Hierarchical Phrase-Based Translation
David Chiang (2006)

Hauptseminar: Hybrid Machine Translation
Instructor: Prof. Dr. Kurt Eberle
Presenters: Roja Irandoost, Mohamed Ouji

Summer Semester 2018
1. Introduction

Advancements in machine translation:
Expanding the basic unit of translation from words to phrases, i.e substrings of potentially unlimited size.

The model learns:
- local reorderings
- translations of multiword expressions
- insertions and deletions sensitive to local context
1. Introduction

The noisy-channel approach

Conventionally:
Source language: French
Target language: English

Translation of a French sentence $f$ into an English sentence $e$ is modeled as:

$$\arg \max_{e} P(e \mid f) = \arg \max_{e} P(e, f)$$

$$= \arg \max_{e} (P(e) \times P(f \mid e))$$
1. Introduction

The noisy-channel approach (Brown et al., 1993)

Statistical translation

First view
Every French string, \( f \), is a possible translation of \( e \).
We assign to every pair of strings \((e,f)\) a number \( Pr(f|e) \) as the probability that a translator, when presented with \( e \), will produce \( f \) as his translation.

Second view
When a native speaker of French produces a string of French words, he has actually conceived of a string of English words, which he translated mentally.
Given a French string \( f \), the job of our translation system is to find the string \( e \) that the native speaker had in mind when he produced \( f \). We minimize our chance of error by choosing that English string \( \hat{e} \) for which \( Pr(e|f) \) is greatest.
1. Introduction

The noisy-channel approach (Brown et al., 1993)

Using Bayes' theorem:

$$
Pr(e|f) = \frac{Pr(e) Pr(f|e)}{Pr(f)}.
$$

Since the denominator here is independent of $e$, finding $\hat{e}$ is the same as finding $e$ so as to make the product $Pr(e)Pr(f|e)$ as large as possible. We arrive at the Fundamental Equation of Machine Translation:

$$
\hat{e} = \arg\max_e Pr(e) Pr(f|e).
$$
1. Introduction

Steps of encoding $e$ into $f$ by the phrase-based translation model $P(f|e)$

1. Segment $e$ into phrases $\tilde{e}_1 \cdots \tilde{e}_I$, typically with a uniform distribution over segmentations;
2. Reorder the $\tilde{e}_i$ according to some distortion model (Al-Onaizan and Papineni, 2006);
3. Translate each of the $\tilde{e}_i$ into French phrases according to a model $P(f | \tilde{e})$ estimated from the training data.

Phrase reordering has limitations.
1. Introduction

An illustration of the limitations of phrase reordering

Mandarin example and its English translation:

澳洲是与北韩有邦交的少数国家之一。
Aozhou shiyu 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.
1. Introduction

Running a phrase-based system

We get the following phrases with translations:

[Aozhou] [shi]$_1$ [yu Beihan]$_2$ [you] [bangjiao] [de shaoshu guojia zhiyi] [.]

[Australia] [has] [dipl. rels.] [with North Korea]$_2$ [is]$_1$ [one of the few countries] [.]

- Correctly ordered by phrase reordering
- Subscripts indicate the reordering of phrases.
- Correctly ordered by a combination of phrase translation and phrase reordering

Not correctly inverted
1. Introduction

Solution

Phrases are good for learning reorderings of words.

Use them to learn reorderings of phrases.

Use hierarchical phrases that can contain other phrases.
1. Introduction

Hierarchical phrase pairs example

\[ \langle \text{you} \, 1, \text{have} \, 2, \text{with} \, 1 \rangle \]

Chinese PPs modify VPs on the left.
English PPs usually modify VPs on the right.

\[ \langle 1 \, \text{de} \, 2, \text{the} \, 2, \text{that} \, 1 \rangle \]

Chinese relative clauses modify NPs on the left.
English relative clauses modify NPs on the right.

\[ \langle 1 \, \text{zhiyi}, \text{one of} \, 1 \rangle \]
The construction \text{zhiyi} in English word order
1. Introduction

Hierarchical phrase pairs example

Some conventional phrase pairs:

\langle Aozhou, Australia \rangle

\langle Beihan, North Korea \rangle

\langle shi, is \rangle

\langle bangjiao, diplomatic relations \rangle

\langle shaoshu guojia, few countries \rangle
1. Introduction

Hierarchical phrase pairs example

The three rules, along with the conventional phrase pairs, suffice to translate the sentence correctly:

[Australia] [is] [one of [the [few countries]_3 that [have [dipl. rels.]_2 with [N. Korea]_1]]]
1. Introduction

Features of our system

- Learning the rules automatically from a parallel text without syntactic annotation
- Using CFG rules, thus, being syntax-based statistical machine translation
- Combining the idea of hierarchical structure with key insights from phrase-based MT namely:
  - Incorporating elementary structures with possibly many words to inherit phrase-based MT’s capacity for memorizing translations from parallel data
  - Using an m-gram language model
  - Using Minimum-error-rate training of Log-linear models
2. Related work

Syntax-based statistical MT

Syntax-based statistical MT vary in:

- Reliance on syntactic theories:
  - Approaches that have no dependence on syntactic theory beyond the idea that natural language is hierarchical.
  - Approaches that make use of parallel data with syntactic annotations, either in the form of phrase-structure trees or dependency trees
- Annotations made according to syntactic theories.

Do NOT distinguish between very many categories
2. Related work

Extraction of larger rules in our system

phrase extraction from word-aligned parallel text
3. Grammar

The model is based on a **synchronous CFG** known as a **syntax-directed transduction grammar**.

3.1. Synchronous CFG

The elementary structures are rewrite rules with aligned pairs of right-hand sides:

\[ X \rightarrow \langle \gamma, \alpha, \sim \rangle \]

- **X**: a nonterminal
- **\gamma** and **\alpha**: strings of terminals and nonterminals
- **\sim**: a one-to-one correspondence between nonterminal occurrences in **\gamma** and nonterminal occurrences in **\alpha**
3. Grammar

3.1. Synchronous CFG

Example

Formalizing the previous **hierarchical phrases** in a synchronous CFG:

\[ X \rightarrow \langle yu \, X_1 \, you \, X_2 \, , \, have \, X_2 \, with \, X_1 \rangle \]

\[ X \rightarrow \langle X_1 \, de \, X_2 \, , \, the \, X_2 \, that \, X_1 \rangle \]

\[ X \rightarrow \langle X_1 \, zhiyi \, , \, one \, of \, X_1 \rangle \]

Boxed indices indicate which nonterminal occurrences are linked by \( \sim \).
3. Grammar

3.1. Synchronous CFG

Example
Formalizing the **conventional phrase pairs** (making rules without nonterminal symbols on the right-hand side):

\[ X \rightarrow \langle \text{Aozhou, Australia} \rangle \]

\[ X \rightarrow \langle \text{Beihan, North Korea} \rangle \]

\[ X \rightarrow \langle \text{shi, is} \rangle \]

\[ X \rightarrow \langle \text{bangjiao, diplomatic relations} \rangle \]

\[ X \rightarrow \langle \text{shaoshu guojia, few countries} \rangle \]
3. Grammar

3.1. Synchronous CFG

A synchronous CFG derivation begins with *a pair of linked start symbols*

\[ S \rightarrow \langle S_1X_2, S_2X_2 \rangle \]

\[ S \rightarrow \langle X_1, X_2 \rangle \]

- At each step, two linked nonterminals are rewritten using the two components of a single rule.
- When denoting links with boxed indices, we must consistently reindex the newly introduced symbols apart from the symbols already present.
\begin{align*}
\langle S_1, S_3 \rangle \\
\Rightarrow \langle S_2 X_5, S_3 X_3 \rangle \\
\Rightarrow \langle S_4 X_5 X_3, S_4 X_5 X_3 \rangle \\
\Rightarrow \langle X_6 X_5 X_3, X_6 X_5 X_3 \rangle \\
\Rightarrow \langle \text{Aozhou } X_5 X_5, \text{ Australia } X_5 X_3 \rangle \\
\Rightarrow \langle \text{Aozhou shi } X_6, \text{ Australia is } X_9 \rangle \\
\Rightarrow \langle \text{Aozhou shi } X_6 \text{ zhiyi, Australia is one of } X_9 \rangle \\
\Rightarrow \langle \text{Aozhou shi } X_6 \text{ de } X_9 \text{ zhiyi, Australia is one of the } X_9 \text{ that } X_9 \rangle \\
\Rightarrow \langle \text{Aozhou shi } X_6 \text{ you } X_9 \text{ de } X_9 \text{ zhiyi, Australia is one of the } X_9 \text{ that have } X_9 \text{ with } X_9 \rangle \\
\Rightarrow \langle \text{Aozhou shi } X_6 \text{ you Beihan you } X_9 \text{ de } X_9 \text{ zhiyi, Australia is one of the } X_9 \text{ that have } X_9 \text{ with North Korea} \rangle \\
\Rightarrow \langle \text{Aozhou shi } X_6 \text{ you bangjiao de } X_9 \text{ zhiyi, Australia is one of the } X_9 \text{ that have diplomatic relations with North Korea} \rangle \\
\Rightarrow \langle \text{Aozhou shi } X_6 \text{ you bangjiao de shaoshu guojia zhiyi, Australia is one of the few countries that have diplomatic relations with North Korea} \rangle
\end{align*}
3. Grammar

3.2. Rule Extraction

(a) **Input word alignment**
   The extraction process begins with a word-aligned corpus

(b) **Initial phrases**
   Identification of initial phrase pairs
   at least one word inside one phrase aligned to a word inside the other, but no word inside one phrase aligned to a word outside the other phrase

(c) **Rule extraction**
   From each word-aligned sentence pair
   Look for phrases that contain other phrases and replace the sub-phrases with nonterminal symbols

by running GIZA++ on the corpus in both directions, and forming the union of the two sets of word alignments
3. Grammar

3.2. Rule Extraction

The extraction process begins with a word-aligned corpus:

A set of triples \(<f, e, \sim>\)

\(f\): a French sentence
\(e\): an English sentence
\(\sim\): a (many-to-many) binary relation between positions of \(f\) and positions of \(e\).

Then a set of rules is extracted from each word-aligned sentence pair.

by running GIZA++ on the corpus in both directions, and forming the union of the two sets of word alignments

consistent with the word alignments
3. Grammar

3.2. Rule Extraction

Extracting the rules in 2 steps:

**STEP 1**

**Identify initial phrase pairs**
There must be at least one word inside one phrase aligned to a word inside the other, but no word inside one phrase can be aligned to a word outside the other phrase.
3. Grammar

3.2. Rule Extraction

Definition 1
Given a word-aligned sentence pair \( <f, e, \sim> \), let \( f^i_j \) and \( e^i_{j'} \) stand for the substring of \( f \) from position \( i \) to position \( j \) inclusive, and similarly for \( e \). Then a rule:

\[
(f^i_j, e^i_{j'})
\]

is an initial phrase pair of \( <f, e, \sim> \) iff:

1. \( f_k \sim e_{k'} \) for some \( k \in [i, j] \) and \( k' \in [i', j'] \);
2. \( f_k \not\sim e_{k'} \) for all \( k \in [i, j] \) and \( k' \not\in [i', j'] \);
3. \( f_k \not\sim e_{k'} \) for all \( k \not\in [i, j] \) and \( k' \in [i', j'] \).
1. \( f_k \sim e_{k'} \) for some \( k \in [i, j] \) and \( k' \in [i', j'] \);

There must be at least one word inside one phrase (in French) aligned to a word inside the other (in English).

2. \( f_k \not\sim e_{k'} \) for all \( k \in [i, j] \) and \( k' \not\in [i', j'] \);

There must be no word inside one phrase (in French) that can be aligned to a word outside the other phrase (in English).

3. \( f_k \not\sim e_{k'} \) for all \( k \not\in [i, j] \) and \( k' \in [i', j'] \).

When a word is not in one phrase (in French), it shouldn't be in the other corresponding phrase (in English) either.
3. Grammar

3.2. Rule Extraction (step b)

Example
If our training data contained the fragment:

30 多年来 的 友好 合作
30 duonianlai de youhao hezou
30 plus-years-past of friendly cooperation

friendly cooperation over the last 30 years

With word alignments as shown in figure:
3. Grammar

3.2. Rule Extraction (step b)

Example

The initial phrases that would be extracted:
3. Grammar

3.2. Rule Extraction

**STEP 2**

Obtain rules from the phrases
Look for phrases that contain other phrases and replace the sub-phrases with nonterminal symbols.
3. Grammar

3.2. Rule Extraction

Definition 2
The set of rules of $<f, e, \sim>$ is the smallest set satisfying the following:

1. If $\langle f_i^j, e_i^j \rangle$ is an initial phrase pair, then $X \rightarrow \langle f_i^j, e_i^j \rangle$ is a rule of $<f, e, \sim>$.

2. If $(X \rightarrow \langle \gamma, \alpha \rangle)$ is a rule of $<f, e, \sim>$ and $\langle f_i^j, e_i^j \rangle$ is an initial phrase pair such that $\gamma = \gamma_1 f_i^j \gamma_2$ and $\alpha = \alpha_1 e_i^j, \alpha_2$, then

$$X \rightarrow \langle \gamma_1 X_k, \gamma_2, \alpha_1 X_k, \alpha_2 \rangle$$

$k$: an index not used in $\gamma$

$\alpha$: a rule of $<f, e, \sim>$.
3. Grammar

3.2. Rule Extraction (step c)

Example

Given the initial phrases:

We could form the rule:

\[ X \rightarrow \langle X_1 \text{ duonianlai de } X_2, X_2 \text{ over the last } X_1 \text{ years} \rangle \]
3. Grammar

3.2. Rule Extraction (step c)

Example

\[ X \rightarrow \langle X_{\Box} \text{ duonianlai de } X_{\Box}, X_{\Box} \text{ over the last } X_{\Box} \text{ years} \rangle \]
3. Grammar

3.2. Rule Extraction

**Problem:**
A very large number of rules is created and consequently:
- very slow training and decoding
- **spurious ambiguity**—a situation where the decoder produces many derivations that are distinct yet have the same model feature vectors and give the same translation.

**Solution:**
Filter the grammar to balance grammar size and performance on our development set.
3. Grammar

3.2. Rule Extraction

Constraints for filtering the grammar:

1. If multiple initial phrase pairs contain the same set of alignments, only keep smallest.
2. Limit Initial phrases to a length of 10 words.
5. Nonterminals cannot be adjacent on the French side.
6. At least one pair of aligned words in each rule (translation decisions always based on some lexical evidence).

Simplifying the decoder
Preventing spurious ambiguity
3. Grammar

3.3. Other Rules

Glue rules

After extracting rules from the training data:
- Let X be the grammar’s start symbol
- Translate new sentences using only the extracted rules
  - Divide a French sentence into a sequence of chunks and translate one chunk at a time.

These rules analyze an S (the start symbol) as a sequence of Xs which are translated without reordering.
3. Grammar

3.3. Other Rules

*Entity rules*

- For each sentence, run some specialized translation modules to translate the numbers, dates, numbers, and bylines in the sentence.
- Insert these translations into the grammar as new rules.
4. Model

Given a French sentence $f$, a synchronous CFG will have, in general, many derivations that yield $f$ on the French side, and therefore (in general) many possible translations $e$.

We define a model over derivations $D$ to predict which translations are more likely than others.
4. Model

Use a general log-linear model over derivations $D$:

$$P(D) \propto \prod_{i} \phi_{i}(D)^{\lambda_{i}}$$

$\phi_{i}$: features defined on derivations

$\lambda_{i}$: feature weights
4. Model

One feature: an m-gram language model PLM(e)
Rest of the features: products of functions on the rules used in a derivation:

$$\phi_i(D) = \prod_{(X \rightarrow \langle \gamma, \alpha \rangle) \in D} \phi_i(X \rightarrow \langle \gamma, \alpha \rangle)$$

So we can rewrite $P(D)$ as:

$$\phi_i(D) = \prod_{(X \rightarrow \langle \gamma, \alpha \rangle) \in D} \phi_i(X \rightarrow \langle \gamma, \alpha \rangle)$$
4. Model

**Weighted synchronous CFG:**

A synchronous CFG together with a function $w$ that assigns weights to rules. This function induces a weight function over derivations:

$$w(D) = \prod_{(X \rightarrow \langle \gamma, \alpha \rangle) \in D} w(X \rightarrow \langle \gamma, \alpha \rangle)$$
4. Model

If we define:

$$w(X \rightarrow \langle \gamma, \alpha \rangle) = \prod_{i \neq LM} \phi_i(X \rightarrow \langle \gamma, \alpha \rangle)^{\lambda_i}$$

then the probability model becomes:

$$P(D) \propto P_{LM}(e)^{\lambda_{LM}} \times w(D)$$
4. Model

4.2 Features

The rules extracted from the training bitext have the following features:

- $P(\gamma | \alpha)$ and $P(\alpha | \gamma)$, the latter of which is not found in the noisy-channel model.
- The lexical weights $P_w(\gamma | \alpha)$ and $P_w(\alpha | \gamma)$ which estimate how well the words in $\alpha$ translate the words in $\gamma$.
- A penalty $\exp(-1)$ for extracted rules which allows the model to learn a preference for longer or shorter derivations.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>language model (large)</td>
<td>1.00</td>
</tr>
<tr>
<td>language model (bitext)</td>
<td>1.03</td>
</tr>
<tr>
<td>$P(\gamma</td>
<td>\alpha)$</td>
</tr>
<tr>
<td>$P(\alpha</td>
<td>\gamma)$</td>
</tr>
<tr>
<td>$P_w(\gamma</td>
<td>\alpha)$</td>
</tr>
<tr>
<td>$P_w(\alpha</td>
<td>\gamma)$</td>
</tr>
<tr>
<td>numbers</td>
<td>0.364</td>
</tr>
<tr>
<td>dates</td>
<td>6.67</td>
</tr>
<tr>
<td>names</td>
<td>2.89</td>
</tr>
<tr>
<td>bylines</td>
<td>-952</td>
</tr>
<tr>
<td>extracted rules</td>
<td>4.32</td>
</tr>
<tr>
<td>glue rule</td>
<td>-0.281</td>
</tr>
<tr>
<td>word penalty</td>
<td>-4.12</td>
</tr>
</tbody>
</table>
4. Model

4.2 Features

Penalties $\exp(-1)$ for various other classes of rules:

- For the glue rule: $S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle$
  
  so that the model can learn a preference for hierarchical phrases over a serial combination of phrases

- For each of the four types of rules (numbers, dates, names, bylines) inserted by the specialized translation modules, so that the model can learn how much to rely on each of them

For all the rules, there is a word penalty $\exp(-\#T(\alpha))$, where $\#T$ just counts terminal symbols. This allows the model to learn a general preference for shorter or longer outputs.
4. Model

4.3. Training

In order to estimate the parameters of the phrase translation and lexical-weighting features, we need counts for the extracted rules. For each sentence pair in the training data, there is in general more than one derivation of the sentence pair using the rules extracted from it. Because we have observed the sentence pair but have not observed the derivations, we do not know how many times each derivation has been seen, and therefore we do not actually know how many times each rule has been seen.
4. Model

4.3. Training

We use heuristics to hypothesize a distribution of possible rules as though we observed them in the training data.

- give a count of one to each initial phrase pair occurrence,
- distribute its weight equally among the rules obtained by subtracting sub-phrases from it,
- use relative-frequency estimation to obtain $P(\gamma | \alpha)$ and $P(\alpha | \gamma)$,
- the parameters $\lambda_i$ of the log-linear model are learned by minimum error-rate training,
- obtain a weighted synchronous CFG ready to be used by the decoder.
5. Decoding

Decoder

A CKY (Cocke-Kasami-Younger) parser with beam search together with a post-processor for mapping French derivations to English derivations

Given a French sentence $f$, it finds the English yield of the single best derivation that has French yield $f$:

$$\hat{e} = e \left( \arg\max_{D \text{ s.t. } f(D) = f} P(D) \right)$$
5. Decoding

5.1 Basic Algorithm

Decoder
Defines:
- a space of weighted items
  - some items: axioms
  - some items: goals (the items to be proven)
- a set of inference rules of the form:

\[
\frac{I_1 : w_1 \quad \cdots \quad I_k : w_k}{I : w} \quad \phi
\]

If all the items \(I_i\) (the antecedents) are provable, with weight \(w_i\), then \(I\) (the consequent) is provable, with weight \(w\), provided the side condition \(\phi\) holds.
5. Decoding

5.1 Basic Algorithm

The parsing process grows a set of provable items:

- LM parser (no language model incorporated)

starts with the axioms

proceeds by applying inference rules

proves more and more items until a goal is proven.
5. Decoding

5.1 Basic Algorithm

The actual search procedure:

- organize the proved items into an array \textit{chart}
- cells: \textit{chart}[X, i, j] are sets of items, ordered such that every item comes after its possible antecedents
- visit the chart cells in order
- prove all the items for each cell
- add every new proved item to the appropriate chart cell
- Store with each item, a tuple of back-pointers to the antecedents from which the item was deduced
- Merge two added items if they are equivalent except for their weights or back-pointers (hypothesis recombination)
- Give the merged item its weight and back-pointers from the better of the two equivalent items