SPEECH TRANSLATION: THEORY AND PRACTICES

Bowen Zhou and Xiaodong He

zhou@us.ibm.com
xiaohe@microsoft.com

May 27th 2013 @ ICASSP 2013
Universal Translator: dream (will) come true or yet another over promise?

- Spoken communication without a language barrier: mankind’s long-standing dreams
  - Translate human speech in one language into another (text or speech)
  - Automatic Speech Recognition: ASR
  - (Statistical) Machine Translation: (S)MT
- We’ve been promised by Sci-fi (Star Trek) for 5 decades
- Serious research efforts have started in early 1990s’
  - When statistical ASR (e.g., HMM-based) starts showing dominance and SMT was emerging
ST Research Evolves Over 2 Decades

<table>
<thead>
<tr>
<th>PROJECT/CAMPAIGN</th>
<th>Active Period</th>
<th>SCOPE, SCENARIOS AND PLATFORMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-STAR</td>
<td>1992-2004</td>
<td>One/Two-way Limited domain spontaneous speech</td>
</tr>
<tr>
<td>VERBMOBIL</td>
<td>1993-2000</td>
<td>One-way Limited domain conversational speech</td>
</tr>
<tr>
<td>DARPA BABYLON</td>
<td>2001-2003</td>
<td>Two-way Limited domain spontaneous speech; Laptop/Handheld</td>
</tr>
<tr>
<td>DARPA TRANSTAC</td>
<td>2005-2010</td>
<td>Two-way small domain spontaneous dialog; Mobile/Smartphone; One-way; Limited domain dialog and unlimited free-style talk</td>
</tr>
<tr>
<td>IWSLT</td>
<td>2004-</td>
<td>One-way; Broadcast speech, political speech; Server-based</td>
</tr>
<tr>
<td>TC-STAR</td>
<td>2004-2007</td>
<td>One-way Broadcast speech, conversational speech; Server-based</td>
</tr>
<tr>
<td>DARPA GALE</td>
<td>2006-2011</td>
<td>One-way Broadcast speech, conversational speech; Server-based</td>
</tr>
<tr>
<td>DARPA BOLT</td>
<td>2011-</td>
<td>One/two-way Conversational speech; Server-based</td>
</tr>
</tbody>
</table>
ST Research Evolves Over 2 Decades

- Broad domains/large vocabulary
- Limited domain
- Formal/concise
- Desktop/laptop
- Conversational/Interactive
- Mobile/Cloud
ST Research Evolves Over 2 Decades

- Broad domains/large vocabulary
- Limited domain
- Formal/concise
- Desktop/laptop
- Conversational/Interactive
- Mobile/Cloud
Are we closer today?

- Speech translation demos:
  - Completely smartphone-hosted speech-to-speech translation (IBM Research)
Are we closer today?

- Speech translation demos:
  - Completely smartphone-hosted speech-to-speech translation (IBM Research)
  - Rashid’s live demo (MSR) in 21st CCC keynote
Are we closer today?

• Speech translation demos:
  – Completely smartphone-hosted speech-to-speech translation (IBM Research)
  – Rashid’s live demo (MSR) in 21st CCC keynote

• Statistical approaches to both ASR/SMT are one of the keys for the success
  – Large data, discriminative training, better models (e.g., DNN for ASR) etc.
Our Objectives

1. Review & analyze state-of-the-art SMT theory, *from ST’s perspective*

2. Unifying ASR and SMT to catalyze joint ASR/SMT for improved ST

3. ST’s *practical issues* & future research topics
In this talk...

1. Overview: ASR, SMT, ST, Metric
2. Learning Problems in SMT
3. Translation Structures for ST [20 min coffee Break]
4. Decoding: A Unified Perspective ASR/SMT
5. Coupling ASR/SMT: Decoding & Modeling
6. Practices
7. Future Directions
In this talk...

1. Overview: ASR, SMT, ST, Metric
2. Learning Problems in SMT
3. Translation Structures for ST [20 min coffee Break]
4. Decoding: A Unified Perspective ASR/SMT
5. Coupling ASR/SMT: Decoding & Modeling
6. Practices
7. Future Directions

• Technical papers accompanying this tutorial:

• Papers are available on authors’ webpages
• Charts coming up soon
Overview & Backgrounds
Speech Translation Process

- **Input:** a source speech signal sequence \( x_T^1 = x_1, \ldots, x_T \)
- **ASR:** recognizes it as a set of source word sequences, \( \{ f_{1j}^j = f_1, \ldots, f_j \} \)
- **SMT:** Translated into the target language sequence of words \( e_I^1 = e_1, \ldots, e_I \)

\[
\hat{e}_I^1 = \arg\max_{e_I^1} P(e_I^1|x_T^1)
\]
\[
= \arg\max_{e_I^1} \left\{ \sum_{f_I^j} P(e_I^1|f_I^j)P(x_T^1|f_I^j)P(f_I^j) \right\}
\]
First comparison: ASR vs. SMT

- **Connection**: sequential pattern recognition
  - Determine a sequence of symbols that is regarded as the optimal equivalent in the target domain
    - ASR: $x_1^T \rightarrow f_1^J$
    - SMT $f_1^J \rightarrow e_1^I$.
  - Hence, many techniques are closely related.
First comparison: ASR vs. SMT

**Connection**: sequential pattern recognition

- Determine a sequence of symbols that is regarded as the optimal equivalent in the target domain

  - ASR: $x_1^T \rightarrow f_1^l$
  - SMT $f_1^l \rightarrow e_1^l$.

- Hence, many techniques are closely related.

**Difference**: ASR is monotonic but SMT is not

- Different modeling/decoding formalisms required
- One of the key issues we address in this tutorial
ASR in a Nutshell

- **ASR Decoding**
  - $\hat{f}_1^j = \arg\max_{f_1^j} P(x_1^T | f_1^j)P(f_1^j)$

- **Acoustic Models (HMM)**
  - $P(x_1^T | f_1^j) = \sum_{q_1^T} \prod_t p(q_t | q_{t-1}) \prod_t p(x_t | q_t)$
  - $p(x_t | q_t)$ by Gaussian Mixture Models or NN

- **Language Models (N-gram)**
  - $p(f_1^j) \approx \prod_{j=1}^I p(f_j | f_{j-N+1}, \ldots, f_{j-1})$
ASR Structures: A Finite-State Problem
ASR Structures: A Finite-State Problem

\[ G: \text{a weighted acceptor that assigns language model probabilities.} \]
ASR Structures: A Finite-State Problem

$G$: a weighted acceptor that assigns language model probabilities.

$L$: transduces context-independent phonetic sequences into words.
ASR Structures: A Finite-State Problem

Search on a composed finite-state space (graph)

$$f_1^L = \text{best_path} \left( O \circ H \circ C \circ L \circ G \right)$$

**L**: transduces context-independent phonetic sequences into words

**G**: a weighted acceptor that assigns language model probabilities.
A Brief History of SMT?

- MT research traced back to 1950s
  - A pioneering work based on source-channel model
  - Same approach succeeded for ASR at IBM and elsewhere
  - A good confluence example of speech and language communities to drive the transition:
    - Rationalism $\Rightarrow$ Empiricism (Church & Mercer, 1993; Knight, 1999)
- A sequence of unsupervised word alignment models
  - Known today as IBM Models 1-5 (Brown et al, 1993)
- Much progress has been made since then
  - Key topics for today’s talk
A Bird view of SMT
How to know Translation X is better than Y?

• It’s a hard problem by itself
  – More than one good translation & even more bad ones
• Assumption: “closer” to human translation(s), the better
• Metrics were proposed to measure the *closeness*, e.g., BLEU, which measures n-gram precisions (Papineni et al., 2002)

\[
    \text{BLEU-}4 = \text{BP} \cdot \exp\left( \frac{1}{4} \sum_{n=1}^{4} \log(p_n) \right)
\]

• ST measurement is more complicated
  – Objective: WER+BLEU (or METEOR, TER etc)
  – Subjective: HTER (GALE/BOLT), concept transfer rate (TransTac/BOLT)
Further Reading

• ASR Backgrounds

• SMT Backgrounds

• SMT Metrics:

• History of MT and SMT:
Learning Problems in SMT

1. Overview: ASR, SMT, ST, Metric
2. Learning Problems in SMT
3. Translation Structures for ST
4. Decoding: A Unified Perspective ASR/SMT
5. Coupling ASR/SMT: Decoding & Modeling
6. Practices
7. Summary and Future Directions
Learning Translation Models

- Word alignment
- From words to phrases
- Log-linear model framework to integrate component models (a.k.a. features)
- Log-linear model training to optimize translation metrics (e.g., MERT)
Word Alignment

Given a set of parallel sentence pairs (e.g., ENU-CHS), find the word-to-word alignment between each sentence pair.

*English*: Finish the task of recruiting students

*Chinese*: 完成 招生 工作

......

word translation models
Word Alignment

Given a set of parallel sentence pairs (e.g., ENU-CHS), find the word-to-word alignment between each sentence pair.

*English*: Finish the task of recruiting students

*Chinese*: 完成 招生 工作

......

word translation models
HMM for Word Alignment

Each word in the Chinese sentence is modeled as a HMM state. The English words, the *observation*, are generated by the HMM one by one.

The generative story for word alignment:

At each step, the current state (a Chinese word) emits one English word; then jump to the next state.

Similar to ASR, HMM can be used to model the word-level translation process. Unlike ASR, note the non-monotonic jumping between states, and the NULL state.

(Vogel et al., 1996)
HMM for Word Alignment

Each word in the Chinese sentence is modeled as a HMM state. The English words, the *observation*, are generated by the HMM one by one.

Similar to ASR, HMM can be used to model the word-level translation process. Unlike ASR, note the non-monotonic jumping between states, and the NULL state.

(Vogel et al., 1996)

The generative story for word alignment:

At each step, the current state (a Chinese word) emits one English word; then jump to the next state.
HMM for Word Alignment

Each word in the Chinese sentence is modeled as a HMM state. The English words, the observation, are generated by the HMM one by one.

The generative story for word alignment:

At each step, the current state (a Chinese word) emits one English word; then jump to the next state.

Similar to ASR, HMM can be used to model the word-level translation process. Unlike ASR, note the non-monotonic jumping between states, and the NULL state.

(Vogel et al., 1996)
HMM for Word Alignment

Each word in the Chinese sentence is modeled as a HMM state. The English words, the *observation*, are generated by the HMM one by one.

The generative story for word alignment:

At each step, the current state (a Chinese word) emits one English word; then jump to the next state.

Similar to ASR, HMM can be used to model the word-level translation process. Unlike ASR, note the non-monotonic jumping between states, and the NULL state.

(Vogel et al., 1996)
HMM for Word Alignment

Each word in the Chinese sentence is modeled as a HMM state. The English words, the *observation*, are generated by the HMM one by one.

The generative story for word alignment:
At each step, the current state (a Chinese word) emits one English word; then jump to the next state.

Similar to ASR, HMM can be used to model the word-level translation process. Unlike ASR, note the non-monotonic jumping between states, and the NULL state.

(Vogel et al., 1996)
HMM for Word Alignment

Each word in the Chinese sentence is modeled as a HMM state. The English words, the *observation*, are generated by the HMM one by one.

The generative story for word alignment:

At each step, the current state (a Chinese word) emits one English word; then jump to the next state.

Similar to ASR, HMM can be used to model the word-level translation process. Unlike ASR, note the non-monotonic jumping between states, and the NULL state.

(Vogel et al., 1996)
HMM for Word Alignment

Each word in the Chinese sentence is modeled as a HMM state. The English words, the *observation*, are generated by the HMM one by one.

The generative story for word alignment:

At each step, the current state (a Chinese word) emits one English word; then jump to the next state.

Similar to ASR, HMM can be used to model the word-level translation process. Unlike ASR, note the non-monotonic jumping between states, and the NULL state.

(Vogel et al., 1996)
Formal Definition
Formal Definition

\[ f_1^J = f_1, \ldots, f_J : \text{source sentence (observation)} \]

- \( f_j \): source word at position \( j \)
- \( J \): length of source sentence
Formal Definition

\[ f_1^J = f_1, \ldots, f_J : \text{source sentence (observation)} \]
\[ f_j : \text{source word at position } j \]
\[ J : \text{length of source sentence} \]

\[ e_1^I = e_1, \ldots, e_I : \text{target sentence (states of HMM)} \]
\[ e_i : \text{target word at position } i \]
\[ I : \text{length of target sentence} \]
Formal Definition

\[ f_1^J = f_1, \ldots, f_J : \text{source sentence (observation)} \]
\[ f_j : \text{source word at position } j \]
\[ J : \text{length of source sentence} \]

\[ e_1^I = e_1, \ldots, e_I : \text{target sentence (states of HMM)} \]
\[ e_i : \text{target word at position } i \]
\[ I : \text{length of target sentence} \]

\[ a_1^J = a_1, \ldots, a_J : \text{alignment (state sequence)} \]
\[ a_j \in [1, I] : f_j \leftrightarrow e_{a_j} \]
Formal Definition

\[ f_1^J = f_1, \ldots, f_J : \text{source sentence (observation)} \]
- \( f_j \): source word at position \( j \)
- \( J \): length of source sentence

\[ e_1^I = e_1, \ldots, e_I : \text{target sentence (states of HMM)} \]
- \( e_i \): target word at position \( i \)
- \( I \): length of target sentence

\[ a_1^J = a_1, \ldots, a_J : \text{alignment (state sequence)} \]
- \( a_j \in [1, I]: f_j \leftrightarrow e_{a_j} \)

\( p(f|e) \): word translation probability
HMM Formulation

Given $f_{i}^{j}$ and $e_{i}^{l}$, $a_{i}^{j}$ is treated as “hidden variable”

$$p(f_{i}^{j} | e_{i}^{l}) = \sum \prod_{j=1}^{J} \left[ p(a_{j} | a_{j-1}, I) p(f_{j} | e_{a_{j}}) \right]$$

Model assumption:

- the emission probability only depends on the target word
- the transition probability only depends on the position of the last state – *relative distortion*
HMM Formulation

Given \( f_{i}^{j} \) and \( e_{i}^{I} \), \( a_{i}^{j} \) is treated as “hidden variable”

\[
p(f_{1}^{J} | e_{1}^{I}) = \sum a_{1}^{j} \prod_{j=1}^{J} \left[ p(a_{j} | a_{j-1}, I) p(f_{j} | e_{a_{j}}) \right]
\]

Emission probability:
Model the word-to-word translation

Model assumption:

- the emission probability only depends on the target word
- the transition probability only depends on the position of the last state – relative distortion
HMM Formulation

Given $f_{1}^{J}$ and $e_{1}^{I}$, $a_{1}^{J}$ is treated as “hidden variable.”

$$p(f_{1}^{J} | e_{1}^{I}) = \sum \prod_{a_{1}^{J}}^{J} \left[ p(a_{j} | a_{j-1}, I) p(f_{j} | e_{a_{j}}) \right]$$

Emission probability: Model the word-to-word translation

Transition probability: Model the distortion of the word order

Model assumption:

- the emission probability only depends on the target word
- the transition probability only depends on the position of the last state – *relative distortion*
ML Training of HMM

Maximum likelihood training:

\[ \Lambda_{ML} = \arg \max_{\Lambda} p(f_1^J | e_1^I, \Lambda) \]

\[ \Lambda = \{ p(a_j = i | a_{j-1} = i', I), p(f_j = f | e_i = e) \} \]

Expectation-Maximization (EM) training for \( \Lambda \).

efficient Forward-Backward algorithm exists.
Find the Optimal Alignment

• Viterbi decoding:

\[ \hat{a}_1^j = \arg \max_{a_1^j} \prod_{j=1}^{J} \left[ p(a_j | a_{j-1}, I) p(f_j | e_{a_j}) \right] \]

• other variations:
  posterior probability based decoding
  max posterior mode decoding
IBM Model 1-5

- IBM model 1-5
  - A series of generative models (Brown et al., 1994)
  - Summarized below (along with the HMM)

<table>
<thead>
<tr>
<th>Model</th>
<th>property</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM-1</td>
<td>Convex model (non-strictly). E-M training. Doesn’t model word order distortion.</td>
</tr>
<tr>
<td>HMM</td>
<td>Relative distortion model (capture the intuition that words move in groups) Efficient E-M training algorithm</td>
</tr>
<tr>
<td>IBM-2</td>
<td>IBM-1 plus an absolute distortion model (measure the divergence of the target word’s position from the ideal monotonic alignment position)</td>
</tr>
<tr>
<td>IBM-3</td>
<td>IBM-2 plus a fertility model (models the number of words that a state generates) Approximate parameter estimation due to the fertility model, costly training</td>
</tr>
<tr>
<td>IBM-4</td>
<td>Like IBM-3, but IBM-4 uses relative distortion model</td>
</tr>
<tr>
<td>IBM-5</td>
<td>IBM-5 addressed the model deficiency issue (while IBM-3&amp;4 are deficient).</td>
</tr>
</tbody>
</table>
## Other Word Alignment Models

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Models</th>
<th>Features/Advantages</th>
</tr>
</thead>
</table>
| Extensions of HMM           | (Tantounova et al. 2002) (Liang et al. 2006) (Deng & Byrne 2005) (He 2007) | • Model the fertility implicitly.  
• Regularize the alignment by agreements of two directions.  
• Better distortion model. |
| Discriminative models       | (Moore et al. 2006) (Taskar et al. 2005) | • Large linear model with many features.  
• Discriminative learning based on annotation |
| Syntax-driven models        | (Zhang & Gildea 2005) (Fossum et al. 2008) (Haghighi et al. 2009) | • Use syntactic features, and/or syntactically structured models |
From Word to Phrase Translation

- Extract phrase translation pairs from word alignment

<table>
<thead>
<tr>
<th>Source phrase</th>
<th>Target phrase</th>
<th>Feature $h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>完成</td>
<td>finish</td>
<td>...</td>
</tr>
<tr>
<td>招生</td>
<td>recruiting</td>
<td>students</td>
</tr>
<tr>
<td>工作</td>
<td>task</td>
<td></td>
</tr>
<tr>
<td>工作</td>
<td>the task</td>
<td></td>
</tr>
<tr>
<td>招生 工作</td>
<td>task of</td>
<td>recruiting</td>
</tr>
<tr>
<td></td>
<td>recruiting</td>
<td>students</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Och & Ney 2004; Koehn, Och and Marcu, 2003)
# Common Phrase Translation Models

<table>
<thead>
<tr>
<th>Model name</th>
<th>Parameterization (decomposed form)</th>
<th>Scoring at the sentence level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward phrase translation model</td>
<td>$p(\tilde{e}</td>
<td>\tilde{f}) = \frac{#(\tilde{e}, \tilde{f}) - d}{#(\ast, \tilde{f})}$</td>
</tr>
<tr>
<td>Forward lexical translation model</td>
<td>$p(e</td>
<td>f) = \frac{\gamma(e, f)}{\gamma(\ast, f)}$</td>
</tr>
<tr>
<td>Backward phrase translation model</td>
<td>Reverse version of the forward counterpart</td>
<td>$h_{bp} = \prod_k p(\tilde{f}_k</td>
</tr>
<tr>
<td>Backward lexical translation model</td>
<td>Reverse version of the forward counterpart</td>
<td>$h_{bl} = \prod_k \prod_n \sum_m p(f_{k,n}</td>
</tr>
</tbody>
</table>
Integration of All Component Models

• Use a log-linear model to integrate all component models (also known as features)

\[ P(E|F) = \frac{1}{Z} \exp \left\{ \sum_{i} \lambda_i h_i(E, F) \right\} \]

– Common features include:
  • Phrase translation models (e.g., the four models discussed before)
  • One or more language models in the target language
  • Counts of words and phrases
  • Distortion model – models word ordering

– All features need to be decomposable to a word/n-gram/phrase level

• Select the translation by the best integrated score

\[ \hat{E} = \arg \max_{E} P(E|F) = \arg \max_{E} \sum_{i} \lambda_i h_i(E, F) \]

(Och & Ney 2002)
Training Feature Weights

Let’s denote by $\lambda$ as $\{\lambda_i\}$, $\lambda$ is trained to optimize translation performance

$$
\hat{\lambda} = \arg\max_{\{\lambda\}} \text{BLEU}(\hat{E}(\lambda, F), E^*)
$$

$$
= \arg\max_{\{\lambda\}} \text{BLEU} \left( \arg\max_{E} \sum_{i} \lambda_i h_i(E, F), E^* \right)
$$

Non-convex problem!

(Och 2003)
Minimum Error Rate Training

- Given a n-best list \( \{E\} \), find the best \( \lambda \)
  - E.g., the top scored \( E \) gives the best BLEU.

<table>
<thead>
<tr>
<th>n-best</th>
<th>( h_1 )</th>
<th>( h_2 )</th>
<th>( h_3 )</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_1 )</td>
<td>( v_{1,1} )</td>
<td>( v_{1,2} )</td>
<td>( v_{1,3} )</td>
<td>( B_1 )</td>
</tr>
<tr>
<td>( E_2 )</td>
<td>( v_{2,1} )</td>
<td>( v_{2,2} )</td>
<td>( v_{2,3} )</td>
<td>( B_2 )</td>
</tr>
<tr>
<td>( E_3 )</td>
<td>( v_{3,1} )</td>
<td>( v_{3,2} )</td>
<td>( v_{3,3} )</td>
<td>( B_3 )</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

MERT (Och 2003):
The score of \( E \) is linear to the value of the \( \lambda \) that is of interest.
Only need to check several key values of \( \lambda \) where lines intersect with each other, since the ranking of hypotheses does not change between the two adjunct key values:

\[ E_1: S = v_{1,1} \lambda_1 + C_1 \]
\[ E_2: S = v_{2,1} \lambda_1 + C_2 \]
\[ E_3: S = v_{3,1} \lambda_1 + C_3 \]

Illustration of the MERT algorithm

\[ S = \sum_i \lambda_i h_i \]

\[ a \quad b \quad c \quad \lambda_1 \]

\( e.g., \text{when } \lambda_1 < a, E_3 \text{ is top scored}; \text{when } a < \lambda_1 < c, E_2 \text{ is top scored}; ... \)
More Advanced Training

• Use a large set of (sparse) features
  – E.g., integrate lexical, POS, syntax features
  – MERT is not effective anymore, use MIRA, PRO etc. for training of feature weights (Watanabe et al., 2007, Chiang et al. 2009, Hopkins & May 2011, Simianer et al. 2012)

• Better estimation of (dense) translation models
  – Train the translation probability distributions discriminatively (He & Deng 2012, Setiawan & Zhou 2013)

• A mix of the above two approaches
  – Build a set of new features for each phrase pair, and train them discriminatively by max expected BLEU (Gao & He 2013)
Further Reading

• Y. Deng and W. Byrne, 2005, HMM Word and Phrase Alignment For Statistical Machine Translation, in Proceedings of HLT/EMNLP.
• J. Gao and X. He, 2013, Training MRF-Based Phrase Translation Models using Gradient Ascent, in Proceedings of NAACL
• X. He and L. Deng, 2012, Maximum Expected BLEU Training of Phrase and Lexicon Translation Models , in Proceedings of ACL
• P. Liang, B. Taskar, and D. Klein, 2006, Alignment by Agreement, in Proceedings of NAACL.
• H. Setiawan and B. Zhou. 2013. Discriminative Training of 150 Million Translation Parameters and Its Application to Pruning, NAACL.
Translation Structures for ST

1. Overview: ASR, SMT, ST, Metric
2. Learning Problems in SMT
3. Translation Structures for ST
4. Decoding: A Unified Perspective ASR/SMT
5. Coupling ASR/SMT: Decoding & Modeling
6. Practices
7. Future Directions
Translation Equivalents (TE)

• Usually represented by some synchronous (source and target) grammar.
Translation Equivalents (TE)

- Usually represented by some synchronous (source and target) grammar.

<table>
<thead>
<tr>
<th>初步 试验</th>
<th>initial experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>的 成功</td>
<td>the success of</td>
</tr>
</tbody>
</table>

初步 $X_1 \leftrightarrow$ initial $X_1$
$X_1$ 的 $X_2 \leftrightarrow$ The $X_2$ of $X_1$
Translation Equivalents (TE)

• Usually represented by some synchronous (source and target) grammar.
• The grammar choices limited by two factors.
  – Expressiveness: is it adequate to model linguistic equivalence between natural language pairs?
  – Computational complexity: is it practical to build machine translation solutions upon it?

<table>
<thead>
<tr>
<th>初步 试验</th>
<th>initial experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>的 成功</td>
<td>the success of</td>
</tr>
</tbody>
</table>

初步 $X_1$ ↔ initial $X_1$
$X_1$ 的 $X_2$ ↔ The $X_2$ of $X_1$
Translation Equivalents (TE)

- Usually represented by some synchronous (source and target) grammar.
- The grammar choices limited by two factors.
  - **Expressiveness**: is it adequate to model linguistic equivalence between natural language pairs?
  - **Computational complexity**: is it practical to build machine translation solutions upon it?
- Need to balance both, particularly for ST,
  - Robust to informal spoken language
  - Critical speed requirement due to ST’s interactive nature
  - Additional bonus if it permits effectively and efficiently integration with ASR

<table>
<thead>
<tr>
<th>初步试验</th>
<th>initial experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>成功</td>
<td>the success of</td>
</tr>
</tbody>
</table>

初步 $X_1 \leftrightarrow$ initial $X_1$
$X_1$ 的 $X_2 \leftrightarrow$ The $X_2$ of $X_1$
Two Dominating Categories of TEs

- Finite-state-based formalism
  - E.g., phrase-based SMT (Och and Ney, 2004; Koehn et al., 2003)
Two Dominating Categories of TEs

- **Finite-state-based formalism**
  - E.g., phrase-based SMT (Och and Ney, 2004; Koehn et al., 2003)

- **Synchronous context-free grammar-based formalism**
  - hierarchical phrase-based (Chiang, 2007),
  - tree-to-string (e.g., Quirk et al., 2005; Huang et al., 2006),
  - forest-to-string (Mi et al., 2008)
  - string-to-tree (e.g., Galley et al., 04; Shen et al., 08)
  - tree-to-tree (e.g., Eisner 2003; Cowan et al, 2006; Zhang et al., 2008; Chiang 2010).

初步 $X_1 \leftrightarrow$ initial $X_1$

$X_1$ 的 $X_2 \leftrightarrow$ The $X_2$ of $X_1$
Two Dominating Categories of TEs

- **Finite-state-based formalism**
  - E.g., phrase-based SMT (Och and Ney, 2004; Koehn et al., 2003)

- **Synchronous context-free grammar-based formalism**
  - hierarchical phrase-based (Chiang, 2007),
  - tree-to-string (e.g., Quirk et al., 2005; Huang et al., 2006),
  - forest-to-string (Mi et al., 2008)
  - string-to-tree (e.g., Galley et al., 04; Shen et al., 08)
  - tree-to-tree (e.g., Eisner 2003; Cowan et al, 2006; Zhang et al., 2008; Chiang 2010).

- **Exceptions:**
  - STAG (DeNeefe and Knight, 2009)
  - Tree Sequences (Zhang et al., 2008)
Phrase-based SMT: a generative process

is segmented into K phrases

The success of initial experiments had inspired us greatly.
Phrase-based SMT: a generative process

1. \( f_1 J_1 f_1 J J f_1 J \) is segmented into \( K \) phrases

2. The input sentence \( f_1 J \) is segmented into \( K \) phrases

\[
\begin{align*}
(1) & \quad \text{初步 试验 的 成功 极大 地 激励 了 我们.} \\
(2) & \quad \text{成功 初步 试验 激励 我们 极大.} \\
(3) & \quad \text{The success of Initial experiments had inspired us greatly.}
\end{align*}
\]
Phrase-based SMT: a generative process

1. \( f_1 J_1 f_1 J_2 f_1 J_3 \) is segmented into \( K \) phrases

2. Permute the source phrases in the appropriate order

The success of initial experiments had inspired us greatly.
Phrase-based SMT: a generative process

1. \( f \ 1 \ J \ 1 \ f \ 1 \ J \ J \ f \ 1 \ J \) is segmented into \( K \) phrases
2. Permute the source phrases in the appropriate order
3. Translate each source phrase into a target phrase

![Diagram of phrase-based SMT process]
Phrase-based SMT: a generative process

1. \( f_1 J_1 f_1 J_2 f_1 J_n \) is segmented into \( K \) phrases
2. Permute the source phrases in the appropriate order
3. Translate each source phrase into a target phrase
4. Concatenate target phrases to form the target sentence
Phrase-based SMT: a generative process

1. The source sentence $f_1 J J f_1 J J f_1 J$ is segmented into $K$ phrases.
2. Permute the source phrases in the appropriate order.
3. Translate each source phrase into a target phrase.
4. Concatenate target phrases to form the target sentence.
5. Score the target sentence by the target language model.

The success of initial experiments had inspired us greatly.
Phrase-based SMT: a generative process

1. $f \mathbf{J} f \mathbf{I} f \mathbf{J} f \mathbf{I} f \mathbf{1} f \mathbf{J}$ is segmented into $K$ phrases
2. Permute the source phrases in the appropriate order
3. Translate each source phrase into a target phrase
4. Concatenate target phrases to form the target sentence
5. Score the target sentence by the target language model

- **Expressiveness**: Any sequence of translation possible if permit arbitrary reordering
- **Complexity**: arbitrary reordering leads to a NP-hard problem (Knight, 1999).
Reordering in Phrasal SMT

- To constrain search space, reordering is usually limited by preset maximum reordering window and skip size (Zens et al., 2004)
- Reordering models usually penalize non-monotonic movement
  - Proportional to the distance of jumping
  - Further refined by lexicalization, i.e., the degrees of penalization vary for different surrounding words (Koehn et al. 2007)
- Reordering is the unit movement at phrasal level
  - i.e., no “gaps” allowed in reordering movement
Backgrounds: Semiring

- Semiring is a 5-tuple: $(\Psi, \oplus, \otimes, 0, 1)$ (Mohri, 2002)
  - $\Psi$ is a Set, and $0, 1 \in \Psi$
  - Two closed and associative operators:
    - $\oplus$: sum to compute the weight of a sequence of edges
      - $0 \oplus a = a$ (unit); $1 \oplus a = 1$ (absorbing), $\forall a \in \Psi$
    - $\otimes$: product to compute the weight of an (optimal) path
      - $0 \otimes a = 0$ (absorbing); $1 \otimes a = a$ (unit)

- Examples used in speech & language
  - Viterbi $([0,1], \max, \times, 0, 1)$
    defined over probabilities,
  - Tropical $(\mathbb{R}^+ \cup \{+\infty\}, \min, +, +\infty, 0)$
    - operates on non-negative weights
    - e.g., negative log probabilities, aka, costs
Backgrounds: Automata/WFST

Weighted Acceptors: 3-gram LM

Weighted Transducers
Backgrounds: Formal Languages

- A finite-state automaton (FSA) is equivalent to a regular grammar, and a regular language
- Weighted finite-state machines closed under the composition operation
- A regular grammar constitutes strict subset of a context-free grammar (CFG) (Hopcroft and Ullman, 1979)
- The composition of CFG with a FSM, is guaranteed to be a CFG
FST-based translation equivalence

- Source-target equivalence is modeled by weighted non-nested mapping between word strings.
- Unit of mapping, \((\vec{e}_i, \vec{f}_i)\), is a modeling choice, and *phrase* is better than *word* due to:
  - reduced word sense ambiguities with surrounding contexts
  - appropriate local reordering encoded in the source-target phrase pair
Phrase-based SMT is Finite-State

- Each step relates input and output as WFST operations
  - $F$ input sentence
  - $P$ segmentation
  - $R$ permutation
  - $T$ translation (phrasal equiv.)
  - $W$ concatenation
  - $G$ target language model
- Doing all steps amounts to composing above WFSTs
- Guaranteed to produce a FSM, since WFSTs are closed under such composition.
Phrase-based SMT is Finite-State

- Each step relates input and output as WFST operations
  - $F$ input sentence
  - $P$ segmentation
  - $R$ permutation
  - $T$ translation (phrasal equiv.)
  - $W$ concatenation
  - $G$ target language model
- Doing all steps amounts to composing above WFSTs
- Guaranteed to produce a FSM, since WFSTs are closed under such composition.

Similar to ASR, phrase-based SMT amounts to the following operations, computable with in a general-purpose FST toolkit (Kumar et al. 2005)

$$\hat{E} = \text{best\_path} (F \circ P \circ R \circ T \circ W \circ G)$$
Why do we care?

• Not only of theoretic convenience to conceptually connect ASR and SMT better
• Practically useful to leverage the mature WFST optimization algorithms.
• Suggests integrated framework to address ST:
  – i.e., composing the WFSTs from ASR/SMT
Make WFST Approach Practical

• WFST-based system in (Kumar et al., 2005) runs significantly slower than the multiple-stack based decoder (Koehn, 2004)
  – large memory requirements
  – heavy online computation for each composition.

• Reordering is a big challenge
  – The number of states needed is $O(2^J)$; cannot be bounded for any arbitrary inputs as finite-state

• Next, we show a framework to address both issues to make it more practically suitable for ST
**Folsom**: phrase-based SMT by FSM

To speed up: first cut the number of online compositions

\[ = \text{best_path~}(F \circ P \circ R \circ T \circ W \circ G) \quad (1) \]

\[ \hat{E} = \text{best_path~}(F' \circ M \circ G) \quad (2) \]

- \(M\): WFST encoding a log-linear phrasal translation model, obtained by

\[ M = \text{Min}(\text{Min}(\text{Det}(P) \circ T) \circ W \]

- Possible by making \(P\) determinizable (Zhou et al., 2006)

- **More flexible reordering**: \(F'\) is a WFSA constructed on-the-fly
  
  - To encode the uncertainty of the input sentence (e.g., reordering); examples on the next Slide

- **Design a dedicated decoder** for further efficiency: Viterbi search on the *lazy 3-way* composed graph as in (2)
Folsom: phrase-based SMT by FSM

\[ \text{best_path} \left( F \circ P \circ R \circ T \circ W \circ G \right) \] (1)

To speed up: first cut the number of online compositions

\[ E = \text{best_path} \left( F \circ P \circ R \circ T \circ W \circ G \right) \] (1)

\[ \hat{E} = \text{best_path} \left( F' \circ M \circ G \right) \] (2)

- \( M \): WFST encoding a log-linear phrasal translation model, obtained by

\[ M = \text{Min} \left( \text{Min} \left( \text{Det} \left( P \right) \circ T \right) \circ W \right) \]

- Possible by making \( P \) determinizable (Zhou et al., 2006)

- **More flexible reordering**: \( F' \) is a WFSA constructed on-the-fly
  - To encode the uncertainty of the input sentence (e.g., reordering); examples on the next Slide

- **Design a dedicated decoder** for further efficiency: Viterbi search on the *lazy 3-way* composed graph as in (2)
**Folsom**: phrase-based SMT by FSM

\[
\text{best_path}(FF' \circ MM \circ GG) \quad (2)
\]

\[
\text{best_path}(FF \circ PP \circ RR \circ TT \circ WW \circ GG) \quad (1)
\]

To **speed up**: first cut the number of online compositions

\[
E = \text{best_path}(F' \circ P \circ R \circ T \circ W \circ G) \quad (1)
\]

\[
E = \text{best_path}(F' \circ M \circ G) \quad (2)
\]

\[
\hat{E} = \text{best_path}(F' \circ M \circ G) \quad (2)
\]

- **M**: WFST encoding a log-linear phrasal translation model, obtained by

\[
M = \text{Min} (\text{Min}(\text{Det}(P) \circ T) \circ W
\]

- Possible by making \(P\) determinizable (Zhou et al., 2006)

- **More flexible reordering**: \(F'\) is a WFSA constructed on-the-fly

  - To encode the uncertainty of the input sentence (e.g., reordering); examples on the next Slide

- **Design a dedicated decoder** for further efficiency: Viterbi search on the **lazy 3-way** composed graph as in (2)
Folsom: phrase-based SMT by FSM

\[ MMiinn \circ DDeett(PP) \circ TT \circ WW \]

WFST encoding a log-linear phrasal translation model, obtained by
best_path \( FF' \circ MM \circ GG \) \( \) (2)
best_path \( FF \circ PP \circ RR \circ TT \circ M \circ GG \) \( \) (1)

To speed up: first cut the number of online compositions
\[ E = \text{best_path} (F \circ P \circ R \circ T \circ W \circ G) \] (1)

– Possible by making \( P \) determinizable (Zhou et al., 2006)

\[ \hat{E} = \text{best_path} (F' \circ M \circ G) \] (2)

– \( M \): WFST encoding a log-linear phrasal translation model, obtained by
\[ M = \text{Min}(\text{Min}(\text{Det}(P) \circ T) \circ W \]

– Possible by making \( P \) determinizable (Zhou et al., 2006)

• More flexible reordering: \( F' \) is a WFSA constructed on-the-fly
  – To encode the uncertainty of the input sentence (e.g., reordering); examples on the next Slide

• Design a dedicated decoder for further efficiency: Viterbi search on the lazy 3-way composed graph as in (2)
**Folsom**: phrase-based SMT by FSM

is a WFSA constructed on-the-fly

\[ M \text{Miinn}(M \text{Miinn}(D \text{Deett}(P \circ T) \circ W) \circ W) \]

WFST encoding a log-linear phrasal translation model, obtained by

\[
\text{best}_{\text{path}}(FF' \circ MM \circ GG) \quad (2)
\]

\[
\text{best}_{\text{path}}(FF \circ PP \circ RR \circ TT \circ WW \circ GG) \quad (1)
\]

To speed up: first cut the number of online compositions

\[ E = \text{best}_{\text{path}}(F \circ P \circ R \circ T \circ W \circ G) \quad (1) \]

- Possible by making \( P \) determinizable (Zhou et al., 2006)
- To encode the uncertainty of the input sentence (e.g., reordering); examples on the next Slide
- \( M \): WFST encoding a log-linear phrasal translation model, obtained by

\[ M = \text{Min}(\text{Min}(\text{Det}(P) \circ T) \circ W) \]

- Possible by making \( P \) determinizable (Zhou et al., 2006)

- **More flexible reordering**: \( F' \) is a WFSA constructed on-the-fly

  - To encode the uncertainty of the input sentence (e.g., reordering); examples on the next Slide

- **Design a dedicated decoder** for further efficiency: Viterbi search on the *lazy 3-way* composed graph as in (2)
Folsom: phrase-based SMT by FSM

is a WFSA constructed on-the-fly

$$MMiinn(MMiinn(DDeett(PP)\circ TT)\circ WW)$$

WFST encoding a log-linear phrasal translation model, obtained by

$$\text{best}_\text{path} \ (FF'\circ MM\circ GG) \quad (2)$$

$$\text{best}_\text{path} \ (FF\circ PP\circ RR\circ TT\circ WW\circ GG) \quad (1)$$

To speed up: first cut the number of online compositions

$$E = = \text{best}_\text{path} \ (F \circ P \circ R \circ T \circ W \circ G) \quad (1)$$

– Possible by making $P$ determinizable (Zhou et al., 2006)
– To encode the uncertainty of the input sentence (e.g., reordering); examples on the next Slide

- Design a dedicated decoder for further efficiency: Viterbi search on the lazy 3-way composed graph as in (2)

$$M = Min(\text{Min}(\text{Det}(P) \circ T) \circ W)$$

– Possible by making $P$ determinizable (Zhou et al., 2006)

- More flexible reordering: $F'$ is a WFSA constructed on-the-fly

– To encode the uncertainty of the input sentence (e.g., reordering); examples on the next Slide

- Design a dedicated decoder for further efficiency: Viterbi search on the lazy 3-way composed graph as in (2)
Reordering, Finite-State & Gaps

- Each state denotes a specific input coverage with a bit-vector.
- Arcs labeled with an input position to be covered in state transition: weights given by reordering models.

(a): Monotonic (like ASR)

\[ f_2f_3f_4 \rightarrow \overline{f_1f_4} \overline{f_2f_3} \]
Reordering, Finite-State & Gaps

(a): Monotonic (like ASR)

(b): Reordering with skip less than three

- Each state denotes a specific input coverage with a bit-vector.
- Arcs labeled with an input position to be covered in state transition: weights given by reordering models.
Reordering, Finite-State & Gaps

- Each state denotes a specific input coverage with a bit-vector.
- Arcs labeled with an input position to be covered in state transition: weights given by reordering models.

(a): Monotonic (like ASR)

(b): Reordering with skip less than three

Every path from the initial to the final state represents an acceptable input permutation
Reordering, Finite-State & Gaps

• Each state denotes a specific input coverage with a bit-vector.
• Arcs labeled with an input position to be covered in state transition: weights given by reordering models.
• Source segmentation allows options with “gaps” in searching for the best-path to transduce the inputs at the phrase level.

(a): Monotonic (like ASR)

(b): Reordering with skip less than three

Every path from the initial to the final state represents an acceptable input permutation.
Reordering, Finite-State & Gaps

(a): Monotonic (like ASR)

(b): Reordering with skip less than three

Every path from the initial to the final state represents an acceptable input permutation.

- Each state denotes a specific input coverage with a bit-vector.
- Arcs labeled with an input position to be covered in state transition: weights given by reordering models.
- Source segmentation allows options with “gaps” in searching for the best-path to transduce the inputs at the phrase level.
- Such a translation phenomenon is beneficial in many language pairs.

Example:

\[ f_1 f_2 f_3 f_4 \quad \text{phrase segmentation} \quad f_1 f_2 f_3 f_4 \]

Example:

\[ f_1 f_2 f_3 f_4 \quad \text{phrase segmentation} \quad f_1 f_2 f_3 f_4 \]

Example:

\[ f_1 f_2 f_3 f_4 \quad \text{phrase segmentation} \quad f_1 f_2 f_3 f_4 \]

Example:

\[ f_1 f_2 f_3 f_4 \quad \text{phrase segmentation} \quad f_1 f_2 f_3 f_4 \]

Example:

\[ f_1 f_2 f_3 f_4 \quad \text{phrase segmentation} \quad f_1 f_2 f_3 f_4 \]
What FST-based models lack

- FST-based models (e.g., phrase-based) remain state-of-the-art for many language pairs/tasks (Zollmann et al., 2008)
- It makes no use that natural language is inherently structural & hierarchical
  - Led to poor long-distance reordering modeling & exponential complexity of permutation
- PCFGs effectively used to model linguistic structures in many monolingual tasks (e.g., parsing)
  - Recall: RL is a subset of context-free language
- The synchronous (bilingual) version of the PCFG (i.e., SCFG) is a alternative to model translation structures
SCFG-based TE

• A SCFG (Lewis and Stearns, 1968) rewrites the non-terminal (NT) on its left-hand side (LHS) into <source, target> on its right-hand side (RHS)
  – s.t. the constraint of one-to-one correspondences (co-indexed by subscripts) between the source and target of every NT occurrence.
    \[ VP \rightarrow {}^p < PP_1 VP_2 , VP_2 PP_1 > \]

• NTs on RHS can be recursively instantiated simultaneously for both the source and the target, by applying any rules with a matched NT on the LHS.

• SCFG captures the hierarchical structure of NL & provides a more principled way to model reordering
  – e.g., the PP and VP will be reordered regardless of their span lengths
Expressiveness vs. Complexity

- Both depend on maximum number of NTs on RHS (aka, rank of the SCFG grammar)
- Complexity is polynomial: higher order for higher rank
  - Cubic $O(|J|^3)$ for rank-two (binary) SCFG
- Expressiveness: more expressive with higher rank; Specifically for a binary SCFG
  - Rare reordering examples exist (Wu 1997; Wellington et al., 2006) that it cannot cover
  - Arguably sufficient in practice
What’s in common? SCFG-MT vs.
What’s in common? SCFG-MT vs.

• Like ice-cream, SCFG models come with different flavors...
What’s in common? SCFG-MT vs.

- Like ice-cream, SCFG models come with different flavors...
- Linguistic syntax based:
  - Utilize structures defined over linguistic theory and annotations (e.g., Penn Treebank)
  - SCFG rules are derived from the parallel corpus guided by explicitly parsing on at least one side of the parallel corpus.
  - E.g., tree-to-string (e.g., Quirk et al., 2005; Huang et al., 2006), forest-to-string (Mi et al., 2008) string-to-tree (e.g., Galley et al., 04; Shen et al., 08) tree-to-tree (e.g., Eisner 2003; Cowan et al, 2006; Zhang et al., 2008; Chiang 2010).
What’s in common? SCFG-MT vs.

• Like ice-cream, SCFG models come with different flavors...

• Linguistic syntax based:
  – Utilize structures defined over linguistic theory and annotations (e.g., Penn Treebank)
  – SCFG rules are derived from the parallel corpus guided by explicitly parsing on at least one side of the parallel corpus.
  – E.g., tree-to-string (e.g., Quirk et al., 2005; Huang et al., 2006), forest-to-string (Mi et al., 2008) string-to-tree (e.g., Galley et al., 04; Shen et al., 08) tree-to-tree (e.g., Eisner 2003; Cowan et al, 2006; Zhang et al., 2008; Chiang 2010).

• Formal syntax based:
  – Utilize the hierarchical structure of natural language only
  – Grammars extracted from the parallel corpus without using any linguistic knowledge or annotations.
  – E.g., ITG (Wu, 1997) and hierarchical models (Chiang, 2007)
Formal syntax based SCFG

• Arguably a better for ST: not relying on parsing, which might be difficult for informal spoken languages

• Grammars: only one universal NT X is used (Chiang, 2007)
  – Phrasal rules:
    \[ X \rightarrow < \text{初步 试验}, \text{initial experiments}> \]
  – Abstract rules: with NTs on RHS
    \[ X \rightarrow < X_1 \text{ 的 } X_2, \text{The } X_2 \text{ of } X_1 > \]
  – Glue rules to generate sentence symbol S
    \[ S \rightarrow < S_1 X_2, S_1 X_2 > \]
    \[ S \rightarrow < X_1, X_1 > \]
Learning of (Abstract) SCFG Rules

• Hierarchical SCFG rule extraction (Chiang 2007):
  – Replace any aligned sub-phrase pair of a PP with co-indexed NT symbols
  – Many-to-many mapping between phrase pairs and derived abstract rules.
  – Conditional probabilities estimated based on heuristic counts.

• Linguistic SCFG rule extraction
  – Syntactic parsing on at least one side of the parallel corpus.
  – Rules are extracted along with the parsing structures: constituency (Galley et al., 2004; Galley et al., 2006), and dependency (Shen et al., 2008)

• Improved extraction and parameterization:
  – Expected counts: forced alignment and inside-outside (Huang and Zhou, 2009);
  – Leaving-one-out smoothing (Wuebker et al., 2010)
  – Extract additional rules:
    • Reduce conflicts between alignment or parsing structures (DeNeefe et al., 2007)
    • From existing ones of high expected counts: rule arithmetic (Cmejrek and Zhou, 2010)
Practical Considerations of Formal Syntax-based SCFG

• On Speed:
  – Finite-state based search with limited reordering usually runs faster than most of the SCFG-based models.
  – Except for tree-to-string (e.g., Huang and Mi, 2010), which is faster in practice.

• On Model:
  – With beam pruning, using higher-rank (>2) SCFG is possible
  – Weakness: no constraints on X often led to over-generalization of SCFG rules

• Improvements: add linguistically motivated constraints
  – Refined NT with direct linguistic annotations (Zollmann and Venugopal, 2006)
  – Soft constraints (Marton et al., 2008)
  – Enriched features tied to NTs (Zhou et al., 2008b, Huang et al., 2010)
Further Reading

• Phase-based SMT

• Computational theory:
Further Reading: SCFG-based SMT

Decoding: A Unified Perspective for ASR/SMT/ST

1. Overview: ASR, SMT, ST, Metric
2. Learning Problems in SMT
3. Translation Structures for ST
4. Decoding: A Unified Perspective ASR/SMT
5. Coupling ASR/SMT: Decoding & Modeling
6. Practices
7. Future Directions
Unifying ASR/SMT/ST Decoding

• Upon first glance, there is dramatic difference between ASR and SMT due to reordering

• Even for SMT, there are a variety of paradigms
  – Each may call for its own decoding algorithm

• Common in all decoding:
  – Search space is usually exponentially large and DP is a must
  – DP: divide-and-conquer w/ reusable sub-solutions.
  – Well-known Viterbi search in HMM-based ASR: an instance of DP

• We try unify them by observing from a higher standpoint

• Benefits of a unified perspective
  – Help us understand concepts better
  – Reveals a close connection between ASR and SMT
  – Beneficial for joint ST decoding
Background: Weighted Directed Acyclic Graph $\Sigma = (V, E, \Omega)$

- A vertices set $V$ and edges set $E$
- A weight mapping function $\Omega: E \rightarrow \Psi$ assigns each edge a weight from $\Psi$
  - $\Psi$ defined in a semiring $(\Psi, \oplus, \otimes, 0, 1)$
- A single source vertex $s \in V$ in the graph.
- A path $\pi$ in $G$ is a sequence of consecutive edges $\pi = e_1 e_2 \cdots e_l$
  - End vertex of one edge is the start vertex of the subsequent edge
  - Weight of path $\Omega(\pi) = \otimes_{i=1}^{l} \Omega(e_i)$
- The shortest distance from $s$ to a vertex $q$, $\delta(q)$, is the “$\oplus$-sum” of the weights of all paths from $s$ to $q$
- We define $\delta(s) = \bar{1}$
Background: Weighted Directed Acyclic Graph $\Sigma = (V, E, \Omega)$

- A vertices set $V$ and edges set $E$
- A weight mapping function $\Omega: E \rightarrow \Psi$ assigns each edge a weight from $\Psi$
  - $\Psi$ defined in a semiring $(\Psi, \oplus, \otimes, 0, 1)$
- A single source vertex $s \in V$ in the graph.
- A path $\pi$ in $G$ is a sequence of consecutive edges $\pi = e_1 e_2 \cdots e_l$
  - End vertex of one edge is the start vertex of the subsequent edge
  - Weight of path $\Omega(\pi) = \otimes_{i=1}^{l} \Omega(e_i)$
- The shortest distance from $s$ to a vertex $q$, $\delta(q)$, is the “$\oplus$-sum” of the weights of all paths from $s$ to $q$
- We define $\delta(s) = 1$

The search space of any finite-state automata, e.g., the one used in HMM-based ASR and FST-based translation, can be represented by such a graph.
Genetic Viterbi Search of DAG

Algorithm 1: Generic Viterbi Search of DAG

1. **Procedure** Viterbi ($\Sigma, s$)
2. Initialize $\Sigma, s$
3. Topological sort vertices of $\Sigma$
4. For each vertex $p$ in $\Sigma$ in topological order do
5. For each edge $e$ such that $\text{start}(e) = p$ do
6. \[ q = \text{end}(e) \]
7. \[ \delta(q) \oplus = \delta(p) \otimes w(e) \]

- Many search problems can be converted to the classical shortest distance problem in the DAG (Cormen et al, 2001)
- Complexity is $O(|V| + |E|)$, as each edge needs to be visited exactly once.
- Terminates when all reachable vertices from $s$ have been visited, or if some predefined destination vertices encountered
Case Studies: 
ASR and Multi-stack Phrasal SMT

HMM-based ASR:
- Composing the input acceptor with a sequence of transducer (Mohri et al., 2002)
- This defines a finite-state search space represented by a graph
- Viterbi search to find shortest-distance path in this graph

\[
\hat{f}_1^J = \text{best\_path} \left( O \circ H \circ C \circ L \circ G \right)
\]

- The best translation collected along the best path with the lowest weight \( \delta(t) \)
Case Studies: ASR and Multi-stack Phrasal SMT

- Multi-stack Phrasal SMT e.g., Moses (Koehn et al., 2007)
  - The best translation collected along the best path with the lowest weight $\delta(t)$
Case Studies:
ASR and Multi-stack Phrasal SMT

- Multi-stack Phrasal SMT e.g., Moses (Koehn et al., 2007)
  - The source vertex: none of the source words has been translated and the translation hypothesis is empty.
  - The target vertex: all the source words have been covered
  - Each edge connect start and end vertices by choosing a consecutive uncovered set of source words, to apply one of the source-matched phrase pairs (PP)
  - The target side of the PP is appended to the hypothesis
  - The weight of each edge is determined by a log-linear model.
  - The best translation collected along the best path with the lowest weight $\delta(t)$
  - The best translation collected along the best path with the lowest weight $\delta(t)$
Hypothesis Recombination in Graph

key idea
The original attempts are

The initial experiments are
Hypothesis Recombination in Graph

key idea

- Only the partial hypothesis with the lowest cost has a possibility to become the best hypothesis in the future
  - So HR is loses-free for 1-best search
Hypothesis Recombination in Graph

- Only the partial hypothesis with the lowest cost has a possibility to become the best hypothesis in the future
  - So HR is loses-free for 1-best search
- It is clear why this works if we view this from graph
  - All partial hypotheses of same signature arrive at the same vertex
  - All future paths leaving this vertex are indistinguishable afterwards.
  - Under “⊕-sum” operation (e.g., the min in the Tropical semiring), only the lowest-cost partial hypothesis is kept
Case Studies II: Folsom Phrasal SMT

\[ \hat{E} = \text{best\_path } (F' \circ M \circ G) \]

Graph expanded by lazy 3-way composition

- \( F_1, M_1, G_1 \)
- Target vertices = each component state is a final state in each individual WFST
- Subsequent vertices visited in topological order
- Weight(e) from \( (F_p, M_p, G_p) \) to \( (F_q, M_q, G_q) \):
  \[ \Omega(e) = \otimes_{m \in \{I,M,G\}} \lambda_m \Omega(e_m) \]
- HR: merge vertices of same \( (F_q, M_q, G_q) \) & keep only the one with lowest cost
- Any path connecting the source to a target vertex is a translation: best one with the shortest distance.

The decoding is optimized lazy 3-way composition + minimization, followed by best-path search.
Case Studies II: Folsom Phrasal SMT

\[ \hat{E} = \text{best\_path}(F' \circ M \circ G) \]

Graph expanded by lazy 3-way composition

- Source vertex = start states comp. \((F_1, M_1, G_1)\)
- Target vertices = each component state is a final state in each individual WFST
- Subsequent vertices visited in topological order
- Weight(e) from \((F_p, M_p, G_p)\) to \((F_q, M_q, G_q)\):
  \[ \Omega(e) = \bigotimes_{m \in \{I, M, G\}} \lambda_m \Omega(e_m) \]
- HR: merge vertices of same \((F_q, M_q, G_q)\) & keep only the one with lowest cost
- Any path connecting the source to a target vertex is a translation: best one with the shortest distance.
- The decoding is optimized lazy 3-way composition + minimization, followed by best-path search.

\[ F 1 F F 1 1 F 1, M 1 M M M 1 1 M 1, G 1 G G 1 1 G 1 \]
Case Studies II: Folsom Phrasal SMT

\[ \hat{E} = \text{best}_\text{path} \left( F' \circ M \circ G \right) \]

\[ e e e e = \otimes m \in \{ I, M, G \} \otimes \otimes m \in \{ I, M, G \} \]  
\[ m \in \{ I I, M M, G G \} \otimes m \in \{ I, M, G \} \lambda m \lambda \lambda \lambda m mm \lambda m \Omega( e m e e e mm m e m ) \]

from \(( F p F F F p p p F p, M p M M M p p M p, G p G G G p p p G p )\) to \(( F q F F F q q q F q, M q M M M q q q M q, G q G G G q q q G q )\):

\( F 1 F F F 1, M 1 M M M 1 1 M 1, G 1 G G G 1 1 G 1 \)

Graph expanded by lazy 3-way composition

- Target vertices = each component state is a final state in each individual WFST

Subsequent vertices visited in topological order

\[ \Omega(e) = \otimes m \in \{ I, M, G \} \lambda m \Omega(e_m) \]

- Weight(e) from \(( F_p, M_p, G_p )\) to \(( F_q, M_q, G_q )\):

\[ \Omega(e) = \otimes m \in \{ I, M, G \} \lambda m \Omega(e_m) \]

- HR: merge vertices of same \(( F_q, M_q, G_q )\) & keep only the one with lowest cost
Case Studies II: Folsom Phrasal SMT

\[ \hat{E} = \text{best_path} \left( F' \circ M \circ G \right) \]

same \( F q, M q, G q \) \( F q FF F q qq F q, M q MM M q qq M q, G q GG G q qq G q \) \( F q, M q, G q \) & keep only the one with lowest cost

\[ \epsilon \epsilon \epsilon = \otimes \ m \in \{I,M,G\} \otimes \ m \in \{I,M,G\} \ m \in \{I,MM,GG\} \otimes \ m \in \{I,M,G\} \lambda m \lambda \lambda m \ m mm \lambda m \ \Omega( e m e e m m m e m ) \]

from ( \( F p FF F p pp F p, M p MM M p pp M p, G p GG G p pp G p \) ) to ( \( F q FF F q qq F q, M q MM M q qq M q, G q GG G q qq G q \) )

\[ F 1 FF F 1 1 F 1, M 1 MM M 1 1 M 1, G 1 G G G 1 1 G 1 \]

Graph expanded by lazy 3-way composition

- Target vertices = each component state is a final state in each individual WFST
- Subsequent vertices visited in topological order
- HR: merge vertices of same (\( F_-, M_-, G_- \)) & keep only the one with lowest cost
Case Studies II: Folsom Phrasal SMT

\[ \hat{E} = \text{best}_\text{path} \left( F' \circ M \circ G \right) \]

same \( F q, M q, G q \) \( F q FF F q qq F q, M q MM M q qq M q, G q GG G q qq G q \)

& keep only the one with lowest cost

\[ e \ e e e = \bigotimes \ m \in \{ I, M, G \} \bigotimes \ m \in \{ I, M, G \} \bigotimes \ m \in \{ I, M, G \} \lambda m \lambda \lambda \lambda m \ m \ m \ m \ \Omega \left( e m \ e e \ e m \ m m \ e m \right) \]

from \( \left( F p FF F p pp F p, M p MM M p pp M p, G p GG G p pp G p \right) \) to \( \left( F q FF F q qq F q, M q MM M q qq M q, G q GG G q qq G q \right) : \)

\[ F q 1 FF F 1 1 F 1, M 1 MM M 1 1 M 1, G 1 G G 1 1 G 1 \]

Graph expanded by lazy 3-way composition

- Target vertices = each component state is a final state in each individual WFST
- Subsequent vertices visited in topological order
- Any path connecting the source to a target vertex is a translation: best one with the
Case Studies II: Folsom Phrasal SMT

\[ \hat{E} = \text{best}_\text{path} \left( F' \circ M \circ G \right) \]

same \( F q, M q, G q \quad F q \quad F q \quad F q \quad F q \quad M q \quad M q \quad q q \quad M q \quad G q \quad G q \quad G q \quad q q \quad G q \quad F q \quad M q \quad G q \) & keep only the one with lowest cost

\[ e e e e = \otimes m \in \{I,M,G\} \otimes \otimes m \in \{I,M,G\} mm \in \{II,MM,GG\} \otimes m \in \{I,M,G\} \lambda m \lambda \lambda m mm \lambda m \Omega \left(e \ m \ e \ e \ m \ m \ m \ e \ m \right) \]

from \( (F \ p \ FF \ F \ p \ pp \ F \ p, M \ p \ MM \ M \ p \ pp \ M \ p, G \ p \ GG \ G \ p \ pp \ G \ p) \) to \( (F q \ FF \ F q \ q q \ F q, M q \ MM \ M q \ q q \ M q, G q \ GG \ G q \ q q \ G q) : \)

\[ F 1 \ FF \ F 1 \ 1 \ F 1, M 1 \ MM \ M 1 \ 1 \ M 1, G 1 \ G G \ 1 \ G 1 \]

*Graph* expanded by lazy 3-way composition

- Target vertices = each component state is a final state in each individual WFST
- Subsequent vertices visited in topological order
- Any path connecting the source to a target vertex is a translation: best one with the
Background: Weighted Directed Hypergraph $H =\langle V, E, \Psi \rangle$

- A vertices set $V$ and hyperedge set $E$
- Each hyperedge links an ordered list of tail vertices to a head vertex
- Arity of $H$ is the maximum number of tail vertices of all $e$
- $f_e: \Psi^{|T(e)|} \rightarrow \Psi$ assigns each hyperedge $e$ a weight from $\Psi$
- A derivation $d$ of a vertex $q$: a sequence of consecutive $e$ connecting source vertices to $q$
  - Weight $\Omega(d)$ recursively computed from weight functions of each $e$.
- The “best” weight of $q$ is the “$\oplus$-sum” of all of its derivations

\[
\delta(q) = \begin{cases} 
1, & \text{if } q \text{ is a source vertex} \\
\oplus_d \Omega(d), & \text{otherwise}
\end{cases}
\]
Background: Weighted Directed Hypergraph $H = \langle V, E, \Psi \rangle$

- A vertices set $V$ and hyperedge set $E$
- Each hyperedge $e$ links an ordered list of tail vertices to a head vertex
- *Arity* of $H$ is the maximum number of tail vertices of all $e$
- $f_e: \Psi^{\mid T(e)\mid} \to \Psi$ assigns each hyperedge $e$ a weight from $\Psi$
- A derivation $d$ of a vertex $q$: a sequence of consecutive $e$ connecting source vertices to $q$
  - Weight $\Omega(d)$ recursively computed from weight functions of each $e$.
- The “best” weight of $q$ is the “$\oplus$-sum” of all of its derivations

$$\delta(q) = \begin{cases} 
1, & \text{q is a source vertex} \\
\bigoplus_d \Omega(d), & \text{otherwise}
\end{cases}$$

A hypergraph, a generalization of a graph, encodes the hierarchical-branching search space expanded over CFG models.
Genetic Viterbi Search of DAH

Algorithm 2: Generic Viterbi Search of DAH

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Procedure Viterbi ( (H, s^n_1) )</td>
</tr>
<tr>
<td>2.</td>
<td>Initialize ( H, s^n_1 )</td>
</tr>
<tr>
<td>3.</td>
<td>Topological sort the vertices of ( H )</td>
</tr>
<tr>
<td>4.</td>
<td>for each vertex ( q ) in ( H ) in topological order do</td>
</tr>
<tr>
<td>5.</td>
<td>for each hyperedge ( e ) such that head( (e) = q ) do</td>
</tr>
<tr>
<td>6.</td>
<td>( {p_1, \ldots, p_{</td>
</tr>
<tr>
<td>7.</td>
<td>( \delta(q) \oplus = f_e(\delta(p_1), \ldots, \delta(p_{</td>
</tr>
</tbody>
</table>

- Decoding of SCFG-based models largely follows CKY, an instance of Viterbi on DAH of arity two
- Complexity is proportional to \( O(|E|) \)
Case Study: Hypergraph decoding
Case Study: Hypergraph decoding

- \( ii, jj \) \( X, i, j \) the LHS NT + span
- Vertices: \( X, i, j \) the LHS NT + span
Case Study: Hypergraph decoding

- \( ii, jj \) \( X, i, j \) the LHS NT + span
- Source vertices: apply phrasal rules matching any consecutive inputs
Case Study: Hypergraph decoding

- *ii are visited earlier than those with longer spans*
- *ii, jj X, i, j* the LHS NT + span
- Source vertices: apply phrasal rules matching any consecutive inputs
- Topological sort ensures that vertices with shorter spans $j - i$ are visited earlier than those with longer spans
Case Study: Hypergraph decoding

- **at head vertex updated per line 7**
- **ii** are visited earlier than those with longer spans
- **ii, jj** X, i, j the LHS NT + span
- Source vertices: apply phrasal rules matching any consecutive inputs
- Derivation weight $\delta q$ at head vertex updated per line 7
Case Study: Hypergraph decoding

- \([SS,0,/]\) is reached.
- at head vertex updated per line 7
- ii are visited earlier than those with longer spans
- \(ii, jj\) \(X,i,j\) the LHS NT + span
- Source vertices: apply phrasal rules matching any consecutive inputs
- Process repeated until the target vertex \([S, 0, J]\) is reached.
Case Study: Hypergraph decoding

- \([SS,0,J]\) is reached.
- at head vertex updated per line 7
- ii are visited earlier than those with longer spans
- ii,jjj X,i,j the LHS NT + span
- Source vertices: apply phrasal rules matching any consecutive inputs
- The best derivation found by back-tracing from the target vertex
Case Study: Hypergraph decoding

- \([SS,0, JJ]\) is reached.
- at head vertex updated per line 7
- \(ii\) are visited earlier than those with longer spans
- \(i, j\) \(X, i, j\) the LHS NT + span
- Source vertices: apply phrasal rules matching any consecutive inputs
- The best derivation found by back-tracing from the target vertex
- Complexity \(O(E) \propto E^E \propto O(|\mathcal{R}|^3)\)
Case Study: DAH with LM

-1 words on both the left-most and right-most side

\[(\mid R \mid J^3 \mid T \mid 4^{n-1})\]
Case Study: DAH with LM

• Search space is still DAH

-1 words on both the left-most and right-most side

\[ (|\mathcal{R}|J^3 |T|^{4(n-1)}) \]
Case Study: DAH with LM

- 1 words on both the left-most and right-most side
- Search space is still DAH
- Tail vertex additionally encodes the n-1 words on both the left-most and right-most side

\[
\left( |\mathcal{R}| f^3 |T|^{4(n-1)} \right)
\]
Case Study: DAH with LM

-1 words on both the left-most and right-most side
Search space is still DAH
Each hyperedge updates LM score and boundary information.

\[
(|\mathcal{R}|^{3} |T|^{4(n-1)})
\]
Case Study: DAH with LM

- $R \ J \ 3 \ J J \ J \ 3 \ 3 \ J \ 3 \ T \ T \ T \ T$
- $R \ J \ 3 \ T \ 4(n-1) \ 4(nn-1) \ R \ J \ 3 \ T \ 4(n-1)$
- $R \ J \ 3 \ T \ 4(n-1)$
- -1 words on both the left-most and right-most side
- Search space is still DAH
- Each hyperedge updates LM score and boundary information.
- Worst-case complexity is $O \ R \ R \ J \ 3 \ T \ 4(n-1)$
Case Study: DAH with LM

- The search space is still DAH.
- Each hyperedge updates the LM score and boundary information.
- Pruning is a must.
- 
  \[ (|R| I^3 |T|^4(n-1)) \]
Pruning: A perspective from graph

Two options to prune in a graph

1. Discard some end vertices
   - $\delta(q)$
   - skipping certain vertices that are outside either
     - a beam from the best (beam pruning)
     - the top $k$ list (histogram pruning).

2. Discard some edges leaving a start vertex (beam or histogram pruning)
   - Apply certain reordering constraints (Zens and Ney, 2004)
   - Discard higher-cost phrase-based translation options.
     - Both, unlike HR, may lead to search errors.
Pruning: A perspective from graph

Two options to prune in a graph

1. Discard some end vertices
   - skipping certain vertices that are outside either
     - a beam from the best (*beam pruning*)
     - the top $k$ list (*histogram pruning*).
     - the top $k$ list (*histogram pruning*).

2. Discard some edges leaving a start vertex (*beam or histogram pruning*)
   - Apply certain reordering constraints (Zens and Ney, 2004)
   - Discard higher-cost phrase-based translation options.
   - Both, unlike HR, may lead to search errors.
Pruning: A perspective from graph

Two options to prune in a graph

1. Discard some end vertices
   - skipping certain vertices that are outside either
     • a beam from the best (*beam pruning*)
     • the top $k$ list (*histogram pruning*).
   - Apply certain reordering constraints (Zens and Ney, 2004)
   - Discard higher-cost phrase-based translation options.
   - Discard higher-cost phrase-based translation options.

• Both, unlike HR, may lead to search errors.
A lazy Histogram Pruning

- Avoid expanding edges *if* they fall outside of the top $k$, if
  - weight function is monotonic in each of its arguments,
  - Input list for each argument is sorted.
- Example: cube pruning for DAH search (Chiang,07):
  - Suppose a *hyperedge bundle* $\{e_j\}$ where they share the same source side and identical tail vertices
  - Hyperedges with lower costs are visited earlier
  - Push $e$ into a priority queue that is sorted by $f_e : \Psi^{T(e)} \rightarrow \Psi$.
  - Hyperedge popped from the priority queue is explored.
  - Stops when the top $k$ hyperedges popped, and all remaining ones discarded
A lazy Histogram Pruning

• Avoid expanding edges if they fall outside of the top $k$, if
  – weight function is monotonic in each of its arguments,
  – Input list for each argument is sorted.
• Example: cube pruning for DAH search (Chiang,07):
  – Suppose a hyperedge bundle $\{e_j\}$ where they share the same
    source side and identical tail vertices
  – Hyperedges with lower costs are visited earlier
  – Push $e$ into a priority queue that is sorted by $f_e : \Psi^{\lvert T(e) \rvert} \rightarrow \Psi$.
  – Hyperedge popped from the priority queue is explored.
  – Stops when the top $k$ hyperedges popped, and all remaining
    ones discarded
A lazy Histogram Pruning

• Avoid expanding edges if they fall outside of the top $k$, if
  – weight function is monotonic in each of its arguments,
  – Input list for each argument is sorted.
• Example: cube pruning for DAH search (Chiang, 07):
  – Suppose a *hyperedge bundle* \{$e_j$\} where they share the same source side and identical tail vertices
  – Hyperedges with lower costs are visited earlier
  – Push $e$ into a priority queue that is sorted by $f_e : \Psi |T(e)| \rightarrow \Psi$.
  – Hyperedge popped from the priority queue is explored.
  – Stops when the top $k$ hyperedges popped, and all remaining ones discarded.
Further Reading

Coupling ASR and SMT for ST – Joint Decoding

1. Overview: ASR, SMT, ST, Metric
2. Learning Problems in SMT
3. Translation Structures for ST
4. Decoding: A Unified Perspective ASR/SMT
5. **Coupling ASR/SMT: Decoding & Modeling**
6. Practices
7. Future Directions
Bayesian Perspective of ST

\[ \hat{e}_1^l = \arg\max_{e_1^l} P(e_1^l | x_1^T) \]

\[ = \arg\max_{e_1^l} \left\{ \sum_{f_1^l} P(e_1^l | f_1^l) P(x_1^T | f_1^l) P(f_1^l) \right\} \]

\[ \approx \arg\max_{e_1^l} \left\{ \max_{f_1^l} P(e_1^l | f_1^l) P(x_1^T | f_1^l) P(f_1^l) \right\} \]

\[ \approx \arg\max_{e_1^l} \left\{ P(e_1^l | \arg\max_{f_1^l} P(x_1^T | f_1^l) P(f_1^l)) \right\} \]
Bayesian Perspective of ST

- Sum replaced by Max
- A common practice

\[
\hat{e}_1^l = \underset{e_1^l}{\text{argmax}} \ P(e_1^l|x_1^T)
\]

\[
= \underset{e_1^l}{\text{argmax}} \ \left\{ \sum_{f_1^I} P(e_1^l|f_1^I)P(x_1^T|f_1^I)P(f_1^I) \right\}
\]

\[
\approx \underset{e_1^l}{\text{argmax}} \ \left\{ \max_{f_1^I} P(e_1^l|f_1^I)P(x_1^T|f_1^I)P(f_1^I) \right\}
\]

\[
\approx \underset{e_1^l}{\text{argmax}} \ \left\{ P[e_1^l|\underset{f_1^I}{\text{argmax}} P(x_1^T|f_1^I)P(f_1^I)] \right\}
\]
Bayesian Perspective of ST

- Sum replaced by Max
- A common practice

\[ \hat{e}_1^l = \arg\max_{e_1^l} P(e_1^l|x_1^T) \]

\[ = \arg\max_{e_1^l} \left\{ \sum_{f_1^l} P(e_1^l|f_1^l)P(x_1^T|f_1^l)P(f_1^l) \right\} \]

\[ \approx \arg\max_{e_1^l} \left\{ \max_{f_1^l} P(e_1^l|f_1^l)P(x_1^T|f_1^l)P(f_1^l) \right\} \]

\[ \approx \arg\max_{e_1^l} \left\{ P[e_1^l|\arg\max_{f_1^l} P(x_1^T|f_1^l)P(f_1^l)] \right\} \]

- \( f_1^l \) determined only by \( x_1^T \), and solved as an isolated problem.
- The foundation of the cascaded approach
Bayesian Perspective of ST

- Sum replaced by Max
- A common practice

\[ \hat{e}_1^l = \arg\max_{e_1^l} P(e_1^l | x_1^T) \]
\[ = \arg\max_{e_1^l} \left\{ \sum_{f_1^l} P(e_1^l | f_1^l) P(x_1^T | f_1^l) P(f_1^l) \right\} \]
\[ \approx \arg\max_{e_1^l} \left\{ \max_{f_1^l} P(e_1^l | f_1^l) P(x_1^T | f_1^l) P(f_1^l) \right\} \]
\[ \approx \arg\max_{e_1^l} \left\{ P[e_1^l | \arg\max_{f_1^l} P(x_1^T | f_1^l) P(f_1^l)] \right\} \]

- \( f_1^l \) determined only by \( x_1^T \), and solved as an isolated problem.
- The foundation of the cascaded approach

- Cascaded approach impaired by compounding of errors propagated from ASR to SMT
- ST improved if ASR/SMT interactions are factored in with better coupled components.
- Coupling achievable with joint decoding and more coherent modeling across components
Tight Joint Decoding

\[ \hat{e}_1^l = \arg\max_{e_1^l} \left\{ \max_{f_1^l} P(e_1^l|f_1^l)P(x_1^T|f_1^l)P(f_1^l) \right\} \]

A fully integrated search over all possible \( e_1^l \) and \( f_1^l \)

- \((e_1^l, f_1^l)\) for the usual source LM used in the ASR WFSTs
- Produce speech translation in the target language.
- Monotonic joint translation model at phrase-level (Casacuberta et al., 2008).
- Tight joint decoding using phrase-based SMT can be achieved by Folsom
  - Composing ASR WFSTs with \( M \) and \( G \) on-the-fly
  - Followed by fully integrated search
  - Limitation is that reordering can only occur within a phrase.
Tight Joint Decoding

\[
\hat{e}_1^l = \arg\max_{e_1^l} \left\{ \max_{f_1^l} P(e_1^l \mid f_1^l) P(x_1^T \mid f_1^l) P(f_1^l) \right\}
\]

A fully integrated search over all possible \(e_1^l\) and \(f_1^l\)

- With the unified view of ASR/SMT, it’s feasible for translation models with simplified reordering

- \((e_1^l, f_1^l)\) for the usual source LM used in the ASR WFSTs
- Produce speech translation in the target language.
  - Monotonic joint translation model at phrase-level (Casacuberta et al., 2008).
  - Tight joint decoding using phrase-based SMT can be achieved by Folsom
    - Composing ASR WFSTs with \(M\) and \(G\) on-the-fly
    - Followed by fully integrated search
    - Limitation is that reordering can only occur within a phrase.
Tight Joint Decoding

\[ \hat{e}_1^l = \arg\max_{e_1^l} \left\{ \max_{f_1^l} P(e_1^l | f_1^l) P(x_1^T | f_1^l) P(f_1^l) \right\} \]

\[ e_1 \parallel f_1 \parallel e_1 \parallel f_1 \parallel f_1 \parallel f_1 \parallel f_1 \parallel f_1 \parallel f_1 \parallel \]

A fully integrated search over all possible \( e_1^l \) and \( f_1^J \)

- With the unified view of ASR/SMT, it’s feasible for translation models with simplified reordering
  - WFST-based word-level ST (Matusov et al. 2006):
    - Produce speech translation in the target language.
    - Produce speech translation in the target language.
  - Monotonic joint translation model at phrase-level (Casacuberta et al., 2008).
  - Tight joint decoding using phrase-based SMT can be achieved by Folsom
    - Composing ASR WFSTs with \( M \) and \( G \) on-the-fly
    - Followed by fully integrated search
    - Limitation is that reordering can only occur within a phrase.
Tight Joint Decoding

\[
\hat{e}_1^l = \arg\max_{e_1^l} \left\{ \max_{f_1^j} P(e_1^l|f_1^j)P(x_1^T|f_1^j)P(f_1^j) \right\}
\]

e 1 I ee e 1 I 1 e 1 I 1 e 1 I, f 1 J f f f 1 J 1 f 1 J J J f 1 J ) for the usual source LM used in the ASR WFSTs

A fully integrated search over all possible \(e_1^l\) and \(f_1^j\)

- With the unified view of ASR/SMT, it’s feasible for translation models with simplified reordering
  - WFST-based word-level ST (Matusov et al. 2006):
    - Produce speech translation in the target language.
  - Monotonic joint translation model at phrase-level (Casacuberta et al., 2008).
  - Monotonic joint translation model at phrase-level (Casacuberta et al., 2008).
  - Tight joint decoding using phrase-based SMT can be achieved by Folsom
    - Composing ASR WFSTs with \(M\) and \(G\) on-the-fly
    - Followed by fully integrated search
    - Limitation is that reordering can only occur within a phrase.
Tight Joint Decoding

\[ \hat{e}_1^l = \arg\max_{e_1^l} \left\{ \max_{f_1^l} P(e_1^l|f_1^l) P(x_1^T|f_1^l) P(f_1^l) \right\} \]

GG on-the-fly

\[ e_1 l e e e 1 l f 1 f f f 1 l f 1 f f f 1 f 1 | f 1 f 1 f 1 f 1 | f 1 f 1 f 1 f 1 | f 1 f 1 \]

A fully integrated search over all possible \( e_1^l \) and \( f_1^l \)

\[ E = C(\mathbf{X} \circ \mathbf{H} \circ \mathbf{C} \circ \mathbf{L} \circ \mathbf{M} \circ \mathbf{C} \circ \mathbf{L} \circ \mathbf{M} \circ \mathbf{G}) \]

— Tight joint decoding using phrase-based SMT can be achieved by Folsom

- Followed by fully integrated search
- Limitation is that reordering can only occur within a phrase.
- Followed by fully integrated search
- Limitation is that reordering can only occur within a phrase.
Loose Joint Decoding

\[
\hat{e}^l_1 = \operatorname{argmax}_{e^l_1} \left\{ \max_{f^l_1} P(e^l_1 | f^l_1) P(x^T_1 | f^l_1) P(f^l_1) \right\}
\]

Approximating full search space with a promising subset

- **N-best ASR hypotheses** (Zhang et al., 2004): Weak
- **Word lattices** produced by ASR recognizers (Matusov et al., 2006; Mathias and Byrne, 2006; Zhou et al., 2007): most general & challenging
- **Confusion networks** (Bertoldi et al., 2008): special case of lattice
  - Better trade-off to address the reordering issue

; here \( i < j \) are the node numbers in the input lattice that are spanned by \( X \).

- If reordering is critical in joint ST, try SCFG-based SMT models
  - Reordering without length constraints
  - Avoids traversing the lattice to compute the distortion costs
Loose Joint Decoding

\[ \hat{e}_1^l = \arg \max_{e_1^l} \left\{ \max_{f_1^l} P(e_1^l | f_1^l) P(x_1^T | f_1^l) P(f_1^l) \right\} \]

\( ii, jj \) \( X, i, j \); here \( ii < jj \) are the node numbers in the input lattice that are spanned by \( XX \).

Approximating full search space with a promising subset

- **N-best** ASR hypotheses (Zhang et al., 2004): **Weak**
- **Word lattices** produced by ASR recognizers (Matusov et al., 2006; Mathias and Byrne, 2006; Zhou et al., 2007): **most general & challenging**
- **Confusion networks** (Bertoldi et al., 2008): special case of lattice
  - Better trade-off to address the reordering issue

- **SCFG models** can take ASR lattice for ST
  - Generalized CKY algorithm for translating lattices (Dyer et al., 2008).
  - Nodes in the lattice numbered such that the end node is always numbered higher than the start node for any edge,
  - The vertex in HG labeled with \( X, i, j \); here \( i < j \) are the node numbers in the input lattice that are spanned by \( X \).

- If reordering is critical in joint ST, try SCFG-based SMT models
  - Reordering without length constraints
Loose Joint Decoding

\[
\hat{e}_1^l = \arg \max_{e_1^l} \left\{ \max_{f_1^l} P(e_1^l | f_1^l) P(x_1^T | f_1^l) P(f_1^l) \right\}
\]

\(ii, jj X, i, j\); here \(ii < jj\) are the node numbers in the input lattice that are spanned by \(XX\).

Approximating full search space with a promising subset

- **N-best** ASR hypotheses (Zhang et al., 2004): Weak
- **Word lattices** produced by ASR recognizers (Matusov et al., 2006; Mathias and Byrne, 2006; Zhou et al., 2007): most general & challenging
- **Confusion networks** (Bertoldi et al., 2008): special case of lattice
  - Better trade-off to address the reordering issue

- **SCFG models** can take ASR lattice for ST
  - Generalized CKY algorithm for translating lattices (Dyer et al., 2008).
  - Nodes in the lattice numbered such that the end node is always numbered higher than the start node for any edge,

- If reordering is critical in joint ST, try SCFG-based SMT models
  - Reordering without length constraints
  - Avoids traversing the lattice to compute the distortion costs
    - Avoids traversing the lattice to compute the distortion costs
Coupling ASR and SMT for ST – Joint Modeling

1. Overview: ASR, SMT, ST, Metric
2. Learning Problems in SMT
3. Translation Structures for ST
4. Decoding: A Unified Perspective ASR/SMT
5. **Coupling ASR/SMT:** Decoding & **Modeling**
6. Practices
7. Future Directions
End-to-end Modeling of Speech Translation

• Two modules in conventional speech translation

- Problems of inconsistency
  - SR and MT are optimized for different criteria, inconsistent to the E2E ST quality (metric discrepancy)
  - SR and MT are trained without considering the interaction between them (train/test condition mismatch)
Why End-to-end Modeling Matters?

Lowest WER not necessarily gives the best BLEU
Fluent English is preferred as the input for MT, despite the fact that this might cause an increase of WER.

(He et al., 2011)
End-to-end ST Model

- A End-to-End log-linear model for ST
  - Enabling incorporation of rich features
  - Enabling a principal way to model the interaction between core components
  - Enabling global optimal modeling/feature training

\[
P(E|X) = \sum_F P(E, F|X) \\
P(E, F|X) = \frac{1}{Z} \exp \left\{ \sum_i \lambda_i \log h_i(E, F, X) \right\}
\]

\[
\hat{E} = \arg\max_E P(E|X)
\]
## Feature Functions for End-to-end ST

- Each feature function is a probabilistic model (except a few count features)
- Include all conventional ASR and MT models as features

<table>
<thead>
<tr>
<th>features</th>
<th>model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acoustic model</td>
<td>$h_{AM} = p(X</td>
</tr>
<tr>
<td>Source language model</td>
<td>$h_{SLM} = P_{LM}(F)$</td>
</tr>
<tr>
<td>ASR hypothesis length</td>
<td>$h_{SWC} = e^{</td>
</tr>
<tr>
<td>Forward phrase trans. Model</td>
<td>$h_{F2Eph} = \prod_k p(\tilde{e}_k</td>
</tr>
<tr>
<td>Forward word trans. Model</td>
<td>$h_{F2Ewd} = \prod_k \prod_m \Sigma_n p(e_{k,m}</td>
</tr>
<tr>
<td>Backward phrase trans. Model</td>
<td>$h_{E2Fph} = \prod_k p(\tilde{f}_k</td>
</tr>
<tr>
<td>Backward word trans. Model</td>
<td>$h_{E2Fwd} = \prod_k \prod_n \Sigma_m p(f_{k,n}</td>
</tr>
<tr>
<td>Translation Phrase count</td>
<td>$h_{PC} = e^K$</td>
</tr>
<tr>
<td>Translation hypothesis length</td>
<td>$h_{TWC} = e^{</td>
</tr>
<tr>
<td>Phrase segment/reorder model</td>
<td>$h_{reorder} = P_{hr}(S</td>
</tr>
<tr>
<td>Target language model</td>
<td>$h_{TLM} = P_{LM}(E)$</td>
</tr>
</tbody>
</table>
Learning Feature Weights In The Log-linear Model

• Jointly optimize the weights of features by MERT:

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (the current ASR and MT system)</td>
<td>33.8%</td>
</tr>
<tr>
<td>Global max-BLEU training (for all features)</td>
<td>35.2% (+1.4%)</td>
</tr>
<tr>
<td>Global max-BLEU training (for ASR-only features)</td>
<td>34.8% (+1.0%)</td>
</tr>
<tr>
<td>Global max-BLEU training (for SMT-only features)</td>
<td>34.2% (+0.4%)</td>
</tr>
</tbody>
</table>

Evaluated on a MS commercial data set
## Case Study:

<table>
<thead>
<tr>
<th>Transcript</th>
<th>it is great seeing you all here today</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation ref.</td>
<td>今天很高兴在这里见到你们</td>
</tr>
<tr>
<td>Reco A.</td>
<td>let’s great see you all here today</td>
</tr>
</tbody>
</table>
| Translation A. | 今天在这里看到你们让我们好
| Reco B. | let’s great to see you all here today |
| Translation B. | 我们今天很高兴在这里见到你们 |

Reco. B contains one more *ins.* error. However, the insertion “to”:

i) makes the MT input grammatically more correct;
ii) provides critical context for “great”;
iii) Provide critical syntactic info for word ordering of the translation.

<table>
<thead>
<tr>
<th>Transcript</th>
<th>i didn't ever really wanna do this</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation ref.</td>
<td>我从来没有真的想要这么做</td>
</tr>
<tr>
<td>Reco A.</td>
<td>i can never really wanna do this</td>
</tr>
<tr>
<td>Translation A.</td>
<td>我永远不能真的想</td>
</tr>
<tr>
<td>Reco B.</td>
<td>i ve never really want to do this</td>
</tr>
<tr>
<td>Translation B.</td>
<td>我从来没有真的要这么做</td>
</tr>
</tbody>
</table>

Reco. B contains two more errors. However, the mis-recognized phrase “want to” is plentifully represented in the formal text that is usually used for MT training and hence leads to correct translation.
Learning Parameters Inside The Features

• First, we define a generic differentiable utility function

\[ U(\Lambda) = \sum_{r=1}^{R} \sum_{E_r \in \text{hyp}(F_r)} \sum_{F_r \in \text{hyp}(X_r)} p(E_r, F_r | X_r, \Lambda) \cdot C(E_r, E'_r) \]

\( \Lambda \): the set of model parameters that are of interest
\( X_r \): the r-th speech input utterance
\( E'_r \): translation reference of the r-th utterance
\( E_r \): translation hypothesis of the r-th utterance
\( F_r \): speech recognition hypothesis of the r-th utterance

• \( U(\Lambda) \) measures the end-to-end quality of ST, e.g.,
  - Choosing \( C(E_r, E'_r) \) properly, \( U(\Lambda) \) covers a variety of ST metrics

<table>
<thead>
<tr>
<th>( C(E_r, E'_r) )</th>
<th>objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{BLEU}(E_r, E'_r) )</td>
<td>Max expected BLEU</td>
</tr>
<tr>
<td>( 1 - \text{TER}(E_r, E'_r) )</td>
<td>Min expected Translation Error Rate</td>
</tr>
</tbody>
</table>

(He & Deng 2013)
Objective, Regularization, and Optimization

• Optimize the utility function directly
  – E.g., max expected BLEU
  – Generic gradient-based methods (Gao & He 2013)
  – Early-stopping/cross-validation on dev set

• Regularize the objective by K-L divergence
  – Suitable for parameters in a probabilistic domain
  – Training objective:
    \[
    O(\Lambda) = \log U(\Lambda) - \tau \cdot KL(\Lambda^0 || \Lambda)
    \]
  – Extended Baum-Welch based optimization
    (He & Deng 2008, 2012, 2013)
EBW Formula for Translation Model

- Use lexicon translation model as an example

\[
p(g|h, \Lambda) = \frac{\sum_{k,m} \sum_{E,F} p(E,F|X,\Lambda') \Delta_E \gamma_h(k,m) + U(\Lambda') \tau_{FP} p(g|h, \Lambda^0) + D_h \cdot p(g|h, \Lambda')}{\sum_{k,m} \sum_{E,F} p(E,F|X,\Lambda') \Delta_E \gamma_h(k,m) + U(\Lambda') \tau_{FP} + D_h}
\]

where \( \Delta_E = [C(E) - U(\Lambda')] \), and \( \gamma_h(k,m) = \frac{\sum_{n:e_{k,n}=h} p(f_{k,m}|e_{k,n},\Lambda')}{\sum_{n} p(f_{k,m}|e_{k,n},\Lambda')} \)

ASR score affects estimation of the translation model

Training is influenced by translation quality
Evaluation on IWSLT’11/TED

• Challenging open-domain SLT: TED Talks (www.ted.com)
  – Public speech on anything (tech, entertainment, arts, science, politics, economics, policy, environment …)

• Data (Chinese-English machine translation track)
  – Parallel Zh-En data: 110K snt. TED transcripts; 7.7M snt. UN corpus
  – Monolingual English data: 115M snt. from Europarl, Gigaword, …
  – Example

  **English:** What I’m going to show you first, as quickly as I can, is some foundational work, some new technology that we brought to …

  **Chinese:** 首先，我要用最快的速度为大家演示一些新技术的基础研究成果 …
Results on IWSLT’11/TED

• Phrase-based system
  – 1st phrase table from the TED parallel corpus
  – 2nd phrase table from 500K parallel snt selected from UN
  – 1st 3-gram LM from TED English transcription
  – 2nd 5-gram LM from 115M supplementary English snt

• Max-BLEU training only applied to the primary (TED) phrase table
  – Fine-tuning of full lambda set is performed at the end

BLEU scores on IWSLT test sets

<table>
<thead>
<tr>
<th>system</th>
<th>Tst2010 (dev)</th>
<th>Tst2011 (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>11.48%</td>
<td>14.68%</td>
</tr>
<tr>
<td>Max Expected BLEU training</td>
<td>12.39% (+0.9%)</td>
<td>15.92% (+1.2%)</td>
</tr>
</tbody>
</table>

(He & Deng 2012)

← best single-system in IWSLT’11/CE_MT
Further Reading


Further Reading


• X. He, L. Deng, and A. Acero, 2011. Why Word Error Rate is not a Good Metric for Speech Recognizer Training for the Speech Translation Task?, in Proc. ICASSP, IEEE


• Y. Zhang, L. Deng, X. He, and A. Acero, 2011, A Novel Decision Function and the Associated Decision-Feedback Learning for Speech Translation, in ICASSP, IEEE
Practices

1. Overview: ASR, SMT, ST, Metric
2. Learning Problems in SMT
3. Translation Structures for ST
4. Decoding: A Unified Perspective ASR/SMT
5. Coupling ASR/SMT: Decoding & Modeling
6. Practices
7. Future Directions
Two techniques in depth:
Two techniques in depth:

1. Domain adaptation

2. System combination
Two techniques in depth:

1. Domain adaptation

2. System combination
Domain adaptation for ST

- Words differ in meaning across domains/contexts
- Domain adaptation is particularly important for ST
  - ST needs to handle spoken language
    - Colloquial style vs. written style
  - ST has interests on particular scenarios/domains
    - E.g., travel
Domain Adaptation by Data Selection for MT

- Selecting data that match the targeting domain from a large general corpus
- MT needs data that match both source & target languages of the targeting domain.
  - The adapted system need to
    - cover domain-specific input
    - produce domain-appropriate output
  - e.g., “Do you know of any restaurants open now?”
    - We need to select data to cover not only “open” at the source side, but also the right translation of “open” at the target side.

- Use bilingual cross-entropy difference:
  \[
  \left[ H_{in\_src}(s) - H_{out\_src}(s) \right] + \left[ H_{in\_tgt}(s) - H_{out\_tgt}(s) \right]
  \]

The perplexity of \(s\) given the in-domain source language model “in_src” (Axelrod, He, and Gao, 2011)
Multi-model and data selection based domain adaptation for ST

Bi-lingual data selection metric:

\[ \left[ H_{I-src}(s) - H_{O-src}(s) \right] + \left[ H_{I-tgt}(s) - H_{O-tgt}(s) \right] \]

**General text parallel data**

**Spoken lang. parallel data**

**Pseudo spk. lang. data**

**Data Selection**

**Model Training**

**Model Training**

**Spoken lang. TM**

**General TM**

Combined in a Log-linear model:

\[
P(E|F) = \frac{1}{Z} \exp \left\{ \sum_{i} \lambda_i \log f_i(E, F) \right\}
\]
Evaluation on commercial ST data

- English-to-Chinese translation
  - Target: dialog style travel domain data
  - Training data available
    - 800K in-domain data
    - 12M general domain data
  - Evaluation
    - Performance on target domain
    - Performance on general domain, e.g., evaluate the robustness on out-of-target-domain.
Results and Analysis

1) Using multiple translation models together as features helps
2) The feature weights, determined by the dev set, play a critical role
3) Big gain on in-domain test (travel), robust on out-of-domain test (general)

(From He & Deng, 2011)

<table>
<thead>
<tr>
<th>Models and ({\lambda}) training</th>
<th>travel</th>
<th>general</th>
</tr>
</thead>
<tbody>
<tr>
<td>General (dev: general)</td>
<td>16.22</td>
<td>18.85</td>
</tr>
<tr>
<td>Travel (dev: travel)</td>
<td>22.32 (+6.1)</td>
<td>10.81 (-8.0)</td>
</tr>
<tr>
<td>Multi-TMs (dev: travel)</td>
<td>22.12 (+5.9)</td>
<td>13.93 (-4.9)</td>
</tr>
<tr>
<td>Multi-TMs (dev: tvl : gen = 1:1)</td>
<td>22.01 (+5.8)</td>
<td>16.89 (-2.0)</td>
</tr>
<tr>
<td>Multi-TMs (dev: tvl : gen = 1:2)</td>
<td>20.24 (+4.0)</td>
<td>18.02 (-0.8)</td>
</tr>
</tbody>
</table>

Results reported in BLEU %
## Case Studies

<table>
<thead>
<tr>
<th>Source English</th>
<th>General model (Chinese)</th>
<th>Travel-domain adapted model</th>
</tr>
</thead>
<tbody>
<tr>
<td>A glass of cold water, please.</td>
<td>一杯冷水请。</td>
<td>请来一杯冷的水。</td>
</tr>
<tr>
<td>Please keep the change.</td>
<td>请保留更改。</td>
<td>不用找了。</td>
</tr>
<tr>
<td>Thank, this is for your tip.</td>
<td>谢谢，这是为您提示。</td>
<td>谢谢，这是给你的小费。</td>
</tr>
<tr>
<td>Do you know of any restaurants open now?</td>
<td>你现在知道的任何打开的餐馆吗？</td>
<td>你知道现在还有餐馆在营业的吗？</td>
</tr>
<tr>
<td>I'd like a restaurant with cheerful atmosphere</td>
<td>我想就食肆的愉悦的气氛</td>
<td>我想要一家气氛活泼的餐厅。</td>
</tr>
</tbody>
</table>

Note the improved translation of ambiguous words (e.g., open, tip, change), and improved processing of colloquial style grammar (e.g., *a glass of cold water, please.*)
From domain adaptation to topic adaptation

• Motivation
  – Topic changes talk to talk, dialog to dialog
    • Meanings of words changes, too.
  – Lots of out-of-domain data
    • Broad coverage, but not all of them are relevant.
    • How to utilize the OOD corpus to enhance the translation performance?

• Method
  – Build topic model on target domain data
  – Select topic relevant data from OOD corpus
  – Combine topic specific model with the general model at testing
Analysis on IWSLT’11: topics of TED talks

• Build LDA-based topic model on TED data
  – estimated on the source side of the parallel data
• Restricted to 4 topics – are they meaningful?
  – Only has 775 talks/110K sentences
• Look at some of the top keywords in each topic:
  – Topic 1 (design, computer, data, system, machine) technology
  – Topic 2 (Africa, dollars, business, market, food, China, society) global
  – Topic 3 (water, earth, universe, ocean, trees, carbon, environment) planet
  – Topic 4 (life, love, god, stories, children, music) abstract

• Reasonable clustering
Topic modeling, data selection, and multi-phrase-table decoding

Training (for each topic)

- Topic model
- UN parallel corpus
- TED parallel corpus

Testing

- Input sentence
- Topic allocation
- Topic specific multi-table decoding

Output

- Topic allocation
- Relevant data
- Model training
- Topic phrase table
- Combined in a Log-linear model

Log-linear combination:

\[ P(E|F) = \frac{1}{Z} \exp\{\lambda_i \log \varphi_i(E,F)\} \]
Experiments on IWSLT’11/TED

- For each topic, select 250~400K sentences from the UN corpus, train a topic-specific phrase table.
- Evaluation results on IWSLT’11 (TED talk dataset):
  - Simply adding an extra UN-driven phrase table didn’t help.
  - Topic specific multi-phrase table decoding helps.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TED only (baseline)</td>
<td>11.3%</td>
<td>13.0%</td>
</tr>
<tr>
<td>TED + UN-all</td>
<td>11.3%</td>
<td>13.0%</td>
</tr>
<tr>
<td>TED + UN-4 topics</td>
<td>11.8%</td>
<td>13.5%</td>
</tr>
</tbody>
</table>

(From Axelrod et al., 2012)
System Combination for MT

\begin{itemize}
\item \textit{Hypotheses from single systems} \\
\begin{align*}
E_1 &: \text{ she bought the Jeep} \\
E_2 &: \text{ she buys the SUV} \\
E_3 &: \text{ she bought the SUV Jeep}
\end{align*}
\item \textit{Combined MT output} \\
\text{ she bought the Jeep SUV}
\end{itemize}

- \textit{System Combination:}
  - \textit{Input:}
    a set of translation hypotheses from multiple single MT systems, for the same source sentence
  - \textit{Output:}
    a \textit{better} translation output derived from input hypotheses
Confusion Network Based Methods

**Hypotheses from single systems**

- $E_1$: she bought the Jeep
- $E_2$: she buys the SUV
- $E_3$: she bought the SUV Jeep

**Combined MT output**

- she bought the SUV

1) **Select the backbone**

$$E_B = \arg \min_{E \in \mathcal{E}} \sum_{E' \in \mathcal{E}'} P(E | F) L(E', E)$$

e.g., $E_2$ is selected

2) **Align hypotheses**

3) **Construct and decode CN**
Hypothesis Alignment

- Hypothesis alignment is crucial
  - GIZA based approach (Matusov et al., 2006)
    - Use GIZA as a generic tool to align hypotheses
  - TER based alignment (Sim et al., 2007, Rosti et al. 2007)
    - Align one hypothesis to another such that the TER is minimized
  - HMM based hypothesis alignment (He et al. 2008, 2009)
    - Use fine-grained statistical model
    - No training needed, HMM’s parameters are derived from pre-trained bilingual word alignment models
  - ITG based approach (Karakos et al., 2008)
  - A latest survey and evaluation (Rosti et al., 2012)
HMM based Hypothesis Alignment

\[ E_B : \ e_1 \ e_2 \ e_3 \]
\[ E_{hyp} : \ e'_1 \ e'_3 \ e'_2 \]

- HMM is built on the backbone side
- HMM aligns the hypothesis to the backbone
- After alignment, a CN is built
HMM Parameter Estimation

- Emitting Probability (via words in source sentence)
  - $P(e'_1|e_1)$ models how likely $e'_1$ and $e_1$ have similar meanings
  - Use the source word sequence $\{f_1, ..., f_M\}$ as a hidden layer, $P(e'_1|e_1)$ takes a mixture-model form, i.e.,
    \[ P_{src}(e'_1|e_1) = \sum_m w_m P(e'_1|f_m) \]
    where $w_m = P(f_m|e_1)/\sum_m P(f_m|e_1)$
  - $P(e'_1|f_m)$ is from the bilingual word alignment model, $F \rightarrow E$ direction
  - $P(f_m|e_1)$ is from that of $E \rightarrow F$

\[ P(e'_1|e_1) = \sum_m w_m P(e'_1|f_m) \]
HMM Parameter Estimation (cont.)

- **Emitting Probability**  (via word surface similarity)
  - Normalized similarity measure \( s \)
    - Based on Levenshtein distance
    - Based on matched prefix length
  - Use an exponential mapping to get
    \[
    P(e'_1|e_1) = \exp[\rho \cdot (s(e'_1, e_1) - 1)]
    \]
  - \( s(e'_1, e_1) \) is normalized to \([0,1]\)

- **Overall Emitting Probability**
  \[
  P(e'_1|e_1) = \alpha \cdot P_{src}(e'_1|e_1) + (1 -\alpha) \cdot P_{simi}(e'_1|e_1)
  \]
HMM Parameter Estimation (cont.)

- Transition Probability
  - $P(a_j|a_{j-1})$ models word ordering
    - Takes the same form as a bilingual word alignment HMM
    - Strongly encourages monotonic word ordering
    - Allows non-monotonic word ordering

$$d(i,i') = \left(1 + |i - i' - 1|\right)^{-\kappa}$$

$$P(a_j | a_{j-1}) = \frac{d(a_j, a_{j-1})}{\sum_i d(i, a_{j-1})}$$

$a_j$ – alignment of the $j$-th word
Find the Optimal Alignment

• Viterbi decoding:

\[
\hat{a}_1^J = \arg\max_{a_1^J} \prod_{j=1}^J \left[ p(a_j | a_{j-1}, I) p(e'_j | e_{a_j}) \right]
\]

• Other variations:
  – posterior probability & threshold based decoding
  – max posterior mode decoding
Decode the Confusion Network

- Log-linear model based decoding (Rosti et al. 2007)
  - Incorporate multiple features (e.g., voting, LM, length, etc.)

\[
E^* = \arg \max_{E' \in E_h} \ln P(E' | F)
\]

where

\[
\ln P(E' | F) = \ln \prod_{l=1}^{L} P_{S-MBR}(e'_l | F) + \nu \ln P_{LM}(E') + \xi |E'| 
\]

Confusion network decoding \((L=5)\)

<table>
<thead>
<tr>
<th>he</th>
<th>have</th>
<th>e</th>
<th>good</th>
<th>car</th>
</tr>
</thead>
<tbody>
<tr>
<td>he</td>
<td>has</td>
<td>e</td>
<td>nice</td>
<td>sedan</td>
</tr>
<tr>
<td>it</td>
<td>e</td>
<td>a</td>
<td>nice</td>
<td>car</td>
</tr>
<tr>
<td>he</td>
<td>has</td>
<td>a</td>
<td>e</td>
<td>sedan</td>
</tr>
</tbody>
</table>

\(e_1\) \(e_2\) \(e_3\) \(e_4\) \(e_5\)
Results on 2008 NIST Open MT Eval

The MSR-NRC-SRI entry for Chinese-to-English

Case-sensitive BLEU-4

Combined System
best score in NIST08 C2E

Individual Systems

(from He et al., 2008)
Related Approaches & Extensions

• Incremental hypothesis alignment
  – (Rosti et al., 2008; Li et al, 2009, Karakos et al., 2010)

• Other approaches
  – Joint decoding (He and Toutanova, 2009)
  – Phrase-level system combination (Feng et al, 2009)
  – System combination as target-to-target decoding (Ma & McKeown, 2012)

• Survey and evaluation
  – (Rosti et al., 2012)
Comparison of aligners

Results reported in BLEU % (best scores in bold)
Significance group ID is marked as superscripts on results

<table>
<thead>
<tr>
<th>Aligner</th>
<th>NIST MT09 Arabic-English</th>
<th>WMT11 German-English</th>
<th>WMT11 Spanish-English</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best single system</td>
<td>51.74</td>
<td>24.16</td>
<td>30.14</td>
</tr>
<tr>
<td>GIZA</td>
<td>57.95(^1)</td>
<td>26.02(^1)</td>
<td>33.62(^1)</td>
</tr>
<tr>
<td>Incr. TER</td>
<td>58.63(^2)</td>
<td>26.39(^2)</td>
<td>33.79(^1)</td>
</tr>
<tr>
<td>Incr. TER++</td>
<td>59.05(^2)</td>
<td>26.10(^1)</td>
<td>33.61(^1)</td>
</tr>
<tr>
<td>Incr. ITG++</td>
<td><strong>59.37(^3)</strong></td>
<td><strong>26.50(^2)</strong></td>
<td>33.85(^1)</td>
</tr>
<tr>
<td>Incr. IHMM</td>
<td><strong>59.27(^3)</strong></td>
<td><strong>26.40(^2)</strong></td>
<td><strong>34.05(^2)</strong></td>
</tr>
</tbody>
</table>

(From Rosti, He, Karakos, Leusch, et al., 2012)
Error Analysis

• Per-sentence errors were analyzed
  – Paired Wilcoxon test measured the probability that the error reduction relative to the best system was real

• Observations
  – All aligners significantly reduce substitution/shift errors
  – No clear trend for insertions/deletions – sometimes worse than the best system

• Further analysis showed that agreement for the non-NULL words is important measure for alignment quality, but agreement on NULL tokens is not
Further Reading

- C. Li, X. He, Y. Liu, and N. Xi, 2009, Incremental HMM Alignment for MT System Combination, in Proceedings of ACL-IJCNLP.
- A. Axelrod, X. He, and J. Gao, Domain Adaptation via Pseudo In-Domain Data Selection, in Proc. EMNLP, 2011.
- B. Haddow and P. Koehn, 2012. Analysing the effect of out-of-domain data on smt systems. in Proceedings of WMT.
Summary

- Fundamental concepts and technologies in speech translation
- Theory of speech translation from a joint modeling perspective
- In-depth discussions on domain adaptation and system combination
- Practical evaluations on major ST/MT benchmarks
ST Future Directions
--Additional to ASR and SMT

• Overcome the low-resource problem:
  – How to rapidly develop ST with limited amount of data, and adapt SMT for ST?
• Confidence measurement:
  – Reduce misleading dialogue turns & lower the risk of miscommunication.
  – Complicated by the combination of two sources of uncertainty in ASR and SMT.
• Active vs. Passive ST
  – More active, e.g., to actively clarify, or warn the user when confidence is low
  – Development of suitable dialogue strategies for ST.
• Context-aware ST
  – ST on smart phones (e.g., location, conversation history).
• End-to-end modeling of speech translation with new features
  – E.g., prosody acoustic features for translation via end-to-end modeling
• Other applications of speech translation
  – E.g., cross-lingual spoken language understanding
Further Reading About this Tutorial

• This tutorial is mainly based on:

• Both provide references for further reading
• Papers and charts available on authors’ webpages
Resources (play it yourself)

• You need also an ASR system
  – HTK: http://htk.eng.cam.ac.uk/
  – Kaldi: http://kaldi.sourceforge.net/
  – HTS (TTS for S2S): http://hts.sp.nitech.ac.jp/
Resources (play it yourself)

- You need also an ASR system
  - HTK: http://htk.eng.cam.ac.uk/
  - Kaldi: http://kaldi.sourceforge.net/
  - HTS (TTS for S2S): http://hts.sp.nitech.ac.jp/

- MT Toolkits (word alignment, decoder, MERT)
  - GIZA++: http://www.statmt.org/moses/giza/GIZA++.html
  - Moses: http://www.statmt.org/moses/
  - Cdec: http://cdec-decoder.org/
Resources (play it yourself)

- You need also an ASR system
  - HTK: http://htk.eng.cam.ac.uk/
  - Kaldi: http://kaldi.sourceforge.net/
  - HTS (TTS for S2S): http://hts.sp.nitech.ac.jp/

- MT Toolkits (word alignment, decoder, MERT)
  - GIZA++: http://www.statmt.org/moses/giza/GIZA++.html
  - Moses: http://www.statmt.org/moses/
  - Cdec: http://cdec-decoder.org/

- Benchmark data sets
  - IWSLT: http://www.iwslt2013.org
  - WMT: http://www.statmt.org (include Europarl)
  - NIST Open MT: http://www.itl.nist.gov/ia/mt/
Bowen Zhou
RESEARCH MANAGER
IBM WATSON RESEARCH CENTER
LinkedIn http://www.linkedin.com/pub/bowen-zhou/18/b5a/436

Xiaodong He
RESEARCHER
MICROSOFT RESEARCH REDMOND
AFFILIATE PROFESSOR
UNIVERSITY OF WASHINGTON, SEATTLE
http://research.microsoft.com/~xiaobe