School of Computing and Information Systems
The University of Melbourne
COMP90042
WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2017)

Sample solutions for discussion exercises: Week 5

Discussion

1. Using typical dependency types, construct (by hand) a dependency parse for the following sentence: *Yesterday, I shot an elephant in my pyjamas.* Check your work against the output of the online GUI for the Stanford Parser (`http://nlp.stanford.edu:8080/parser/index.jsp`).

   - Dependency parses lend themselves to a flat representation, from which you can derive the tree if you wish:

```
ID Token Head Relation
1 Yesterday 4 TMP
2 , 1 PUNCT
3 I 4 NSUBJ
4 shot 0 ROOT
5 an 6 DET
6 elephant 4 DOBJ
7 in 9 CASE
8 my 9 POSS
9 pyjamas 4 NMOD
10 . 4 PUNCT
```

2. In what ways is (transition–based, probabilistic) dependency parsing similar to (probabilistic) Earley parsing? In what ways is it different?

   - The connections are a little tenuous, but let's see what we can come up with:
     - Both methods are attempting to determine the structure of a sentence; both methods are attempting to disambiguate amongst the (perhaps many) possible structures licensed by the grammar by using a probabilistic grammar to determine the most probable structure.
     - Both methods process the tokens in the sentence one–by–one, left–to–right.
     - Both methods posit edges to the "left" and "right" (for tokens earlier in the sentence, and for tokens later in the sentence) — as opposed to CYK, which only looks "left".
   - There are numerous differences (probably too many to enumerate here), for example:
     - Although POS tags are implicitly used in constructing the "oracle" (training), the depedency parser doesn't explicitly tag the sentence.
     - The transition–based dependency parser can potentially take into account other (non–local) relations in the sentence, whereas Earley's probabilities depend only on the (local) sub-tree.

– Earley adds numerous fragments to the chart, which don't end up getting used in the final parse structure, whereas the transition–based dependency parser only adds edges that will be in the final structure.

3. Give illustrative examples that show the difference between:

   (a) **Synonyms** and **hypernyms**
   - Two words are synonyms when they share (mostly) the same meaning, for example: *snake* and *serpent* are synonyms.
   - One word is a hypernym of a second word when it is a more general instance ("higher up" in the hierarchy) of the latter, for example, *reptile* is the hypernym of *snake* (in its animal sense).

   (b) **Hyponyms** and **meronyms**
   - One word is a hyponym of a second word when it is a more specific instance ("lower down" in the hierarchy) of the latter, for example, *snake* is one hyponym of *reptile*. (The opposite of hypernymy.)
   - One word is a meronym of a second word when it is a part of the whole defined by the latter, for example, *scales* (the skin structure) is a meronym of *reptile*.

4. One possible step of text normalisation (tokenisation) is conflating synonyms as a single representation. Give a couple of reasons why this doesn't usually happen.
   - The most compelling reason is that identifying synonyms is **difficult** to do accurately.
   - To identify synonyms in general, we need to know the **sense** of a given token, which means solving the **word sense disambiguation** problem (see below), because typically synonyms operate on the sense level, rather than the token level.
   - We usually need context to identify the sense with any degree of accuracy — but we need to tokenise the text to identify the context, which means that tokenisation typically needs to be performed (well) before sense disambiguation.

5. Using some Wordnet visualisation tool, for example, `http://wordnetweb.princeton.edu/perl/webwn` and the Wu & Palmer definition of **word similarity**, check whether the word *information* more similar to the word *retrieval* or the word *science* (choose the sense which minimises the distance). Does this mesh with your intuition?
   - The word *information* has five different senses in Wordnet; I've reproduce the fragment of the hierarchy above these senses below:
   - Here's the corresponding fragment of the three senses above `retrieval`:
   - To find the Wu & Palmer similarity, we need to find the **lowest common subsumer** — the lowest node in the hierarchy shared by the two senses, and then apply the following formula:

$$\text{sim}(c_1, c_2) = \frac{2 \times \text{depth}(\text{LCS}(c_1, c_2))}{\text{depth}(c_1) + \text{depth}(c_2)}$$

		entity abstraction... communication message...		
entity abstraction... communication message...	entity abstraction... psychological... cognition...	statement pleading charge... accusation...	entity abstraction... group... collection...	entity abstraction... measure system of meas... information meas...

information

entity physical... process... processing data process... operation computer op...	entity abstraction... psychological... cognition... process... basic cog... memory...	entity abstraction... psychological... event act...

retrieval

- The question asks us to choose the senses which minimise the distance, so we need to check them all.
- The "message" sense of *information* lies at depth 5; the "data retrieval" sense of *retrieval* is at depth 8; the lowest node in the hierarchy that they share is `entity` (the root node) so the similarity is:

$$\begin{aligned} \text{sim}(\text{information}, \text{retrieval}) &= \frac{2 \times 1}{5 + 8} \\ &= \frac{2}{13} \approx 0.154 \end{aligned}$$

- What about the "message" sense of *information* with the other senses of *retrieval*?
  - The "memory" sense of *retrieval* is at depth 8, and the lowest node shared is `abstraction, abstract entity` (at depth 2); this means that the similarity is $\frac{2\times2}{8+5} \approx 0.308$.
  - The "event" sense of *retrieval* is at depth 5, and the lowest node shared is also `abstraction, abstract entity`, so the similarity is $\frac{2\times2}{5+5} = 0.4$.
- Let me summarise these results in a table, where I've numbered the senses according to the Wordnet ordering (left-to-right above):

		information				
		1	2	3	4	5
retrieval	1	0.154	0.154	0.118	0.154	0.143
	2	0.308	**0.615**	0.235	0.308	0.286
	3	0.4	0.6	0.286	0.4	0.364

- The maximum similarity (in bold in the table above) is 0.615, for the second sense of *information* — "knowledge acquired through study or experience or instruction" — and the second sense of *retrieval* — "the cognitive operation of accessing information in memory" (because they are both cognitive processes).

- I will leave *science* as an exercise (there are only two senses this time), but the maximum similarity is 0.727 for the "knowledge acquired..." sense of *information*, and the "ability to produce solutions in some problem domain" sense of *science*.
- *science* is clearly the more similar word. This does match with my personal expectations, however, this probably isn't the sense of *science* I had in mind!

6. What is **word sense disambiguation**?

   - Word sense disambiguation is the computational problem of automatically determining which sense (usually, Wordnet synset) of a word is intended for a given token instance with a document.

   (a) The **Yarowsky** method from the lectures uses two heuristics — what do they mean and why are they significant? Can you find counter–examples?

   - **One sense per collocation**: a given sense of a word co-occurs with (at the document level, or the local context level) other words that are related to its meaning.
     - The lecture example is that the "manufacturing" sense of *plant* occurs with other words related to manufacturing, like "factory" or "industrial" whereas the "vegetation" sense of *plant* occurs with words like "life" or "soil".
     - This notion is fairly strongly supported in the literature; in fact, it is closely related to the notion of **distributional semantics** (more next week).
     - One possible problem is that Wordnet synsets are too-fine grained, so that the "different" senses actually co-occur with the same words (because they aren't very different in practice!).
     - Another problem is that some common words can co-occur with multiple senses; for example, "ground" or "growth" in the *plant* senses above — because of polysemy in the collocated words, or because of word play (newspaper headlines are well-known for this!), or because of **semantic convergence** between the unrelated senses.
   - **One sense per discourse**: all instances of a word in a discourse (conventionally a conversation, but more typically construed as a document) use the same sense.
     - At first glance, this seems reasonable, as a document is usually about a small number of **topics** (more next week), and chances are that the senses of both topics don't occur in the same document. For the example above, a document about gardening will (presumably) not refer to a manufacturing *plant*.
     - Since documents don't typically have a single topic associated with them, in a large collection, you'll probably see some (hopefully small number) of documents with both senses attested. Hopefully the algorithm will notice that it can't make a clear decision — this is another motivation for the **bootstrap** method, where we start by focussing on the instances where we're most confident.

  - This also has the same problem about the Wordnet synsets being more fine-grained that the way people normally use them — so that a given document might contain multiple senses of a given word because the writer doesn't perceive them as being different.
- Generally speaking, making these two assumptions allows us to perform word sense disambiguation on some unknown token by comparing the context of many instances (within the same document) with instances of the same word in other documents, to hopefully find some comparable instance where we are confident of its sense.
- It's a nice idea, and it works ... okay ... in practice (word sense disambiguation is **hard**!), but natural language (especially in casual settings) is frustratingly ambiguous. Let me leave you with a quote that I overhead the other day: "I'm banking on finding an ATM nearby..."