Parameter estimation for Statistical Machine Translation

Amrutha Varshini Ramesh

ECE 531 Final Project

Keywords: SMT, Expectation maximization, Machine Translation

1. Introduction

Machine Translation(MT) is a classical problem in language processing, where the task is to translate sentences from language l_1 to l_2. A simple translator that substitutes a word in l_1 to a word in l_2, would not be sufficient for a good translation. A good translator must handle the changes in the language structure, identify the compound words, two-word verbs and words with multiple meanings and provide the correct translation. Considering all the above mentioned challenges, the problems in machine learning can be broadly classified into 2 types.

- Insertion/Deletion : One of the first problems that a Machine Translation system is expected to handle is the fact that sentences in l_1 need not have the same length as sentences in l_2. Assuming the system translates sentences one at a time, it is required to predict, for a given sentence s_1 from l_1, what would be the length of the translated sentence(number of words) in s_2 from l_2.
- Mis-alignment: A much harder problem is to find a syntactic match between sentences. This problem is called alignment. More specifically, given words from s_1(s_1 is known) and s_2, we are required to align these words in such a way that the translation is meaningful and the generated sentence s_2 is syntactically correct at the same time.

2. Notation

Through out the report, we use the following notation

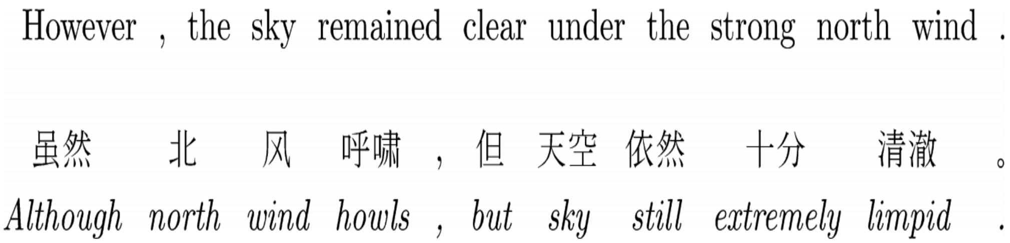
- s_1 is the source/foreign sentence of length l_{s_1}
- s_2 is the target sentence of length l_{s_2}
- ws_i is a word in s_1
- wt_j is a word in s_2

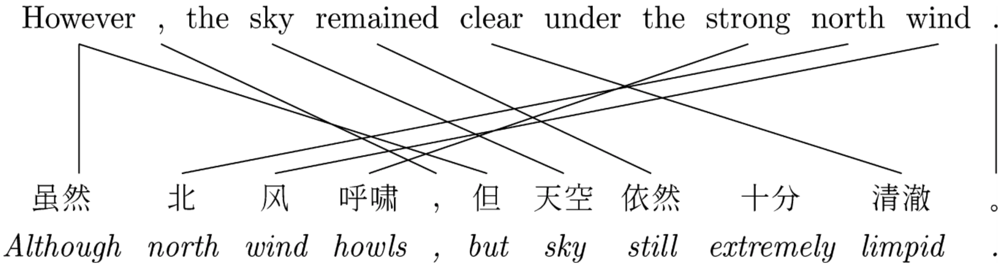
3. Machine translation

The goal of MT is to translate s_1 to s_2, where s_2 is the translationally equivalent sentence in the target language l_2. Considering the above mentioned challenges, the solution to the machine translation problem comprises of the following steps

- Translation of words from source language to target language
- Alignment of the translated words to translationally equivalent sentence

An example of translationally equivalent sentences are


However , the sky remained clear under the strong north wind .
虽然 北 风 呼啸 ， 但 天空 依然 十分 清澈 。
Although north wind howls , but sky still extremely limpid .


However , the sky remained clear under the strong north wind .
虽然 北 风 呼啸 ， 但 天空 依然 十分 清澈 。
Although north wind howls , but sky still extremely limpid .

- The translation of ws_I can have more than 1 equivalent $wt_J \in l_2$, without the loss of meaning. The decision of choosing the right wt_J depends on the context. The devised algorithm should be able to analyse the context and choose the right word.

4. Approaches to MT

There are 2 major categories of technologies that approaches to solve the MT problems

- Rule-based Machine translation(RBMT)
- Corpus-based Machine translation(CBMT)

4.1. Rule-based Machine translation

RBMT is the classical approach to machine translation, built on dictionaries and linguistic rules of the source and target languages. The linguistic rules are usually manually created based on the morphological, syntactic and semantic map between the two languages. Thus it is a time consuming and labor intensive knowledge acquisition problem. The three different types of RBMT systems are

- Direct approaches
- Transfer-based approaches
- Interlingua-based approaches

In the direct approach, each word is translated directly from l_1 to l_2. The other aspects of translations like variations in the meaning, differences in the sentence structure are not taken into consideration, leading to significant error rates. In the transfer-based approaches, the morphological and the syntactical variations are taken into consideration. The grammatical structure of s_1 is analyzed and it is transferred to an intermediate representation, from which the translation to s_2 is made. In the interlingua-based approaches, like the transfer-based approach, s_1 is transferred to an intermediate representation called interlingua, but this representation is not related to the l_1 structure, it is language neutral. The translation to the target language is then performed from interlingua representation.

4.2. Corpus-based Machine translation

CBMT is the most used approach to the translation problem today. The bilingual mapped corpora, that is, a large dataset of already translated examples, is the basis of CBMT. This data-driven approach is broadly classified into two types,

- Statistical Machine translation(SMT)
- Example based Machine translation(EBMT)

. **EBMT** is an approach where the machine translation is performed based on the idea of analogy. When an unseen sentence is provided to the EBMT, the sentence is divided into phrases. The corpus is searched for similar phrases, which are identified by the measure of distance of the meaning. Therefore the EBMT approach is divided into 3 tasks, source sentence decomposition into phrases, matching the phrases with the translation examples and selecting similar ones, adaptation and recombination of the target translated sentence.

. **SMT** is the most popular and currently dominating approach in the machine translation research, where the problem is solved by constructing statistical models, whose parameters are estimated by analyzing the bilingual text corpora. When an unseen sentence is provided to the trained model, it generates a translated sentence in the target language based on the model's training from the corpus.
This report targets to survey several models/algorithms that the SMT uses for machine translation.

5. Statistical machine translation

The idea behind the statistical machine translation approach is modeling the probability distribution $p(s_2|s_1)$, that is probability of the sentence in the target language l_2, when the sentence s_1 in the source language l_1 is seen.
According to Bayes' theorem

$$p(s_2|s_1) = p(s_1|s_2) * p(s_2)$$

where,
$p(s_1|s_2)$ is the probability that the source sentence s_1 is the translation of target sentence s_2
$p(s_2)$ is the target language l_2 model.
Now the problem of machine translation is finding the target sentence s_2 that maximizes the probability

$$\operatorname*{argmax}_{s_2 \in l_2} p(s_2|s_1) = \operatorname*{argmax}_{s_2 \in l_2} p(s_1|s_2) * p(s_2) \tag{1}$$

In equation 1, $p(s_1|s_2)$ is called translation model, and $p(s_2)$ is called language model.

- The translation model, models the word to word translation and alignment of the translated words.

- The language model, models the correctness of the target sentence, it gives us how possible is the target sentence s_2, under the rules of the target language.

Ideally, the entire search space, that is, all the sentences in l_2 has to searched to find the s_2 that maximizes the probability, which is out of the question. So, good approximations for $p(s_1|s_2)$ and $p(s_2)$, that gives acceptable quality of translation has to be constructed.

5.1. Translation model

The translation model, models the relationship between source and target sentences. This comprises of the word to word translation of the given sentence and the alignment of the translated sentence. The approximation to the language model is constructed by estimating the parameters of the translation model, by learning them from the dataset. In our model, the parameter is alignment, a.
Formally, estimating $p(s_1|s_2)$ comprises of two stages

- Stage 1: Word-by-word translation - $p(wt_j|ws_i)$, Given $ws_i \in s_1$, finding the best $wt_j \in s_2$.

- Stage 2: Given the word-by-word translations, finding the best alignment between $ws_i's$ and $wt_j's$.

No initial information on both the stages are available to begin the estimation. In our model, we solve this problem by using Expectation Maximization algorithm.

6. Expectation Maximization

To obtain the translation, ideally, one must count the number of times wt_j is assigned to ws_i. Recall that, it cannot be observed because we do not have information about both stage 1 and stage 2. The Expectation Maximization algorithm computes the expected number of times wt_j is aligned to ws_i, for an initial word-to-word translation. With the learned alignment, it computes the maximum likelihood function of word-to-word translation.

6.1. Expectation

The expectation step learns the best alignment function a that maps words ws_i to wt_j. To perform this step, one would require the probability of the word-to-word translation, $p(wt_n|ws_n)$. Assuming it as uniform distribution, the alignment a is learnt.

$$p(a|s_1, s_2) = \frac{p(s_2, a|s_1)}{p(s_2|s_1)} \tag{2}$$

$$\begin{aligned} p(s_2|s_1) &= \sum_a p(s_2, a|s_1) \\ &= \frac{\prod_{j=1}^{l_{s_2}} \sum_{i=0}^{l_{s_1}} p(wt_j|ws_i)}{(l_{s_1}+1)^{l_{s_2}}} \end{aligned} \tag{3}$$

$$p(s_2, a|s_1) = \frac{\prod_{j=1}^{l_{s_2}} p(wt_j|ws_{k(j)})}{(l_{s_1}+1)^{l_{s_2}}} \tag{4}$$

Combining 3 and 4, we get the alignment probability distribution as

$$p(a|s_1, s_2) = \prod_{j=1}^{l_{s_2}} \frac{p(wt_j|ws_{k(j)})}{\sum_{i=0}^{l_{s_1}} p(wt_j|ws_i)} \tag{5}$$

Here $k(j)$ is an iterative function, that iterates from 0 to l_{s_1}, summing the product $\prod_{j=1}^{l_{s_2}} p(wt_j|ws_{k(j)})$ of each iteration. Intuitively, sum of the probability of those alignments that contain the decision that we are interested in, is divided by the sum of the probability of all possible alignments.

6.2. Maximization

This is the problem of finding the wt_j that maximizes the probablity $p(wt_j|ws_i)$. This is now a simple counting problem, given the alignment probabilities from the Expectation step. The probabilities of $p(a|s_1, s_2)$ for which there is an alignment between wt_j and ws_i are summed and normalized.

$$wt_j = \operatorname*{argmax}_{wt \in l_2} p(wt_j|ws_i)$$

The Expectation and the maximization steps are repeated until convergence.

7. Algorithm

Algorithm 1 EM Algorithm for SMT

```
Input: Set of sentence pairs $(s_2, s_1)$
Output: Translation prob. $p(wt_j|ws_i)$

initialize $p(wt|ws)$ uniformly
// initialize
while not converged do
    $\text{count}(wt_j|ws_i) = 0$ for all $i, j$
    $\text{total}(ws_i) = 0$ for all $j$
    for all sentence pairs $(s_1, s_2)$ do
        // Compute normalization
        for all words $wt_j$ in $s_2$ do
            $\text{sumprob}(wt_j) = 0$
            for all words $ws_i$ in $s_1$ do
                $\text{sumprob}(wt_j) += p(wt_j|ws_i)$
            end for
        end for
    end for
    // Counting
    for all words $wt_j$ in $s_2$ do
        for all words $ws_i$ in $s_1$ do
            $\text{count}(wt_j|ws_i) += \frac{p(wt_j|ws_i)}{\text{sumprob}(wt_j)}$
            $\text{total}(ws_i) += \frac{p(wt_j|ws_i)}{\text{sumprob}(wt_j)}$
        end for
    end for
    // estimate probabilities
    for all target words $wt_j$ do
        for all source words $ws_i$ do
            $p(wt_j|ws_i) = \frac{\text{count}(wt_j|ws_i)}{\text{total}(ws_i)}$
        end for
    end for
end while
```

8. Experiment

For evaluating the performance of the SMT algorithm(EM), we tested it on two standard Machine Translation datasets from [7]. The first dataset contains translations from English to French and the second one translates from English to Spanish. Dataset statistics are given below.

English-French Europarl dataset

# sentences	50000
# words in the vocabulary	37000
Avg. sent. length	6

English-Spanish Europarl dataset

# sentences	50000
# words in the vocabulary	74000
Avg. sent. length	5

Table 1: Dataset descriptions

8.1. Evaluation

The performance of the EM based SMT algorithm was evaluated using standard machine learning metrics such as **Precision**, **Recall** and **F-1 score**. Precision is a statistic that measures how accurate, the guesses of the algorithms are. Recall measures the fraction of guesses that the algorithm correctly identified. F-1 score is a measure that combines precision and recall. Their formulations are given below.

$$\texttt{Precision} = \frac{\texttt{True Positive}}{\texttt{True positive} + \texttt{False Positive}} \quad \texttt{Recall} = \frac{\texttt{True positive}}{\texttt{True Positive } + \texttt{False Negative}}$$

$$\texttt{F1-score} = \frac{2 * \texttt{Precision} * \texttt{Recall}}{\texttt{Precision} + \texttt{Recall}}$$

In the context of Machine Translation, if X is the set of source sentences and Y is the set of target sentences, the precision and Recall definitions given above are equivalent to the following definitions.

$$\texttt{Precision} = \frac{X \cap Y}{X} \qquad \texttt{Recall} = \frac{X \cap Y}{Y}$$

With the above definitions, the results of the SMT algorithm on the English-French & English-Spanish datasets are given below.

Table 2: Performance of EM-SMT on English to Spanish Europarl dataset

Precision	0.596
Recall	0.487
F1 score	0.536

Table 3: Performance of EM-SMT on English to French Europarl dataset

Precision	0.613
Recall	0.53
F1 score	0.5685

Some example translated sentences. Here we present some results, that were hand-picked to show-case the ability and inefficiencies of our current Machine Translation system. For reference, we also compared the results with one of the more sophisticated Translation Systems available - Google's Translation Service.

- s_1: De région en région, les situations sont très, très différentes
  s_2: The situation varies to an enormous degree throughout the regions.
  Google translate: From region to region, situations are very, very different

- s_1: Je ne le crois pas.
  s_2: I do not believe so
  Google translate: I do not believe that

- s_1: La procédure a connu quelques ralentissements au niveau du Conseil, ralentissements dus notamment à des divergences de vues concernant l' accord sur la libre circulation des personnes
  s_2: The procedure has undergone some delays in the Council due, in particular, to differences of opinion regarding the free movement of persons.
  Google translate: The procedure has seen some slowdowns at Council level, slowdowns due in particular to differences of opinion concerning the agreement on the free movement of persons

8.2. Discussion

From the results, we found that the SMT translation system does well on short sentences with very few connecting words or verbs inbetween. However on larger sentences, the system suffers to find the right translation and alignment. This is probably one of the drawbacks of the EM procedure. Since the parameter estimation is performed in an alternating manner, poor optimization of one set of variables affect the results of the other.

9. Proposed Improvements to the current model

Just to recall, in the Machine translation model introduced in the previous section had two sub-models.

1. *An Translation model*, given by $p(wt_j|ws_i)$, which captures the word by word translation between words wt_j and ws_i in sentences s_2 and s_1 respectively.
2. *An Alignment model*, given by $p(a|s_1, s_2)$, that gives the probability of alignment, given the sentences s_1 and s_1.

There are many improvements in terms of both modeling and optimization, that one could propose. Perhaps one of the simplest extension that one could propose is to make the Translation model richer. Recall that the tranining procedure for learning the parameters of the model(s) was done using EM and the parameters of the Translation model are optimized during the *Maximization* step. Here, we update the translation probabilities $p(wt_j|ws_i)$, by merely finding support for the joint occurance of wt_j and ws_i. This so-called "*generative model*" maximizes the joint probability of pairs of words. Though effective, humans use various linguistic cues to make accurate translations.

1. Consider the following sentence for instance. `He went to school yesterday`. Here, the fact that `He` is a *pronoun* and it is *Capitalized*, could all be useful in disambiguating what `He` translates into in the target language. In short, the syntactic information associated with words could be useful. This information is usually given by *Part-of-speech* tags, which can be estimate using a different model.
2. Let's look at another sentence. "`Brown won the chess championship emphatically`". Here, it is useful to know that the word `Brown` refers to a person and not a color. This kind of *Sense Disambiguation* is usually given by solving another linguistic task called *Named entity recognition*, for which numerous algorithms are available.

These two examples tell us that additional structural / semantic information about the words in the sentence, could give us clues about what the word would mean in the target language. In other words, if for a given source language word ws_i, there are multiple target translations, a "richer" word model could help us disambiguate the target translation better. All this is done by enriching the features space of the words. For instance, in case of the first example sentence, we could come up with the following feature representation for the first word, `He`.

{ *Is_Capitalized* : **Yes**, *Is_first_word*: **Yes**, *Is_verb* : **No** }

However, the question now is, whether the current language model enables us to encode this information. The short answer is *No*. Generative models, introduced in the previous section, are in general, not amenable to arbitrary feature expansion. Typically, problems of such kind are handled by a class of sophisticated *Discriminative probabilites*, which aim to estimate the conditional probability of $p(w_{t_j}|w_{s_i})$ directly, instead of modeling the joint probability. One such model that is frequently used in modeling sequentially structured data is *Conditional Random Fields*.

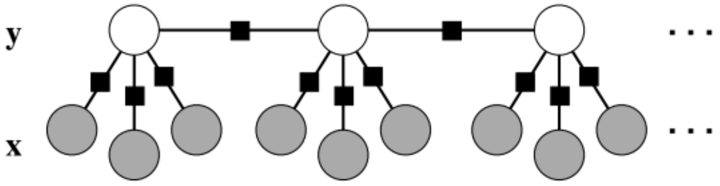
9.1. Conditional Random Fields

Conditional Random Fields(CRFs) belong to a class of statistical models called *Directed Graphical Models*. Here, the random variables in question are assumed to have structured relationship, commonly represented by a graph. One of the attractive features of CRFs is that, it allows us to express arbitrary features on the random variables. Mathematially, CRFs have the following form.

$$p(y|x) = \frac{1}{Z(x)} \exp \left\{ \sum_{k=1}^{K} \theta_k f_k(y, \mathbf{x}) \right\}$$

where $f_k(y, \mathbf{x})$ may be the set of linguistic features we want encode about ws_i and wt_j and $Z(x)$ is a normalization factor, commonly referred to as the *Partition function.* A simple linear-chain CRF is given below. For a detailed introduction to CRFs, the reader is referred to [11].

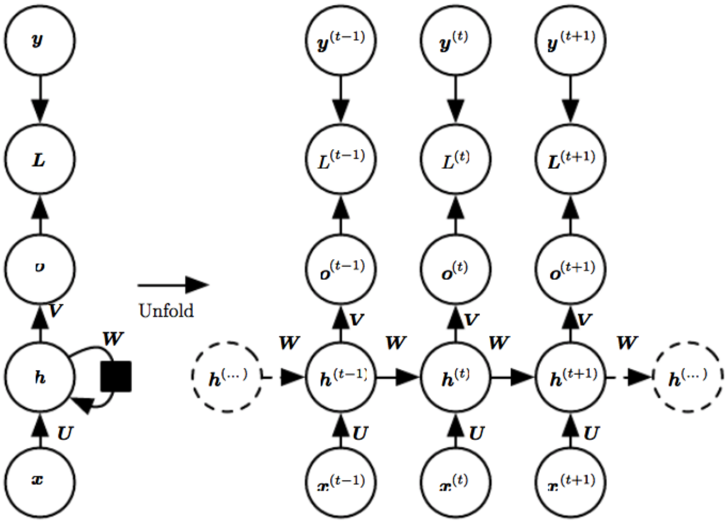
The main advantage of CRFs is also it's greatest weakness. The blown up feature space, as a result of adding arbitrary features, makes training a CRF model extremely hard and inference almost always intractable. However, a more severe problem for a practioner, is obtaining training data for CRFs. Usually, data collection for CRFs involve a combination of crowd-sourcing, bootstrapping and manual correction. This makes CRF training impractical for large datasets. Having said that, on availability of such datasets, CRFs are probably one of the best sequential models available. Subsequently, CRFs also fit the bill as an excellent alignment model. [2] show how to use CRFs for word alignments.



9.2. Neural network based translation

The problem of hand-tuning feature representations for learning algorithms is partly alleviated by what are called *Deep Neural Network(DNN)* models, which have the ability to automatically learn feature representations, suitable for the task at hand. They do this using hierarchical features that capture subtle patterns in the input, suitably guided by a loss function, specifically designed for the task at hand.

Recurrant Neural Networks.. Sequential learning problems in neural networks are typically handled by incorporating a temporal dimension to the conventional neural network, introduced earlier. Typical applications of such a model would be to model distributions of words over time in text/speech, protein sequences in a DNA molecule etc. A snapshot of a simple RNN architecture is given below. [4] gives a comprehensive introduction to RNNs.



Naturally, RNN's are a natural model for capturing relationships between words over time. For our task, we simply extend the existing word-based model to a phrase based model. Specifically, features of a word at position i, would now be phrases, which are groups of words in the context of the given word w_i.

10. References

[1] Statistical-Machine-Translation/tp3.sujet.pdf at master · Mandarancio/Statistical-Machine-Translation · GitHub.

[2] Phil Blunsom and Trevor Cohn. Discriminative Word Alignment with Conditional Random Fields. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 65–72, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[3] Chris Callison-Burch. Machine translation: Word-based models and the EM algorithm Chris Callison-Burch Word-based translation models and EM. 2007.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[5] Andrej Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks, 2015.

[6] Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation.

[7] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

[8] Thomas Lavergne, Josep Maria Crego, Alexandre Allauzen, and François Yvon. From n-gram-based to CRF-based Translation Models. pages 542–553.

[9] Adam Lopez. A Survey of Statistical Machine Translation. 2007.

[10] I Dan Melamed, Ryan Green, and Joseph P Turian. Precision and recall of machine translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers-Volume 2*, pages 61–63. Association for Computational Linguistics, 2003.

[11] Charles Sutton and Andrew Mccallum. An Introduction to Conditional Random Fields. *Machine Learning*, 4(4):267–373, 2011.

[12] Chong Wu, Can Yang, Hongyu Zhao, and Ji Zhu. On the Convergence of the EM Algorithm: A Data-Adaptive Analysis. nov 2016.