INTRODUCTION TO NATURAL LANGUAGE PROCESSING
HOMEWORK ASSIGNMENT #1

CORPUS AND DATA WILL BE PUT ON THE WEBSITE A WEEK PRIOR TO THE ASSIGNMENT.
PLEASE SEND ANY QUESTIONS TO TSARFATY@STP.LINGFIL.UU.SE

1. Probability Theory Primer (30pt)

Let $\Omega$ be an event space, i.e., a set of possible outcomes.
Let $E$ be the set of all subsets of $\Omega$, i.e.,
$$ E = \{ A \mid A \subset \Omega \}, |E| = 2^{\Omega} $$

For a specific event $A \in E$, we define the following

- **The complement event:** $\bar{A} = \Omega - A$
- **The impossible event:** $A = \emptyset$
- **The certain event:** $A = \Omega$

Let $P$ be a probability mass function that fulfills the following axioms

- **The Range Axiom:**
  $$ \forall A \in E : 0 < P(A) < 1 $$

- **The Certain Event Axiom:**
  $$ A = \Omega \Rightarrow P(A) = 1 $$

- **The Mutually Exclusive Events Axiom:**
  $$ \forall A, B \in E : A \cap B = \emptyset \Rightarrow P(A \cup B) = P(A) + P(B) $$

1. Prove, using the above axioms and definitions, the following equalities (3pt each)

   (1) $P(\emptyset) = 0$
   (2) $P(\bar{A}) = 1 - P(A)$
   (3) $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
   (4) $P(B - A) = P(B) - P(A \cap B)$
   (5) if $\cup_i A_i = A$ and $\forall A_i, A_j : A_j \cap A_j = \emptyset$ then $P(A) = \Sigma_i P(A \cap A_i)$
Let $A, B$ be two events in $E$. The probability of event $A$ given event $B$ is called the conditional probability of $A$ given $B$ and is defined as follows.

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

2. Prove, using the above axioms and definitions, the following rules (5pt each)

1. **Chain rule:**

   $$P(A \cap B) = P(A)P(B|A) = P(B)P(A|B)$$

2. **Bayes rule:**

   $$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

3. **Useful rule:**

   for $\bigcup_{i=1}^{n} A_i = \Omega$ show that

   $$P(B) = P(\Omega \cap B) = \Sigma_{i=1}^{n} P(A_i \cap B) = \Sigma_{i=1}^{n} P(A_i)P(B|A_i)$$

2. **Words and Morphemes (30pt)**

   Let us define a simple unigram model in which the probability of a space delimited token $P(w)$ is estimated as the relative frequency of this word in a body of text.

   1. Let a “word” be a sequence of symbols separated by white-spaces. Take a large text and extract all words and their counts from a steadily growing part of the text. Start with 10% of the text and add another 10% every time until you count the full text. What is the number of word types in the full corpus? what is the number of word tokens in the full corpus? Plot the relative frequency estimate (RFE) of the word probability for certain words and observe whether the RFE is converging around a certain value as the data grows large. Discuss the differences between the convergence of different words (10pt)

   2. Let a ”morpheme” be a minimal unit of sound-meaning correspondence. Assume a corpus of text in which the space-delimited tokens are morphemes, not words. Take a large text thus annotated and extract all morphemes and their counts from a steadily growing part of the text. Start with 10% of the text and add another 10% every time until you count full text. What is the number of morpheme types in the full corpus? what is the number of morpheme tokens in the full corpus? Plot the relative frequency estimate (RFE) of the word probability for certain words and observe whether the RFE is converging around a certain value as the data grows large. Discuss the differences between the convergence of different morphemes (10pt)

   3. Discuss the similarities and difference in the convergence patterns as observed for sequences of words vs. sequence of morphemes. Can you think of an explanation/justification for your observation? What is the linguistic significance of your observation? (10pt)
3. N-Grams and Language Modeling (40pt)

In this exercise your goal is to build a simple language model over word sequences. This will be a first-order Markov Model (i.e., based on n-grams such as \( n = 2 \)). You are required to write a computer program which takes a sentence of words delimited with white spaces as input and returns its estimated probability based on the language model, as output.

- **Note 1:** remember to differentiate upper-case and lower-case occurrences
- **Note 2:** every sentence should end with a punctuation mark
- **Note 3:** every sentence should start with an artificial START symbol

1. Before proceeding with implementation, formalize the model to implement (6pt)
   (1) Provide the formula for the probability of a given sentence based on a first-order language model
   (2) Provide a formula for the relative frequency estimates for \( P(w_x|w_y) \) and for \( P(w_y, w_x) \).

2. Given these formulae, you may proceed with the implementation (24pt)
   (1) Extract and store, as efficiently as you can, a table of all n-grams with their frequencies from the text, where \( n = 1, 2 \)
   (2) Implement a function which returns the relative frequency estimate for \( P(w_x|w_y) \) for any given bigram \( \langle w_y, w_x \rangle \) from the counts in the table.
   (3) Implement a function that takes as input a sentence and assigns to it a probability according to the 1st-order Markov model.

3. Given the computer program thus described, answer the following questions (10pt)
   (1) Assign a probability for the sentence
       “EFRWT ANFIM MGIRIM M TAILND L IFRAL”
       (tens of people arrive from Thailand to Israel)
   (2) Assign a probability for the sentence
       “EFRWT ANFIM MGIRIM M IFRAL L TAILND”
       (tens of people arrive from Israel to Thailand)

   Explain what causes the difference in the probabilities of these two sentences. What are the modeling assumptions that this difference points to? How could you implement a model that can improve on that? (Hint and Bonus (10pt): implement a smoothing component for the language model and repeat the experiment. What are your new findings?)