CS585: Homework #3

December 10, 2009

Due to my lateness with this assignment, you may mail your results to me up till December 22. To get feedback before the final, however, you can mail me any non-programming question by Tuesday, December 15 at 10am, and I will grade it for you by the review session at 4pm. (Programming questions are marked by an asterisk.)

1 Formal Semantics

For background on formal semantics, see the lecture slides and chapter 10 of the NLTK book.

1. Translate the following sentences into quantified formulas of first-order logic:
   (a) Angus likes someone and someone likes Julia.
   (b) Nobody smiles at Pat.
   (c) Nobody coughed or sneezed.
2. Translate the following *verb phrases* using λ-abstracts and quantified formulas of first-order logic:
   (a) feed Cyril and give a cappucino to Angus
   (b) be loved by everyone
   (c) be loved by everyone and detested by no-one
3. Let $g = \mathit{chase}$ and $h = \lambda x.\forall y.(\mathit{dog}(y) \Rightarrow \mathit{chase}(x, y))$. If $h = f(g)$, write down a λ-abstract for f.
4. Let $g = \mathit{give}$ and $h = \lambda z.\lambda x.\exists y.(\mathit{present}(y) \wedge \mathit{give}(x, y, z))$. If $h = f(g)$, write down a λ-abstract for f.

2 Parsing and Translation

Dekai Wu's inversion transduction grammar is a special case of synchronous context-free grammar. A special case of ITG is a bracketing transduction grammar with only one nonterminal. See the slides for more on ITG and bilingual parsing. If you're interested, you could also look at Dekai Wu's original paper from *Computational Linguistics* back in 1997: `http://aclweb.org/anthology-new/J/J97/J97-3002.pdf`.

For this section, we will be aligning English and German sentences with the following BTG:

$$\begin{aligned}
A &\xrightarrow{e^{-1}} [A\ A] \\
A &\xrightarrow{e^{-2}} \langle A\ A \rangle \\
A &\xrightarrow{e^{-20}} <\mathit{English\ word}> /\epsilon \\
A &\xrightarrow{e^{-21}} \epsilon / <\mathit{German\ word}> \\
&\ldots
\end{aligned}$$

The rest of the grammar consists of word-to-word translation probabilities. For the present purposes, note that we assign all English-to-ϵ (empty word) alignments the uniform, small probability of e^{-20}; ϵ-to-German alignments have probability e^{-21}. Recall that the rule $A \rightarrow \langle A\ A \rangle$ reverse the order of its children in German compared to English.

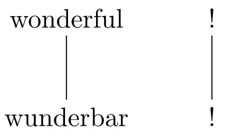
1. Suppose that the grammar contains the following translation rules:

$$\begin{aligned}
A &\xrightarrow{e^{-7.82}} \text{!/!} \\
A &\xrightarrow{e^{-15.4}} \text{wonderful/wunderbar}
\end{aligned}$$

   Given the following English-German sentence pair:

```
wonderful !
wunderbar !
```

   write down an expression for the probability of the following word alignment:



2. Given the grammar and sentence pair above, write down an expression for the probability assigned to *all possible* word alignments of that sentence pair. Are there some derivations that result in identical word alignments? Which alignments?

3. *The next few questions use the following data:

   - `http://www.cs.umass.edu/~dasmith/inlp2009/test.en`: 100 short English sentences, of no more than 10 words each, from the European Parliament corpus, with all words lowercased;
   - `http://www.cs.umass.edu/~dasmith/inlp2009/test.de`: German translations of those 100 sentences; and
   - `http://www.cs.umass.edu/~dasmith/inlp2009/itg.dict`: a probabilistic translation dictionary with 1874 entries. This dictionary was estimated by aligning parallel text from the Europarl corpus (not including the sentences above). Entries that contained words not in the 100 sentence pairs above were removed, and not all word pairs are included. Each line in this file contains an English word, a German word, and a log probability, separated by tab characters.

   Write a parsing program that outputs the best word alignment for each sentence pair, given the grammar and dictionary. As with monolingual CKY, all rules are either binary, or unary terminal productions. In monolingual CKY, you need to keep track of the beginning and ending points of your constituents. Here, you need to keep track of constituent spans in two languages simultaneously. Please email your source code and any instructions for compiling and running it.

4. *In the alignments produced by your program, what percentage of the German words were aligned to ϵ? of the English words?

5. *What percentage of the time was the "swap" rule $A \rightarrow \langle A\ A \rangle$ used, compared to the "in-order" rule?

6. *Disable the swap rule in your program. Now, alignments will be "monotonic" in MT jargon. What percentage of German and English words are aligned to ϵ?