FNLP Tutorial 1

1 Ambiguities

Ambiguities are pervasive in natural language, but often go unnoticed when we use language because humans are so good at resolving them. In this exercise, we want you to find ambiguities in the example sentences and attempt to articulate a paraphrase that as much as possible removes the ambiguity (similar to section 1.2 of J&M, 2nd edition). Categorise the different ambiguities you observe: e.g., word sense ambiguity, structural ambiguity, phonetic ambiguity and so on.

1. At the bank, Mary noticed her sister.

   Solution

   (a) Mary noticed her own sister at the financial institute
   (b) Mary noticed her own sister at the river bank
   (c) Mary noticed someone's sister at the financial institute
   (d) Mary noticed someone's sister at the river bank

   Note that there are two ambiguities: what does "bank" mean? (this is a word sense ambiguity) and who does "her" refer to? (reference ambiguity), and we can combine them freely, so there are $2 \cdot 2 = 4$ possible readings in total.

2. Every student wants to win the first prize in a programming competition with a robot.

   Solution There are at least the following ambiguities:

   (a) A "programming competition" could be a competition that involves programming (plausible) or a competition 'that programs' (implausible). This is a semantic ambiguity.
   (b) "with a robot" could describe the competition, i.e. a competition involving a robot, or it could describe the winning, i.e. winning the competition by using a robot or with a robot as a teammate. This is a structural ambiguity.
   (c) It could be the case that speaker was trying to say that there is (at least) one competition in which every student wants to win the first prize or there could be multiple competitions and every student wants to win first prize in (at least) one of them. This is a semantic ambiguity, more specifically, a scope ambiguity (because the scope of the quantifiers are ambiguous).
   (d) It could be the case that speaker was trying to say that the students want to win the first prize because it has the property of being the first prize (they want to come first), no matter what the prize is or the speaker meant that they are keen on the first prize because it is something specific (e.g. a laptop) that they all would like to win. This is a semantic ambiguity, more specifically it's a de re/de dicto ambiguity[1].

   Note that the ambiguity described in (d) does *not* arise from the problem that we cannot infer the students' intentions in the competition but rather that we cannot unambiguously infer what the *speaker* of (2) meant.

[1] You can find more examples at https://en.wikipedia.org/wiki/De_dicto_and_de_re

2 Corpora and annotation

In this exercise, we want you to get some insights into the challenges that humans and machines face when it comes to annotation. Consider the following corpus:

1. Paris Hilton stayed at the Hilton in Paris.
2. Donald Fucking Trump.
3. James Clerk Maxwell was educated at Edinburgh and Cambridge.
4. Tom works for the Dumfries & Galloway Standard.

1. Annotate the above utterances with named entities. For our purposes, a named entity is a single word or multiple words that refer to a person (`PER`), location (`LOC`) or organisation (`ORG`). Are there cases that you found difficult? Which cases do you think are difficult for an automated system? And why?

   Solution There is often no single optimal solution for annotation; whether something is a good annotation depends on what it is used for. For example, should ”the Hilton” be a named entity or just “Hilton”? Can named entities overlap?

   The following should be a reasonable annotation though for many applications:

   (a) [Paris Hilton]$_{\text{PER}}$ stayed at the [Hilton]$_{\text{ORG}}$ in [Paris]$_{\text{LOC}}$.

   (b) [Donald]$_{\text{PER}}$ Fucking [Trump]$_{\text{PER}}$.

   (c) [James Clerk Maxwell]$_{\text{PER}}$ was educated at [Edinburgh]$_{\text{ORG}}$ and [Cambridge]$_{\text{ORG}}$.

   (d) [Tom]$_{\text{PER}}$ works for the [[[Dumfries]$_{\text{LOC}}$ & [Galloway]$_{\text{LOC}}$]$_{\text{LOC}}$ Standard]$_{\text{ORG}}$.

   1b can be considered a *single* discontinuous named entity, rather than two named entities. In 1d, there is an ambiguity if *Dumfries & Galloway* is a location or an organisation (the council) that is difficult to resolve.

   An automated system might struggle with cases that require context: the token “Paris” can refer to a location or a person, “Hilton” can refer to a person or to an organisation (the hotel chain), Edinburgh and Cambridge are locations but the text refers to universities in those cities, which are organisations. A named entity recogniser that does not use a list of council areas of Scotland might also struggle with detecting that the span *Dumfries & Galloway* is one named entity.

2. Annotations in NLP must be mathematical objects in order to be machine-readable. How would you formalise the annotation of a named entity (e.g. using tuples, sets, lists, strings and natural numbers)? Provide formalised examples of two of the sentences. Are there any examples where you cannot unambiguously represent your annotations?

   Solution

   Again, there is no single best way to formalise the annotation but your formalisation should be able to represent your annotations well. In the sample solution, we said that we consider *Donald Fucking Trump* to be a discontinuous named entity and we have to accommodate this now in our formal notation.

   We formalise the annotation of a single sentence as a set A. Each element $a \in A$ represents a named entity and is itself a tuple $a = \langle S, \ell \rangle$ where $\ell \in \{\texttt{PER}, \texttt{LOC}, \texttt{ORG}\}$ is its label and $S \subseteq \mathbb{N} \times \mathbb{N}$ is a set of spans in the input, indicating where the named entity (or its fragments, in case there are multiple) can be found in the tokenised text.

   Examples:

- $[\text{Donald}]_{\text{PER}}$ Fucking $[\text{Trump}]_{\text{PER}}$:

$$\{\langle\{\langle 0,1\rangle,\langle 2,3\rangle\},\mathtt{PER}\rangle\}$$

- $[\text{Tom}]_{\text{PER}}$ works for the $[[[\text{Dumfries}]_{\text{LOC}}$ & $[\text{Galloway}]_{\text{LOC}}]_{\text{LOC}}$ $\text{Standard}]_{\text{ORG}}$:

$$\{\langle\{0,1\},\mathtt{PER}\rangle,\langle\{\langle 4,5\rangle\},\mathtt{LOC}\rangle,\langle\{\langle 6,7\rangle\},\mathtt{LOC}\rangle,\langle\{\langle 4,7\rangle\},\mathtt{LOC}\rangle,\langle\{\langle 4,8\rangle\},\mathtt{ORG}\rangle\}$$

As you can see, we pay a price in terms of complexity for accommodating the discontinuous named entity. Basing the annotation on an already tokenised text is convenient but if later on a different tokenisation is desired, this will pose challenges to map the annotations to the new tokenisation.

A different and commonly used way to represent named entities is using BIO-tagging (J&M section 8.3, 3rd edition). BIO-tagging uses beginning tokens (indicated with a `B`), tokens inside of a span (indicated with a `I`) and tokens for anything that is not part of a named entity (indicated with a `O`). In our case, we would formalise the annotation of a single sentence as a list B. The i-th element of B (b_i) represents the tag assigned to the i-th word in the sentence, with $b_i \in \{\mathtt{B\text{-}PER},\mathtt{B\text{-}LOC},\mathtt{B\text{-}ORG},\mathtt{I\text{-}PER},\mathtt{I\text{-}LOC},\mathtt{I\text{-}ORG},\mathtt{O}\}$. When using BIO-tagging, we run into issues with our little corpus due to discontinuous and overlapping named entities in the two sentences listed above:

- To capture “Donald Trump” as entity, we are forced to choose between including “Fucking”, or creating two separate entities: `[B-PER, I-PER, I-PER]` vs. `[B-PER, O, B-PER]`.
- We cannot both capture “Dumfries & Galloway Standard” as an organisation while capturing the remaining named entities as locations: `[B-PER, O, O, O, B-ORG, I-ORG, I-ORG, I-ORG]`.

3 Preprocessing Twitter data

Read through the ten messages from the social media platform Twitter, below.[2]

Mini Twitter corpus

1. @fakeusername cool!I always lookd for one but only found Haighs and Koko Black -not that there’s anything wrong with that ;)
2. sorry might not b back 4 a while i have alot coming up ,pantomine,option choices,holiday,footie play offs AND all my ruin your halfterm work
3. Let me sneak out to kitchen. I’m hella hungry. Brb!
4. @fakeusername yo im bored imma bout to call ya ass on skype
5. http://fakeurl.com - BITCHES DONT KNOW BOUT MY MILLENNIUM FALCON #fakehashtag
6. RT @fakeusername: RT @fakeusername: RT @fakeusername: Pleaseeeee, i want to be taller :(http://fakeurl.com
7. OH SNAP. my 6 year old cousin snores SO loud.
8. the an-noy-ing cramps is getting my way! I wanna kill eeuuu!
9. Daughter’s been *kil-ling* it on video chat with Grandma. I’d put her on that ChatRoulette thing if that weren’t, like, awful parenting.
10. Thr R times, tht I test yr faith,’til U think U might surrender. Baby Im, Im not ashamed 2 say, tht my hopes wr (cont) http://fakeurl.com

1. Imagine you want to POS-tag these sentences with a computational model trained using white-space tokenised articles from news papers. Rewrite the ten tweets in the format you think is best given this application.

   Solution This question is open-ended and can have many adequate answers. Below, an example is provided. A couple of things to keep in mind are:

   - The documents will feed into a model trained on white-space tokenised text. Match that tokenisation.
   - News paper articles will have conventional casing, so start a sentence with a capital letter and decapitalise words like MILLENIUM FALCON.
   - The hashtags, mentions and URLs are Twitter-specific and will throw the computational model off, which is why it is good to exclude them.
   - The computational model would certainly benefit from spelling correction, but that is a rather expensive computational preprocessing step. It is up to the student whether or not to do that here.

   (a) Cool ! I always looked for one but only found Haighs and Koko Black - not that there ’s anything wrong with that .

   (b) Sorry might not be back for a while I have a lot coming up , pantomine , option choices , holiday , footie play offs and all my ruin your halfterm work .

[2] Tweets are adapted from the SentiStrength corpus, that is free for academic use.

(c) Let me sneak out to kitchen . I ’m hella hungry. Brb !

(d) Yo I ’m bored I ’m about to call your ass on Skype

(e) Bitches don ’t know about my millenium falcon .

(f) Please , I want to be taller .

(g) Oh snap . My 6 year old cousin snores so loud .

(h) The annoying cramps is getting my way ! I wanna kill you !

(i) Daughter ’s been killing it on video chat with grandma . I ’d put her on that ChatRoulette thing if that weren ’t , like , awful parenting .

(j) The R times , that I test your faith , ’til you think you might surrender. Baby I ’m , I ’m not ashamed to say , that my hopes were (cont) .

2. In practice, preprocessing need not only be effective, but also cheap in terms of resources that are involved. Provide five rules for string manipulation that, in your opinion, will have the greatest impact. You can simply phrase them in text, or use regular expressions if you are familiar with them (see J&M, chapter 2, 3rd edition).

Solution Below are some suggestions for how to handle the punctuation and white space in the tweets.

(a) Remove URLs, mentions, hashtags, that could be found using the following expression: replace `(http\S*)|(@\S*)|(#\S*)` with the empty string.

(b) For this specific text, some punctuation marks can be removed (e.g. ‘*’ and ‘-’ in “*killing*”) by replacing `[-;):(*]` with the empty string.

(c) Other punctuation marks should be surrounded by white space, by replacing `(?=[!?.,’])` and `(?<=[!?.,’])` with white space.

(d) To transform the capitalised words, we could lowercase all letters, but that harms the representation of the beginnings of sentences and named entities like ‘Haighs’ and ‘Koko Black’. A better heuristic would be to lowercase all letters that do not start a word or sentence, found by `(?<!\b)[A-Z]`.

(e) Not all tweets started with uppercased letters to begin with. We can remove white space from the beginning of tweets (remove `^\s+`), and, afterwards, make the first letters of sentences (`^[a-z]`) uppercased.

(f) We can add punctuation to tweets that do not end in punctuation `$(?<![.!?])`.