Feature-Rich Translation by Quasi-Synchronous Lattice Parsing

Kevin Gimpel and Noah A. Smith
Introduction

- Two trends in machine translation research
  - Many approaches to decoding
    - Phrase-based
    - Hierarchical phrase-based
    - Tree-to-string
    - String-to-tree
    - Tree-to-tree
  - Regardless of decoding approach, addition of richer features can improve translation quality
Introduction

- Two trends in machine translation research
  - Many approaches to decoding
    - Phrase-based
    - Hierarchical phrase-based
    - Tree-to-string
    - String-to-tree
    - Tree-to-tree
  - Regardless of decoding approach, addition of richer features can improve translation quality
- Decoding algorithms are strongly tied to features permitted
konnten sie es übersetzen?
could you translate it?
konnten sie es übersetzen?

Could you translate it?

Phrase Table

1 konnten → could
2 konnten sie → could you
3 es übersetzen → translate it
4 sie es übersetzen → you translate it
5 es → it
6 ? → ?

...
konnten sie es übersetzen?

could you translate it?

Phrase Table

<table>
<thead>
<tr>
<th></th>
<th>konnten → could</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>konnten sie → could you</td>
</tr>
<tr>
<td>3</td>
<td>es übersetzen → translate it</td>
</tr>
<tr>
<td>4</td>
<td>sie es übersetzen → you translate it</td>
</tr>
<tr>
<td>5</td>
<td>es → it</td>
</tr>
<tr>
<td>6</td>
<td>? → ?</td>
</tr>
</tbody>
</table>

...
konnten sie es übersetzen?

could you translate it?

Phrase-Based Decoding

konnten → could
konnten sie → could you
es übersetzen → translate it
sie es übersetzen → you translate it
es → it
? → ?

...
Hierarchical Phrase-Based Decoding

0 konnten 1 sie 2 es 3 übersetzen 4 ?

could you translate it?

<table>
<thead>
<tr>
<th>SCFG Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  X → es übersetzen / translate it</td>
</tr>
<tr>
<td>2  X → es / it</td>
</tr>
<tr>
<td>3  X → übersetzen / translate</td>
</tr>
<tr>
<td>4  X → konnten sie X ? / could you X ?</td>
</tr>
<tr>
<td>5  X → konnten sie X₁ X₂ ? / could you X₂ X₁ ?</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>
Konnten sie es übersetzen?

Could you translate it?

SCFG Rules
1. X → es übersetzen / translate it
2. X → es / it
3. X → übersetzen / translate
4. X → konnten sie X? / could you X?
5. X → konnten sie X₁ X₂? / could you X₂ X₁?

...
konnten sie es übersetzen?

could you translate it?

Hierarchical Phrase-Based Decoding

<table>
<thead>
<tr>
<th>SCFG Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 X → es übersetzen / translate it</td>
</tr>
<tr>
<td>2 X → es / it</td>
</tr>
<tr>
<td>3 X → übersetzen / translate</td>
</tr>
<tr>
<td>4 X → konnten sie X? / could you X?</td>
</tr>
<tr>
<td>5 X → konnten sie X₁ X₂? / could you X₂ X₁?</td>
</tr>
</tbody>
</table>

...
Our goal:

An MT framework that allows as many features as possible without committing to any particular decoding approach
Overview

- Initial step towards a “universal decoder” that can permit any feature of source and target words/trees/alignments

- Experimental platform for comparison of formalisms, feature sets, and training methods

- Building blocks:
  - Quasi-synchronous grammar (Smith & Eisner 2006)
  - Generic approximate inference methods for non-local features (Chiang 2007; Gimpel & Smith 2009)
Outline

- Introduction
- Model
- Quasi-Synchronous Grammar
- Training and Decoding
- Experiments
- Conclusions and Future Work
\[ \langle t^*, \tau_t^*, a^* \rangle = \underset{\langle t, \tau_t, a \rangle}{\operatorname{argmax}} p(t, \tau_t, a \mid s, \tau_s) \]
Parameterization

\[ \langle t^*, \tau^*_t, a^* \rangle = \operatorname{argmax} p(t, \tau_t, a \mid s, \tau_s) \langle t, \tau_t, a \rangle \]

- We use a single globally-normalized log-linear model:

\[
p(t, \tau_t, a \mid s, \tau_s) = \frac{\exp\{\theta^\top g(s, \tau_s, a, t, \tau_t)\}}{\sum_{a', t', \tau'_t} \exp\{\theta^\top g(s, \tau_s, a', t', \tau'_t)\}}
\]

- Features can look at any part of any structure
Features

- Log-linear models allow “arbitrary” features, but in practice inference algorithms must be developed to support feature sets.

- Many types of features appear in MT:
  - lexical word and phrase mappings
  - N-gram and syntactic language models
  - distortion/reordering
  - hierarchical phrase mappings
  - syntactic transfer rules

- We want to use all of these!
Outline

- Introduction
- Model
- Quasi-Synchronous Grammar
- Training and Decoding
- Experiments
- Conclusions and Future Work
A quasi-synchronous grammar (QG) is a model of
\[ p(t, \tau_t, a \mid s, \tau_s) \]
Quasi-Synchronous Grammar
(Smith & Eisner 06)

- A quasi-synchronous grammar (QG) is a model of

\[ p(t, \tau_t, a \mid s, \tau_s) \]

- To model target trees, any monolingual formalism can be used
- We use a quasi-synchronous dependency grammar (QDG)
Quasi-Synchronous Grammar
(Smith & Eisner 06)

- A quasi-synchronous grammar (QG) is a model of
  \[ p(t, \tau_t, \alpha \mid s, \tau_s) \]

- $\tau_t$
  - To model target trees, any monolingual formalism can be used
  - We use a quasi-synchronous dependency grammar (QDG)

- $\alpha$
  - Each node in the target tree is aligned to zero or more nodes in the source tree (for a QDG, nodes = words)
  - Constraints on the alignments $\rightarrow$ synchronous grammar
  - In QG, departures from synchrony are penalized softly using features
Konnten sie es übersetzen?

Could you translate it?
For every parent-child pair in the target sentence, what is the relationship of the source words they are linked to?

$ konnten$ sie$ es$ übersetzen$ ?

$ could$ you$ translate$ it$ ?
For every parent-child pair in the target sentence, what is the relationship of the source words they are linked to?
For every parent-child pair in the **target** sentence, what is the relationship of the **source** words they are linked to?

$ \textit{konnten sie es übersetzen?} $  

$ \textit{could you translate it?} $  

**Parent-child**
For every parent-child pair in the **target** sentence, what is the relationship of the **source** words they are linked to?

All “parent-child” configurations $\rightarrow$ synchronous dependency grammar

\[
\begin{align*}
$ konnten$ & \quad\text{sie es übersetzen ?} \\
$ could$ & \quad\text{you translate it ?}
\end{align*}
\]
Many other configurations are possible:

$\text{wo kann ich untergrundbahnkarten kaufen ?}$

$\text{where can i buy subway tickets ?}$
Many other configurations are possible:

- Parent-child
- Child-parent
- Same node
- Sibling
- Grandparent/child
- Grandchild/parent
- C-Command
- Parent null
- Child null
- Both null
- Other
Coverage Features

- There are no hard constraints to ensure that all source words get translated.
- While QG has been used for several tasks, it has not previously been used for generation.
- We add **coverage features** and learn their weights.
<table>
<thead>
<tr>
<th>Coverage Feature</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word never translated</td>
<td>-2.21</td>
</tr>
<tr>
<td>Word translated that was translated at least $N$ times already:</td>
<td></td>
</tr>
<tr>
<td>$N = 0$</td>
<td>1.48</td>
</tr>
<tr>
<td>$N = 1$</td>
<td>-3.04</td>
</tr>
<tr>
<td>$N = 2$</td>
<td>-0.22</td>
</tr>
<tr>
<td>$N = 3$</td>
<td>-0.05</td>
</tr>
<tr>
<td>Coverage Feature</td>
<td>Weight</td>
</tr>
<tr>
<td>-------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Word never translated</td>
<td>-2.21</td>
</tr>
<tr>
<td>Word translated that was translated at least $N$ times already:</td>
<td></td>
</tr>
<tr>
<td>$N = 0$</td>
<td>1.48</td>
</tr>
<tr>
<td>$N = 1$</td>
<td>-3.04</td>
</tr>
<tr>
<td>$N = 2$</td>
<td>-0.22</td>
</tr>
<tr>
<td>$N = 3$</td>
<td>-0.05</td>
</tr>
</tbody>
</table>

![Graph showing the relationship between the number of times a word is translated and its score. The x-axis represents the number of times a word is translated, ranging from 0 to 5. The y-axis represents the score, ranging from -6 to 2. The graph shows a peak at $N = 1$ and a decline as $N$ increases.]
Outline

- Introduction
- Model
- Quasi-Synchronous Grammar
- Training and Decoding
- Experiments
- Conclusions and Future Work
Decoding

- A QDG induces a monolingual grammar for a source sentence whose language consists of all possible translations

- Decoding:
  - Build a weighted lattice encoding the language of this grammar
  - Perform lattice parsing with a dependency grammar
    - Extension of dependency parsing algs for strings (Eisner 97)
  - Integrate non-local features via cube pruning/decoding (Chiang 07, Gimpel & Smith 09)
konnten sie es übersetzen?
could you translate it?
$ konnten sie es übersetzen?
could you translate it?
Could you translate it?
konnten sie es übersetzen?
could you translate it?
$ konnten sie es übersetzen?
could you translate it?
konnten sie es übersetzen?

Could you translate it?
Recall that we use a single globally-normalized log-linear model:

\[
p(t, \tau_t, a \mid s, \tau_s) = \frac{\exp\{\theta^\top g(s, \tau_s, a, t, \tau_t)\}}{\sum_{a', t', \tau_t'} \exp\{\theta^\top g(s, \tau_s, a', t', \tau_t')\}}
\]

- If all structures are given, this becomes a convex, supervised learning problem
- If a structure is not given, it can be marginalized out during training (or simply ignored during both training and testing)
- Here, we assume alignments are not given and marginalize them out during training
Training

- Standard approach is to optimize conditional likelihood

\[
\text{LL}(\theta) = \sum_{i=1}^{N} \log p(t^{(i)}, \tau_{t}^{(i)} \mid s^{(i)}, \tau_{s}^{(i)})
\]

\[
= \sum_{i=1}^{N} \log \frac{\sum_{a} \exp\{\theta^{\top} g(s^{(i)}, \tau_{s}^{(i)}, a, t^{(i)}, \tau_{t}^{(i)})\}}{\sum_{t, \tau_{t}, a} \exp\{\theta^{\top} g(s^{(i)}, \tau_{s}^{(i)}, a, t, \tau_{t})\}}
\]

*problem: must sum over words + trees + alignments!*
Pseudo-likelihood

Solution: optimize pseudo-likelihood (Besag, 1975) by making the following approximation:

\[ p(t, \tau_t \mid s, \tau_s) \approx p(t \mid \tau_t, s, \tau_s) \times p(\tau_t \mid t, s, \tau_s) \]

The objective function becomes:

\[
\text{PL}(\theta) = \sum_{i=1}^{N} \log \left( \sum_{a} p(t^{(i)}, a \mid \tau_t^{(i)}, s^{(i)}, \tau_s^{(i)}) \right) + \sum_{i=1}^{N} \log \left( \sum_{a} p(\tau_t^{(i)}, a \mid t^{(i)}, s^{(i)}, \tau_s^{(i)}) \right)
\]

Integrate non-local features via “cube summing” [Gimpel & Smith 09]
Outline

- Introduction
- Model
- Quasi-Synchronous Grammar
- Training and Decoding
- Experiments
- Conclusions and Future Work
Experiments

One of our goals was an experimental platform to address questions like the following:

- How do phrase features interact with syntactic features?
- How do synchronous (isomorphism) constraints affect translation quality?
- How do string-to-tree, tree-to-string, and tree-to-tree approaches compare in terms of runtime and translation quality?
- Does a small number of feature templates work better than a large number of binary features?
- How do MERT/MIRA compare with optimization of conditional likelihood?
Experiments

One of our goals was an experimental platform to address questions like the following:

- How do phrase features interact with syntactic features?
- How do synchronous (isomorphism) constraints affect translation quality?
- How do string-to-tree, tree-to-string, and tree-to-tree approaches compare in terms of runtime and translation quality?
- Does a small number of feature templates work better than a large number of binary features?
- How do MERT/MIRA compare with optimization of conditional likelihood?
Experimental Setup

- **Data**
  - German-English Basic Travel Expression Corpus (BTEC)
  - Only sentences of length ≤ 15
  - 80k sentences for training, 1k for tuning, 500 for testing

- **Features**
  - Parsed source and target text using Stanford parser
  - Phrase extraction using Moses (max phrase length = 3)
  - Trigram language model
Experiments

- This is not a state-of-the-art MT system
  - Moses obtains 68.4 BLEU and 85.2 METEOR on this dataset
  - Our best scores are 52 BLEU and 75 METEOR
Features

Lexical Translation
\[ p(s \mid t) \]
\[ p(t \mid s) \]

Language Model
Bigram Probability
Trigram Probability

Reordering
Absolute Distortion

Coverage
Word Left Untranslated
Used Word Already Used N Times
(N in \{0, 1, 2, 3\})

Phrase Translation
\[ p(s \mid t) \]
\[ p(t \mid s) \]
\[ lex(s \mid t) \]
\[ lex(t \mid s) \]

Target Dependency
words & word classes
\[ p(root) \]
\[ p(child \mid parent, left) \]
\[ p(child \mid parent, right) \]
(+ 4 valence distributions)

QG Configuration
(14 binary features, one for each configuration)
# Features

## Lexical Translation
- $p(s \mid t)$
- $p(t \mid s)$

## Language Model
- Bigram Probability
- Trigram Probability

## Reordering
- Absolute Distortion

## Coverage
- Word Left Untranslated
- Used Word Already Used NTimes
  ($N$ in $\{0,1,2,3\}$)

## Phrase Translation
- $p(s \mid t)$
- $p(t \mid s)$
- $lex(s \mid t)$
- $lex(t \mid s)$

## Target Dependency
- \[ p(root) \]
- \[ p(child \mid parent, left) \]
- \[ p(child \mid parent, right) \]
  (+ 4 valence distributions)

## QG Configuration
- (14 binary features, one for each configuration)
## Feature Set Comparison: BLEU Scores

<table>
<thead>
<tr>
<th></th>
<th>No Syntax Features</th>
<th>Target Syntax Features Only</th>
<th>Source &amp; Target Syntax Features</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>No Phrase Features</strong></td>
<td>37.3</td>
<td>44.6</td>
<td>44.2</td>
</tr>
<tr>
<td><strong>Phrase Features</strong></td>
<td>46.8</td>
<td>49.7</td>
<td>51.4</td>
</tr>
</tbody>
</table>
## QDG Configuration Comparison

<table>
<thead>
<tr>
<th>Configuration</th>
<th>BLEU</th>
<th>METEOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>synchronous</td>
<td>40.1</td>
<td>69.5</td>
</tr>
<tr>
<td>+ nulls, root-any</td>
<td>41.1</td>
<td>69.3</td>
</tr>
<tr>
<td>+ child-parent, same-node</td>
<td>43.4</td>
<td>68.2</td>
</tr>
<tr>
<td>+ sibling</td>
<td>48.8</td>
<td>72.2</td>
</tr>
<tr>
<td>+ grandparent/child</td>
<td>50.2</td>
<td>73.7</td>
</tr>
<tr>
<td>+ c-command</td>
<td>51.6</td>
<td>74.4</td>
</tr>
<tr>
<td>+ other</td>
<td>51.4</td>
<td>74.7</td>
</tr>
</tbody>
</table>
Conclusions and Ongoing Work

- We have described an MT system based on quasi-synchronous grammar that can use features from many types of MT systems.

- We reported on preliminary experiments comparing feature sets and synchronous dependency constraints.

- Ongoing work in improving decoder efficiency, adding features, and conducting additional experiments.
Thanks!