Statistical Machine Translation Using IBM Translation Models

Richard Samuelson

March 31, 2020

1 Introduction

The IBM translation models are a family of statistical machine translation algorithms that date to the late 1980s. The models were created as part of IBM's "Candide" project, which aimed to automatically translate French to English. As an excercise in theory, I decided to try implementing the models myself.

The IBM models work by estimating the conditional probability that a given sentence in one language is the translation of a given sentence in the other. Learning a distribution like this is difficult because if its sparsity. An English sentence is just a sequence of English words, and set of all such sequences is vast even if you limit their length. Given a French sentence, a very small portion of this vastness will be decent English translations of that sentence. A very small portion of this vastness will be decent English at all: most sequences of English words are ungrammatical.

This problem is compounded by lack of a decent metric. If one could give strings a notion of distance that put similar translations close together, than a translator could point more-or-less at the right spot and do pretty well. But this is not the case: most vectorizations of strings represent good English as rare islands admist a sea of nonsense. If a translator misses even a little bit, grammar and sense can see the door.

The translation models produced by the Candide team did not solve this problem. The team's approach was to draw a large sample of aligned French and English strings, and then search within small range of reasonable distributions for the one with which the sample best agreed. The distributions generated by this technique generally assigned high probabilites to good translations of sentences, but they also assigned high probabilities to ungrammatical strings with the right words in approximately the right order.

To ensure sensible translations, the Candide team combined their translation models with prexisting "language models," which had been developed to aid in speech transcription. This strategy is called the "noisy channel" approach to translation. It works as follows: rather than directly estimating the probability

$$\mathbb{P}(\mathbf{e}|\mathbf{f})$$

that an English string $\mathbf{e}$ is a translation of a French string $\mathbf{f}$, one chooses instead to estimate the product

$$\mathbb{P}(\mathbf{e})\mathbb{P}(\mathbf{f}|\mathbf{e}),$$

where $\mathbb{P}(\mathbf{e})$ is the probability that $\mathbf{e}$ might show up in a random English document, and $\mathbb{P}(\mathbf{f}|\mathbf{e})$ is the probability that $\mathbf{f}$ is a translation of $\mathbf{e}$. Recall that per Bayes' Theorem, we have the equality

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

Therefore, the English string $\mathbf{e}$ that maximizes $\mathbb{P}(\mathbf{e}|\mathbf{f})$ is the same English string maximizing $\mathbb{P}(\mathbf{e})\mathbb{P}(\mathbf{f}|\mathbf{e})$.

In a 1993 article, the Candide team playfully described their reasoning as follows:

> A string of English words, $\mathbf{e}$, can be translated into a string of French words in many different ways. Often, knowing the broader context in which $\mathbf{e}$ occurs may serve to winnow the field of acceptable French translations, but even so, many acceptable translations will remain; the choice among them is largely a matter of taste. In statistical translation, we take the view than every French string, $\mathbf{f}$, is a possible translation of $\mathbf{e}$. We assign to every pair of strings $(\mathbf{e}, \mathbf{f})$ a number $\mathbb{P}(\mathbf{e}|\mathbf{f})$, which we interpret as the probability that a translator, when presented with $\mathbf{e}$, will produce $\mathbf{f}$ as his translation. We further take the view that when a native speaker of French produces a string of French words, he has actually concieved of a string of English words, which he translated mentally. Given a French string $\mathbf{f}$, the job of our translation sustem is to find the string $\mathbf{e}$ that the native speaker had in mind then he produced $\mathbf{f}$. We minimize our chance of error by choosing that string $\hat{\mathbf{e}}$ for which $\mathbb{P}(\mathbf{e}|\mathbf{f})$ is greatest. (Brown et. al. 1993)

Practically, the noisy channel approach allows one to shift responsibility for producing grammatical English strings onto the distribution $\mathbb{P}(\mathbf{e})$. That distribution is estimated by an English "language model," which is chosen to assign very low probabilites to ungrammatical strings. This frees up the "translation model," which estimates $\mathbb{P}(\mathbf{f}|\mathbf{e})$, to focus on choosing the right words in the right number in approximately the right order. In this project, I estimate $\mathbb{P}(\mathbf{e})$ with a bigram language model and $\mathbb{P}(\mathbf{f}|\mathbf{e})$ with IBM models one and two. I maximize the product $\mathbb{P}(\mathbf{e})\mathbb{P}(\mathbf{f}|\mathbf{e})$ with a greedy decoder, which starts with a best guess and searches neighboring English strings for better and better translations.

In my choice of decoder, I depart from the history. The Candide team used a ngram language model, as I do, but they used a "beam-search" method to choose a best translation. I decided to use a greedy decoder instead because it lets you watch the translation process as it occurs, and I thought that would be pretty cool.

2 IBM Model One

2.1 Model One Independence Assumptions

The Candide team's simplest translation model assumes that the probability $\mathbb{P}(\mathbf{f}|\mathbf{e})$ is a conditional distribution of the form

$$\mathbb{P}(\mathbf{f}|\mathbf{e}) = \epsilon(m|l) \prod_{f \in \mathcal{F}} \left(\sum_{e \in \mathcal{E}} c(e, \mathbf{e}) t(f|e) \right)^{c(f,\mathbf{f})},$$

where

- the set $\mathcal{E}$ consists of all English words,
- the set $\mathcal{F}$ consists of all French words,
- the function $c(w, \mathbf{w})$ counts the appearences of a word w in a string $\mathbf{w}$,
- the conditional distribution $t(f|e)$ estimates the probability that an English word e translates to a French word f, and
- the conditional distribution $\epsilon(m|l)$ estimates the probability that an English string of length l translates to a French string of length m.

This model can be justified by the following assumptions. We assume that the probability that an English string $\mathbf{e}$ translates to a French string containing the French word f is a conditional distibution of the form

$$\mathbb{P}(f|\mathbf{e}) = \sum_{e \in \mathcal{E}} c(e, \mathbf{e}) t(f|e).$$

We then imagine that translation of $\mathbf{e}$ into $\mathbf{f}$ is a sequence of independent trials. First, a length m is chosen according to the conditional distribution $\epsilon(m|l)$. Next, m French words are independently drawn one-by-one per the conditional distribution $\mathbb{P}(f|\mathbf{e})$.

Every distribution $\mathbb{P}(\mathbf{f}|\mathbf{e})$ of this form is uniquely determined by the distributions ϵ and t. Our choice of ϵ and t is informed by a sample S we draw of n translated strings $(\mathbf{e}, \mathbf{f})$. Given such a sample, we choose ϵ and t to maximize the cross-entropy function

$$\frac{1}{n} \sum_{(\mathbf{e},\mathbf{f}) \in S} \ln(\mathbb{P}(\mathbf{f}|\mathbf{e})) = \frac{1}{n} \sum_{(\mathbf{e},\mathbf{f}) \in S} \left(\ln(\epsilon(m|l)) + \sum_{f \in \mathcal{F}} c(f, \mathbf{f}) \ln \left(\sum_{e \in \mathcal{E}} c(e, \mathbf{e}) t(f|e) \right) \right).$$

2.2 Cross Entropy

For those unfamiliar with cross-entropy, this may seem an odd thing to maximize. We justify here it as follows. We can define a distribution $\hat{\mu}$ over english strings with the formula

$$\hat{\mu}(\mathbf{e}) = \frac{c(\mathbf{e}, S)}{n},$$

where $c(\mathbf{e}, S)$ is the number of times the string $\mathbf{e}$ appears in the sample S. We will call $\hat{\mu}$ the empirical marginal distribution of our sample S, and we will think of it as a very simple English language model. It assigns low probabilities to English strings that appear in S, and it assigns no probability at all to strings that do not appear in S.

Similarly, we can define a conditional distribution $\hat{K}$ from the set of all English strings to the set of all French strings by

$$\hat{K}(\mathbf{f}|\mathbf{e}) = \frac{c((\mathbf{e}, \mathbf{f}), S)}{c(\mathbf{e}, S)},$$

where $c((\mathbf{e}, \mathbf{f}), S)$ is the number of times the pair $(\mathbf{e}, \mathbf{f})$ appears in the sample S. We wll call $\hat{K}$ the empirical kernel of S, and we will think of $\hat{K}$ as a very simple English-to-French translation model. Given an english string $\mathbf{e}$ in the sample S, $\hat{K}$ assigns probability 1 to its translation and probability 0 to all other French strings. Given an english string $\mathbf{e}$ that does not appear in our sample, the formula fails to define $\hat{K}$, and we can define it however we like so long as $\hat{K}$ remains a conditional probability distribution.

Though the empirical kernel $\hat{K}$ is a very simple translation model, it is, in fact, all that our sample tells us about the relationship between French and English. We will choose the parameters t and ϵ so that the resulting conditional distribution is as close as possible to the empirical kernel.

The measure of distance we use is Kullback-Liebler divergence. KL divergence is a convex, positive-definite function of two distributions. It is not symmetric, and it does not satsify the triangle inequality, so it is not a metric but merely a "divergence." However, it does upper bound the total variation metric per Pinsker's inequality.

Given two distributions μ, ν over some set, with μ absolutely continuous with respect to ν, the KL divergence $D(\mu\|\nu)$ is given by

$$D(\mu\|\nu) = \int \mathrm{d}\mu \ln\left(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}\right),$$

where $\frac{\mathrm{d}\mu}{\mathrm{d}\nu}$ is the Radon-Nikodym derivative of μ with respect to ν. In our case, we seek t and ϵ that minimize the divergence

$$D\left(\hat{K}(\cdot|\mathbf{e}) \,\|\, \mathbb{P}(\cdot|\mathbf{e})\right)$$

for all $\mathbf{e}$. Of course, we should prioritize minimizing this expression over strings $\mathbf{e}$ that are well formed. For this reason, we weight each divergence $D\left(\hat{K}(\cdot|\mathbf{e}) \,\|\, \mathbb{P}(\cdot|\mathbf{e})\right)$ in our loss function by the probability that the string $\mathbf{e}$ is well-formed, which we approximate by the empirical marginal distribution $\hat{\mu}$.

We choose, then, to mininize the expression

$$\int \hat{\mu}(\mathrm{d}\mathbf{e})\, \mathrm{D}(\hat{\mathrm{K}}\,(\cdot|\mathbf{e}) \,\|\, \mathbb{P}(\cdot|\mathbf{e})) = \sum_{\mathbf{e}} \hat{\mu}(\mathbf{e}) \sum_{\mathbf{f}} \hat{K}(\mathbf{f}|\mathbf{e}) \ln \left(\frac{\hat{K}(\mathbf{f}|\mathbf{e})}{\mathbb{P}(\mathbf{f}|\mathbf{e})} \right)$$
$$= \frac{1}{n} \sum_{(\mathbf{e},\mathbf{f}) \in S} \ln \left(\frac{\hat{K}(\mathbf{f}|\mathbf{e})}{\mathbb{P}(\mathbf{f}|\mathbf{e})} \right),$$

the minimization of which is equivalent to maximizing the cross-entropy shown above. Our choice of $\hat{\mu}$ as a language model is justified by the fact that $\hat{K}(\cdot|\mathbf{e})$ tells us nothing new when $\mathbf{e}$ is not in the sample S, since in that case we define the empirical kernel ourselves.

2.3 EM Algorithm

The careful reader has already noticed that when maximizing the cross-entropy over t and ϵ, the two variables can be solved seperately. The optimal distribution ϵ is merely the empirical kernel $\hat{\epsilon}$ defined by

$$\hat{\epsilon}(m|l) = \frac{c((l,m),S)}{c(l,S)},$$

where $c(l,S)$ is the number of English strings in the sample S of length l, and $c((l,m),S)$ is the number of pairs $(\mathbf{e},\mathbf{f}) \in S$ with respective lengths l and m.

Choosing t is somewhat trickier. The Candide team used the update rule

$$\frac{t(f|e)}{n} \sum_{(\mathbf{e},\mathbf{f}) \in S} \frac{c(e,\mathbf{e})c(f,\mathbf{f})}{\sum_{e \in \mathcal{E}} c(e,\mathbf{e})t(f|e)} \mapsto t(f|e),$$

which eventually converges to an optimal solution.