**LIN3022 Natural Language Processing**

**Practical Task I**

# 1 Introduction

This task involves the practical application of some concepts related to probability, language modelling and smoothing. We shall use the British National Corpus (BNC) online. The BNC is a balanced corpus of British English text (90%) and speech (10%) of ca. 100 million words.

For this tutorial, you will be making some probability computations, so it’s useful to have a calculator handy (e.g. the Windows calculator).

* 1. ***Accessing the BNC***

The BNC can be accessed online here: [http://sara.natcorp.ox.ac.uk](http://sara.natcorp.ox.ac.uk/lookup.html)

## 1.2 An example application

Automatic Speech Recognition (ASR) software usually proceeds by first analysing an incoming speech signal, and then using some form of statistical estimation to resolve ambiguities. Here is an example of the latter. Suppose a user has uttered the sentence:

* *He ate two apples.*

ASR software assigns an interpretation to an utterance, and to do so it needs to rely on purely phonetic input. There may be several possible interpretations, because words are phonetically ambiguous. For example, from a purely phonetic point of view, *two* is could be either the word *two*, or *to*, or *too*.

**Q1:** Using purely phonetic/phonological input, how many possible hypotheses does the ASR program need to test? Write down the sentences corresponding to the different interpretations.

# 2 Resolving ambiguities using simple probabilities

Probably the simplest way to resolve the ambiguity is to make a guess using raw frequencies. Let us assume that the ASR software has recognised *he* and *apples* correctly. The problem is the ambiguous words *two* and *ate*. Remember that the corpus has 100 million words, and so this is your “corpus size”.

**Q2.** Look up the frequencies of *ate* and *eight* in the BNC. Estimate the frequency-based probability of these two words in the corpus (also known as the *Maximum Likelihood Estimate*).

**Q3.** Repeat the process for *to, too,* and *two.*

**Q4.** Note that you are essentially estimating a **unigram** model, i.e. a model based on single words with no context. Based on the probabilities for the alternative interpretations of the two ambiguous words, which of the possible interpretations for the whole sentence that you listed in Q1 would the ASR software choose?

## 2.1 Computing the probability of a sentence

Having identified the “most likely” words based on a unigram model, we now want to compute the probability of the chosen sentence as a whole.

**Q5.** Estimate the probability for all the remaining words (*he*, *apples*) in the sentence that you identified as most likely in Q4.

**Q6.** Compute the overall probability of the sentence. (Hint: use the multiplication rule.) Intuitively, do you think this is a likely sentence based on this model?

**Q7.** Compare the probability of this sentence to the probability of the correct sentence (*he ate two apples*). Which comes out highest?

**Q8.** If the ASR software used unigrams only, as you have been doing, it could analyse the sentence by using only the probability of the ambiguous words. It wouldn’t need to compute the probability of the non-ambiguous words. Explain why this is the case.

## Using bigram statistics

A bigram is a pair of words that can occur together, i.e. a pair which is *attested* in a corpus. We will now compute the probability of the two sentences based on bigrams, and compare the results to those obtained using the simple unigram model. In what follows, the wrong sentence chosen using the unigram model is referred to as S1, the correct one is referred to as S2.

In a bigram model, the probability of a sentence is the product of P(w(i+1) | w(i)), where w(i) is the ith word of the sentence. We also assume that there is an additional special "start of sentence" word <s> at the start of a sentence. The probabilities for each bigram are estimated from a corpus using the standard conditional probability formula.

**Q9.** List the set of bigrams in S1 and S2. Be sure to include the start-of-sentence symbols.

**Q10.** We’ll now compute the probability of each bigram. Since it’s not straightforward to estimate the likelihood of <s>+he, just assume that it has a conditional probability P(he|<s>) = 0.004.

Using the web interface, conduct a multi-word query for each bigram in S1 and S2. For each bigram <w1, w2>, you need to:

1. Find the probability P(w1);
2. Find the probability P(w1 AND w2), using a multiword query where you search for w1 and w2 in that order.
3. If a bigram has frequency of 0, then P(w2|w1) is 0.

**Q11.** Compute the overall probabilities of S1 and S2 based on bigrams, using the multiplication rule. Which comes out as the most probable?

**Note:** ASR models often use trigrams. However, this example should suffice to indicate the utility of n-gram models, and particularly of conditional probabilities.

## Smoothing

We earlier assumed that if a bigram did not occur at all in the corpus, then it had a probability of 0. Assuming that “unseens” have zero probability is a problematic assumption.

**Q11**. List some of the problems with assuming zero probability for unseens in the sample.

**Q12.** For these reasons, our calculations of the frequency (and hence the probability) of what we actually observe are overestimated. Why?

To get around the problem of rare words or phrases, which may exist even though our corpus doesn’t have them, we use *smoothing*. This exercise will only give you an introduction to one, simple method for smoothing. You’ll encounter others in later lectures.

### Add-one smoothing

Add-one smoothing is the simplest form of smoothing. This is calculated as follows:

1. Add 1 to the frequency of the event of interest (in our case, a bigram);
2. Add a coefficient V to the denominator, where V is the number of unique words (types) in the corpus.
3. Compute the probability:
   1. P(w2|w1) = (number of times w2 follows w1 + 1)/(number of times w1 occurs + V)
   2. You can assume that V is about 1 million in the BNC.

**Q13.** Repeat the steps used in the earlier exercise, to re-compute the probabilities of S1 and S2 using the new method. How do the new estimates differ?

# Acknowledgement

The first part of this tutorial is based on some ideas from a practical given by Chris Mellish at the University of Aberdeen.