George Maratos
Homework 2 part a

Part 1 Automatic POS tagging:

1. "I am sitting in Mindy's restaurant putting on the gefillte fish, which is a dish I am very fond of"

Results: nlp.stanford.edu

Tagging

```
I/PRP
am/VBP
sitting/VBG
in/IN
Mindy/NNP
's/POS
restaurant/NN
putting/VBG
on/IN
the/DT
gefillte/JJ
fish/NN
,/,
which/WDT
is/VBZ
a/DT
dish/NN
I/PRP
am/VBP
very/RB
fond/JJ
of/IN
```

Parse

```
(ROOT
  (S
    (NP (PRP I))
    (VP (VBP am)
      (VP (VBG sitting)
        (PP (IN in)
          (NP
            (NP (NNP Mindy) (POS 's))
            (NN restaurant)))
```

```
(S
   (VP (VBG putting)
      (PP (IN on)
         (NP
            (NP (DT the) (JJ gefillte) (NN fish))
            (, ,)
            (SBAR
               (WHNP (WDT which))
               (S
                  (VP (VBZ is)
                     (NP
                        (NP (DT a) (NN dish))
                        (SBAR
                           (S
                              (NP (PRP I))
                              (VP (VBP am)
                                 (ADJP (RB very) (JJ fond)
                                    (PP (IN of)))))))))))))))))))
```

Typed dependencies

nsubj(sitting-3, I-1)
aux(sitting-3, am-2)
root(ROOT-0, sitting-3)
prep(sitting-3, in-4)
poss(restaurant-7, Mindy-5)
possessive(Mindy-5, 's-6)
pobj(in-4, restaurant-7)
vmod(sitting-3, putting-8)
prep(putting-8, on-9)
det(fish-12, the-10)
amod(fish-12, gefillte-11)
pobj(on-9, fish-12)
nsubj(dish-17, which-14)
cop(dish-17, is-15)
det(dish-17, a-16)
rcmod(fish-12, dish-17)
nsubj(fond-21, I-18)
cop(fond-21, am-19)
advmod(fond-21, very-20)
rcmod(dish-17, fond-21)
prep(fond-21, of-22)

Typed dependencies, collapsed

nsubj(sitting-3, I-1)
aux(sitting-3, am-2)
root(ROOT-0, sitting-3)
poss(restaurant-7, Mindy-5)
prep_in(sitting-3, restaurant-7)
vmod(sitting-3, putting-8)
det(fish-12, the-10)
amod(fish-12, gefillte-11)
prep_on(putting-8, fish-12)
nsubj(dish-17, fish-12)
cop(dish-17, is-15)

det(dish-17, a-16)
rcmod(fish-12, dish-17)
prep_of(fond-21, dish-17)
nsubj(fond-21, I-18)
cop(fond-21, am-19)
advmod(fond-21, very-20)
rcmod(dish-17, fond-21)

Results from cogcomp.cs.illinois.edu:

PRP|I **VBP**|am **VBG**|sitting **IN**|in **NNP**|Mindy **POS**|'s **NN**|restaurant **VBG**|putting **IN**|on|**DT**|the **NN**|gefillte **NN**|fish ,
WDT|which **VBZ**|is **DT**|a **NN**|dish **PRP**|I **VBP**|am **RB**|very **JJ**|fond **IN**|of

2. Sentence: "Wearn a suit tomorrow for da interview ;+) dats how serious da money is"

Results: nlp.stanford.edu

Your query

Wearn a suit tomorrow for da interview ;+) dats how serious da money is

Tagging

```
Wearn/VB
a/DT
suit/NN
tomorrow/NN
for/IN
da/NN
interview/NN
;/:
+/VBN
-RRB-/-RRB-
dats/VBZ
how/WRB
serious/JJ
da/NN
money/NN
is/VBZ
```

Parse

```
(ROOT
   (S
      (VP (VB Wearn)
         (SBAR
            (S
               (NP
                  (NP (DT a) (NN suit) (NN tomorrow))
                  (PP (IN for)
                     (NP (NN da) (NN interview)))
                  (: ;)
                  (VP (VBN +))
                  (-RRB- -RRB-))
               (VP (VBZ dats)
                  (SBAR
                     (WHADJP (WRB how) (JJ serious))
                     (S
                        (NP (NN da) (NN money))
                        (VP (VBZ is)))))))))))
```

Typed dependencies

```
root(ROOT-0, Wearn-1)
det(tomorrow-4, a-2)
nn(tomorrow-4, suit-3)
nsubj(dats-11, tomorrow-4)
prep(tomorrow-4, for-5)
```

nn(interview-7, da-6)
pobj(for-5, interview-7)
vmod(tomorrow-4, +-9)
ccomp(Wearn-1, dats-11)
advmod(serious-13, how-12)
dep(is-16, serious-13)
nn(money-15, da-14)
nsubj(is-16, money-15)
ccomp(dats-11, is-16)

Typed dependencies, collapsed

root(ROOT-0, Wearn-1)
det(tomorrow-4, a-2)
nn(tomorrow-4, suit-3)
nsubj(dats-11, tomorrow-4)
nn(interview-7, da-6)
prep_for(tomorrow-4, interview-7)
vmod(tomorrow-4, +-9)
ccomp(Wearn-1, dats-11)
advmod(serious-13, how-12)
dep(is-16, serious-13)
nn(money-15, da-14)
nsubj(is-16, money-15)
ccomp(dats-11, is-16)

Results from cogcomp.cs.illinois.edu:

NNP|Wearn **DT**|a **NN**|suit **NN**|tomorrow **IN**|for **NNP**|da **NN**|interview **: ;** **NN**|+ **- RRB-** |) **NN**| dats **WRB**|how **JJ**|
serious **NNP**|da **NN**|money **VBZ**|is

3. Sentence

 "The single most important factor affecting allophonic variation is the

identity of the surrounding phones."

(Taken from the book)

Stanford Results:

Your query

The single most important factor affecting allophonic variation is the identity of the surrounding phones.

Tagging

The/DT
single/JJ
most/RBS
important/JJ
factor/NN
affecting/VBG
allophonic/JJ
variation/NN
is/VBZ
the/DT
identity/NN
of/IN
the/DT
surrounding/VBG
phones/NNS
./.

Parse

(ROOT
  (S
    (NP
      (NP (DT The) (JJ single)
        (ADJP (RBS most) (JJ important))
        (NN factor))
      (VP (VBG affecting)
        (NP (JJ allophonic) (NN variation))))
    (VP (VBZ is)
      (NP
        (NP (DT the) (NN identity))
        (PP (IN of)
          (NP (DT the) (VBG surrounding) (NNS phones)))))
    (. .)))

Typed dependencies

det(factor-5, The-1)
amod(factor-5, single-2)

advmod(important-4, most-3)
amod(factor-5, important-4)
nsubj(identity-11, factor-5)
vmod(factor-5, affecting-6)
amod(variation-8, allophonic-7)
dobj(affecting-6, variation-8)
cop(identity-11, is-9)
det(identity-11, the-10)
root(ROOT-0, identity-11)
prep(identity-11, of-12)
det(phones-15, the-13)
amod(phones-15, surrounding-14)
pobj(of-12, phones-15)

Typed dependencies, collapsed

det(factor-5, The-1)
amod(factor-5, single-2)
advmod(important-4, most-3)
amod(factor-5, important-4)
nsubj(identity-11, factor-5)
vmod(factor-5, affecting-6)
amod(variation-8, allophonic-7)
dobj(affecting-6, variation-8)
cop(identity-11, is-9)
det(identity-11, the-10)
root(ROOT-0, identity-11)
det(phones-15, the-13)
amod(phones-15, surrounding-14)
prep_of(identity-11, phones-15)

SNOW results:

DT|The JJ|single RBS|most JJ|important NN|factor VBG|affecting JJ|allophonic NN|variation VBZ|is DT|the NN|identity IN|of DT|the VBG|surrounding NNS|phones . .

Follow up Questions to First Part:

1.

The taggers to not agree on certain tags. In the first sentence, the taggers did not agree on the tag for the word gefillte. Stanford's tagger believed that the tag should have been JJ, while the SNOW tagger tagged it as NN. In the third sentence both taggers agreed with the structures of the sentence. In the second sentence the taggers did not agree on many different words. Below is a list of words tagged differently followed by their tag.

Different tags for Sentence 2

Stanford:	SNOW:
wearn / VB	wearn / NNP
+ / VBN	+ / NN
dats = VBZ	dats / NN
da / NN	da / NNP

Format: (word) / (tag)

2.

Analysis of First Sentence:

I personally agree with the tagging done by the Stanford tagger, I believe that the SNOW tagger did not tag gefillte right as it is an adjective describing a kind of fish.

Analysis of the Second Sentence:

SNOW tagged "wearn" as a proper noun, but as it is slang for "wearing", it shoud have been tagged as a verb. Stanford and SNOW both tagged the word "da" wrong, as it is slang for the word "the , in this case it should be DT. Both taggers had issues with interpreting the ascii smiley face ";+)" which really has no tag, unless Penn Treebank decides to at a tag for emoticons. The word "dats" is really a contraction of the words, "that" and "is"; therefore, we could give it its own tag, which would solve the tagging problem, or treat it as two words for which we would tag it as both DT for "that" and VBZ for "is". And, in conclusion, the word "da" was tagged as a noun in both systems, but since "da" is really slang for "the" it sould be tagged as a DT.

Analysis of the third sentence:

Both taggers did a relatively good job of tagging the sentence although the word "surrounding" was tagged as a present verb when it should have been tagged as an adjective (JJ) describing the noun "phones".