Statistical Translation Models: A Literature Survey

Piyush Dilip Dungarwal
123050083

June 19, 2014

In this survey, we briefly study Phrase-based, Factored and Hierarchical translation models. First we learn basics of *Phrase-based model*. Then we get introduced to an interesting SMT approach called *Factored translation models*. We also study mathematical modeling of the Factored models. Finally, we compare Factored models with Phrase-based models and know their disadvantages which are pulling them back in the race with Phrase-based models. Finally, we study a comparatively different approach called *Hierarchical Phrase-based models*.

1 Phrase-based models

Firstly we will discuss the state-of-the-art approach to statistical machine translation called *Phrase-based models* [Koehn et al., 2003]. The objective of Phrase-based models is to reduce the restrictions of word-based models by translating chunks of words which are contiguous, also called *Phrases*. Note that these phrases need not be linguistic phrases.

1.1 Motivation

- Going beyond word-to-word translation: words may not be the best candidates for the smallest units of translation
- Experiments show that phrase-based models outperform word-based models and results hold for almost all language pairs
- Phrase learning helps resolving ambiguities, as context can provide useful clues about translation

- Model becomes simpler as we do away with the complex notions of fertility, insertion, deletion, etc.
- Intuitively, phrase-based models should take MT closer to the syntax of the languages

1.2 Mathematical model

Phrase-based models use noisy channel approach as shown below. Bayes rule is used to create a Generative model in which translation probability is split into two parts: Reverse translation probability and probability of target sentence.

$$argmax_e p(e|f) = argmax_e p(f|e).p(e)$$

$p(f|e)$ is further decomposed into translation model and distortion model.

$$p(f|e) = \Pi_i \phi(f_i|e_i).d(a_i, b_{i-1})$$

$p(e)$ acts as a n-gram languge model.

$$p(e) = p(e_1).p(e_2|e_1).p(e_3|e_2)...p(e_m|e_{m-1})$$

1.3 Decoding

Decoding of phrase-based models is based on Beam search algorithm. Target sentence is generated left-to-right in form of partial translations.

Decoding starts with an empty hypothesis. A new hypothesis is expanded from an existing hypothesis as follows: A sequence of untranslated words and a possible target phrase translation for them is selected. The target phrase is attached to the existing output sequence. The source words are marked as translated and the probability cost of the hypothesis is updated. The cheapest (highest probability) final hypothesis with no untranslated source words is the output of the search. The hypotheses are stored in stacks. Each stack can hold only a beam of the best n hypotheses. Future cost is determined using the estimated phrase translation cost without considering expected distortion cost.

Time complexity of the beam search is quadratic in sentence length, and linear in beam size.

1.4 Phrase learning

There are three basic methods suggested to learn the phrases.

- Generate all phrases consistent with the word alignments
- Generate phrases from the word sequences that are covered by a single subtree in a syntactic parse tree
- Directly learn phrase alignments using phrase-based joint probability model

2 Factored Translation models

Phrase-based models are limited to the mapping of small contiguous word chunks without using any linguistic information such as morphology, syntax, or semantics. Therefore, phrase-based models were extended to factored models [Koehn and Hoang, 2007b] to include this type of information. The factored approach allows additional annotation at the word level. A word in this framework is not only a token, but a vector of factors that represent different levels of annotation as shown in Figure 1.

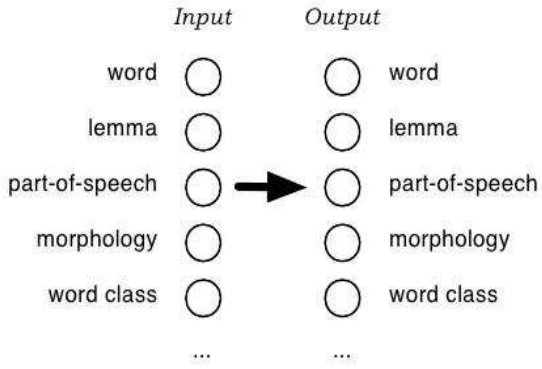

Figure 1: Factored representations of input and output words [Koehn and Hoang, 2007b]

2.1 Motivation

We will consider a case where we are translating from English to Hindi. Now, if the input word is *plays*, then possible outputs may be: खेलता है (*khelta hai*) or खेलती है (*khelti hai*). Both output are equally probable. As the number of translation options increases for English phrase 'e', probability of 'e' being translated to Hindi phrase 'f' ($p(f|e)$) decreases. Thus, if we use phrase-based model in this case, then the system can not decide the correct output just based on input English phrase. It requires some extra information about the input. In our example, we require geneder information to decide whether output should be खेलता है (*khelta hai*) or खेलती है (*khelti hai*). This is in general true for translating from morphologically poor language to morphologically richer language. Factored models can incorporate this extra information together with surface words.

2.2 Decomposition of Factored translation

The single translation step in phrase-based model is broken down into a sequence of mapping steps that either translate source factors into target factors, or generate additional target factors from existing target factors. For example, we consider two translation mappings and one generation mapping as follows [Koehn and Hoang, 2007b]:

- Translation step 1: Translate lemmas
- Translation step 2: Translate parts of speech (POS) and other morphological tags
- Generation step: Generate target surface form given target lemma, target POS tag and target morphology These mappings are shown in Figure 2.

Note that Translation steps map factors in source phrases to factors in target phrases and Generation steps map target factors within individual target words. As all mapping steps operate on the same phrase segmentation of the input and output sentence into phrase pairs, thes models are called *synchronous factored models*.

Let us consider an example to understand these mapping steps. Suppose we are translting a word *boys* from English to Hindi. Then the three mapping steps in our morphological analysis and generation model may provide the following applicable mappings:

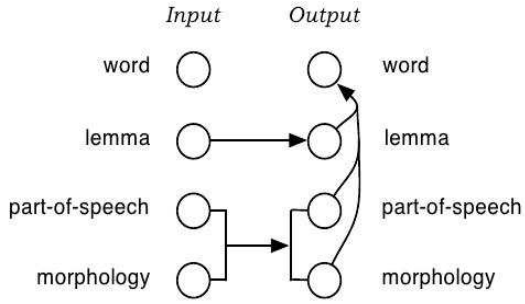

Figure 2: A sequence of mapping steps in Factored models

- Translation: Mapping lemmas
  boy → **लड़का** (*ladka*), **युवक** (*yuvak*), etc.
- Translation: Mapping morphology
  NN|directCase|plural → NN|-e, NN|-on, etc.
- Generation: Generating surface forms
  लड़का|NN|-e → **लड़के** (*ladke*)
  लड़का|NN|-on → **लड़कों** (*ladkon*)
  युवक|NN|-e → **युवक** (*yuvak*)
  युवक|NN|-on → **युवकों** (*yuvakon*)

The application of these mapping steps to an input phrase is called *expansion*. Given the multiple choices for each step (reflecting the ambiguity in translation), each input phrase may be expanded into a list of translation options. English *boys*|*NN*|*directCase*|*plural* may be expanded as follows:

- Translation: Mapping lemmas
  {?|**लड़का**|?|?, ?|**युवक**|?|?}
- Translation: Mapping morphology
  {?|**लड़का**|NN|-e, ?|**लड़का**|NN|-on, ?|**युवक**|NN|-e, ?|**युवक**|NN|-on}
- Generation: Generating surface forms
  {**लड़के**|**लड़का**|NN|-e, **लड़कों**|**लड़का**|NN|-on, **युवक**|**युवक**|NN|-e, **युवकों**|**युवक**|NN|-on}

2.3 Statistical modeling of Factored models

Factored translation modeling is very much similar to the statistical modeling approach of phrase-based models. In fact, in Section 2.4 we will see that phrase-based models are a special case of factored models. The main difference lies in the preparation of the training data and the type of models learned from the data [Koehn and Hoang, 2007b]. **Training:**

- Additional factors are generated for the words in the training data. As manual annotation is expensive, these factors are typically generated using automatic annotation tools such as POS tagger.

- Word aligner is used to learn the alignemnts between words or factors in training data. Generally same method is used as in phrase-based models (GIZA++ alignments). Note that the word alignment methods may operate on the surface forms of words or on any other factors.

- Then, we learn translation and generation tables from the word-aligned parallel corpus and define scoring methods that help us to choose between ambiguous mappings.

- The models for the translation steps are acquired from a word-aligned parallel corpus in the same manner as that of phrase-based models. For the specified factors in the input and output, phrase mappings are extracted. The set of phrase mappings (now over factored representations) is scored based on relative counts and word-based translation probabilities.

- The generation distributions are estimated on the output side only. The word alignment plays no role here. In fact, additional monolingual data may be used. The generation model is learned on a word-for-word basis.

- An important component of statistical machine translation is the language model. Typically an n-gram model over surface forms of words is learned on target side corpus. In the framework of factored translation models, such sequence models may be defined over any factor, or any set of factors.

Combination of components:

Factored translation models can be seen as the combination of several components (language model, reordering model, translation steps, generation steps). These components define one or more feature functions that are combined in a log-linear model [Koehn and Hoang, 2007b]:

$$p(e|f) = \frac{1}{Z} exp \sum_{i=1}^{n} \lambda_i h_i(e, f)$$

Z is a normalization constant that is ignored in practice. To compute the probability of a translation e given an input sentence f, we have to evaluate each feature function hi.

The feature function for a bigram language model component is (m is the number of words ei in the sentence e):

$$h_{LM}(e, f) = p_{LM}(e) = p(e_1).p(e_2|e_1).p(e_3|e_2)...p(e_m|e_{m-1})$$

The translation of the input sentence f into the output sentence e breaks down to a set of phrase translations $(\bar{f}_j, \bar{e_j})$. For a translation step component, each feature function h_T is defined over the phrase pairs $(\bar{f}_j, \bar{e_j})$ given a scoring function τ:

$$h_T(e, f) = \sum_j \tau(\bar{f}_j, \bar{e_j})$$

For a generation step component, each feature function h_G given a scoring function γ is defined over the output words e_k only:

$$h_G(e, f) = \sum_k \gamma(e_k)$$

The feature functions follow from the scoring functions (τ , γ) acquired during the training of translation and generation tables. The feature weights λ_i in the log-linear model are determined using a minimum error rate training method.

Decoding:

Compared to phrase-based models, the decomposition of phrase translation into several mapping steps leads to additional computational complexity. Multiple tables have to be searched instead of a single table look-up to obtain the possible translations for an input phrase. Entries in the phrase table that may be potentially used for a specific input sentence are called *Translation options* [Koehn and Hoang, 2007b].

Decoding algorithm is similar to that of a Phrase-based model (Beam search). The beam search decoding algorithm starts with an empty hypothesis. Then new hypotheses are generated by using all applicable translation options. Hypotheses are created until we get the hypotheses that covers the full input sentence. The highest scoring complete hypothesis indicates the best translation according to the model.

Since all mapping steps operate on the same phrase segmentation, the expansions of these mapping steps can be efficiently precomputed prior to the heuristic beam search and stored as translation options. Given input phrase, all possible translation options are thus computed before decoding.

But we face a problem of combinatorial explosion of the number of translation options given a sequence of mapping steps. This problem is currently solved by heavy pruning of expansions and limiting the number of translation options per input phrase to a maximum number, by default 50. This is, however, not a perfect solution.

2.4 Phrase based models: A special case of Factored models

Phrase-based models are a special case of factored models, i.e., we can derive phrase-based model by using appropriate feature functions in factored model. First we will see the noisy channel modeling of phrase-based models [Koehn et al., 2003]:

$$argmax_e p(e|f) = argmax_e p(f|e).p(e)$$

Here, $p(e)$ is modeled as a Language model:

$$p(e) = p(e_1).p(e_2|e_1).p(e_3|e_2)...p(e_m|e_{m-1})$$

Whereas $p(f|e)$ is expressed in terms of translation and distortion models:

$$p(f|e) = \Pi_i \phi(f_i|e_i).d(a_i, b_{i-1})$$

Here, ϕ is a translation probability function and d is a distortion function. a_i and b_{i-1} denote the starting position of the current phrase and ending position of previous phrase respectively. To derive phrase-based model from factored model, we will consider following feature functions:

- Language model: $h_{LM}(e) = log(p(e))$
- Translation model: $h_T(e, f) = \sum_j log(\phi(f_j|e_j))$
- Distortion model: $h_R(e, f) = \sum_j log(d(a_j, b_{j-1}))$

Now, to prove:

$$argmax_e(\frac{1}{Z}exp\sum_j \lambda_j h_j(e,f)) = argmax_e(p(e).\Pi_j \phi(f_j|e_j).d(a_j,b_{j-1}))$$

Derivation:

$LHS = argmax_e(\frac{1}{Z}exp\sum_j \lambda_j h_j(e,f))$

$= argmax_e(\frac{1}{Z}exp(\lambda_1 \sum_j log(\phi(f_j|e_j)) + \lambda_2 \sum_j log(d(a_j,b_{j-1})) + \lambda_3 log(p(e))))$

Z can be ignored as it is a normalization constant.

$= argmax_e(exp(\lambda_1 \sum_j log(\phi(f_j|e_j)) + \lambda_2 \sum_j log(d(a_j,b_{j-1})) + \lambda_3 log(p(e))))$

Considering weight of each model as 1, i.e., $\lambda_1 = \lambda_2 = \lambda_3 = 1$

$= argmax_e(exp(\sum_j log(\phi(f_j|e_j)) + \sum_j log(d(a_j,b_{j-1})) + log(p(e))))$

Expanding exponential function over summation,

$= argmax_e(exp(\sum_j log(\phi(f_j|e_j)).exp(\sum_j log(d(a_j,b_{j-1}))).exp(log(p(e)))))$

As $exp(log(p(e)) = p(e)$,

$= argmax_e(exp(\sum_j log(\phi(f_j|e_j)).exp(\sum_j log(d(a_j,b_{j-1}))).p(e)))$

Also, $exp(\sum_j log(\phi(f_j|e_j)) = \Pi_j log(\phi(f_j|e_j))$ and $exp(\sum_j log(d(a_j,b_{j-1}))) = \Pi_j log(d(a_j,b_{j-1}))$,

$= argmax_e(p(e).\Pi_j \phi(f_j|e_j).d(a_j,b_{j-1}))$

$= RHS$

Thus, we proved that Phrase-based model can be derived from Factored model by using three feature functions: Language model, translation model and distortion model. Hence, we can say that Phrase-based models are a special case of Factored models.

2.5 Disadvantages of Factored models

Factored models create more accurate translations but also create many more unknowns compared to phrase-based model.
For example, consider two pairs of sentences,

1. Factored: Ram|null eats|+musc food|null → राम खाना खाता है (*raam khana khata hai*)
   Unfactored: Ram eats food → राम खाना खाता है (*raam khana khata hai*)

2. Factored: Sita|null eats|-musc food|null → सीता खाना खाती है (*sita khana khati hai*)
   Unfactored: Sita eats food → सीता खाना खाती है (*sita khana khati hai*)

Now, consider two different cases:

- *Case 1: With only sentence 1 in training data*
  Test phrase: eats|-musc (for factored model), eats (for phrase-based model)
  As combination eats|-musc will be absent in the phrase table of factored model, output will be unknown. But phrase-based model has eats in its phrase table. Hence, output will be खाता है (*khata hai*), even though it is incorrect.

- *Case 2: With both, sentence 1 and 2 in training data*
  Test phrase: eats|-musc (for factored model), eats (for phrase-based model)
  Now, combination eats|-musc will be present in the phrase table of factored model and hence output will be खाती है (*khati hai*), which is correct. Whereas, phrase-based model has two possible outputs for eats with equal probability. Hence, probability of correct translation for factored model is 1, whereas for phrase-based model, it is 0.5.

Data sparseness:

Even though factored models generate accurate translations, they face severe problem of data sparseness which limits their performance. The example of data sparseness with factored models can be seen in case 1 above.

Data sparseness can be classified as follows:

- **Sparseness in translation:** Combination of factors does not exist on the source side in the training data
  E.g. Consider case 1 of above example. Phrase table of factored model did not have factor combination eat|-musc on the source side. Hence, output generated was unknown.

- **Sparseness in Generation:** Combination of factors does not exist on the target side in the training data
  E.g. Consider generation step of generating Hindi surface form from Hindi lemma and suffix. Let translations steps be English surface word to Hindi lemma and English gendet to Hindi suffix.
  Let training data contain two sentences:

  – Ram|null eats|+musc food|null → राम|.|. खाना|.|. खाताहै|खा|ताहै
  – Sita|null runs|-musc → सीता|.|. दौड़तीहै|दौड़|तीहै

  Now, let the test sentence be: Sita|null eats|-musc.
  The output will be: सीता eats|-musc
  Because, even though translation tables have entry for Sita to सीता, eats to खा and -musc to तीहै, generation table does not have entry for खा|तीहै to खातीहै.

Decoding complexity:

As we saw in Section 2.3, decoding of factored models may generate huge number of translation options. This is obvious from the fact that factored models consider extra information apart of surface word. The number of translation options increase exponentially with number of factors used. As it is unmanageable for a system to handle large number of options, it either results in degraded translation output or it takes large time to translate.

Hence, it is not suggested to use many factors while designing a factored model, as it may degrade the translation system performance. Moses decoder allows four factors by default.

More complex setups of factored models can dramatically increase the complexity of factored models. Combination of translation options of various steps can cause combinatorial explosion. During decoding, pruning will likely discard good hypotheses, as stacks will be filled with too many factor combinations.

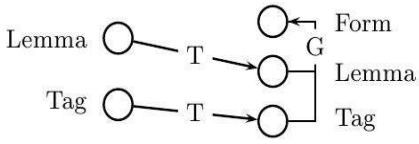

Figure 3: Sample Factored model [Tamchyna and Bojar, 2013]

Consider the factored setup shown in Figure 3. This setup uses two translation steps (lemma → lemma, tag → tag). It also uses one generation step (Target lemma, tag → Form). For each source phrase, the decoder generates all possible translations of the lemmas. Then it combines each lemma with all *consistent* translations of the tags (resulting in a subset of Cartesian product of the lemma/tag options). Finally, each combination generates zero, one or more phrases of target forms.

An expansion is considered consistent if the target side has the same length and if the shared factors match. If the steps share some of the output factors, the order of application of mapping step plays a significant role. In this case, only consistent translation options can be generated during expansion [Tamchyna and Bojar, 2013].

3 Hierarchical Phrase-based models

Hierarchical phrases are phrases which contain subphrases. Hierarchical model is based on Synchronous-CFG. This model is learnt from the parallel data without any syntactic annotations. This model combines Phrase-based and Syntax-based translation. Hierarchical models are described in detail in [Chiang, 2005].

3.1 Problem with Phrase-based model

Even though phrase-based approach learns reordering well, there is often need of learning reordering between phrases itself. Thus, the need of hierarchical phrases arises.

Example: Mandarin to English phrase-based translation

Aozhou shi yu Beihan you bangjiao de shaoshu guojia zhiyi .

Australia is with North Korea have dipl. rels. that few countries one of .

Australia is one of the few countries that have diplomatic relations with North Korea.

Translation output of ATS (Alignment Template System) phrase-based system:

[Aozhou] [shi] [yu Beihan] [you] [bangjiao] [de shaoshu guojia zhiyi] [.]
[Australia] [has] [dipl. rels.] [with North Korea] [is] [one of the few countries] [.]

Words in brackets are phrases learnt by ATS. The output translation is wrong. The phrase-based model is able to order *has diplomatic relations with North Korea* correctly (using phrase reordering) and *is one of the few countries* correctly (using a combination of phrase translation and phrase reordering), but does not invert these two groups as it should.

Hence, to get correct translation, we also need to learn hierarchical phrases of the form:
<yu X1 you X2, have X2 with X1>
<X1 de X2, the X2 that X1>
<X1 zhiyi, one of X1>
Here, X1 and X2 are place-holders for subphrases.

3.2 Hierarchical Model

Hierarchical model consists of a set of SCFG rules of the form as shown above. In SCFG, the elementary structures are rewrite rules with aligned pairs of right-hand sides: X $\rightarrow$ <γ, α, ~>
Here, X is a nonterminal. γ and α are strings of nonterminals and terminals. ~ is one-to-one correspondences between nonterminals in γ and α.

Rewrite rules for previous example:
X $\rightarrow$ <yu X1 you X2, have X2 with X1>
X $\rightarrow$ <X1 de X2, the X2 that X1>
X $\rightarrow$ <X1 zhiyi, one of X1>

Other rules (similar to phrase-table entries in phrase-based model):
X $\rightarrow$ <Azhou, Australia>
X $\rightarrow$ <Beihan, North Korea>
X $\rightarrow$ <shi, is>
X $\rightarrow$ <bangjiao, diplomatic relations>
X $\rightarrow$ <shaoshu guojia, few countries>

Glue rules:

All of the rules in SCFG use only X as a nonterminal, except for two special glue rules, which combine a sequence of Xs to form an S:

S → <S1 X2, S1 X2>
S → <X1, X1>

These rules give the model the option to build only partial translations using hierarchical phrases, and then combine them serially as in a standard phrase-based model.

Hierachical model is a log-linear model:
$w(X \rightarrow <\gamma, \alpha>) = \Pi_i \phi_i(X \rightarrow <\gamma, \alpha>)^{\lambda_i}$

Where the ϕ_i are features defined on rules. Features analogous to Pharaohs default feature set are:

- $P(\gamma|\alpha)$ and $P(\alpha|\gamma)$: $P(\gamma|\alpha)$ is similar to the phrase translation probability used in noisy channel model.
- Lexical weights: $P_w(\gamma|\alpha)$ and $P_w(\alpha|\gamma)$ to analyze how well the words in α translate the words in γ.
- Phrase penalty (exp(1)): Allows model to learn preference for longer or shorter derivations.

Let D be the derivation of the grammar. We can represent D as a set of triples <r, i, j> , each of which stands for an application of a grammar rule r to rewrite a nonterminal that spans i to j on the source side. Then, the weight of D is decided as:
$w(D) = \Pi_{<r,\ i,\ j> \epsilon D} * p_{lm}(e)^{\lambda_{lm}} * \exp(-\lambda_{wp}|e|)$

Weight of a derivation is product of the weights of the rules used in the translation, multiplied by the language model factor and word penalty. Word penalty is to control the length of target output sentence.

Training of this model includes word alignment of bilingual corpus and extraction of set of rules consistent with word alignments. Extraction of rules is divided into two steps:

- Identifying Initial phrase pairs: Find phrases from the phrase-table which are consistent with word alignments.
- Obtain rules from phrases: Find phrases that contain other phrases and replace the contained phrases (subphrases) with nonterminal symbols.

As this method generates large number of rules, some constraints are applied on the extracted rules to filter them out. Parameter estimation of the model and decoding process is not discussed here.

Thus, we studied three different approaches of statistical machine translation out of which Phrase-based and Factored models share many processes and parameters that are used in training and decoding as they are based on the concept of *phrases*. Whereas, Hierarchical models are based on the concept of Hierarchical phrases and follow completely different methodology.

Summary

- We studied Phrase-based models: Mathematical modeling, decoding, phrase learning
- We studied Factored models: Mathematical modeling, training and decoding
- We also studied how Phrase-based models are a special case of Factored models
- We discusses some disadvantages of Factored models such as Data sparseness and huge decoding complexity
- We studied Hierarchical phrase-based models and Synchronous CFG

References

Avramidis, Eleftherios, and Philipp Koehn. Enriching Morphologically Poor Languages for Statistical Machine Translation. *ACL*, 2008.

Chiang, David. A hierarchical phrase-based model for statistical machine translation. *ACL*, 2005.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. *Proceedings of the 2003 Conference of the North American Association for Computational Linguistics on Human Language Technology*, pages 48–54, 2003.

Koehn, Philipp and Hieu Hoang. Factored translation models. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, ACL*, pages 868–876, 2007.

Koehn, Philipp. Statistical machine translation. *Cambridge University Press*, 2010

Ramanathan, Ananthakrishnan, Bhattacharyya P., Hegde J.J., Shah R.M., and Sasikumar M. Simple syntactic and morphological processing can help english-hindi statistical machine translation. *Proceedings of IJCNLP*, 2008.

Ramanathan, Ananthakrishnan, Hansraj Choudhary, Avishek Ghosh, and Pushpak Bhattacharyya. Case markers and morphology: Addressing the crux of the fluency problem in english-hindi smt. *Proceedings of ACL/IJCNLP, ACL*, 2:800–808, 2009.

Tamchyna, Ale and Ondej Bojar. No free lunch in factored phrase-based machine translation. In *Computational Linguistics and Intelligent Text Processing*, volume 7817, pages 210–223. Springer Berlin Heidelberg, 2013. URL `http://dx.doi.org/10.1007/978-3-642-37256-8_18`.