Statistical Machine Translation

Part I: Khalil Sima’an
Data and Models
Universiteit van Amsterdam

Part II: Trevor Cohn
Decoding and efficiency
University of Sheffield
Statistical Machine Translation: PART I

Dr. Khalil Sima’an
Statistical Language Processing and Learning
Institute for Logic, Language and Computation
Universiteit van Amsterdam

Some slides use figures from Philipp Koehn, Barry Haddow and Sophie Arnoult
Data and Models: Structure of lecture

- General statistical framework
- Word-based models: word alignments
- Phrase-based models: phrase-alignments
- Tree-based models: tree-alignments
Task: Translate a source sentence $f$ to a target sentence $e$.

Data: Parallel corpus (source-target sentence pairs).

Source-Channel Approach: IBM Models (1990’s)
Parallel Corpus Example

Parallel corpus $C = \text{a collection of text-chunks and their translations.}$

Parallel corpora are the by-product of \textit{human translation}. Every source chunk is paired with a target chunk.

<table>
<thead>
<tr>
<th>Dutch</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>De prijs van het huis is gestegen.</td>
<td>The price of the house has risen.</td>
</tr>
<tr>
<td>Het huis kan worden verkocht.</td>
<td>The house can be sold.</td>
</tr>
<tr>
<td>Als het de marktprijs daalt zullen sommige gezinnen een zware tijd doormaken.</td>
<td>If the market price goes down, some families will go through difficult times.</td>
</tr>
</tbody>
</table>

- United Nations documents.
- Newspapers: Chinese-English; Arabic-English; Urdu-English.
- TAUS corpora.
Given source sentence $f$, select target sentence $e$

$$\arg\max_{e \in E(f)} \{ P(e \mid f) \} = \arg\max_{e \in E(f)} \{ P(e) \times P(f \mid e) \}$$

Set $E(f)$ is the set of hypothesized translations of $f$.

$P(f \mid e)$: accounts for divergence in . . .
- word order
- morphology
- syntactic relations
- idiomatic ways of expression

How to estimate $P(e \mid f)$? **Sparse-data problem!**
Inducing The Structure of Translation Data

\[ e = \text{Mary did not slap the green witch .} \]
\[ f = \text{Maria no dio una bofetada a la bruja verde .} \]

The latent structure of translation equivalence

Graphical representations \( \Delta_f \) and \( \Delta_e \) for \( f \) and \( e \)

Relation \( a \) between \( \Delta_f \) and \( \Delta_e \)

\[
\arg \max_{e \in E(f)} \left\{ P(e \mid f) \right\} = \arg \max_{e \in E(f)} \left\{ \sum_{\langle \Delta_f, a, \Delta_e \rangle} P(e, \Delta_f, \Delta_e, a \mid f) \right\}
\]

The difficult question: Which \( \Delta_f/e \) and \( a \) fit data best?
Structure in current models

\[ \Delta_f \xrightarrow{a} \Delta_e \]

In most current models structure of reordering:
- \( \Delta_{f/e} \) are structures over word positions.
- \( a \) is an **alignment** between groups of word positions in \( \Delta_f \) and \( \Delta_e \).

Challenge: Number of permutations of \( n \) words is \( n! \)

Structure shows translation units **composing** together
- What are the atomic translation units?
- How these compose together **efficiently**?
- How to put probs. on these structures?

Structure helps combat sparsity and complexity
Word-based

Phrase-based

Tree-based

Problem: No sufficient stats to estimate $P(e \mid f)$ from data
Word-Based Models: Word Alignments
Some History and References

Statistical models with word-alignments:

Word-Based Models and Word-Alignment

- $a$ is a mapping between word positions.

- $\Delta_f$ and $\Delta_e$ are sequences of word positions. 
  
  $e = e_1^l = e_1 \ldots e_l$ and $f = f_1^m = f_1 \ldots f_m$

- A hidden word-alignment $a$:

  $$P(f \mid e) = \sum_a P(a, f \mid e)$$

- Assume that a target word-position $e_i$ translates into zero or more source word-positions.

  $$a : \{\text{pos}_f\} \rightarrow (\{\text{pos}_e\} \cup \{0\})$$

- $a_i$ or $a(i)$, i.e., word position in $e$ with which $f_i$ is aligned.
Word Alignment Example
Word Alignment Example

The balance was the territory of the aboriginal people.
Le reste appartenait aux autochtones.
Word Alignment Example: Not covered in this setting

- These words:
  - The
  - poor
  - don’t
  - have
  - any
  - money

- These words:
  - Les
  - pauvres
  - sont
  - diminis

Limitations of PB Models
Syntax
Translation model with word alignment

\[
\arg\max_e P(e \mid f) = \arg\max_e P(e) \times P(f \mid e)
\]

\[
P(f \mid e) = \sum_a P(a, f \mid e) = \sum_a P(a \mid e) \times P(f \mid a, e)
\]

Questions

- How to parametrize the model?
  How are \(e\), \(f\) and \(a\) composed from basic units?

- How to train the model?
  How to acquire word alignment?

- How to translate with this model?
  Decoding and computational issues (for second part)
We need to decompose

- The alignment \( a \) and the length \( m: P(a \mid e) \)
- “Translation dictionary” \( P(f \mid e, a) \)
Word Alignment Models: General Scheme

Alignment of positions in \( f \) with positions in \( e \):
\[ a = a_1^m = a_1 \ldots a_m \]

Markov process over \( a \)

\[
P(a_1^m, f_1^m | e_1^l) = P(m | e) \times \\
\prod_{j=1}^{m} P(a_j | a_1^{j-1}, f_1^{j-1}, m, e) \times P(f_j | a_1^j, f_1^{j-1}, m, e)
\]

In words: to generate alignment \( a \) and foreign sentence \( f \)

1. Choose a length \( m \) for \( f \)
2. Generate alignment \( a_j \) given the preceding alignments, words in \( f, m, \) and \( e \)
3. Generate word \( f_j \) conditioned on structure so far and \( e \).

IBM models are obtained by simplifications of this formula.
IBM Model I

\[
P(a_{1}^{m}, f_{1}^{m} \mid e_{1} \ldots e_{l}) = P(m \mid e) \times \\
\prod_{j=1}^{m} P(a_{j} \mid a_{1}^{j-1}, f_{1}^{j-1}, m, e) \times P(f_{j} \mid a_{1}^{j}, f_{1}^{j-1}, m, e)
\]

IBM Model I:

**Length:** \( P(m \mid e) \approx \approx P(m \mid l) \approx = \epsilon \)  \( \text{A fixed probability } \epsilon. \)

**Align** with uniform probability \( j \) with any \( a_{j} \) in \( e_1^{l} \) or

**NULL:** \( P(a_{j} \mid a_{1}^{j-1}, f_{1}^{j-1}, m, e) \approx (l + 1)^{-1} \)

Note that \( a_{j} \) can be linked with \( l \) positions in \( e \) or with NULL.

**Lexicon:** lexicon parameters \( \pi_{t}(f \mid e) \)

\[
P(f_{j} \mid a_{1}^{j}, f_{1}^{j-1}, m, e) \approx P(f_{j} \mid e_{a_{j}}) = \pi_{t}(f_{j} \mid e_{a_{j}})
\]

Parameters: \( \epsilon \) and \( \{\pi_{t}(f \mid e) \mid \langle f, e \rangle \in C\}. \)
Sketch IBM Model I

- Label every f position with an e position
- Choose length m given l

Language Model on e

"Translation dictionary"
IBM Model I Parameters and Data Likelihood

Data Likelihood:

\[ P(f | e) = \sum_{a_1^m} P(a_1^m, f_1^m | e_1 \ldots e_l) \]

\[ = \frac{\epsilon}{(l+1)^m} \times \sum_{a_1=0}^{l} \ldots \sum_{a_m=0}^{l} \prod_{j=1}^{m} \pi_t(f_j | e_{a_j}) \]

Parameters: \( \epsilon \) and \( \{\pi_t(f | e) | \langle f, e \rangle \in C\} \).

Fix \( \epsilon \), i.e., in practice put a uniform probability over a range \([1..m]\), for some natural number \( m \).

Dilemma

To estimate these parameters we need word-alignment.
To get word-alignment we need these parameters.
IBM Model II

Extends IBM Model I at alignment probs:

$$P(a^m_1, f^m_1 \mid e_1 \ldots e_l) \approx \epsilon \times \prod_{j=1}^{m} P(a_j \mid a^{j-1}_1, f^{j-1}_1, m, e) \times \pi_t(f_j \mid e_{a_j})$$

IBM Model II: changes only one element in IBM Model I:

- IBM Model I does not take into account the position of words in both strings

$$P(a_j \mid a^{j-1}_1, f^{j-1}_1, m, e) = P(a_j \mid j, l, m) := \pi_A(a_j \mid j, l, m)$$

Where $\pi_A(\cdot \mid \cdot)$ are parameters to be learned from data.

IBM Models III, IV and V concentrate on more complex alignments allowing, e.g., $1 \to to \to n$ (fertility)
IBM Model II Parameters

\[ P(a^m_1, f^m_1 \mid e_1 \ldots e_l) \approx \epsilon \times \prod_{j=1}^{m} \pi_A(a_j \mid j, l, m) \times \pi_t(f_j \mid e_{a_j}) \]

Parameters: \( \{\pi_A(a_j \mid j, l, m)\} \) and \( \{\pi_t(f_j \mid e_{a_j})\} \)

Dilemma

To estimate these parameters we need word-alignment. To get word-alignment, we need these parameters.
Maximum-Likelihood Estimation of model M on parallel corpus C

$$\arg \max_{m \in M} P(C \mid m) = \arg \max_{m \in M} \prod_{\langle e, f \rangle \text{in } C} P_m(e \mid f)$$

Example IBM Model I:
- Model $M$ is defined by model parameters.
- Data is incomplete: no closed form solution.
- Expectation-Maximization (EM) sketch
  - Init: Set the parameters at some $m_0$ and let $i = 0$
  - Repeat until convergence (in perplexity)
    $$EM_i(C) = C \text{ completed using estimate } m_i$$
    $$EM_i(C) \text{ contains } m_i\text{-expectations over } \langle e, f, a \rangle: P(a \mid f, e)$$
    $$m_{i+1} = \text{Relative Frequency Estimates from } EM_i(C).$$
EM for Lexicon and Word Alignment Probs

... la maison ... la maison blue ... la fleur ...

... the house ... the blue house ... the flower ...
EM for Lexicon and Word Alignment Probs

... la maison ... la maison blue ... la fleur ...

... the house ... the blue house ... the flower ...
EM for Lexicon and Word Alignment Probs

... la maison ... la maison bleu ... la fleur ...

... the house ... the blue house ... the flower ...
EM for Lexicon and Word Alignment Probs

... la maison ... la maison bleu ... la fleur ...
... the house ... the blue house ... the flower ...
Translation Using EM Estimates

- Lexicon probability estimates: \( \{\hat{\pi}_t(f_j | e_{aj})\} \)
- Alignment probabilities: \( \{\hat{\pi}_A(a_j | j, m, l)\} \)
- Translation Model + Language Model + Decoder

\[
\arg\max_e P(e | f) = \arg\max_e P(e) \times \sum_a P(a, f | e)
\]
Viterbi Word-Alignment using EM estimates

After EM has stabilized on estimates

\[ \{ \hat{\pi}_t(f_j \mid e_{a_j}) \} \quad \text{and} \quad \{ \hat{\pi}_A(a_j \mid j, m, l) \} \]

For every \( \langle f, e \rangle \) in \( C \) apply the following

\[
\arg \max_{a_1^m} P(a_1^m, f_1^m \mid e_1 \dots e_l) \approx \arg \max_{a_1^m} \epsilon \times \prod_{j=1}^{m} \hat{\pi}_A(a_j \mid j, m, l) \times \hat{\pi}_t(f_j \mid e_{a_j})
\]
HMM Alignment Model: General Form

\[
P(a_1^m, f_1^m \mid e_1 \ldots e_l) \approx \epsilon \times \prod_{j=1}^{m} P(a_j \mid a_{j-1}^{j-1}, f_{j-1}^{j-1}, m, e) \times \pi_t(f_j \mid e_{a_j})
\]

- Words do not move independently of each other: condition word movement on previous word movement

\[
P(a_j \mid a_{j-1}^{j-1}, f_{j-1}^{j-1}, m, e) \approx P(a_j \mid a_{j-1}, m)
\]
IBM Model III (and IV): Example

- A hidden word-alignment $a$: $P(f | e) = \sum_a P(a, f | e)$

```
However, the sky remained clear under the strong north wind.

Fertility: (1)
1 2 1 0 1 1 1 0 0 1 1 1 1

Translate words independently (2)

Reorder words: (3)

Although north wind howls, but sky still extremely limpid.
```

Estimate alignment + lexicon + reordering + fertility parameters.
# Word-based Models (Och & Ney 2003)

## Table 1
Overview of the alignment models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Alignment model</th>
<th>Fertility model</th>
<th>E-step</th>
<th>Deficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>uniform</td>
<td>no</td>
<td>exact</td>
<td>no</td>
</tr>
<tr>
<td>Model 2</td>
<td>zero-order</td>
<td>no</td>
<td>exact</td>
<td>no</td>
</tr>
<tr>
<td>HMM</td>
<td>first-order</td>
<td>no</td>
<td>exact</td>
<td>no</td>
</tr>
<tr>
<td>Model 3</td>
<td>zero-order</td>
<td>yes</td>
<td>approximative</td>
<td>yes</td>
</tr>
<tr>
<td>Model 4</td>
<td>first-order</td>
<td>yes</td>
<td>approximative</td>
<td>yes</td>
</tr>
<tr>
<td>Model 5</td>
<td>first-order</td>
<td>yes</td>
<td>approximative</td>
<td>no</td>
</tr>
<tr>
<td>Model 6</td>
<td>first-order</td>
<td>yes</td>
<td>approximative</td>
<td>yes</td>
</tr>
</tbody>
</table>
Word-Alignment As Hidden Structure: Sufficient?

We assumed alignment between words and dictionary:

- Alignment $a$ and the length $m$: $P(a \mid e)$
- Dictionary $P(f \mid e, a)$
Limitations of word-based translation:

- Many-to-one and many-to-many is common: “Makes more difficult”/bemoeilijkt “Dat richtte (hen) ten gronde”/”That destroyed (them)”
- Reordering takes place (often) by whole blocks. Reordering individual words increases ambiguity. “The (big heavy) cow/la vaca (pesada grande)”
- Translation works by “fixed expressions” (idiomatic). Concatenating word-translations increases ambiguity.

Estimates of $P(f \mid e)$ by word-based models are inaccurate.

Instead of words as basic events: multi-word events in corpus.
Obtaining Symmetrized Word Alignments
Asymmetric Alignments: Limitations

- Word-based models presented so far are based on asymmetric word alignment. Each position $i$ in $f$ is aligned with at most one position in $e$: $a_i$
- What about such word alignments?
- Or when a word in $f$ translated into two or more in $e$?
Symmetrization Heuristic

Obtain $A_{f \rightarrow e}$ and $A_{e \rightarrow f}$
From Intersection $A_{f \rightarrow e} \cap A_{e \rightarrow f}$ to Union $A_{f \rightarrow e} \cup A_{e \rightarrow f}$
Symmetrization Heuristic Algorithm

Obtain $A_{f\rightarrow e}$ and $A_{e\rightarrow f}$
From Intersection $A_{f\rightarrow e} \cap A_{e\rightarrow f}$ to Union $A_{f\rightarrow e} \cup A_{e\rightarrow f}$

- From intersection $A = A_{f\rightarrow e} \cap A_{f\rightarrow e}$ to union $A_{f\rightarrow e} \cup A_{f\rightarrow e}$
- step 1 (diagonal): add neighbouring points $(f, e)$ in union s.t. $(f, e') \in A$ or $(f', e) \in A$
- step 2 (finalize): add remaining points in union s.t. $(f, e') \in A$ and $(f', e) \in A$
Phrase-based Models: Alignment between Phrases
Statistical “Memory-based” Translation

*Store arbitrary length source-target translation units from training parallel corpus.*

*Translate new input by “covering” it with translation units replayed from memory.*

**Idiomatic = Tiling: Phrase-Based SMT**

- Assume word-alignment \( a \) is given in parallel corpus.
- Phrase-pair = contiguous source-target \( \langle n, m \rangle \)-grams that are *translational equivalents* under \( a \).
- Estimate phrase-pair probabilities \( \Theta(f_i \mid e_i) \)
- Translate \( f \) by “tiling it with phrases with order permutation”
PBSMT some references

- Phrase-based statistical machine translation (Zens, Och and Ney 2002)
- Phrase based SMT (Koehn, Och and Marcu 2003)
- Joint Phrase-based SMT (Wang and Marcu 2005)

Relation to EBMT 1984; Data-Oriented Translation (2000).
Phrase-Based Models: Conceptual

Segment foreign sentence \( f \) into \( l \) phrases \( \tilde{f}_1 \)

\[
\text{arg max}_e P(e \mid f) = \text{arg max}_e P(e) \times P(f \mid e)
\]

\[
P(f \mid e) = \sum_{\langle \tilde{f}_1, \bar{e}_1 \rangle} P(\tilde{f}_1, \bar{e}_1 \mid e) \prod_{i=1}^{l} P(\tilde{f}_i \mid \bar{e}_i) \times d(\text{start}_i - \text{end}_i - 1 - 1)
\]

\[
\text{arg max}_e P(f \mid e) \approx \text{arg max}_{\langle \tilde{f}_1, \bar{e}_1 \rangle} \prod_{i=1}^{l} \Theta(\tilde{f}_i \mid \bar{e}_i) \times d(\text{start}_i - \text{end}_i - 1 - 1)
\]

\( \text{start}_i/\text{end}_i \) are positions of first/last words of \( \tilde{f}_i \) (translating to \( \bar{e}_i \)).

\( d(x) = \alpha^x \) exponentially decaying in words skipped (\( \alpha \in (0, 1] \)).
Phrase-Based Models: Linear-interpolation

Segment foreign sentence $f$ into $l$ phrases $\tilde{f}_1^l$

Log-linear interpolation of factors:

$$score(e|f) = \sum_{f \in F} \lambda_f \times \log H_f(e, f)$$

Where set $F$ consists of:

- Bag of phrases translation $= \prod_{i=1}^{l} \Theta(\tilde{f}_i | \tilde{e}_i)$
- $d(.)$ Phrases reordered with reordering model $d(.)$
- $lm$ Language model (5-grams or even 7-grams).
- other Smoothing + length penalty terms.
Topics to discuss

- Phrase table extraction
- Estimating \( \Theta(f_i | e_i) \) and \( \Theta(e_i | f_i) \)
- Lexicalized and hierarchical phrase reordering models
- Other: phrase, length penalty ...
- Log-linear interpolation and minimum error-rate training
- Decoding and optimization
Extracting phrase pairs

A phrase pair $\langle f, e \rangle$ is consistent with alignment $a$ iff

- Non-empty: at least one alignment pair from $a$ is in $\langle f, e \rangle$
- No foreign positions inside $\langle f, e \rangle$ aligned to positions outside it
- No english positions inside $\langle f, e \rangle$ aligned to positions outside it
Extracting phrase pairs
Extracting phrase pairs

Word Alignment Induced Phrases (2)

(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green)
Extracting phrase pairs

Word Alignment Induced Phrases (3)

(Maria, Mary), (no, did not), (slap, daba una bofetada), (a la, the), (bruja, witch), (verde, green), (Maria no, Mary did not), (no daba una bofetada, did not slap), (daba una bofetada a la, slap the), (bruja verde, green witch)
Extracting phrases from permutations
Phrase pair weights

Extract phrase pairs from corpus into multiset

\[ Tab = \{\langle f, e \rangle, \text{freq}(f, e)\} \]

Weights for \( \langle f, e \rangle \)

- \( \Theta(f | e) = \frac{\text{freq}(f, e)}{\sum_{\langle f', e \rangle \in Tab} \text{freq}(f', e)} \)
- \( \Theta(e | f) = \frac{\text{freq}(e, f)}{\sum_{\langle f, e' \rangle \in Tab} \text{freq}(f, e')} \)

Smoothing with lexical word alignment estimates from IBM models
Distance-Based Reordering Sketch

- **Diagram**: Shows a comparison between foreign and English phrases, illustrating the reordering process.

- **Table**:

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Translates</th>
<th>Movement</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1–3</td>
<td>start at beginning</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>skip over 4–5</td>
<td>+2</td>
</tr>
<tr>
<td>3</td>
<td>4–5</td>
<td>move back over 4–6</td>
<td>-3</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>skip over 6</td>
<td>+1</td>
</tr>
</tbody>
</table>

**Total**: 6
Lexicalized Reordering Sketch (Tillmann 2004)

Three types: monotone, swap, discontinuous

Condition on phrase pair: $p(o|e, f)$

Gives six features (3 orientations, current and next phrase pair).
Limited generalization over parallel data (1)

Non-productive Phrase Table: Phrase Variants?

**Morphological** e.g., changing inflection, agreement

<table>
<thead>
<tr>
<th>Arabic</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Al$arekat Alhindiyya</td>
<td>$areka hindiyya company Indian</td>
</tr>
<tr>
<td>the-companies the-Indian</td>
<td>(an) Indian company</td>
</tr>
<tr>
<td>the Indian companies</td>
<td></td>
</tr>
</tbody>
</table>

**Syntactic** e.g. adding adjective/proposition/adverbials

<table>
<thead>
<tr>
<th>English</th>
<th>Arabic</th>
</tr>
</thead>
<tbody>
<tr>
<td>the fish in the deep sea swims</td>
<td>the fish swims</td>
</tr>
</tbody>
</table>

**Reordering** minor reordering of same words not allowed

In Arabic V-S-O and S-V-O are allowed.

**Semantic** e.g. synonyms, paraphrases

Non-productive Phrase Table = Data Sparseness
Limited generalization over parallel data (2)

Reordering

Local, monotone, almost non-lexicalized reordering.

What about long range reordering?

Five phrases need to be reversed: see Chiang 2007 (J. CL).

Reordering target phrases with a coarse “source road map”?
Limitations: Data-Sparseness

Non-productive phrase table + Local, *Uncharted* reordering

⇓

Data-sparseness: Shorter phrases will apply down to word level.

⇓

Shorter phrases combined assuming independence.

⇓

Target phrase selection hard due to large hypotheses lattice
Target Language Model = Only “GLUE” over target phrases.

The Shorter the Phrases, the Greater the Risk
Idiomatic Approach: GOOD, BAD and UGLY

<table>
<thead>
<tr>
<th>Phrases as atomic units</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Good:</strong> Less ambiguity in lexical choice and reordering. Match-Retrieve exactly is largely safe.</td>
</tr>
<tr>
<td><strong>Bad:</strong> Weak generalization over data. No phrase variants, weak reordering</td>
</tr>
<tr>
<td><strong>Ugly:</strong> Fall-back on shorter phrases down to word-to-word. LM as “glue” is insufficient.</td>
</tr>
</tbody>
</table>

Idiomatic approach does not alleviate data-sparseness

How Should We Translate Novel Phrases?
Towards the land of bi-trees
Alignments between Tree Pairs, ITG, Hierarchical Models and Syntax
Hidden Structure of Translation: Tree Pairs
Reordering \( n \) words

- Permutations of \( n \) words: \( n! \)

- Surely **not all permutations** are needed! (Wu 1995)

- Use trees and allow permutations on the nodes? There is an exponential number of trees in \( n \)

- **ITG hypothesis** (Wu 1995)

  Assume binary trees with two operations

- Phrase-based forms of ITG (Chiang 2005; 2007): Hiero
Syntax-Driven Phrase Translation

Syntax-driven Re-Ordering

Hierarchical (ITG)

Linguistic Syntax

Is translation syntactically cohesive?

Reordering == Moving children in parse tree?

- Binary: monotone or inverted order at every node.
- Lexical elements can be phrase pairs.
- Covers word-alignments in parallel corpora?
Word order difference and syntax: Impression

Extracting phrase-pairs with gaps (hierarchical trees):

\[ X \rightarrow X_1 \text{ dar una bufetada a } X_2 / X_1 \text{ slap } X_2 \]

\[ X \rightarrow \text{ Maria no } X \text{ la bruja verde } / \text{ Mary did not } X \text{ the green witch} \]
ITG with syntactic labels

\[
\begin{align*}
S\text{BAR} & \rightarrow \{W\text{HNP} \ S\text{BAR} \backslash W\text{HNP}\} & (a) \\
S\text{BAR} \backslash W\text{HNP} & \rightarrow \{V\text{P} / N\text{P}^L \ N\text{P}^R\} & (b) \\
N\text{P}^R & \rightarrow \{N\text{P} \ PP\} & (c) \\
W\text{HNP} & \rightarrow W\text{HNP}_p & (d) \\
W\text{HNP}_p & \rightarrow \text{which} / \text{der} & (e) \\
V\text{P} / N\text{P}^L_p & \rightarrow V\text{P} / N\text{P}^L_p & (f) \\
V\text{P} / N\text{P}^L_p & \rightarrow \text{is} / \text{ist} & (g) \\
N\text{P}^R & \rightarrow N\text{P}^R_p & (h) \\
N\text{P}^R_p & \rightarrow \text{the solution} / \text{die Lösung} & (i) \\
N\text{P} & \rightarrow N\text{P}_p & (j) \\
N\text{P}_p & \rightarrow \text{the solution} / \text{die Lösung} & (k) \\
P\text{P} & \rightarrow P\text{P}_p & (l) \\
P\text{P}_p & \rightarrow \text{to the problem} / \text{für das Problem} & (m)
\end{align*}
\]
Part II: Trevor Cohn
Decoding algorithms and efficiency