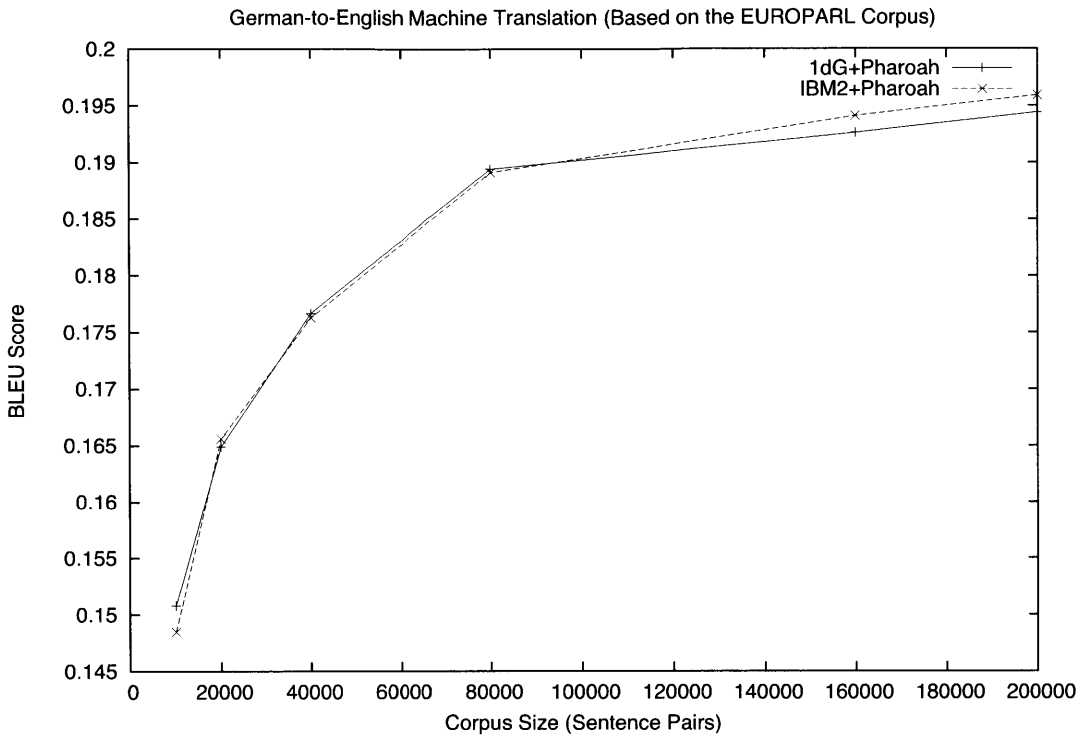


Figure D-5: The performance of the alignments induced by the one-dimensional gaussian model is similar to those induced by the IBM-2 model. This graph shows the BLEU metric of the two models when applied to the EUROPARL training data and standard test set for the German-English language pair.

model, intuition suggests that it should converge to its limiting BLEU score with less data. If this is true, it is an insignificant effect, as Figure D-5 shows; we believe that this is due to the still overwhelming number of translation parameters in the model. The performance of the one-dimensional gaussian model is, in fact, indistinguishable from that of Model 2.

Bibliography

[1] Second international workshop on evaluating word sense disambiguation systems. http://www.senseval.org/, July 2001.

[2] Danit Ben-Ari, Daniel M. Berry, and Mori Rimon. Translational ambiguity rephrased. In *Proceedings of Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, pages 257–262, Pittsburgh, PA, 1988.

[3] Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. *Bracketing Guidelines for Treebank II Style Penn Treebank Project.* Unpublished, January 1995.

[4] Dan Bikel. A multilingual statistical parsing engine. http://www.cis.upenn.edu/~dbikel/software.html#stat-parser.

[5] Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[6] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Jennifer C. Lai, and Robert L. Mercer. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1), 1992.

[7] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Association for Computational Linguistics*, 19(2):263–311, 1993.

[8] Chris Callison-Burch. Linear b system description for the 2005 nist mt evaluation exercise. In *Proceedings of Machine Translation Evaluation Workshop*, 2005.

[9] Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. (meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[10] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the role of bleu in machine translation research. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy, 2006.

[11] Marine Carpuat and Dekai Wu. Word sense disambiguation vs. statistical machine translation. In *Proceedings of the 43rd Annual meeting of the Association for Computational Linguistics*, pages 387–394, Ann Arbor, July 2005.

[12] Marine Carpuat and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[13] David Carter. The treebanker: a tool for supervised training of parsed corpora. In *Computational Environments for Grammar Development and Linguistic Engineering*. Association for Computational Linguistics, July 1997.

[14] Yee Seng Chan, Hwee Tou Ng, and David Chiang. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th Annual Meet-*

ing of the Association of Computational Linguistics, pages 33–40, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[15] Kenneth W. Church and Eduard H. Hovy. Good applications for crummy machine translation. *Machine Translation*, 8:239–258, 1993.

[16] Michael Collins. *Head-Driven Statistical Models for Natural Language Processing.* PhD thesis, University of Pennsylvania, 1999.

[17] Loïc Dugast, Jean Senellart, and Philipp Koehn. Statistical post-editing on systrans rule-based translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 220–223, Prague, June 2007. Association for Computational Linguistics.

[18] Andreas Fink and Aljoscha C. Neubauer. Speed of information processing, psychometric intelligence and time estimation as an index of cognitive load. *Personality and Individual Differences*, 30:1009–1021, 2001.

[19] Sandiway Fong and Robert C. Berwick. New approaches to parsing conjunctions using prolog. 1985.

[20] Ryan Gabbard, Mitchell Marcus, and Seth Kulick. Fully parsing the penn treebank. In *Proceedings of the Human Language Technology Conference of the north American Chapter of the ACL*, pages 184–191, New York, NY, 2006.

[21] Ralf Herbrich and Thore Graepel. Trueskill(tm): A bayesian skill rating system. 2006.

[22] F. Jelinek, R. L. Mercer, L. Bahl, and J. Baker. Interpolated estimation of Markov source parameters from sparse data. In *Pattern Recognition in Practice: Proceedings, International Workshop.* Pattern Recognition in Practice, May 1980.

[23] Frederick Jelinek. Up from trigrams! the struggle for improved language models. In *Proceedings of the Second European Conference on Speech and Technology*,

pages 1037–1040, Genova, September 1991. International Speech Communication Association.

[24] Frederick Jelinek. *Statistical methods for speech recognition.* MIT Press, Cambridge, MA, USA, 1997.

[25] Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[26] Martin Kay. *The MIND System*, pages 155–188. Algorithmics Press, New York, 1973.

[27] Martin Kay. Machine translation will not work. In *Proceedings of the 24th Annual meeting of the Association for Computational Linguistics*, page 268, New York, July 1986. Association for Computational Linguistics.

[28] Philipp Koehn. Pharaoh: a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In *The 6th Conference of the Association for Machine Translation in the Americas*, pages 115–124, Washington DC, September 2004. Association for Machine Translation in the Americas.

[29] Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *The Tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, September 2005. Asia-Pacific Association for Machine Translation.

[30] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion*

Volume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[31] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, AB Canada, May–June 2003. Association for Computational Linguistics.

[32] Beth Levin. *English Verb Classes And Alternations: A Preliminary Investigation.* University of Chicago Press, Chicago, 1993.

[33] Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 133–139, Philadelphia, July 2002. Association for Computational Linguistics.

[34] Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert Macintyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The Penn Treebank: Annotating Predicate Argument Structure. pages 114–119, March 1994.

[35] Hiroshi Maruyama, Hideo Watanbe, and Shiho Ogino. An interactive japanese parser for machine translation. In *International Conference On Computational Linguistics*, pages 257–262, Stanford, CA, 1990.

[36] Barack Obama. A strategy for american innovation. September 2009.

[37] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July 2003. Association for Computational Linguistics.

[38] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[39] Kishore Papineni, Salim Roukos, Todd Ward, John Henderson, and Florence Reeder. Corpus-based comprehensive and diagnostic mt evaluation: Initial arabic, chinese, french, and spanish results. In *Proceedings of HLT 2002, Second International Conference on Human Language Technology Research*, pages 132–137, San Francisco, March 2002. Association for Computational Linguistics.

[40] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, July 2002. Association for Computational Linguistics.

[41] John R. Pierce. Whither speech recognition? pages 1049–1051, 1949.

[42] John R. Pierce, John B. Carroll, Eric P. Hamp, David G. Hays, Charles F. Hockett, Anthony G. Oettinger, and Alan Perlis. *Language and Machines: Computers in Translation and Linguistics.* National Academy of Sciences, Washington, D.C., 1966.

[43] E. Polak. *Optimization: Algorithms and Consistent Approximations.* Springer, New York, 1997.

[44] Matt Post and Daniel Gildea. Parsers as language moels for statistical machine translation. In *Proceedings of the Eighth AMTA Conference*, pages 172–181, Hawaii, October 2008. Association for Computational Linguistics.

[45] Benoit Thouin. *The METEO System*, pages 39–44. 1981.

[46] Masaru Tomita. Disambiguating grammatically ambiguous sentences by asking. In *Proceedings of the 22nd Annual Meeting of the Association for Comuptational Linguistics*, pages 475–480, Stanford, CA, 1984.

[47] Warren Weaver. *Translation*, pages 15–23. Technology Press of the Massachusetts Institute of Technology, Cambridge, MA, 1949.

[48] Eric W. Weisstein. Central limit theorem. http://mathworld.wolfram.com/CentralLimitTheorem.html.