Automatic Determination of Number of clusters for creating templates in Example-Based Machine Translation

Rashmi Gangadharaih, Ralf D. Brown and Jaime Carbonell

Presentation: Bob Frederking
Outline of this talk

1. Our EBMT System
2. G-EBMT: Use of templates
3. Automatically determine the number of clusters
   - Word-Generalized Templates in TM
   - Word-Generalized Templates in LM
4. Results
Outline of this talk

1. Our EBMT System

2. G-EBMT: Use of templates

3. Automatically determine the number of clusters
   - Word-Generalized Templates in TM
   - Word-Generalized Templates in LM

4. Results
R. D. Brown et. al., 2003
EBMT requires large amounts of data to function well.

- Decoding is expensive with long input sentences and short phrasal candidates.
  - Place restrictions on the decoder
  - Obtain local reordering information
    - Increase corpus size to obtain longer target phrasal matches.

Hence, **EBMT requires large amounts of data to function well.**
Sparse Data

- EBMT systems like other corpus-based methods require large amounts of data to function well.
  - *But*, obtaining parallel text is time-consuming, expensive and difficult.
  - Effect of less data on EBMT:
    - Reduces translation quality due to absence of longer phrasal matches.

*How do we obtain longer phrasal matches in data sparse conditions?*
Outline of this talk

1. Our EBMT System
2. G-EBMT: Use of templates
3. Automatically determine the number of clusters
   - Word-Generalized Templates in TM
   - Word-Generalized Templates in LM
4. Results
How do templates help in data sparse conditions

S1: The session opened at 6 pm. ↔ La séance est ouverte à 6 heures.
T1: The <event> opened at <time> ↔ La <event> est ouverte à <time>.

• If, “session(séance)”,”seminar(séminaire)” belong to <event> and, “6 pm(6 heures),2pm(2 heures),9am(9 heures)” belong to <time> class.

• T1 can now translate:
  - The session opened at 2 pm.
  - The seminar opened at 9 am.
Templates in TM

- Example training corpus:
  - $S_1$: The Minister gave a speech on Wednesday.
    $T_1$: Le ministre a donné un discours mercredi.
  - $S_2$: The President gave a speech on Monday.
    $T_2$: Le président a donné un discours lundi.

- Example word-pair Clusters:
  - <CL0>: Minister-$ministre$, President-$président$, ...
  - <CL1>: Wednesday-$mercredi$, Monday-$lundi$, ...

- Generalized template (T):
  - The <CL0> gave a speech on <CL1>.
    Le $<CL0>$ a donné un discours $<CL1>$.

- $I$: The President gave a speech on Wednesday.
Templates in TM

- I: The **President** gave a speech on **Wednesday**.

Example word-pair Clusters:
- <CL0>: **Minister**-ministre, **President**-président, ..
- <CL1>: **Wednesday**-mercredi, **Monday**-lundi, ..

Generalized template (T):
- The <CL0> gave a speech on <CL1>.
  
  *Le <CL0> a donné un discours <CL1>.*
Templates in TM

- _I_: The President gave a speech on Wednesday.

- _ITS_: The `<CL0>` gave a speech on `<CL1>`.
  - _ITT_: `Le `<CL0>` a donné un discours `<CL1>`.

Example word-pair Clusters:
- `<CL0>`: Minister-`ministre`, President-`président`, ..
- `<CL1>`: Wednesday-`mercredi`, Monday-`lundi`, ..

Generalized template (T):
- The `<CL0>` gave a speech on `<CL1>`.
  - `Le `<CL0>` a donné un discours `<CL1>`.

Templates in TM

- 1: The President gave a speech on Wednesday.

- ITS: The <CL0> gave a speech on <CL1>.
  ITT: Le <CL0> a donné un discours <CL1>.

- O: Le président a donné un discours mercredi.

Example word-pair Clusters:

- <CL0>: Minister-ministre, President-président,..
- <CL1>: Wednesday-mercredi, Monday-lundi,..

Generalized template (T):

- The <CL0> gave a speech on <CL1>.
  Le <CL0> a donné un discours <CL1>.
Usefullness of templates in G-EBMT systems that use Statistical decoders

- EBMT systems that use statistical decoders.
  - Constraints on decoder.
  - Extract longer phrasal matches.

“Le président a donné un discours mercredi” vs. “Le président a donné” and “mercredi”
Related Work: Templates resemble Transfer Rules

- **Traditional Rule-based MT (trad. RBMT)**
  - includes Xfer-based MT and interlingua-based MT
  - transformations based on structural rules or interlingua
  - manually built transfer rules made up of non-terminal (NT) labels with constraints and lexicon to translate source words.

- **Xfer-based MT (Lavie, 2008)**
  - similar to trad. RBMT with manually/automatically built transfer rules containing T and NT labels with constraints.
  - rules extracted by aligning source and target parse trees.

- **Syntax-based SMT**
  - Yamada and Knight (2001) statistical model containing transfer rules of NT labels to reorder child nodes, insert extra words and translate leaf words in the source parse tree.
  - Heiro (Chiang et. al., 2005) is a stochastic synchronous CFG consisting of pairs of CFG rules with aligned NT labels.
EBMT templates provide more flexibility

- Flat (not nested) structural templates contain both T and NT labels with fewer or no constraints
- NT labels not necessarily linguistics-based syntactic phrases
- any sequence of one or more words forms a phrase
Related Work

- Methods that generalize differences and similarities
  - ([Cicekli and Guvenir, 2001]; [McTait, 2001]) use only similar and dissimilar portions limiting the amount of generalization
  - Recursive transfer-rule induction process (Brown, 2001) combining (Cicekli and Guvenir, 2001) and word clustering (Brown, 2000) based on context, but finds the number of clusters empirically.

- Methods that generalize chunk translations
  - (Kaji et al., 1992) extract phrase pairs from parse trees hence, templates created are less controllable
  - (Block, 2000) extracts chunk pairs from word alignments, can cause over-generalization increasing decoding time
  - (Carl, 2001) similar to (Block, 2000) but use bracketing Gaijin (Veale and Way, 1997) uses only marker hypothesis
Outline of this talk

1. Our EBMT System
2. G-EBMT: Use of templates
3. Automatically determine the number of clusters
   - Word-Generalized Templates in TM
   - Word-Generalized Templates in LM
4. Results
MT in Data Sparse Conditions: EBMT

Automatically determine the number of clusters

Clustering Algorithm to obtain templates

- Automatically cluster words based on context
  - Selecting a clustering algorithm
    - simple in design
    - automatically determine the number of clusters
    - high quality clusters
Clustering Algorithm

- Automatically cluster words based on context
  - Spectral Clustering (NJW algorithm)
    - Cluster points using the eigenvectors of distance matrices obtained from data. Features: form *term vectors* for each word-pair by accumulating counts for tokens in its context.
    - Superior to Group Average Clustering (Gangadharaiyah et. al., 2006)
  - Automatically determine the number of clusters [modified (Sanguinetti et al., 2005)].
Finding number of clusters (N)

Modified algorithm of (Sanguinetti et al., 2005):
Artificially generated data

Real data
### Cluster Purity

<table>
<thead>
<tr>
<th>Impure clusters</th>
<th>Pure clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&quot;almost&quot; &quot;presque&quot;)</td>
<td></td>
</tr>
<tr>
<td>(&quot;certain&quot; &quot;certains&quot;)</td>
<td></td>
</tr>
<tr>
<td>(&quot;his&quot; &quot;sa&quot;)</td>
<td>(&quot;his&quot; &quot;sa&quot;)</td>
</tr>
<tr>
<td>(&quot;his&quot; &quot;son&quot;)</td>
<td>(&quot;his&quot; &quot;son&quot;)</td>
</tr>
<tr>
<td>(&quot;its&quot; &quot;sa&quot;)</td>
<td>(&quot;its&quot; &quot;sa&quot;)</td>
</tr>
<tr>
<td>(&quot;its&quot; &quot;ses&quot;)</td>
<td>(&quot;its&quot; &quot;ses&quot;)</td>
</tr>
<tr>
<td>(&quot;last&quot; &quot;hier&quot;)</td>
<td></td>
</tr>
<tr>
<td>(&quot;my&quot; &quot;mes&quot;)</td>
<td>(&quot;my&quot; &quot;mes&quot;)</td>
</tr>
<tr>
<td>(&quot;my&quot; &quot;mon&quot;)</td>
<td>(&quot;my&quot; &quot;mon&quot;)</td>
</tr>
<tr>
<td>(&quot;our&quot; &quot;nos&quot;)</td>
<td>(&quot;our&quot; &quot;nos&quot;)</td>
</tr>
<tr>
<td>(&quot;our&quot; &quot;notre&quot;)</td>
<td>(&quot;our&quot; &quot;notre&quot;)</td>
</tr>
<tr>
<td>(&quot;their&quot; &quot;leur&quot;)</td>
<td>(&quot;their&quot; &quot;leur&quot;)</td>
</tr>
<tr>
<td>(&quot;their&quot; &quot;leurs&quot;)</td>
<td>(&quot;their&quot; &quot;leurs&quot;)</td>
</tr>
<tr>
<td>(&quot;these&quot; &quot;ces&quot;)</td>
<td>(&quot;these&quot; &quot;ces&quot;)</td>
</tr>
<tr>
<td>(&quot;too&quot; &quot;trop&quot;)</td>
<td></td>
</tr>
<tr>
<td>(&quot;without&quot; &quot;sans&quot;)</td>
<td>(&quot;his&quot; &quot;ses&quot;)</td>
</tr>
</tbody>
</table>

**Table:** Cluster purity before and after removal of oscillating points with 10k Eng-Fre ($th_1 > 9$)
Previous Approaches

- Data sparsity is a big challenge in statistical LM.
- n-gram Class-based (CB) Language Models (Brown et al., 1992)

\[ p(w_i | h) = p(w_i | c_i) \times p(c_i | c_{i-1}, \ldots, c_{i-n+1}) \]

- Words grouped based on POS tags or automatically clustered
- Require all words present in the training data to be clustered
  - Unreliable clusters if errors in the data (e.g., segmentation)
- Factored Language Models (Kirchhoff and Yang, 2005)
  - Word represented by linguistic features
  - Extremely large model space with many backoff paths
Our approach: Template-based (TB)

Alternate approach

- based on using short reusable sequences or ‘templates’ made up of words and class labels
- Does not require all words to be clustered
  - Helpful when a small set of manually built clusters are present
- How to form reliable clusters when manually built clusters are not available?
  - use clustering approach adopted in the TM

Note: CB can be made equivalent to TB

- when unreliable words are treated as singleton clusters.
Template-based Model

- Assume corpus C contains S1 and S2
  - S1: the school reopens on Monday
  - S2: the office is too far
Template-based Model

- Assume corpus C contains S1 and S2
  - S1: the school reopens on Monday
  - S2: the office is too far

- Assume <ORG> and <WEEKDAY> are obtained either manually or automatically
  - <ORG>: school, company, office
  - <WEEKDAY>: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.
Template-based Model

- Assume corpus C contains S1 and S2
  
  S1: the school reopens on Monday
  S2: the office is too far

- Assume <ORG> and <WEEKDAY> are obtained either manually or automatically
  
  <ORG>: school, company, office
  <WEEKDAY>: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.

- Templates T1 and T2 are obtained from S1 and S2
  
  T1: the <ORG> reopens on <WEEKDAY>
  T2: the <ORG> is too far
Template-based Model

- Assume corpus C contains S1 and S2
  S1: the school reopens on Monday
  S2: the office is too far

- Assume <ORG> and <WEEKDAY> are obtained either manually or automatically
  <ORG>: school, company, office
  <WEEKDAY>: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday.

- Templates T1 and T2 are obtained from S1 and S2
  T1: the <ORG> reopens on <WEEKDAY>
  T2: the <ORG> is too far

- If “p(reopens | the office)” is encountered during decoding
  - Word-based model: backs-off to unigram score, p(reopens)
  - Template-based model: gives a more reliable score, p(reopens | the <ORG>)
Formal Description

\[ p(w_i | h) \approx x p(f_i | f_{i-1}, \ldots, f_{i-n+1}) \]

\[ f_j = \begin{cases} 
  c(w_j), & \text{if } w_j^{th} \text{ class is present} \\
  w_j, & \text{otherwise}
\end{cases} \]

\[ x = \begin{cases} 
  p(w_i | c(w_i)), & \text{if } w_i^{th} \text{ class is present} \\
  1, & \text{otherwise}
\end{cases} \]

- The probability of the \( i^{th} \) word \( (w_i) \) given its history \( h \) is represented as the probability of feature \( f_i \) corresponding to \( w_i \) given its previous history of features.
- Each \( f_i \) can represent a word \( w_j \) or its class \( c(w_j) \).
Incorporating Template-based models

- EBMT engine assigns a quality score \( q_i \) to phrasal translations
  - Log-linear combination of alignment and translation score
- Our decoder works on a lattice of phrasal translations
  - Total score for a path

\[
\text{total score} = \frac{1}{n} \sum_{i=1}^{n} \left[ wt_1 \times \log(b_i) + wt_2 \times \log(pen_i) + wt_3 \times \log(q_i) + wt_4 \times \log(P(w_i|w_{i-2}, w_{i-1})) \right]
\]

- \( n \): number of target words in the path, \( wt_j \): importance of each score, \( b_i \): bonus factor, \( pen_i \): penalty factor, \( P(w_i|w_{i-2}, w_{i-1}) \): LM score.
- Template-based and word-based language model scores are interpolated.
Outline of this talk

1. Our EBMT System
2. G-EBMT: Use of templates
3. Automatically determine the number of clusters
   - Word-Generalized Templates in TM
   - Word-Generalized Templates in LM
4. Results
Experimental Setup(1)

- **English-Haitian**: The English–Haitian medical domain data (Haitian Creole, CMU, 2010)
  - Training Data: 1219 sentence pairs.
  - Tune Set: 200 sentence pairs, Test Data: 200 sentence pairs.

- **English–Chinese**: FBIS (NIST 2003)
  - Training Data: 15k, 30k and 200k sentence pairs.
  - Tune Set: 200 sentence pairs, Test Data: 4000 sentence pairs.

- **English-French**: Hansard Corpus (LDC)
  - Training Data: 10k, 30k and 100k sentence pairs.
  - Tune Set: 200 sentence pairs, Test Data: 4000 sentence pairs.
Experimental Setup(2)

- Language Models:
  - the target half of the training data.
  - 5-grams Language Models

- Statistical significance: Wilcoxon Signed-Rank Test.
# Results

<table>
<thead>
<tr>
<th>Lang-Pair</th>
<th>data</th>
<th>Manual</th>
<th>SangAlgo</th>
<th>Mod Algo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eng-Fre(TM)</td>
<td>10k</td>
<td>0.1777</td>
<td>0.1641</td>
<td>0.1790</td>
</tr>
<tr>
<td>Eng-Chi(LM)</td>
<td>30k</td>
<td>0.1290</td>
<td>0.1257</td>
<td>0.1300</td>
</tr>
</tbody>
</table>

**Table:** BLEU scores with templates created using manually, SangAlgo and the modified algorithm to find $N$ on 10k English-French and 30k English-Chinese training data.
### Results

<table>
<thead>
<tr>
<th>Lang-Pair</th>
<th>Baseline</th>
<th>LM</th>
<th>TM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eng-Chi 15k</td>
<td>0.1076</td>
<td>0.1098</td>
<td>0.1102</td>
</tr>
<tr>
<td>Eng-Chi 30k</td>
<td>0.1245</td>
<td>0.1300</td>
<td>0.1338</td>
</tr>
<tr>
<td>Eng-Chi 200k</td>
<td>0.1905</td>
<td>0.1936</td>
<td>0.1913</td>
</tr>
<tr>
<td>Eng-Haitian</td>
<td>0.2182</td>
<td>0.2370</td>
<td>0.229</td>
</tr>
</tbody>
</table>

**Table:** BLEU scores with templates applied in LM and TM with 15k, 30k and 200k English-Chinese, and English-Haitian training data.
Conclusion and Future Work

- introduced a method for automatically finding the number of clusters (N) for a real world problem.
- refined the clustering process by removing incoherent points and showed that discarding these points boosts the translation quality.
- showed significant improvements by adding generalized templates.

Future Work:
Template-based systems with larger training data sets.
Backup Slides: Term Vectors

- A rough mapping between source and target words is created.
- For each word pair accumulate counts for each word in the surrounding context of its occurrences (N=3).
- Weigh the counts w.r.t distance from occurrence with a linear decay.

<table>
<thead>
<tr>
<th>word</th>
<th>occur</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NULL&gt;(-3)</td>
<td>1</td>
<td>0.333</td>
</tr>
<tr>
<td>&lt;NULL&gt;(-2)</td>
<td>1</td>
<td>0.667</td>
</tr>
<tr>
<td>commenceront(-2)</td>
<td>1</td>
<td>0.667</td>
</tr>
<tr>
<td>Le(-1)</td>
<td>1</td>
<td>1.000</td>
</tr>
<tr>
<td>en(-1)</td>
<td>1</td>
<td>1.000</td>
</tr>
<tr>
<td>jours(1)</td>
<td>2</td>
<td>2.000</td>
</tr>
<tr>
<td>depuis(2)</td>
<td>1</td>
<td>0.667</td>
</tr>
<tr>
<td>.(2)</td>
<td>1</td>
<td>0.667</td>
</tr>
<tr>
<td>la(3)</td>
<td>1</td>
<td>0.333</td>
</tr>
<tr>
<td>&lt;NULL&gt;(3)</td>
<td>1</td>
<td>0.333</td>
</tr>
</tbody>
</table>

MT in Data Sparse Conditions: EBMT

Results

S1: <NULL><NULL> Le cinq jours depuis la
T1:<NULL><NULL> The five days since the elles
S2: elles commenceront en cinq jours . <NULL>
T2: They will begin in five days . <NULL>
Automatically determine the number of clusters: Modified algorithm of (Sanguinetti et al., 2005):

- runs iteratively starting with three clusters and performs a modified version of $k$-means clustering to detect if points are assigned to the origin
- When $q$ is less than best, points that are not close to any of the $q$ centers, get assigned to the origin.
- $q = q + 1$ if points assigned to the origin and repeat
- Halt if there are no points assigned to the origin