Statistical Machine Translation of French and German into English Using IBM Model 2 Greedy Decoding

Michael Turitzin

Department of Computer Science
Stanford University, Stanford, CA
turitzin@stanford.edu

Abstract

The job of a decoder in statistical machine translation is to find the most probable translation of a given sentence, as defined by a set of previously learned parameters. Because the search space of potential translations is essentially infinite, there is always a trade-off between accuracy and speed when designing a decoder. Germann et al. [4] recently presented a fast, greedy decoder that starts with an initial guess and then refines that guess through small "mutations" that produce more probable translations. The greedy decoder in [4] was designed to work with the IBM Model 4 translation model, which, while being a sophisticated model of the translation process, is also quite complex and therefore difficult to implement and fairly slow in training and decoding. We present modifications to the greedy decoder presented in [4] that allow it to work with the simpler and more efficient IBM Model 2. We have tested our modified decoder by having it translate equivalent French and German sentences into English, and we present the results and translation accuracies that we have obtained. Because we are interested in the relative effectiveness of our decoder in translating between different languages, we discuss the discrepancies between the results we obtained when performing French-to-English and German-to-English translation, and we speculate on the factors inherent to these languages that may have contributed to these discrepancies.

1. Introduction and Related Work

The goal of statistical machine translation is to translate a sentence in one language, say French, into another language, say English. The statistical machine translation process is typically divided into three parts: a language model, a translation model, and a decoder. A language model assigns probalities to English strings, a translation model assigns probabilities to English-French sentence pairs based on the likelihood that one is a translation of the other, and a decoder attempts to find the English sentence for which the product of the language and translation model probabilities is highest. The details and derivation of this type of machine translation system are described in Section 2.

Much research has been done on language models, but we will not discuss such work here as it is only tangential to our work. Our language model, which is described in Section 2.1, was chosen for its favorable balance between simplicity, resource requirements, and effectiveness. Brown et al. [2] introduced a set of translation models that have since been put into use by many researchers in machine translation. These models are *generative* in that they model a process that generates a French sentence from an English one; to translate from French to English, it is necessary to reverse this process and determine which English sentence was most likely to have generated the French one that is to be translated. This generative process is related to the *noisy channel model* of translation, which postulates (falsely) that French speakers "have in mind" an English sentence that is garbled in some way when spoken so as to become a French one.

We employ the second of the models presented in [2], IBM Model 2, and we have adapted the greedy decoder presented in [4] to work with this model. Brown et al. did not include a decoding algorithm in their original paper, and their only public work to date on the subject was published in the form of a patent application [3], which describes a priority-queue ("stack") based IBM Model 3 decoder. Priority-queue based decoders typically enumerate a large subset of possible decodings, forming hypotheses that are inserted into priority queues and ordered based on heuristics. Such decoders are typically extremely slow, especially for longer sentences (i.e., sentences over 15 words in length). Wang and Waibel [7] have presented a similar priority-queue based decoder that was designed to work with IBM Model 2, and Jahr [5] has presented a sophisticated priority-queue based decoder designed to work with IBM Model 4. Germann et al. [4] recently presented an entirely new type of decoder: a decoder that is "greedy" in the sense that it very aggressively prunes paths from the search space, only following the search direction that is locally best, as determined by a number of simple sentence "mutation" operations. Germann et al. showed their greedy decoder to be extremely fast in comparison to stack-based decoders and only marginally less accurate. We have chosen to base our work on this decoder. To our knowledge, no previous work has been done in analyzing the relative effectiveness of a particular decoder in obtaining translations between different languages.

In Section 2 of our paper, we describe the machine translation process and our choices of language model and translation model. In Section 3, we describe our IBM Model 2 greedy decoder, which is a modified version of the IBM Model 4 greedy decoder presented in [4]. In Section 4, we describe several experiments we have performed to test our decoder's ability to translate French and German sentences into English, and we present the translation accuracies we were able to obtain. Finally, in Section 5, we discuss the issue of decoding errors and look at language-specific issues that may have contributed to the discrepancy in accuracy that we observed between translations from French and translations from German in our experiments.

2. Statistical Machine Translation

Say we wish to translate French sentences into English ones. Letting $\mathbf{f}$ be a French sentence and $\mathbf{e}$ be a possible English translation, we therefore wish to find the most likely English translation

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} P(\mathbf{e}|\mathbf{f}), \tag{1}$$

where $P(\mathbf{e}|\mathbf{f})$ is the probability that a given French sentence $\mathbf{f}$ was produced from an English sentence $\mathbf{e}$ (under the noisy channel model of translation).

By Bayes' theorem, we can rewrite $P(\mathbf{e}|\mathbf{f})$ as

$$P(\mathbf{e}|\mathbf{f}) = \frac{P(\mathbf{e})P(\mathbf{f}|\mathbf{e})}{P(\mathbf{f})}. \tag{2}$$

Thus, equation (1) becomes:

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \frac{P(\mathbf{e})P(\mathbf{f}|\mathbf{e})}{P(\mathbf{f})}. \tag{3}$$

We can think of $P(\mathbf{e})$ (or $P(\mathbf{f})$) as the probability that an English (or French) speaker utters the phrase $\mathbf{e}$ (or $\mathbf{f}$) out of all possible utterances. Because $P(\mathbf{f})$ is independent of $\mathbf{e}$, we can simplify equation (3) to obtain what the authors of [2] refer to as the "Fundamental Equation of Machine Translation":

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} P(\mathbf{e})P(\mathbf{f}|\mathbf{e}). \tag{4}$$

Thus, in order to evaluate a particular English sentence translation candidate $\mathbf{e}$, we must have some way of computing $P(\mathbf{e})$ and $P(\mathbf{f}|\mathbf{e})$. The computation of $P(\mathbf{e})$, which is done by a language model, will be discussed in Section 2.1. Because we are using a word alignment model (specifically, IBM Model 2) for translation modeling, we do not explicitly compute $P(\mathbf{f}|\mathbf{e})$. As will be discussed in Section 2.2, we instead generate an alignment $\mathbf{a}$ along with each English sentence $\mathbf{e}$ and compute

$$\langle\hat{\mathbf{e}}, \hat{\mathbf{a}}\rangle = \operatorname*{argmax}_{\langle\mathbf{e},\mathbf{a}\rangle} P(\mathbf{e})P(\mathbf{a},\mathbf{f}|\mathbf{e}) \tag{5}$$

The concept of an alignment between a French and English sentence will be defined in Section 2.2.

2.1. The Language Model

We have chosen to use a trigram mixture model (with interpolation) as our language model for computing $P(\mathbf{e})$ for a given English sentence $\mathbf{e}$. To compute $P(\mathbf{e})$, we iterate through the words of $\mathbf{e}$, multiplying together the conditional trigram probabilities $P_{CT}(w)$ for each word w as we go.

Say that we encounter a word w_3 after having seen the words w_1 and w_2 (in that order). We define the conditional trigram probability of w_3 to be:

$$P_{CT}(w_3) = \lambda_1 P_{abs}(w_3) + \lambda_2 P_{abs}(w_3|w_2) + \lambda_3 P_{abs}(w_3|w_1, w_2), \tag{6}$$

where λ_1, λ_2, and λ_3 are manually chosen interpolation parameters (we use $\lambda_1 = 0.5$, $\lambda_2 = 0.3$, and $\lambda_3 = 0.2$), and $P_{abs}(w_3)$, $P_{abs}(w_3|w_2)$, and $P_{abs}(w_3|w_1, w_2)$ are the unigram and conditional bigram and trigram probabilities of w_3 derived from English training sentences, using *absolute discounting* for smoothing.

Under our absolute discounting smoothing scheme, all previously unseen words encountered in testing data are viewed as instances of one $\langle unk \rangle$ token. Using the unigram case as an example, we compute $P_{abs}(w_3)$ as

$$P_{abs}(w_3) = \begin{cases} (r-\delta)/N & \text{if } r > 0 \\ |S|\delta/N & \text{otherwise} \end{cases}, \tag{7}$$

where N is the total number of words seen in the training data, S is the *set* of unique words seen in the training data, $C(w_3) = r$ is the number of times word w_3 appears in the training data, and δ is a manually adjusted parameter (we used $\delta = 0.75$).

2.2. The Translation Model: IBM Model 2

The goal of a translation model is to compute the probability $P(\mathbf{f}|\mathbf{e})$ of a French sentence $\mathbf{f}$ being the translation of a given English sentence $\mathbf{e}$ (or, under the noisy channel model, having been produced *from* $\mathbf{e}$). The IBM translation models (of which there are five) rely on the concept of *sentence alignment* to compute translation probabilities. An alignment $\mathbf{a}$ between a French sentence $\mathbf{f}$ and an English sentence $\mathbf{e}$ is the mapping between words in the two sentences; if an English word e_i and a French word f_j are aligned, we say that f_j was produced by e_i in the translation process. We define the length of $\mathbf{e}$ to be l and the length of $\mathbf{f}$ to be m; thus, $i = 0, 1, 2, \ldots, l$ and $j = 1, 2, \ldots, m$. We define there to be a "imaginary" word called *NULL* at the 0th position of $\mathbf{e}$, which is why i may equal 0. If a French word is aligned to the *NULL* English word, that French word is said to have been spontaneously generated; that is, no word in the English sentence generated it during the translation process.

An alignment $\mathbf{a}$ corresponding to a French-English sentence pair $(\mathbf{f}, \mathbf{e})$ is a vector of alignment indices a_j mapping words of the French sentence to words of the English sentence (or the *NULL* English word). Each a_j (where $j = 1, 2, \ldots, m$) takes one of the values $0, 1, 2, \ldots, l$. Thus, if (for example) $a_2 = 3$, then the French sentence word f_2 is aligned to the English sentence word e_3. In general, the French sentence word f_j is aligned to the English sentence word e_{a_j}.

Because the IBM models deal with alignments, it is much more efficient to compute $P(\mathbf{a}, \mathbf{f}|\mathbf{e})$ than it is to compute $P(\mathbf{f}|\mathbf{e})$ (which equals $\sum_{\mathbf{a}} P(\mathbf{a}, \mathbf{f}|\mathbf{e})$). We therefore attempt to find the most probable sentence and alignment *pair* $\langle\mathbf{e}, \mathbf{a}\rangle$ for a translation, rather than just the most probable sentence $\mathbf{e}$ (as described in Section 2).

We have chosen to use IBM Model 2 because it is quite a bit simpler and more efficient than the higher IBM models, yet it is still reasonably effective in modeling sentence translation. IBM Model 2 defines $P(\mathbf{a}, \mathbf{f}|\mathbf{e})$ according to the following equation:

$$P(\mathbf{a}, \mathbf{f}|\mathbf{e}) = \prod_{j=1}^{m} P(a_j|j, m, l)P(f_j|e_{a_j}), \tag{8}$$

where $P(a_j|j, m, l)$ is an *alignment probability*—specifically, the probability that for a French sentence of length m and an English sentence of length l the jth French sentence word will align to the a_jth English sentence word—and $P(f_j|e_{a_j})$ is a *translation probability*—specifically, the probability that a particular French word f_j is the translation of a particular English word e_{a_j}.

IBM Model 2 is trained on a set of French-English sentence translation pairs. The *expectation maximization* (EM) algorithm, which works in an iterative fashion, is used to estimate the optimal value for each alignment and translation probability. Each iteration of the EM algorithm computes the expected number of co-occurrences of a particular pair of French and English words (or of a type of alignment) and compares this value to the actual number of co-occurrences (or alignments). The probabilities are then adjusted to account for the differences in these numbers. A thorough discussion of how the EM algorithm works for training IBM Model 2 is beyond the scope of this paper, and we refer the reader to [2] for a full treatment of this topic.

The net effect of the IBM Model 2 translation model is that unlikely translations—such as *dog* translating to *banane* (English: *banana*)—and unlikely alignments—such as those involving sentences of widely varying lengths—are penalized. Some anomalies are still not penalized however; for example, under IBM Model 2 it is just as likely for a particular English word to align with 10 French words as it is for it align with 1 French word. This deficiency is dealt with in the higher IBM models, at the cost of greater complexity and decreased efficiency.

3. IBM Model 2 Greedy Decoding

As was discussed in Sections 2 and 2.2, our goal in performing machine translation using IBM Model 2 is to find the most likely English sentence and alignment pair for a given French sentence:

$$\langle \hat{\mathbf{e}}, \hat{\mathbf{a}} \rangle = \operatorname*{argmax}_{\langle \mathbf{e}, \mathbf{a} \rangle} P(\mathbf{e}) P(\mathbf{a}, \mathbf{f} | \mathbf{e})$$

Finding $\langle \hat{\mathbf{e}}, \hat{\mathbf{a}} \rangle$ (or, more importantly, just $\hat{\mathbf{e}}$) is known as the decoding problem in machine translation. Because the search space of possible English sentences and alignments is essentially infinite, it is clear that computing the probabilities of all such pairs for each French sentence one wishes to translate is infeasible. Various approaches to enumerating likely sentence-alignment pairs have been developed; most such strategies perform a heuristic-based search using one or more priority queues, in which partially completed sentence-alignment hypotheses are stored (e.g., [7], [5]). We have chosen to base our decoding strategy on the greedy decoding algorithm described in Germann et al. [4], which is orders of magnitude faster than existing priority-queue ("stack") based algorithms but has been shown to be only marginally less effective in finding highly probable sentence-alignment pairs.

The greedy decoder described in [4] was designed to work with IBM Model 4, which is more complex than IBM Model 2, the translation model we use. We thus have developed a modified version of this decoder designed to work with IBM Model 2; although our changes are relatively minor, the decoding process would be impossible without them. Because our decoder uses IBM Model 2 (rather than 4), theoretically it should run faster than the decoder described in [4].

3.1. Greedy Decoder Operation

The general greedy decoding strategy described in [4] is as follows:

1. Starting with the French sentence that we wish to translate (into English), we create a preliminary English "gloss" of that sentence. For each word f in the French sentence, we find the English word e for which $P(e|f)$ is highest. Because the *NULL* English word may be the most probable, it is possible for the English gloss to be shorter than the French sentence, but it will never be longer. Because when training the translation model, we obtain only probabilities of the form $P(f|e)$, we must do some addition work to obtain the "inverse" probabilities $P(e|f)$. We will discuss our method of doing this below.
2. Set the current translation $\langle \mathbf{e_T}, \mathbf{a_T} \rangle$ to be the English gloss (which is a sentence and alignment pair).
3. Compute the probability $P(\mathbf{a_T}, \mathbf{f} | \mathbf{e_T})$ of the current translation.
4. Apply a series of "mutations" to the current translation and compute the probability $P(\mathbf{a_{T_i}}, \mathbf{f} | \mathbf{e_{T_i}})$ of the new sentence-alignment pair generated by each mutation. The mutation strategies we use involve adding and removing words from the translation, changing the translations of words, and moving words around. These strategies are described in Section 3.2. We iterate through all possible mutations of the current translation, keeping track of the best (most probable) one $\langle \mathbf{e_{T_M}}, \mathbf{a_{T_M}} \rangle$ we have encountered as we go.
5. If the most probable mutation $\langle \mathbf{e_{T_M}}, \mathbf{a_{T_M}} \rangle$ is *more* probable than the current translation $\langle \mathbf{e_T}, \mathbf{a_T} \rangle$, set $\langle \mathbf{e_T}, \mathbf{a_T} \rangle := \langle \mathbf{e_{T_M}}, \mathbf{a_{T_M}} \rangle$ and go to step 3. Otherwise, stop.

Of the steps listed above, steps 1 and 4 require further explanation. When constructing the English gloss of the French sentence we wish to translate, we require "inverse" probabilities of the form $P(e|f)$, which are not estimated during the IBM Model 2 training process. The authors of [4] have not described specifically how they computed these probabilities. We tried two methods. The first method was simply to apply the translation model training process in reverse, which will automatically estimate the inverse probabilities. We found that this method did not work well at all, largely because it is often the case that $P(f|e)$ is estimated to be low while $P(e|f)$ is estimated to be high (or vice versa). The cause of this problem is largely inherent to the languages involved in the translation process. For example, there may be a common French word f_1 that usually translates to a common English word e_1 and a *rare* French word f_2 that *nearly always* translates to e_1. In this case, $P(e_1|f_2)$ will be very high, while $P(f_2|e_1)$ while be very low (because f_2 is rare). Thus, while the English gloss that is generated may *look* good, it will actually be of very low probability and may be quite far from the optimal translation. This issue rises from the fact that our translation model is generative: we are trying to find the English sentence most likely *to have generated* the given French sentence. Thus, if a particular English word almost never generates a particular French word, that word should not be chosen when constructing the English gloss.

The second method we tried for calculating the inverse probabilities is also quite straightforward and has worked well in practice. First, we note that by Bayes' rule,

$$P(e|f) = \frac{P(f|e)P(e)}{P(f)},$$

where e and f are English and French words, respectively. Second, we note that when creating the English gloss, for a given French word f we are interested in the English word e for which $P(e|f)$ is *highest*, but we are not interested in the actual value of $P(e|f)$. Thus, we can simply compute $P(f|e)P(e)$ for each English word co-occurring with f and find the maximum of these values (and use the corresponding English word). In the case discussed earlier, it is extremely unlikely that we will choose e_1 as a translation for f_2 in the English gloss, as $P(f_2|e_1)$ will be very low.

3.2. Mutation Strategies

We will next discuss the mutation strategies that we employ in step 4 of the process described in the previous section (3.1). We use essentially the same mutation strategies as are described in [4]; however, we needed to make modifications to some of them to get them to work with IBM Model 2.

English e	$P(e$ generated from $NULL)$
.	0.4832
the	0.1635
to	0.0499
is	0.0484
of	0.0350
it	0.0318
that	0.0301
a	0.0235
,	0.0198
in	0.0160
this	0.0155
for	0.0118
have	0.0113
are	0.0088
we	0.0066
on	0.0057
be	0.0055
and	0.0040
there	0.0029
with	0.0028
will	0.0026
has	0.0021
i	0.0019
do	0.0016
an	0.0013
’	0.0012
about	0.0011
they	0.0011
as	0.0011
would	0.0010

Table 1: *The top 30 English words we estimate to be most likely to generate no French words in the translation process (i.e., to have fertility 0). These words were determined to be the most likely to have been generated from the French* NULL *word in a reverse training process.*

We use the following mutation strategies:

- **translate-one-or-two-words**(j_1,w_1,j_2,w_2)
  Changes the translation of the French words at positions j_1 and j_2 to the English words e_1 and e_2. If the English word previously aligned to one of the French words f_{j_i} ($i = 1, 2$) is *only* aligned to that word and w_i is the *NULL* English word, then that English word is deleted from the translation. If one of the French words f_{j_i} is aligned to *NULL* already, then the new English word w_i is inserted into the translation in the position that yields the highest probability. If the English words previously aligned to one of the French words f_{j_i} is *equal* to w_i, then this operation amounts to changing the translation of one word.

  As was suggested in [4], for efficiency reasons we only attempt to change the translation of each French word its 10 most likely English translations (as determined by the inverse translation probabilities $P(e|f)$).

- **translate-and-insert**(j,w_1,w_2)
  Changes the translation of the French word at position j to w_1 and then inserts w_2 into the English sentence at the position that yields the highest probability. If the English word previously aligned to f_j is equal to w_1, this operation amounts to the insertion of one word.

  For efficiency concerns, we choose w_2 from a list of 1024 English words as the authors of [4] suggest. While [4] prescribes that this list should consist of the 1024 English words most likely to have fertility 0 (i.e., not to generate any French words in the translation process), we are using IBM Model 2 and thus do not have this data, as it is not estimated in the model 2 training process. We have devised an alternative method for finding words likely to have fertility 0: we run the iterative model 2 training process in reverse, generating inverse probabilities $P(e|f)$ for each English/French word pair. We then pick the 1024 English words most likely to have been generated from the *French NULL* word. Intuitively, it makes sense that English words unlikely to have been generated by French words are not likely to generate French words *themselves* when the process is reversed. Table 1 shows the 30 words deemed most likely to have been generated from the French *NULL* word according to this method; the results are very similar to those obtained in [5] for the same corpus. We have also experimented with modifying this operation so that it only inserts a new word and does no translation; we have found that this approach is *much* faster and very rarely yields poorer results.

- **remove-word-of-fertility-0**(i)
  If the English word at position i is not aligned to any French words (i.e., has fertility 0), it is deleted from the sentence.

- **swap-segments**(i_1,i_2,j_1,j_2)
  Swaps the (non-overlapping) English word segments $[i_1, i_2]$ and $[j_1, j_2]$; each of these segments can be as short as a word and as long as $l - 1$ words (where l is the length of the English sentence). All alignments between French and English words remain unchanged during the swap.

- **join-words**(i_1,i_2)
  Deletes the English word at position i_1 and aligns all the French words that *were* aligned with this word (e_{i_1}) to the English word at position i_2.

4. Results

We have tested our greedy decoder on multiple corpora and for two types of translation: French to English, and German to English. Our first experiment involved the Hansard corpus, which contains parallel French and English texts of Canadian parliamentary proceedings. We used the sentence-aligned data produced from this corpus by Brown et al. [1]. We trained our translation model (see Section 2.2) on approximately 100,000 sentence translation pairs, and we trained our language model (see Section 2.1) on approximately 200,000 English sentences, or about 4 million words of text.

We ran our decoder on 30 French sentences from the Hansard corpus, each of which was no greater than 20 words in length. For evaluation purposes, we have rated the quality of each decoded sentence. Each sentence was judged either to be *fully understandable and correct*, *fully understandable*, *mainly understandable*, or *not understandable*. Sentences that are *fully understandable and correct* must convey the same idea that is conveyed in the original sentence (as judged from the “gold standard” translation) and must be grammatically correct. Sentences in the *fully understandable* and *mainly understandable* categories need not be grammatically correct, but should not be too far off; their meanings will also be slightly distorted from

Rating	**Example**	
Fully understandable/correct	Original french	*Mon collègue de Toronto dit encore 53 ans.*
	Gold standard	*My friend from Toronto says 53 more years.*
	Decoded translation	***My colleague from Toronto says 53 more years.***
Fully understandable	Original french	*Ils ont dèjà indiquè les embranchements que ils songeaient à fermer.*
	Gold standard	*They have already given an indication of the branches they intend to close.*
	Decoded translation	***They have already indicated that the branches they want to close.***
Mainly understandable	Original french	*Si je comprends bien, on preévoit un taux de intérêts annuel de 70%.*
	Gold standard	*I understand that a rate of at least 70 per cent per annum is being contemplated.*
	Decoded translation	***If I understand it, it provides an annual rate of interest 70 per cent.***
Not understandable	Original french	*Il laissait une femme et une famille.*
	Gold standard	*He left a wife and family.*
	Decoded translation	***It hearing a woman and her family.***

Table 2: *To gauge the accuracy of our decoder (in combination with our language and translation models), we have assigned one of four different ranks to each decoded translation. This table shows typical examples of translations falling into each rank. For each example, the original sentence, the "gold standard" English translation, and the translation our decoder produced are listed.*

the meaning of the original sentence or missing some parts. Sentences in the *not understandable* category have essentially no value as translations. Examples of each translation rating are shown in Table 2. We assign a score from 0 to 1 for each rating: 1.0 for *fully understandable and correct*, 0.75 for *fully understandable*, 0.5 for *mainly understandable*, and 0.0 for *not understandable*. We obtained an average score of **54.2%** for the 30 French sentences we tested our decoder on. Of these 30 sentences, 9 of the decoded translations were rated *fully understandable and correct*, 5 were rated *fully understandable*, 7 were rated *mainly understandable*, and 9 were rated *not understandable*.

Our decoder generally takes less than a second on sentences of less than 5 words, less than 5 seconds on sentences of less than 10 words, and less than 20 seconds on sentences of 20 words or less. The decoder typically takes anywhere from 2 to 15 iterations before it is unable to find a more probable translation using its mutation strategies. Figure 1 shows the decoding process for one sentence in the Hansard corpus. This sentence, which is 8 words long, took 3 iterations to decode.

Because we wanted to see how the performance of our decoder differs when translating between different languages, we performed a second experiment, which involved translating both French and German sentences into English. For this experiment, we used sentence-aligned text derived from the proceedings of the European Parliament [6]. This data set contains parallel texts in French, German, and English (among other languages), which means that we can train our language and translation models on equalivalent text and then test by decoding equivalent sentences (i.e., French and German sentences with identical English translations). Because of memory constraints, we were not able to train our models on quite as many sentence translation pairs as for the previous experiment: we trained our translation model on approximately 60,000 sentence pairs for both French and German, and we trained our language model on 150,000 English sentences, or about 4.5 million words of text.

We ran our decoder on 30 French sentences and 30 German sentences; these sentences were equivalent, in that they were translations of each other and had identical English translations. After rating each decoded translation using the scheme described eariier, we obtained the results shown in Table 3. Of the 30 translations from French, 8 were rated *fully understandable and correct*, 5 were rated *fully understandable*, 10 were rated *mainly understandable*, and 7 were rated *not understandable*; the average score for the French translations was **55.8%**, which is very similar to the result obtained for the first experiment. Of the 30 translations from German, 7 were rated *fully understandable and correct*, 6 were rated *fully understandable*, 7 were rated *mainly understandable*, and 10 were rated *not understandable*; the average score for the German translations was **50.0%**.

	French	**German**
Fully understandable/correct	8	7
Fully understandable	5	6
Mainly understandable	10	7
Not understandable	7	10
Average score	**55.8%**	**50.0%**

Table 3: *The results of our decoding of 30 equivalent French and German sentences into English. Counts of the number of decoded sentences rated in each category are shown as well as the average score for each language.*

5. Analysis

5.1. Decoding Errors

When there are errors in the sentence that a decoder produces, it is difficult to ascertain whether these errors have come as a result of language or translation modeling errors, or as a result of a decoding error. A decoding error is when the decoder fails to find the English sentence that is most likely to have generated the sentence we wish to translate. As the authors of [7] note, decoding errors are difficult to identify: the only way to do so is to come up with an alternative sentence for which the language and translation models assign higher probability. One method to detect decoding errors is to evaluate the probability of the "gold standard" translation having generated the sentence that is being translated. If the probability of this sentence is higher than that of the decoded sentence, there certainly was a decoding error; however, if the probability is *not* higher, which was the case for most of our decodings, we still do not know whether or not there was a decoding error. This sort of finding does, however, give us some indication that our language and translation models may sometimes be failing to work adequately.

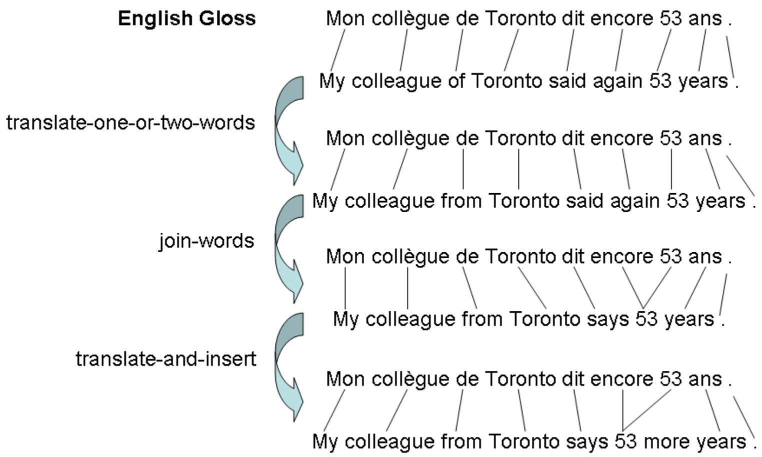
English Gloss
Mon collègue de Toronto dit encore 53 ans .
My colleague of Toronto said again 53 years .
translate-one-or-two-words
Mon collègue de Toronto dit encore 53 ans .
My colleague from Toronto said again 53 years .
join-words
Mon collègue de Toronto dit encore 53 ans .
My colleague from Toronto says 53 years .
translate-and-insert
Mon collègue de Toronto dit encore 53 ans .
My colleague from Toronto says 53 more years .

Figure 1: *This figure depicts the translation process of our greedy decoder for a particular French sentence, "Mon collègue de Toronto dit encore 53 ans." The first English sentence is the gloss generated in the first stage of the decoding process. Each additional English sentence and alignment pair was generated using the mutation operation indicated next to the arrow pointing to it. The lines connecting French and English words represent the alignment at each stage of the process. The "gold standard" translation of the French sentence was: "My friend from Toronto says 53 more years."*

5.2. Language-specific Issues

Although there was not a huge difference in the average scores obtained by our decoder for French-to-English and German-to-English translation in the second experiment described in Section 4, the decoder did score about 6% higher for its translations from French, and the translations from German *were* of noticeably poorer quality. For example, our decoder translated the German sentence "In Tadschikistan wird damit kein einziges Problem gelöst" into ", Tajikistan is no problem is solved," whereas it translated the equivalent French sentence "Cette aide ne permettra de résoudre aucun problème au sein du Tadjikistan" into "This is not capable of solving the problem within Tajikistan." The "gold standard" English translation of this sentence was "As such, this will not solve any problems within Tajikistan," which is similar to the French translation.

We suspect that much of the discrepancy in performance between our translations between French and German and English can be accounted for by looking at characteristics of the languages being translated. While some of this discrepancy is likely caused by language and/or translation modeling errors, we suspect that much of it is actually caused by errors in decoding (see Section 5.1). One very important aspect of greedy decoding is the choice of initial state for the first iteration. As we discussed in 3.1, the algorithm we use forms an English "gloss" of the French (foreign) sentence that we wish to translate. Importantly, this gloss is aligned directly with the French sentence, which is to say that the first word in the gloss is aligned with the first word in the French sentence, and so on. If the most probable translation is aligned similarly, this strategy works well; if, on the other hand, the most probable translation is aligned in a substantially different manner, it is likely that the greedy decoding algorithm will fail to make the "jump" to the optimal solution and will instead get caught at a local maximum.

Although the word order of French and German sentences is generally not that far off of the word order of their English counterparts, both languages do have features that may give our greedy decoder a hard time. We suspect that the word order discrepancies between German and English are more serious than the discrepancies between French and English, and the results of our second experiment seem to support this conclusion. In the following two sections, we discuss some of the word order issues that arise in French-to-English and German-to-English translation.

5.2.1. *French-to-English alignment issues*

Most (but not all) French adjectives come *after* the noun that they modify. Thus, every (successful) greedy decoding of a French sentence containing adjectives will usually involve one or more uses of the **swap-segments** mutation strategy (see Section 3.2). In addition, pronouns in French will often appear *before* the verb when they serve as the direct or indirect object of a sentence. This issue also generally requires that **swap-segments** be used during greedy decoding for the sentences it impacts.

Although in the interest of brevity we will not discuss all of the discrepancies between French and English word orderings, it appears to us that most, if not all, such discrepancies can be remedied using a single mutation operation—most often **swap-segments**. As we will see in the next section, this is sometimes not that case in German to English translation.

5.2.2. *German-to-English alignment issues*

The word order issue in German-to-English translation is a serious one. In general, German verbs that are not in the present tense appear in a very different place than they would in equivalent English sentences. For example, one German equivalent of "I ate at the restaurant on Monday" is "Ich habe an der Gaststätte am Montag gegessen," where *gegessen* is essentially the German translation of *ate*. The English gloss of this sentence might be "I have on the restaurant on Monday ate," which is likely to be assigned low probability by the language model. We suspect that our decoder has a hard time properly translating sentences like this because it generally takes more than one mutation operation to see an improvement in translation probability. For example, if the **join-words** operation were used to replace *have* with *ate* in the English gloss above, we might obtain the sentence "I ate on the restaurant on Monday." While this sentence still has problems, it will likely be gauged as much more probable by the language model; *however*, because we have moved the word *ate* so far from the diagonal, the translation (alignment) model may penalize the new sentence enough that it is not seen as an improvement over the gloss. On the other hand, if **translate-one-or-two-words** is used to translate *on* to *at* in the gloss (obtaining the new sentence "I have at the restaurant on Monday ate") there will likely be no substantial improvement to either the language or translation model probability. If both of these changes had been made simultaneously, there *would* be a substantial improvement in probability, but the greedy decoder only performs one operation per iteration, so such a jump is impossible.

In Section 5.2 we mentioned that our decoder did a very poor job translating the German sentence "In Tadschikistan wird damit kein einziges Problem gelöst." This sentence exhibits the problem we have just described: the verb is not in the present tense, so it appears at the end of the sentence. The bizarre translation that our decoder produced (", Tajikistan is no problem is solved,") is evidence for our suspicion that this word ordering issue is a significant problem for the greedy decoding strategy that we employ. There are certainly other discrepancies between German and English word order that (in the interest of brevity) we will not discuss, but we feel that this one is likely one of the most problematic.

6. Conclusion

We have presented a modified greedy decoder for IBM Model 2 that was based on the greedy decoder presented in [4]. According to a rating scale that we devised, our decoder—in combination with our language and translation models—obtained a translation accuracy of about 55% for French-to-English translation and of about 50% for German-to-English translation. We have discussed language-specific issues that may have contributed to the discrepancy in these accuracies, and we have argued that it is likely mainly our decoder itself—rather than our language and translation models—that interacted negatively with these language features.

7. Acknowledgements

We wish to thank Prof. Chris Manning for his helpful comments and suggestions regarding our project proposal and for pointing us toward the parallel French-German-English text we used in our experiments.

8. References

[1] Brown, P., Lai, J., and Mercer, R. 1991. *Aligning sentences in parallel corpora*. In Proceedings, 29th Annual Meeting of the Association for Computational Linguistics. Berkeley CA, June 1991, 169-176.

[2] Brown, P., Della Pietra, S., Della Pietra, V., Mercer, R. 1993. *The mathematics of statistical machine translation: parameter estimation*. Computational Linguistics, 19(2).

[3] Brown P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Lai, J., and Mercer, R.. 1995. *Method and system for natural language translation*. U.S. Patent 5,477,451.

[4] Germann, U., Jahr, M., Knight, K., Marcu, D. and Yamada, K. 2003. *Fast decoding and optimal decoding for machine translation*. Artificial Intelligence, 2003.

[5] Jahr, Michael E. *Multistack decoding in statistical machine translation*. Honors Thesis, Symbolic Systems Program, Stanford University. June 2001.

[6] Koehn, Philipp. *Europarl: a multilingual corpus for evaluation of machine translation*. Unpublished draft.

[7] Wang, Y. and Waibel, A. 1997. *Decoding algorithm in statistical machine translation*. In Proc. ACL.