CSME 206A – Natural Language & Speech Processing **Spring Semester**

Lecture 17: Statistical machine Translation-II

Lecturer: K.R. Chowdhary *: Professor of CS*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

17.1 Global lexical Selection

For global *lexical selection*, in contrast to the local approaches of associating target words to the source words, the target words are associated with the entire source sentence. There are two primary advantages in doing this: (1) For languages such as English-Hindi, there is a high degree of *word-reordering* (see Fig. **??**). Hence, it is difficult to obtain high-quality alignments, consequently there is a poor-quality local associations. (2) There are *lexico-syntactic* features of the source sentence (not necessarily a single source word) that trigger the presence of a target word in the target sentence. So, while translating a sentence from English to Hindi, the lexical form of the verb in Hindi language depends both on the English verb as well as the subject/object of the English sentence. This is one of the cases where global lexical selection is helpful in picking the right lexical items in the target language. For example, in the sentence "The girl whom we met yesterday sings well", the global subject-verb pairs are "girl-sings", while the local are "we-met". This phenomenon cannot be captured by a system which relies only on local associations, as is the case in a *phrase-based* SMT system.

In Fig. 17.1(a), (b), we can see two sentences where the translations of the words "plays" and "sings" have different word forms in the target language for feminine and masculine subjects. In the first sentence pair, we can see that "sings" translates in Hindi to form a feminine word form "gaatii" while in the second pair, "plays" appeared as the masculine form "kheltaa."

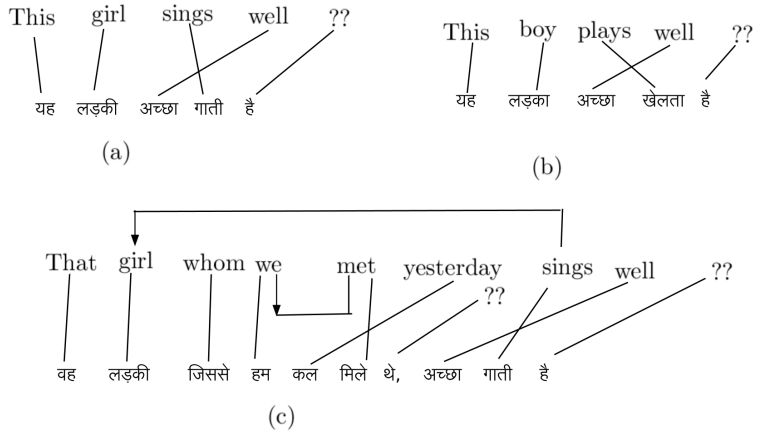

Figure 17.1: An example of Global Lexical Selection

Furthermore, there are cases where it is difficult to exactly associate a target word to a source word/phrase: (a) when sentence translations are not exact but are *paraphrases*, and (b) the target language does not have one lexical item to express the same concept that is expressed in the source word. The extensions of word

alignments to phrasal alignments attempt to address some of these situations in addition to alleviating the noise in word-level alignments.

As a consequence of the *global lexical selection approach*, we no longer have a tight association between source language words/phrases and target language words/phrases. The result of lexical selection is simply a *bag of words*/phrases in the target language and the target sentence has to be reconstructed using this bag of words/phrases.

The target words in the bag, however, might be enhanced with rich syntactic information that could aid in the reconstruction of the target sentence. This approach to lexical selection and sentence reconstruction has the potential to circumvent the limitations of word-alignment based methods for translation between significantly different word order languages.

17.2 Statistical Machine Translation

Statistical machine translation (SMT) is an approach to MT that is characterized by the use of machine learning methods. In less than two decades, SMT has come to dominate academic MT research, and has gained a share of the commercial MT market. Progress is rapid, and the state of the art is a moving target. However, as the field has matured, some common themes have emerged.

SMT treats translation as a machine learning problem. This means that we apply a learning algorithm to a large body of previously translated text, known variously as a *parallel corpus*, or *parallel text* (see Fig. 17.2). The learner is then able translate previously unseen sentences. With an SMT toolkit and enough enough parallel text, we can build an MT system for a new language pair within a very short period of time – perhaps as little as a day.

Assuming that we are given a sentence $\mathbf{s}$ in a source language, the text-to-text translation problem can be stated as finding its translation $\mathbf{t}$ in a target language. Using *statistical decision theory*, the best translation is given by the equation:

$$\hat{\mathbf{t}} = argmax_t P(\mathbf{t}|\mathbf{s}) \tag{17.1}$$

By using the Bayes theorem:

$$\hat{\mathbf{t}} = argmax_t P(\mathbf{s}|\mathbf{t}).P(\mathbf{t}) \tag{17.2}$$

Th equation (17.2) is interpreted as follows: The best translation is one in the target language which is correct sentence that conveys the meaning of the source sentence. The probability $P(\mathbf{t})$ represents the well-formedness of $\mathbf{t}$, called *language model* probability. For this, n-gram models are usually adopted. On the other hand, $P(\mathbf{s}|\mathbf{t})$ represents the relationship between the two sentences (the source and its translation). It should be of a high value if the source is a good translation of the target and of a low value otherwise. Note that the translation direction is inverted from what would be normally expected; correspondingly, the models built around this equation are often called *inverted translation models*. These models are based on the notion of alignment. It is interesting to note that if we had perfect models, the use of Equation 17.1 would suffice. Given that we have only approximations, the use of Equation 17.2 allows the language model to correct deficiencies in the translation model.

In practice all of these models are often combined into a *log-linear model*[1] for $P(\mathbf{t}|\mathbf{s})$:

[1]The log-linear is a mathematical model, which takes the form of a function, whose logarithm equals to a linear combination of the parameters of the model, which makes it possible to apply linear regression, including the multivariate linear regression.

$$\hat{\mathbf{t}} = argmax_t \left\{ \sum_{i=1}^{N} \lambda_i . \log f_i(\mathbf{t}, \mathbf{s}) \right\} \tag{17.3}$$

where $f_i(\mathbf{t}, \mathbf{s})$ can be a model for $P(\mathbf{s}|\mathbf{t})$, or a model for $P(\mathbf{t}|\mathbf{s})$, or a target language model for $P(\mathbf{t})$, or any model that represents an important feature for the translation. The N is the number of models (or features) and λ_i are the weights of the *log-linear* combination.

When using SFSTs (statistical FST), a different transformation can be used. These transducers have an implicit target language model (which can be obtained from the finite-state transducer by dropping the source symbols of each transition. Therefore, this separation is no longer needed. SFSTs model joint probability distributions, therefore, Equation (17.1) has to be rewritten as:

$$\hat{\mathbf{t}} = argmax_t P(\mathbf{s}, \mathbf{t}) \tag{17.4}$$

17.2.1 Components of SMT System

The Fig. 17.2 shows the process steps and resources of statistical machine translation system for natural language. The statistical translation systems strat by learning how various languages work. A system begin with minimal dictionary and language resources, like, Corpus, and then it is trained before it handles extensive translations. During the training, the system is fed with documents in source language and correct translation documents of the target language decoded by human. The system uses its existing resources to estimate the documents' translation, and a separate application compares these two, and results are output to improve the system's performance.

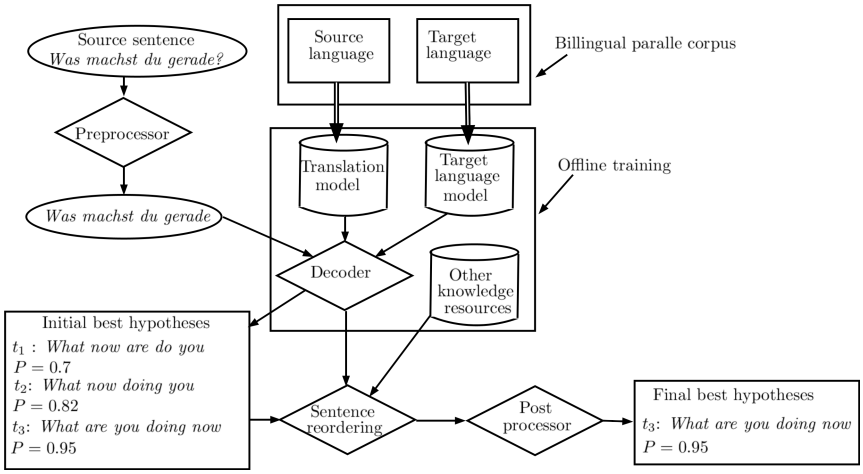

Figure 17.2: Process of statistical machine translation

As explained above, the statistical translation systems work by dividing the sentence into n-grams. Because, the n-gram studies the words as parts of phrases, instead of independent, it improves the accuracy of translation. The n-gram is usually bi-gram or tri-gram. The position of the n-gram in the sentence is also analyzed, which helps in re-ordering of the sentence.

The statistical tool kits were usually written in C++ and Java, due to speed requirements, but now Python is also common.

17.2.2 Statistical Alignment Models

The translation models discussed above deal with $P(\mathbf{s}|\mathbf{t})$ (Equation 17.2) are based on the concept of alignment between the components of a pair $(\mathbf{s}, \mathbf{t})$, hence they are called *statistical alignment models*. Formally, if the number of the source words in $\mathbf{s}$ is J and the number of target words in $\mathbf{t}$ is I, an alignment is a mapping function:

$$\mathbf{a} : \{1, ..., J\} \rightarrow \{0, ..., I\} \tag{17.5}$$

The image of j by $\mathbf{a}$ will be denoted as $\mathbf{a}_j$, in which the particular case $\mathbf{a}_j = 0$ means that the position j in $\mathbf{s}$ is not aligned with any position of $\mathbf{t}$. By introducing the alignment as a hidden variable in $P(\mathbf{s}|\mathbf{t})$,

$$P(\mathbf{s}|\mathbf{t}) = \sum_{a} P(\mathbf{s}, \mathbf{a}|\mathbf{t}) \tag{17.6}$$

The alignment that maximizes $P(\mathbf{s}, \mathbf{a}|\mathbf{t})$ is shown to be useful in practice for training and for searching.The summation over function variable $\mathbf{a}$ indicates that, that sentence in the target is best which has maximum number of mappings with the words in the source sentence, as well as, the probability of source sentence $\mathbf{s}$ is maximum.

In all these models, single words are taken into account. Moreover, in practice the summation operator is replaced with the maximization operator, which in turn reduces the contribution of each individual source word in generating a target word. On the other hand, modeling word sequences rather than single words in both the alignment and lexicon models cause significant improvement in translation quality. Both models are based on bilingual phrases (pairs of segments or word sequences) in which all words within the source-language phrase are aligned only to words of the target-language phrase and vice versa. Note that at least one word in the source-language phrase must be aligned to one word of the target-language phrase, that is, there are no empty phrases similar to the empty word of the word-based models. In addition, no gaps and no overlaps between phrases are allowed.

The statistical models for *Alignment Templates* (AT) are based on the bilingual phrases but they are generalized by replacing words with word classes and by storing the alignment information for each phrase pair. Formally, an AT Z is a triple $(S, T, \tilde{\mathbf{a}})$, where S and T are a source class sequence and a target class sequence, respectively, and $\tilde{\mathbf{a}}$ is an alignment from the set of positions in S to set of positions in T. Mapping of source and target words to bilingual word classes is automatically trained. This training method is *unsupervised clustering* method which partitions the source and target vocabularies, so that assigning words to classes is deterministic operation. It is also possible to employ the part-of-speech or semantic categories instead of unsupervised clustering method.

To arrive at translation model, we first perform a segmentation of the source and target sentences into K "blocks" $d_k = (i_k; b_k, j_k)(i_k \in \{1, ..., I\}$ and $j_k, b_k \in \{1, ..., J\}$ for $1 \leq k \leq K)$. For a given sentence pair $(\mathbf{s}_1^J, \mathbf{t}_1^I)$, the kth bilingual segment $(\tilde{\mathbf{s}}_k, \tilde{\mathbf{t}}_k)$ is $(\mathbf{s}_{b_{k-1}+1}^{j_k}, \mathbf{t}_{i_{k-1}+1}^{i_k})$. The alignment templates $Z_k = (S_k, T_k, \tilde{\mathbf{a}}_k)$ associated with the kth bilingual segment is: S_k the sequence of word classes in $\tilde{\mathbf{s}}_k$; T_k the sequence of word classes in $\tilde{\mathbf{t}}_k$, and $\tilde{\mathbf{a}}_k$ the alignment between positions in a source class sequence S and the positions in the target class sequence T.

For translating a given sentence $\mathbf{s}$ we use the following decision rule as an approximation to the following equation:

$$(\hat{I}, \hat{\mathbf{t}}_1^{\hat{I}}) = argmax_{I,t_1^I}\left\{ max_{K,d_1^K,\tilde{\mathbf{a}}_1^K} \ \log P_{AT}(\mathbf{s}_1^J, \mathbf{t}_1^I; d_1^K, \tilde{\mathbf{a}}_1^K) \right\} \tag{17.7}$$

The generation of the best translation for a given source sentence $\mathbf{s}$ is carried out by producing the target sentence in left-to-right order using the model of Equation 17.7. At each step of the generation algorithm a set of active hypotheses is maintained and chosen the one of them for extension. A word of the target language is then added to the chosen hypothesis and its costs get updated. This kind of generation fits well into a dynamic programming framework, as hypotheses which are indistinguishable by both language and translation models, and that have covered the same source positions, can be recombined. Because the dynamic programming search space grows exponentially with the size of the input, standard dynamic programming search is not used, instead the heuristic, called *beam-search* is used.

For generation of target sentence, a word graph that represents possible translations of the given source sentence, is generated. This word graph is generated once for each source sentence. During the process of human – machine interaction the system makes use of this word graph in order to complete the prefixes accepted by the human translator. In other words, after the human translator has accepted a prefix string, the system finds the best path in the word graph associated with this prefix string so that it is able to complete the target sentence. Using the word graph in such a way, the system is able to interact with the human translator in a time efficient way.

A word graph is a weighted directed acyclic graph, in which each node represents a partial translation hypothesis and each edge is labeled with a word of the target sentence and is weighted according to the language and translation model scores. The word is produced as a by-product of the search process. An example of a word graph is shown in Fig. 17.3.

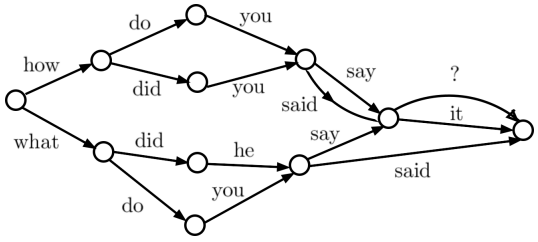

Figure 17.3: Word graph for "How do you say it"

The computational cost of this approach is much lower, as the whole search for the translation must be carried out only once, and the generated word graph can be reused for further completion requests. For a fixed source sentence, if no pruning is applied in the production of the word graph, it represents all possible sequences of target words for which the posterior probability is greater than zero, as per the model. However, because of the pruning helps to make the problem computationally feasible, the resulting word graph only represents a subset of the possible translations. Therefore, it may happen that the user sets prefixes which cannot be found in the word graph. This requires some heuristics to be implemented.