Taaltheorie en Taalverwerking 2017

Homework 5

Note: Some of the exercises here and in the NLTK set require you to consult WordNet online at `http://wordnetweb.princeton.edu/perl/webwn`. When you search for a word in WordNet, you will see a list of senses. Each sense is a "synset" in WordNet terminology: a set of synonyms and possibly a definition in brackets (called a gloss). Each synset starts with S:. If you click on this S:, a list of new links appears. To explore the inherited hypernym chair of a synset click on *inherited hypernym*. This will show you all the hypernyms up to the root node of WordNet (*entity*).

1. Give one first-order logic formula for each of the following sentences using event representations. As you can see, the sentences include two verbs (*make* and *see*), each of which introduces an event. Make sure your formulas capture the temporal relations between the two events present in each sentence. Use appropriate thematic roles for the arguments of each verb, taken from the set of roles shown in Figure 19.5 and 19.6 of *Speech & Language Processing*. [6 points]

   (a) After Bill made a coffee, Anne saw a shooting-star.

   (b) When Bill made a coffee, Anne had seen a shooting-star.

2. Exercise 19.8 from *Speech & Language Processing* (simplified): Using WordNet, describe appropriate selectional restrictions on the verbs *drink* and *write*. [4 points]

   Hint: Each of these verbs has two arguments. You first need to indicate the semantic role of each argument (from the list in Figure 19.5 and 19.6 of *Speech & Language Processing*) and then find a synset in WordNet that captures the selectional restrictions the verb imposes on each argument. To find the right synset, you can search in WordNet for words that are appropriate arguments for each of these verbs and explore the inherited hypernym chain of the right sense until you find the appropriate level of generalisation.

3. Exercises 19.4 and 19.6 from *Speech & Language Processing* (slightly modified): Consult the WordNet entry for the word *scrap*. As you will see, WordNet lists 4 noun senses and 3 verb senses for this word. [2 points]

   (a) Classify the four noun senses into homonymous and polysemous senses given the definitions of homonymy and polysemy that we have seen in class and that can be found in chapter 19 of *Speech & Language Processing*.

   (b) Examine the inherited hypernym chain of each noun sense. If the hypernym chain has more than one path, consider only the first one. Are the chains consistent with your classification of these senses in terms of homonymy and polysemy? Explain why.

4. Exercise 20.4 from *Speech & Language Processing* (slightly modified): Simulate the original (not simplified!) Lesk word-overlap disambiguation algorithm on the phrase *Time flies like an arrow* (see p. 681). Use WordNet's senses and glosses for your contexts and signatures (but ignore WordNet's example sentences). Disambiguate the words one at a time, from left to right, and after you have chosen a sense for a word, use only the gloss from your chosen sense when disambiguating later words. [7 points]

While disambiguating and computing overlap, ignore the frequently occurring English *stop words*:

> *a, able, about, across, after, all, almost, also, am, among, an, and, any, are, as, at, be, because, been, but, by, can, cannot, could, dear, did, do, does, either, else, ever, every, for, from, get, got, had, has, have, he, her, hers, him, his, how, however, i, if, in, into, is, it, its, just, least, let, like, likely, may, me, might, most, must, my, neither, no, nor, not, of, off, often, on, only, or, other, our, own, rather, said, say, says, she, should, since, so, some, than, that, the, their, them, then, there, these, they, this, tis, to, too, twas, us, wants, was, we, were, what, when, where, which, while, who, whom, why, will, with, would, yet, you, your*

You may assume that there has already been a *syntactic* parse, and so you only have to consider WordNet senses for the correct part of speech.

Show each step of the algorithm, including the results of all assignment statements. Does the algorithm return the correct senses for each word of the sentence?