Section 1: Semantics

Question 1A: Consider the following frequency vectors

black = (4, 4, 2, 6, 0), white = (4, 8, 0, 2, 10)

Using cosine similarity, compute the distributional semantic distance between 'black' and 'white'

The cosine similarity of two words can be calculated as $\cos\alpha = \frac{|w_1 \cdot w_2|}{\|w_1\| \cdot \|w_2\|}$

In this case we have:

$$\left.\begin{aligned} \|black\| &= \sqrt{4^2+4^2+2^2+6^2} = \sqrt{16+16+4+36} = \sqrt{72} \\ \|white\| &= \sqrt{4^2+8^2+2^2+10^2} = \sqrt{16+64+4+100} = \sqrt{184} \\ |black \cdot white| &= 4\cdot4 + 4\cdot8 + 6\cdot2 = 16+32+12 = 60 \end{aligned}\right\} \Rightarrow \cos\alpha = \frac{60}{\sqrt{72}\sqrt{184}} = \frac{60}{\sqrt{13284}} \simeq 0.52$$

Therefore, in this context, the cosine similarity between white and black is 0.52.

Question 1B: Consider the following sense definitions for 'bank':

bank#1 - the slope beside a body of water

bank#2 - the financial institution that accepts deposits and channels the money into lending activities

Now consider the following occurrence of bank in this sentence:

"The bank was left free to offer interest on demand deposits"

How would you apply the lesk algorithm to disambiguate 'bank' between the two ~~sentences~~ senses given above?

Given a word and a sentence, lesk algorithm returns the best sense of the word.

```
function SIMPLIFIED LESK (word, sentence):
    best_sense ← most frequent sense for word
    max_overlap ← 0
    context ← set of words in sentence
    for each sense in senses of word do
        signature ← set of words in the gloss and
                    examples of sense
        overlap ← COMPUTEOVERLAP(signature, context)
        If overlap > max_overlap then:
            max_overlap ← overlap
            best_sense ← sense
end return (best_sense)
```

The second definition would have a higher overlap with the context of the word bank that we are refering to. Therefore, LESK algorithm would return the second ~~sense~~ sense.

Question 1C: Define homonymy, synonymy, and antonymy.

homonymy: two words that have the same spelling or pronunciation but different meaning

synonymy: a words that means exactly or nearly the same than another word

antonymy: a word that means exacty or nearly the opposite than another word

Section 2: Part-of-speech tagging

Question 2A: Consider a Hidden Markov Model with the following probabilities (S designates the start state):

$p(w_i \mid t)$	w_i = the	w_i = University	w_i = of	w_i = Ireland
t_i = B	0.2	0.7	0.2	0.4
t_i = I	0.2	0.2	0.5	0.4
t_i = O	0.6	0.1	0.3	0.2

$p(t_i \mid t_{i-1})$	t_{i-1} = B	t_{i-1} = I	t_{i-1} = O	t_{i-1} = S
t_i = B	0.1	0.3	0.3	0.3
t_i = I	0.6	0.4	0.0	0.0
t_i = O	0.1	0.2	0.3	0.4

What is the probability of the sequence "the University of Ireland" being tagged as "O B I I"?

Viterbi algorithm can be used to calculate the ~~maximum~~ most likely sequence of tags for that sentence. However, in this case, is enough to apply the following approximation of probability:

$$p(w_1, \ldots, w_N, t_1, \ldots, t_N) \cong p(w_N \mid t_N)\, p(t_N \mid t_{N-1})\, p(w_{N-1} \mid t_{N-1}) \cdots p(t_2 \mid t_1)\, p(w_1 \mid t_1)\, p(t_1)$$

In this specific case, this is:

$$p(\text{"the, University of Ireland"}, \text{"O B I I"}) = p(\text{"Ireland"} \mid \text{"I"})\, p(\text{"I"} \mid \text{"I"})\, p(\text{"of"} \mid \text{"I"})\, p(\text{"I"} \mid \text{"B"})\, p(\text{"University"} \mid \text{"B"})\, p(\text{"B"} \mid \text{"O"})\, p(\text{"the"} \mid \text{"O"})$$

$$= 0.4 \cdot 0.4 \cdot 0.5 \cdot 0.6 \cdot 0.7 \cdot 0.3 \cdot 0.6 \cdot 0.4 = 0.0024192$$

Therefore, the probability of "the University of Ireland" being tagged as "O B I I" is of 0.0024192.

- Question 2B: Given an annotated text corpus, describe how would you find the probabilities such as given in the table above. Write any algorithms you would use in pseudocode.

Using an annotated text corpus, the previous probabilities can be calculated in the following way:

$$p(S_i|S_j) = \frac{c(t_{i-1}=s_j, t_i=s_i)}{\sum_{s'} c(t_{i-1}=s_j, t_i=s')} \quad (1)$$

$$p(w|s) = \frac{c(w_i=w, t_i=s)}{c(t_i=s)} \quad (2)$$

(1) Pseudocode:

Let's assume that corpus is given in a tokenize format where each token is a pair (word, tag).

```
count_tags ← array with zeros of size #tags x #tags
for n=1,..., len(corpus):
    (previous_word, previous_tag) ← corpus[n-1]
    (word, tag) ← corpus[n]
    array[tag, previous_tag] += 1

for row in tags:
    total_row = sum(array[tag_i, :])
    array[tag_i, :] = array[tag_i, :] / total row
```

(2) Pseudocode

```
count_word_tags ← array with zeros of size #words x #tags
for n=0,..., len(corpus):
    (word, tag) ← corpus[n]
    array[word, tag] += 1

for col in tags:
    total_col = sum(array[:, tag])
    array[:, tag] = array[:, tag] / total_col
```

Section 3: Sentiment Analysis

Question 3A: Explain two challenges for automatic approaches to sentiment analysis

Implicit sentiment is a challenge of sentiment Analysis. While neutral words are used in a sentence the sentiment of the sentence can still be positive or negative.

A second challenge is word ambiguity in which the same word can be used to express a positive and negative feeling.

Question 3B: What is a sentiment lexicon and how it may be used as a feature in a sentiment analysis classifier?

Sentiment Lexicon is a database that provides a lists of positive and negative words. In a simple way, counting the positive vs negative words in a sentence can give an estimation of the overall probability for that sentence being pos/neg.

Question 3C: Provide a suggestion of one way in which negation may be handled in a sentiment analysis.

The simplest approach is to append 'NOT' to all words after the occurrence of a 'negation word' until next punctuation. This forms new words that can be identified during the training of a model

Question 3D: What is meant with aspect-based sentiment analysis? Give an example of an aspect.

Aspect-based sentiment analysis ~~aims~~ is a technique that aims to extract both, the identity described in the text (e.g., product or service) and the sentiment expressed towards such entities.

Example:

In a customer review that says:

"The restaurant had great food but the service was disappointing".

Aspect-based sentiment analysis could identify two categories: food and service. The first one expressing a positive opinion and the second a negative one.

Section 4: Information Extraction

Consider the following corpus of sentences about company acquisitions, with named entity annotation (COM: company) and gold standard labelling of the sentence does or does not express a company acquisition.

	company acquisition (Y/N)	Sentence
1	Y	[COM Saleforce] to acquire data analytics firm [COM Tableau] in $157 billion deal
2	Y	[COM Bird] confirms acquisition of [COM Scoot]
3	Y	[COM Shutterfly] to be merged with [COM Snapfish] after $27B acquisition
4	N	A $3.3 billion [COM walmart] acquisition of [COM jet.com] is under discussion
5	Y	[COM Medtronic] acquisition of [COM INM] approved by regulator

Question 4A: What is the Precision, Recall and F-score of an information extraction system that has just one pattern, applied to the 5 sentences given above:

[COM X]* acquisition of* [COM Y]

This model would return

	pred	true
1	N	Y
2	Y	Y
3	N	Y
4	Y	N
5	Y	Y

This gives us the following confusion matrix:

	Positive Pred	Neg Pred
Positive Class	2	2
Negative Class	1	0

$$\text{Precision} = \text{True Positives} / (\text{True Pos} + \text{False Pos}) = \frac{2}{2+2} = \frac{1}{2}$$

$$\text{Recall} = \text{True Positives} / (\text{True Pos} + \text{False Neg}) = \frac{2}{2+1} = \frac{2}{3}$$

$$\text{F-measure} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall}) = \frac{2 \cdot 1/2 \cdot 2/3}{1/2 + 2/3} = \frac{4/6}{\frac{3+4}{6}} = \frac{4/6}{7/6} = 4/7$$

- Question 4B Give the formula for Cohen's Kappa coefficient. What is it used for in information extraction?

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}$$ where a is the observed agreement and e the expected agreement

κ is used as a measure of inter-annotator agreement and helps to know if the annotation used in training is reliable for the specific case.