Winter School

Day 5: Discriminative Training and Factored Translation Models

MT Marathon

30 January 2009
The birth of SMT: generative models

• The definition of translation probability follows a mathematical derivation

$$\arg\max_e p(e|f) = \arg\max_e p(f|e) \ p(e)$$

• Occasionally, some independence assumptions are thrown in for instance IBM Model 1: word translations are independent of each other

$$p(e|f,a) = \frac{1}{Z} \prod_i p(e_i|f_{a(i)})$$

• Generative story leads to straight-forward estimation
  – maximum likelihood estimation of component probability distribution
  – EM algorithm for discovering hidden variables (alignment)
Log-linear models

• IBM Models provided mathematical justification for factoring components together

\[ p_{LM} \times p_{TM} \times p_D \]

• These may be weighted

\[ p_{LM}^{\lambda_{LM}} \times p_{TM}^{\lambda_{TM}} \times p_D^{\lambda_D} \]

• Many components \( p_i \) with weights \( \lambda_i \)

\[ \prod_i p_i^{\lambda_i} = \exp(\sum_i \lambda_i \log(p_i)) \]

\[ \log \prod_i p_i^{\lambda_i} = \sum_i \lambda_i \log(p_i) \]
Knowledge sources

- Many different knowledge sources useful
  - language model
  - reordering (distortion) model
  - phrase translation model
  - word translation model
  - word count
  - phrase count
  - drop word feature
  - phrase pair frequency
  - additional language models
  - additional features
Set feature weights

• Contribution of components $p_i$ determined by weight $\lambda_i$

• Methods
  – *manual setting* of weights: try a few, take best
  – *automate* this process

• Learn weights
  – set aside a *development corpus*
  – set the weights, so that *optimal translation performance* on this development corpus is achieved
  – requires *automatic scoring* method (e.g., BLEU)
Discriminative training

- Training set (*development set*)
  - different from original training set
  - small (maybe 1000 sentences)
  - must be different from test set

- Current model *translates* this development set
  - *n-best list* of translations (n=100, 10000)
  - translations in n-best list can be *scored*

- Feature weights are *adjusted*

- N-Best list generation and feature weight adjustment repeated for a number of iterations
Discriminative training

Model

- Change feature weights
- Find feature weights that move up good translations
- Score translations
- Generate n-best list

1 2 3 4 5 6

MT Marathon Winter School, Lecture 5 30 January 2009
Discriminative vs. generative models

- Generative models
  - translation process is broken down to steps
  - each step is modeled by a probability distribution
  - each probability distribution is estimated from the data by maximum likelihood

- Discriminative models
  - model consist of a number of features (e.g. the language model score)
  - each feature has a weight, measuring its value for judging a translation as correct
  - feature weights are optimized on development data, so that the system output matches correct translations as close as possible
Learning task

- Task: *find weights*, so that feature vector of best translations *ranked first*
- Input: *Er geht ja nicht nach Hause*, Ref: *He does not go home*

<table>
<thead>
<tr>
<th>Translation</th>
<th>Feature values</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>it is not under house</td>
<td>-32.22</td>
<td>-5</td>
</tr>
<tr>
<td>he is not under house</td>
<td>-34.50</td>
<td>-5</td>
</tr>
<tr>
<td>it is not a home</td>
<td>-28.49</td>
<td>-5</td>
</tr>
<tr>
<td>it is not to go home</td>
<td>-32.53</td>
<td>-6</td>
</tr>
<tr>
<td>it is not for house</td>
<td>-31.75</td>
<td>-5</td>
</tr>
<tr>
<td>he is not to go home</td>
<td>-35.79</td>
<td>-6</td>
</tr>
<tr>
<td><strong>he does not home</strong></td>
<td><strong>-32.64</strong></td>
<td><strong>-4</strong></td>
</tr>
<tr>
<td>it is not packing</td>
<td>-32.26</td>
<td>-4</td>
</tr>
<tr>
<td>he is not packing</td>
<td>-34.55</td>
<td>-4</td>
</tr>
<tr>
<td>he is not for home</td>
<td>-36.70</td>
<td>-5</td>
</tr>
</tbody>
</table>
Och’s minimum error rate training (MERT)

- **Line search** for best feature weights

  - given: sentences with n-best list of translations
  - iterate n times
    - randomize starting feature weights
    - iterate until convergences
      - for each feature
        - find best feature weight
        - update if different from current
  - return best feature weights found in any iteration
Find Best Feature Weight

• Core task:
  – find optimal value for one parameter weight $\lambda$
  – ... while leaving all other weights constant

• Score of translation $i$ for a sentence $f$:

$$p(e_i|f) = \lambda a_i + b_i$$

• Recall that:
  – we deal with 100s of translations $e_i$ per sentence $f$
  – we deal with 100s or 1000s of sentences $f$
  – we are trying to find the value $\lambda$ so that over all sentences, the error score is optimized
Translations for one Sentence

- each translation is a line $p(e_i|f) = \lambda a_i + b_i$
- the model-best translation for a given $\lambda$ (x-axis), is highest line at that point
- there are one a few threshold points $t_j$ where the model-best line changes
Finding the Optimal Value for $\lambda$

- Real-valued $\lambda$ can have infinite number of values
- But only on threshold points, one of the model-best translation changes

⇒ Algorithm:
  - find the threshold points
  - for each interval between threshold points
    * find best translations
    * compute error-score
  - pick interval with best error-score
• Varying one parameter: a rugged line with many local optima
Unstable outcomes: weights vary

<table>
<thead>
<tr>
<th>component</th>
<th>run 1</th>
<th>run 2</th>
<th>run 3</th>
<th>run 4</th>
<th>run 5</th>
<th>run 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>distance</td>
<td>0.059531</td>
<td>0.071025</td>
<td>0.069061</td>
<td>0.120828</td>
<td>0.120828</td>
<td>0.072891</td>
</tr>
<tr>
<td>lexdist 1</td>
<td>0.093565</td>
<td>0.044724</td>
<td>0.097312</td>
<td>0.108922</td>
<td>0.108922</td>
<td>0.062848</td>
</tr>
<tr>
<td>lexdist 2</td>
<td>0.021165</td>
<td>0.008882</td>
<td>0.008607</td>
<td>0.013950</td>
<td>0.013950</td>
<td>0.030890</td>
</tr>
<tr>
<td>lexdist 3</td>
<td>0.083298</td>
<td>0.049741</td>
<td>0.024822</td>
<td>-0.000598</td>
<td>-0.000598</td>
<td>0.023018</td>
</tr>
<tr>
<td>lexdist 4</td>
<td>0.051842</td>
<td>0.108107</td>
<td>0.090298</td>
<td>0.111243</td>
<td>0.111243</td>
<td>0.047508</td>
</tr>
<tr>
<td>lexdist 5</td>
<td>0.043290</td>
<td>0.047801</td>
<td>0.020211</td>
<td>0.028672</td>
<td>0.028672</td>
<td>0.050748</td>
</tr>
<tr>
<td>lexdist 6</td>
<td>0.083848</td>
<td>0.056161</td>
<td>0.103767</td>
<td>0.032869</td>
<td>0.032869</td>
<td>0.050240</td>
</tr>
<tr>
<td>lm 1</td>
<td>0.042750</td>
<td>0.056124</td>
<td>0.052090</td>
<td>0.049561</td>
<td>0.049561</td>
<td>0.059518</td>
</tr>
<tr>
<td>lm 2</td>
<td>0.019881</td>
<td>0.012075</td>
<td>0.022896</td>
<td>0.035769</td>
<td>0.035769</td>
<td>0.026414</td>
</tr>
<tr>
<td>lm 3</td>
<td>0.059497</td>
<td>0.054580</td>
<td>0.044363</td>
<td>0.048321</td>
<td>0.048321</td>
<td>0.056282</td>
</tr>
<tr>
<td>ttable 1</td>
<td>0.052111</td>
<td>0.045096</td>
<td>0.046655</td>
<td>0.054519</td>
<td>0.054519</td>
<td>0.046538</td>
</tr>
<tr>
<td>ttable 1</td>
<td>0.052888</td>
<td>0.036831</td>
<td>0.040820</td>
<td>0.058003</td>
<td>0.058003</td>
<td>0.066308</td>
</tr>
<tr>
<td>ttable 1</td>
<td>0.042151</td>
<td>0.066256</td>
<td>0.043265</td>
<td>0.047271</td>
<td>0.047271</td>
<td>0.052853</td>
</tr>
<tr>
<td>ttable 1</td>
<td>0.034067</td>
<td>0.031048</td>
<td>0.050794</td>
<td>0.037589</td>
<td>0.037589</td>
<td>0.031939</td>
</tr>
<tr>
<td>phrase-pen.</td>
<td>0.059151</td>
<td>0.062019</td>
<td>-0.037950</td>
<td>0.023414</td>
<td>0.023414</td>
<td>-0.069425</td>
</tr>
<tr>
<td>word-pen</td>
<td>-0.200963</td>
<td>-0.249531</td>
<td>-0.247089</td>
<td>-0.228469</td>
<td>-0.228469</td>
<td>-0.252579</td>
</tr>
</tbody>
</table>
Unstable outcomes: scores vary

- Even different scores with different runs (varying 0.40 on dev, 0.89 on test)

<table>
<thead>
<tr>
<th>run</th>
<th>iterations</th>
<th>dev score</th>
<th>test score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>50.16</td>
<td>51.99</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>50.26</td>
<td>51.78</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>50.13</td>
<td>51.59</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>50.10</td>
<td>51.20</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>50.16</td>
<td>51.43</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>50.02</td>
<td>51.66</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>50.25</td>
<td>51.10</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>50.21</td>
<td>51.32</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>50.42</td>
<td>51.79</td>
</tr>
</tbody>
</table>
More features: more components

- We would like to add **more components** to our model
  - multiple language models
  - domain adaptation features
  - various special handling features
  - using linguistic information

→ MERT becomes even **less reliable**
  - runs many more iterations
  - fails more frequently
More features: factored models

- Factored translation models break up phrase mapping into smaller steps
  - multiple translation tables
  - multiple generation tables
  - multiple language models and sequence models on factors

→ Many more features
Millions of features

• Why **mix** of discriminative training and generative models?

• Discriminative training of all components
  – phrase table [Liang et al., 2006]
  – language model [Roark et al, 2004]
  – additional features

• **Large-scale** discriminative training
  – millions of features
  – training of full training set, not just a small development corpus
Perceptron algorithm

- Translate each sentence
- If no match with reference translation: update features

\[
\text{set all } \lambda = 0 \\
\text{do until convergence} \\
\quad \text{for all foreign sentences } f \\
\quad \quad \text{set } e\text{-best to best translation according to model} \\
\quad \quad \text{set } e\text{-ref to reference translation} \\
\quad \quad \text{if } e\text{-best } \neq e\text{-ref} \\
\quad \quad \quad \text{for all features } \text{feature-}i \\
\quad \quad \quad \quad \lambda_{-i} += \text{feature-}i(f,e\text{-ref}) \\
\quad \quad \quad \quad - \text{feature-}i(f,e\text{-best})
\]
Problem: overfitting

- Fundamental problem in machine learning
  - what works best for training data, may not work well in general
  - rare, unrepresentative features may get too much weight
- Especially severe problem in phrase-based models
  - long phrase pairs explain well *individual sentences*
  - ... but are less general, *suspect to noise*
  - EM training of phrase models [Marcu and Wong, 2002] has same problem
Solutions

• **Restrict to short phrases**, e.g., maximum 3 words (current approach)
  – limits the power of phrase-based models
  – ... but not very much [Koehn et al, 2003]

• **Jackknife**
  – collect phrase pairs from one part of corpus
  – optimize their feature weights on another part

• IBM direct model: **only one-to-many** phrases [Ittycheriah and Salim Roukos, 2007]
Problem: reference translation

- Reference translation may be anywhere in this box

- If produceable by model → we can compute feature scores
- If not → we can not
Some solutions

- **Skip sentences**, for which reference can not be produced
  - invalidates large amounts of training data
  - biases model to shorter sentences

- Declare candidate translations closest to reference as **surrogate**
  - closeness measured for instance by smoothed BLEU score
  - may be not a very good translation: odd feature values, training is severely distorted
Experiment

- Skipping sentences with unproduceable reference **hurts**

<table>
<thead>
<tr>
<th>Handling of reference</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>with skipping</td>
<td>25.81</td>
</tr>
<tr>
<td>w/o skipping</td>
<td>29.61</td>
</tr>
</tbody>
</table>

- When including all sentences: surrogate reference picked from 1000-best list using maximum *smoothed BLEU score* with respect to reference translation

- Czech-English task, **only binary features**
  - phrase table features
  - lexicalized reordering features
  - source and target phrase bigram

- See also [Liang et al., 2006] for similar approach
Better solution: early updating?

- At some point the reference translation falls out of the search space
  - for instance, due to unknown words:

  Reference: The group attended the meeting in Najaf ...
  System: The group meeting was attended in UNKNOWN ...

- Early updating [Collins et al., 2005]:
  - stop search, when reference translation is not covered by model
  - only update features involved in partial reference / system output
Conclusions

• Currently have proof-of-concept implementation
• Future work: Overcome various technical challenges
  – reference translation may not be produceable
  – overfitting
  – mix of binary and real-valued features
  – scaling up
• More and more features are unavoidable, let’s deal with them
Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- Experiments
Statistical machine translation today

- Best performing methods based on **phrases**
  - short sequences of words
  - no use of explicit syntactic information
  - no use of morphological information
  - currently best performing method

- Progress in **syntax-based** translation
  - tree transfer models using syntactic annotation
  - still shallow representation of words and non-terminals
  - active research, improving performance
One motivation: morphology

- Models treat *car* and *cars* as completely different words
  - training occurrences of *car* have no effect on learning translation of *cars*
  - if we only see *car*, we do not know how to translate *cars*
  - rich morphology (German, Arabic, Finnish, Czech, ...) → many word forms

- Better approach
  - analyze surface word forms into **lemma** and **morphology**, e.g.: *car +plural*
  - translate lemma and morphology separately
  - generate target surface form
Factored translation models

- **Factored representation** of words

  \[
  \begin{array}{c|c|c|c}
  \text{Input} & \text{Output} \\
  \hline
  \text{word} & & \text{word} \\
  \text{lemma} & & \text{lemma} \\
  \text{part-of-speech} & \rightarrow & \text{part-of-speech} \\
  \text{morphology} & & \text{morphology} \\
  \text{word class} & & \text{word class} \\
  \vdots & \vdots & \vdots \\
  \end{array}
  \]

- **Goals**
  - **Generalization**, e.g. by translating lemmas, not surface forms
  - **Richer model**, e.g. using syntax for reordering, language modeling)
Related work

- **Back off** to representations with richer statistics (lemma, etc.)

- Use of additional annotation in **pre-processing** (POS, syntax trees, etc.)
  [Collins et al., 2005, Crego et al, 2006]

- Use of additional annotation in **re-ranking** (morphological features, POS, syntax trees, etc.)
  [Och et al. 2004, Koehn and Knight, 2005]

→ we pursue an **integrated approach**

- Use of syntactic **tree structure**

→ may be **combined** with our approach
Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- Experiments
Decomposing translation: example

- **Translate** lemma and syntactic information *separately*

  ![Diagram]

  - [ ] lemma → [ ] lemma
  - [ ] part-of-speech
    - morphology → [ ] part-of-speech
    - morphology
Decomposing translation: example

- **Generate surface** form on target side

```
  surface
  ┌──────┐
  │     │
  │↑     │
  │ lemma│
  │      │
  │ part-of-speech │
  │      │
  │    │
  │  │
  │ morphology │
  │      │
```

Translation process: example

Input: \((Autos, Auto, NNS)\)

1. Translation step: lemma \(\Rightarrow\) lemma
\((?, car, ?), (?, auto, ?)\)

2. Generation step: lemma \(\Rightarrow\) part-of-speech
\((?, car, NN), (?, car, NNS), (?, auto, NN), (?, auto, NNS)\)

3. Translation step: part-of-speech \(\Rightarrow\) part-of-speech
\((?, car, NN), (?, car, NNS), (?, auto, NNP), (?, auto, NNS)\)

4. Generation step: lemma,part-of-speech \(\Rightarrow\) surface
\((car, car, NN), (cars, car, NNS), (auto, auto, NN), (autos, auto, NNS)\)
Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- Experiments
Model

- Extension of *phrase model*
- Mapping of foreign words into English words broken up into steps
  - **translation step**: maps foreign factors into English factors
    (on the phrasal level)
  - **generation step**: maps English factors into English factors
    (for each word)
- Each step is modeled by one or more *feature functions*
  - fits nicely into log-linear model
  - weight set by discriminative training method
- Order of mapping steps is chosen to optimize search
Phrase-based training

- Establish word alignment (GIZA++ and symmetrization)
Phrase-based training

- Extract phrase

⇒ natürlich hat john — naturally john has
Factored training

- Annotate training with factors, extract phrase

⇒ ADV V NNP — ADV NNP V
Training of generation steps

- Generation steps map target factors to target factors
  - typically trained on target side of parallel corpus
  - may be trained on additional monolingual data

- Example: *The*/DET *man*/NN *sleeps*/VBZ
  - count collection
    - count(*the*,DET)++
    - count(*man*,NN)++
    - count(*sleeps*,VBZ)++
  - evidence for probability distributions (max. likelihood estimation)
    - \( p(\text{DET}|\text{the}) \), \( p(\text{the}|\text{DET}) \)
    - \( p(\text{NN}|\text{man}) \), \( p(\text{man}|\text{NN}) \)
    - \( p(\text{VBZ}|\text{sleeps}) \), \( p(\text{sleeps}|\text{VBZ}) \)
Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- Experiments
Phrase-based translation

- Task: *translate this sentence* from German into English

\[
\begin{array}{cccccccc}
er & geht & ja & nicht & nach & hause
\end{array}
\]
Translation step 1

• Task: translate this sentence from German into English

er geht ja nicht nach hause

• Pick phrase in input, translate

he
Translation step 2

- Task: translate this sentence from German into English

\[ \text{er geht ja nicht nach hause} \]

\[ \begin{align*}
\text{he} & \quad \text{does not} \\
\text{ja nicht} & \\
\text{er} & \\
\text{geht} &
\end{align*} \]

- Pick phrase in input, translate
  - it is allowed to pick words *out of sequence* (reordering)
  - phrases may have multiple words: *many-to-many* translation
Translation step 3

- Task: translate this sentence from German into English

```
er geht ja nicht nach hause
```

- Pick phrase in input, translate

```
he does not go
```
Translation step 4

- Task: translate this sentence from German into English

er geht ja nicht nach hause

he does not go home

- Pick phrase in input, translate
### Translation options

Here are some translation options for the phrase "er geht ja nicht nach hause":

<table>
<thead>
<tr>
<th>er</th>
<th>geht</th>
<th>ja</th>
<th>nicht</th>
<th>nach</th>
<th>hause</th>
</tr>
</thead>
<tbody>
<tr>
<td>he</td>
<td>is</td>
<td>yes</td>
<td>not</td>
<td>after</td>
<td>house</td>
</tr>
<tr>
<td>it</td>
<td>are</td>
<td>is</td>
<td>do not</td>
<td>to</td>
<td>home</td>
</tr>
<tr>
<td>, it</td>
<td>goes</td>
<td>, of course</td>
<td>does not</td>
<td>according to</td>
<td>under house</td>
</tr>
<tr>
<td>, he</td>
<td>go</td>
<td></td>
<td>is not</td>
<td>in</td>
<td>return home</td>
</tr>
</tbody>
</table>

- **Many translation options** to choose from
- in Europarl phrase table: **2727 matching phrase pairs** for this sentence
- by pruning to the top 20 per phrase, **202 translation options** remain
The machine translation decoder does not know the right answer

→ **Search problem** solved by heuristic beam search
Decoding process: precompute translation options

er  geht  ja  nicht  nach  hause
Decoding process: start with initial hypothesis

er geht ja nicht nach hause
Decoding process: hypothesis expansion

er geht ja nicht nach hause

are
Decoding process: hypothesis expansion

er    geht    ja    nicht    nach    hause

he    are    it
Decoding process: hypothesis expansion

er geht ja nicht nach hause
Decoding process: find best path

er geht ja nicht nach hause
Factored model decoding

- Factored model decoding introduces *additional complexity*
- Hypothesis expansion not any more according to simple translation table, but by *executing a number of mapping steps*, e.g.:
  1. translating of *lemma* → *lemma*
  2. translating of *part-of-speech, morphology* → *part-of-speech, morphology*
  3. generation of *surface form*
- Example: *haus|NN|neutral|plural|nominative* → { *houses|house|NN|plural, homes|home|NN|plural,*
  *buildings|building|NN|plural, shells|shell|NN|plural *}
- Each time, a hypothesis is expanded, these mapping steps have to applied
Efficient factored model decoding

- Key insight: executing of mapping steps can be *pre-computed* and stored as translation options
  - apply mapping steps to all input phrases
  - store results as *translation options*
→ decoding algorithm *unchanged*
Efficient factored model decoding

- Problem: *Explosion* of translation options
  - originally limited to 20 per input phrase
  - even with simple model, now 1000s of mapping expansions possible

- Solution: *Additional pruning* of translation options
  - *keep only the best* expanded translation options
  - current default 50 per input phrase
  - decoding only about 2-3 times slower than with surface model
Factored Translation Models

- Motivation
- Example
- Model and Training
- Decoding
- Experiments
Adding linguistic markup to output

- Generation of POS tags on the target side
- Use of high order language models over POS (7-gram, 9-gram)
- Motivation: syntactic tags should enforce syntactic sentence structure model not strong enough to support major restructuring
Some experiments

• English–German, Europarl, 30 million word, test2006

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>best published result</td>
<td>18.15</td>
</tr>
<tr>
<td>baseline (surface)</td>
<td>18.04</td>
</tr>
<tr>
<td>surface + POS</td>
<td>18.15</td>
</tr>
</tbody>
</table>

• German–English, News Commentary data (WMT 2007), 1 million word

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>18.19</td>
</tr>
<tr>
<td>With POS LM</td>
<td>19.05</td>
</tr>
</tbody>
</table>

• Improvements under sparse data conditions

• Similar results with CCG supertags [Birch et al., 2007]
Sequence models over morphological tags

<table>
<thead>
<tr>
<th>die</th>
<th>hellen</th>
<th>Sterne</th>
<th>erleuchten</th>
<th>das</th>
<th>schwarze</th>
<th>Himmel</th>
</tr>
</thead>
<tbody>
<tr>
<td>fem</td>
<td>fem</td>
<td>fem</td>
<td>-</td>
<td>neutral</td>
<td>neutral</td>
<td>male</td>
</tr>
<tr>
<td>plural</td>
<td>plural</td>
<td>plural</td>
<td>plural</td>
<td>sgl.</td>
<td>sgl.</td>
<td>sgl.</td>
</tr>
<tr>
<td>nom.</td>
<td>nom.</td>
<td>nom.</td>
<td>-</td>
<td>acc.</td>
<td>acc.</td>
<td>acc.</td>
</tr>
</tbody>
</table>

- Violation of noun phrase agreement in gender
  - *das schwarze* and *schwarze Himmel* are perfectly fine bigrams
  - but: *das schwarze Himmel* is not

- If relevant n-grams does not occur in the corpus, a lexical n-gram model would *fail to detect* this mistake

- Morphological sequence model: \( p(N\text{-male}|J\text{-male}) > p(N\text{-male}|J\text{-neutral}) \)
Local agreement (esp. within noun phrases)

- High order language models over POS and morphology
- Motivation
  - \textit{DET-sgl NOUN-sgl} good sequence
  - \textit{DET-sgl NOUN-plural} bad sequence
Agreement within noun phrases

• Experiment: 7-gram POS, morph LM in addition to 3-gram word LM
• Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Agreement errors in NP</th>
<th>devtest</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>15% in NP ≥ 3 words</td>
<td>18.22 BLEU</td>
<td>18.04 BLEU</td>
</tr>
<tr>
<td>factored model</td>
<td>4% in NP ≥ 3 words</td>
<td>18.25 BLEU</td>
<td>18.22 BLEU</td>
</tr>
</tbody>
</table>

• Example
  – baseline:  ... zur zwischenstaatlichen methoden ...
  – factored model:  ... zu zwischenstaatlichen methoden ...

• Example
  – baseline:  ... das zweite wichtige änderung ...
  – factored model:  ... die zweite wichtige änderung ...
Morphological generation model

- Our motivating example
- Translating lemma and morphological information more robust
Initial results

• Results on 1 million word News Commentary corpus (German–English)

<table>
<thead>
<tr>
<th>System</th>
<th>In-doman</th>
<th>Out-of-domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>18.19</td>
<td>15.01</td>
</tr>
<tr>
<td>With POS LM</td>
<td>19.05</td>
<td>15.03</td>
</tr>
<tr>
<td>Morphgen model</td>
<td>14.38</td>
<td>11.65</td>
</tr>
</tbody>
</table>

• What went wrong?
  – why back-off to lemma, when we know how to translate surface forms?
  → loss of information
Solution: alternative decoding paths

- Allow both surface form translation and morphgen model
  - prefer surface model for known words
  - morphgen model acts as back-off
Results

- Model now beats the baseline:

<table>
<thead>
<tr>
<th>System</th>
<th>In-domain</th>
<th>Out-of-domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>18.19</td>
<td>15.01</td>
</tr>
<tr>
<td>With POS LM</td>
<td>19.05</td>
<td>15.03</td>
</tr>
<tr>
<td>Morphgen model</td>
<td>14.38</td>
<td>11.65</td>
</tr>
<tr>
<td>Both model paths</td>
<td>19.47</td>
<td>15.23</td>
</tr>
</tbody>
</table>
Adding annotation to the source

- Source words may lack sufficient information to map phrases
  - English-German: what case for noun phrases?
  - Chinese-English: plural or singular
  - pronoun translation: what do they refer to?

- Idea: add additional information to the source that makes the required information available locally (where it is needed)

- see [Avramidis and Koehn, ACL 2008] for details
Case Information for English–Greek

- Detect in English, if noun phrase is subject/object (using parse tree)
- Map information into case morphology of Greek
- Use case morphology to generate correct word form
Obtaining Case Information

- Use syntactic parse of English input
  (method similar to semantic role labeling)
Results English-Greek

- Automatic BLEU scores

<table>
<thead>
<tr>
<th>System</th>
<th>devtest</th>
<th>test07</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>18.13</td>
<td>18.05</td>
</tr>
<tr>
<td>enriched</td>
<td>18.21</td>
<td>18.20</td>
</tr>
</tbody>
</table>

- Improvement in verb inflection

<table>
<thead>
<tr>
<th>System</th>
<th>Verb count</th>
<th>Errors</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>311</td>
<td>19.0%</td>
<td>7.4%</td>
</tr>
<tr>
<td>enriched</td>
<td>294</td>
<td>5.4%</td>
<td>2.7%</td>
</tr>
</tbody>
</table>

- Improvement in noun phrase inflection

<table>
<thead>
<tr>
<th>System</th>
<th>NPs</th>
<th>Errors</th>
<th>Missing</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>247</td>
<td>8.1%</td>
<td>3.2%</td>
</tr>
<tr>
<td>enriched</td>
<td>239</td>
<td>5.0%</td>
<td>5.0%</td>
</tr>
</tbody>
</table>

- Also successfully applied to English-Czech
Factored Template Models

- **Long range** reordering
  - movement often not limited to local changes
  - German-English: $SBJ \ AUX \ OBJ \ V \rightarrow SBJ \ AUX \ V \ OBJ$

- Template models
  - some factor mappings (POS, syntactic chunks) may have longer scope than others (words)
  - larger mappings form template for shorter mappings
  - computational problems with this

- published in [Hoang and Koehn, EACL 2009]
Shallow syntactic tasks have been formulated as sequence labeling tasks
- base noun phrase chunking
- syntactic role labeling

Results presented in [Cettolo et al., AMTA 2008]