In this lecture....

• More on word alignment from parallel corpora
• Word alignment from comparable corpora
• Cross-linguistic induction of linguistic knowledge
• Feedback on HW1
• HW 2

Other alignment models: linguistic knowledge

• [Niessen 2004]: different morphological forms usually treated as different words
• Model relations between words by explicitly incorporating linguistic features
• Word is enriched with POS and lemma information
• Used in a different lexicon model:

\[ P(u_j | x_j) = \sum P(u_j, m_j | x_j) \]

• \( m_j \) = possible morphological analysis out of all possible analyses for \( t_j \)
• \( P(u_j, m_j | x_j) \) is trained using MaxEnt model

Other alignment models: linguistic knowledge

• Not evaluated in terms of AER or F-score
• Improved translation performance significantly in overall system

<table>
<thead>
<tr>
<th>Number of candidates</th>
<th>BEU</th>
<th>en-NER</th>
<th>GER</th>
<th>DEU</th>
<th>ESU</th>
</tr>
</thead>
<tbody>
<tr>
<td>9,000</td>
<td>52.1</td>
<td>54.8</td>
<td>56</td>
<td>57.3</td>
<td>57.3</td>
</tr>
<tr>
<td>5,000</td>
<td>52.1</td>
<td>54.8</td>
<td>56</td>
<td>57.3</td>
<td>57.3</td>
</tr>
<tr>
<td>0</td>
<td>52.1</td>
<td>54.8</td>
<td>56</td>
<td>57.3</td>
<td>57.3</td>
</tr>
</tbody>
</table>

Other alignment models

• More than on null word in IBM model 1
• Other ways of incorporating linguistic knowledge (eg stemming before training standard models)
• Knowledge-based vs. data-driven morphology models
• Other discriminative training procedures (eg max-margin bipartite graph matching)
• Combination of several alignment techniques
  - Neural networks
  - Boosting
• Etc.
Word alignment benchmark evaluations
• Benchmark evaluations sponsored by ACL at
  – At HLT-NAACL 2003
    (http://www.cse.unt.edu/~rada/wpt/)
  – At ACL 2005 (http://www.statmt.org/wpt05)
• Comparison of alignment systems
  – English-French
  – Romanian-English
  – Inuktitut-English
• Useful resources, evaluation code, etc.

Word alignment from comparable corpora
• Find word translations in non-aligned, non-parallel corpora (lexicon extraction)
• Methods:
  – Cognates
  – Bridge languages
  – Transliteration
  – Aggregate word statistics (context)

Using cognates
• What are cognates?
  – words in different languages sharing meaning and similar surface form, usually historically related
  – E.g. father (En) - Vater (De)
  – Can be used when dealing with two related languages
• Identifying cognates:
  \[
  \forall s \in S, \forall t_1, t_2 \in T
  
  \text{If } \text{cognate}(s, t_1) \rightarrow \neg \text{cognate}(s, t_2)
  
  \text{Then } D(s, t_1) < D(s, t_2)
  \]
• Distance function \(D(s, t)\): edit distance based on characters or phonetic transcription
  – Levenshtein distance: insertions, deletions, and substitutions with cost of 1 each
  – Cost functions can be adjusted based on character/phoneme class (e.g. penalize vowel-consonant substitutions)
  – Cost functions can be learned automatically
    – e.g. Ristad & Yianilos 1998, Satta & Henderson 1997
  – Pick word with lowest distance as translation:
    \[
    t^* = \arg \min\{D(s, t)\}
    \]
• How well does it work?
  – Studied for Romance and Slavic language families by Mann & Yarowsky 2001:
    | Language Pair | % Correct |
    |---------------|-----------|
    | ru-cz         | 47.1      |
    | ru-sr         | 37.8      |
    | ru-pl         | 42.3      |
    | ru-uk         | 37.8      |
    | 23.3          |
    | 27.3          |
    | 29.9          |
    | 32.8          |
    | 35.8          |
  – Performance correlates with language relatedness
  – Some learned correspondences:
    | fr | n | p | b | s | c |
    |---|---|---|---|---|---|
    | pt | m | f | v | c | q |

Using cognates
• Bridge languages
  – What if languages are not closely related?
    – Exploit resources for a third language (bridge language) related to target language
    – Bridge language is in same family as target language but has bilingual resources with source language
    – Can use one or more bridge languages

Using cognates
• Distance function \(D(s, t)\): edit distance based on characters or phonetic transcription
  – Levenshtein distance: insertions, deletions, and substitutions with cost of 1 each
  – Cost functions can be adjusted based on character/phoneme class (e.g. penalize vowel-consonant substitutions)
  – Cost functions can be learned automatically
    – e.g. Ristad & Yianilos 1998, Satta & Henderson 1997
  – Pick word with lowest distance as translation:
    \[
    t^* = \arg \min\{D(s, t)\}
    \]
Bridge languages

- Algorithm:
  for each word $s \in S$
  for each bridge language $B$
  translate $s \rightarrow B$
  $D_b(t, s)$ = compute $D_b(t, s)$

- Experiments using multiple bridge languages:
  - E.g. En $\rightarrow$ Pt, using Romance languages as bridges
  - En $\rightarrow$ No, using Germanic languages as bridges
  - Adding more bridges usually helps
  - If the single closest bridge language is closely related to target,
    difficult to get additional improvements

<table>
<thead>
<tr>
<th>Phrase</th>
<th>SALDAGS</th>
<th>Eval Lex</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\times$</td>
<td>94.7</td>
<td>94.0</td>
</tr>
<tr>
<td>$\times$</td>
<td>92.6</td>
<td>91.7</td>
</tr>
<tr>
<td>$\times$</td>
<td>83.0</td>
<td>81.7</td>
</tr>
<tr>
<td>$\times$</td>
<td>80.0</td>
<td>79.1</td>
</tr>
<tr>
<td>$\times$</td>
<td>75.0</td>
<td>73.6</td>
</tr>
<tr>
<td>$\times$</td>
<td>72.0</td>
<td>71.0</td>
</tr>
<tr>
<td>$\times$</td>
<td>70.1</td>
<td>69.1</td>
</tr>
<tr>
<td>$\times$</td>
<td>65.0</td>
<td>63.1</td>
</tr>
<tr>
<td>$\times$</td>
<td>61.0</td>
<td>58.1</td>
</tr>
<tr>
<td>$\times$</td>
<td>58.0</td>
<td>55.3</td>
</tr>
<tr>
<td>$\times$</td>
<td>50.0</td>
<td>47.0</td>
</tr>
</tbody>
</table>

Transliteration

- Applied to learning of translations of new names (organizations, places,...) in languages with non-ascii alphabets
- E.g. Japanese:
  - $\text{Angela Johnson}$ vs $\text{Angela Johanna}$
  - $\text{New York Times}$ vs $\text{The New York Times}$
  - $\text{NY Times}$ vs $\text{The New York Times}$

- Usually many possible spellings on foreign language side, only one possible spelling on English side

Transliteration

- Ambiguity problem
  - Arabic $\rightarrow$ English
    - Svartsynyr $\rightarrow$ Schwarzenegger
    - Svtrenyynr $\rightarrow$ Arnold Schwarzenegger
  - Often more complicated than 1-to-1 character mapping
  - Need language-independent, trainable approach

Transliteration

- Noisy channel model:
  $P(e)$ vs $P(f\,|\,e)$
  $P(p\,|\,e)$ vs $P(p\,|\,f)$
  $P(p) = \sum_{e,f} P(e)P(f\,|\,e)P(p\,|\,e)P(p\,|\,f)$

- Each model can be implemented as a weighted finite-state transducer
- Can be applied in reverse
- Individual models can be composed into one transducer
- Probability estimation:
  - sound-letter models can be estimated from pronunciation dictionary
  - Sound-to-sound model requires preprocessed parallel data, EM training to find phone alignments
Transliteration

- Evaluation on Japanese-English transliteration [Knight & Graehl 1998]
- Out of 100, 64% correct, 12% phonetically identical to target string but misspelled, 24% wrong
- Human performance: 27%/7%/66% (but had no knowledge of Japanese phonology)
- Frequent mistakes: segmentation of English character string into words, e.g.
  - nancy care again ⇒ Nancy Kerrigan

Word Context

- “you shall know a word by the company it keeps”
- If two words co-occur with high frequency in language 1, their translations should occur with high frequency in language 2
- Collocations
- Word-cooccurrence statistics
- Trivially holds for parallel corpora, what about non-parallel corpora?
  - Empirical study by Rapp [1995]: assumption holds for unrelated English and German texts

Word Context

- Study of 100 German-English word pairs in 40-60M word corpora

Word Context

- Pick a window of length n around current word (e.g. n = 3)
- Count how often each pair of words occurs within this window
  - Removal of function words
  - Lemmatization
  - Position-dependent vs. position-independent counts:
    - compute one count vector by adding all counts regardless of position
    - Compute multiple vectors depending on distance to center word and concatenate
- From raw co-occurrence counts, compute measure of “word association strength”:
  - mutual information,
  - conditional probabilities, etc.

Word Context

- From vectors of word association strength values, compute similarity between words in L1 and L2
  - For each word in L1 matrix, compare association vector to all vectors in L2 matrix
  - Problem: word order! - works only when related words are in same order
  - Solutions:
    - Search over all possible permutations
    - Constrain space according to initial seed lexicon
  - Select word with highest context vector similarity as translation

Word Context

- Effect of non-corresponding word order
- Search over possible permutations was suggested by [Rapp 1995] but not evaluated
Word Context

- Search over all possible permutations is prohibitive
- Narrow search space by assuming that small initial seed lexicon is available [Rapp 2000, Fung & Yee 1998]
- Compute similarity of vectors over subset of matrix only

Measures of similarity:
- Cosine similarity:
  \[ S(X, Y) = \frac{X \cdot Y}{|X||Y|} \]
- Jaccard coefficient:
  \[ S(X, Y) = \frac{\sum x_i \cap y_i}{\sum x_i + \sum y_i - \sum x_i \cap y_i} \]
- Dice coefficient:
  \[ S(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \]
- City-block metric (Hamming distance)
  \[ S(X, Y) = \sum |x_i - y_i| \]

Alternative to simple similarity measures:
- IR model: view context of L1, C(L1), as query, and context of L1, C(L2), as document
- [Shao & Ng 2004]: language modeling approach to IR: use a LM to compute \( P(Q|D) \)
  \[ P(Q|D) = \prod_{t} P(t|D) \cdot \prod_{t} P(t|Q) \]
  Normalization term, can be omitted
- In this context:
  \[ P(C(L1)|C(L2)) = \prod_{t \in L1} P(t|C(L2))^{\text{count}(t)} \]
  Translation of context in L2 according to bilingual dictionary

Evaluation measures:
- % correct of 1-best translations
- % correct of N-best translations
- Mean rank of correct translation in N-best list of returned translations
- Recall?

Some example results:
- 72% 1-best accuracy, German-English [Rapp 2000]
- 60% to 90% 10-best accuracy, Chinese-English [Shan & Ng 2004]

Use of alignments
- Use automatically extracted parallel sentences/words to update translation model
- Use alignments to automatically train annotation tools for new languages
- Use alignments to improve tools for known languages
Updating translation model

- [Munteanu & Marcu 2004] study: detection of parallel sentences in comparable corpora using maxEnt classifier
- Use extracted data to increase training set for existing machine translation system

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Correct</th>
<th>Incorrect</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>the DT import</td>
<td>0.75</td>
<td>0.25</td>
<td>0.05</td>
</tr>
<tr>
<td>oil NP</td>
<td>0.80</td>
<td>0.20</td>
<td>0.02</td>
</tr>
<tr>
<td>crude JJ</td>
<td>0.82</td>
<td>0.18</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Cross-linguistic knowledge induction

- Given a (partially) aligned bitext and existing annotations for one language, how to infer annotations for the other language?
- Project information via word alignments
- Use noisy projections to train annotation tools
- What type of annotations are amenable to projection?
  - Shallow syntactic information? POS, bracketing
  - Deep syntactic information? (parse trees)
  - Semantic information? e.g. predicate-argument structure

Induction of POS tagger

- Inducing a POS tagger:
  - Align a bitext using standard models (eg IBM 4)
  - Annotate input language with POS
  - Select a set of POS tags to project
    - problem: languages have different POS tags
      - different grammatical distinctions
      - grammatical features are realized differently (synthetically or by morphol. inflections)
  - Project tags
    - Distill noise-robust tagger from noisy projection data

Induction of POS tagger

- Noisyness of initial projection (eg English-French from [Yarowsky et al. 2000]):
  - Accuracy on core tags (N,V,A,Adv): 76%
  - With manual correction of alignments: 85%
  - Weak generalization: 86% accuracy on training data

Induction of POS tagger

- Handling 1-to-many alignments:
  - Interpolation of tag probabilities estimated from 1-1 alignments and 1-to-N alignments:
    \[ P(w | t) = \lambda P_{1-1}(w | t) + (1 - \lambda) P_{1-N}(w | t) \]
  - Prevents e.g. function words from absorbing too much probability mass
Induction of POS tagger

- Improving tag sequence model:
  - Filter noisy alignments according to confidence (proportional to word alignment score)

- Evaluation on English-French POS induction:
  - Training data: 2M words (Canadian Hansard)
  - Bigram POS tagger
  - Accuracy on aligned data: 96% (core tags)
  - Accuracy on unseen monolingual data: 98%

Induction of NP bracketer

- Noun phrase bracketing:
  - NPs tend to be translated as coherent units even in free word-order languages
  - Project NP bracketing and train learning algorithm (TBL)
  - Simple cases: assign indices to each NP in input language; NP in other language = maximum span of projected index
  - Difficult cases: interwoven NPs (often alignment errors)

Induction of NP bracketer

- Improvement of noisy projections:
  - Focus on highest-quality alignments
    - Exclude N% of alignments with lowest alignment score
    - Exclude sentences with many alignment crossings

- Evaluation on set of 40 hand-bracketed sentences (English-French, English-Chinese):