INGI2263 – Computational Linguistics

Assignment 3

Semantic Analysis

In this assignment, you will be given the opportunity to dabble with semantic similarities of words. More precisely, our aim is to exploit the co-occurence contexts of words in order to measure pairwise similarities.

1 General description

1.1 The task

For this assignment, we rely on a corpus of most frequent 5-grams. It has the advantage to spare you the preprocessing workload and make your results more comparable with ours. In general, we would have considered a corpus of documents. In that case, the corpus would have undergone the usual steps of preprocessing and contexts would have been extracted with a sliding window.

The corpus is provided as a `tsv` (tab-separated values) file. Each line begins by the number of occurences of the 5-gram, followed by the sequence of tokens. Here is an excerpt:

```
12   a   badge   of     honor    to
8    a   bag     of     chips    and
7    a   bag     of     frozen   peas
8    a   bag     of     frozen   vegetables
7    a   bag     of     ice      on
45   a   bag     of     potato   chips
9    a   bag     over   her      head
21   a   bag     over   his      head
8    a   bag     over   your     head
7    a   bagel   with   cream    cheese
```

The rules of the game are quite simple. Given a term in the lexicon (more on that below), your program should be able to produce a list of the most similar terms with a score. We insist here on the fact that all the words must be present in the corpus : the query term as well as the results. Here is the kind of results one could obtain with the query term *sister*:

1.	sister	1.0000
2.	mother	0.6912
3.	father	0.6897
4.	brother	0.6578
5.	husband	0.6341
6.	grandmother	0.6146

7.	daughter	0.5338
8.	boyfriend	0.5156
9.	friends	0.4914
10.	mom	0.4896

1.2 Crash course on vectorial representation of documents

In order to do that, we can represent the context of each word as a vector w and evaluate the similarity between two representations as

$$sim(w_1, w_2) = \frac{w_1 \cdot w_2}{||w_1||\,||w_2||},$$

that is, as the cosine of the angle between their corresponding vectors.

There exists reams of ways to compute the vectorial representation of a word from its context. The most basic approach consists in choosing a fixed lexicon and representing the term as a bag-of-words. More precisely, each entry in the vector corresponds to a word type in the lexicon and its value is the number of times it appears in the context. Many other approaches still rely on a fixed vocabulary, but instead of merely plugging the word count, each entry in the vector is computed as the product of two factors :

- A local weight that only depends on the context at hand. This weight intends to represent the importance of the word type in the context.
- A global weight specific to each word type, that is computed from the set of contexts. This one represents the discriminatory power of the word type. To give you the intuition, it is easy to understand that words that occur in each context might not be as relevant as less frequent words to measure the similarity.

Hence, the general form of the j^{th} entry of the i^{th} context-vector can be mathematically expressed as $w_{ij} = l_{ij} \times g_j$ with l_{ij}, the local factor and g_j, the global factor that depends only on the word type j. There are virtually an infinity of ways to define these two factors. A popular weighting scheme is tf-idf. But before defining it, let's introduce a few notations:

- n, the number of contexts (which, incidentally, is equal to the lexicon size in our case).
- tf_{ij}, the number of times the j^{th} word type occurs in the i^{th} context (*tf* stands for *term frequency*).
- df_j, the proportion of contexts in which the j^{th} word type occurs (*df* stands for *document frequency*). For instance, if the term j appears in 735 out of 1000 contexts, $df_j = 0.735$.

In the case of tf-idf scheme, the weights are defined as :

- $l_{ij} = tf_{ij}$. So, the importance of a term in the vectorial representation of a context is proportional to number of times it occurs in it.
- $g_j = -\log df_j$. This weight is usually called the inverse document frequency ($g_j = idf_j$). As a result, the weight of a term decreases if it appears in high proportion of contexts.

Consistently with the general description, we have $d_{ij} = tf_{ij} \times idf_j$.

2 Dataset

For this assignment, we will use a free sample of the 5-grams frequency list of COCA (Corpus of Contemporary American English). It can be found at:

http://www.ngrams.info/samples_coca1.asp.

You should download the non-case sensitive corpus of 1,000,000 most frequent 5-grams. We suggest you to use your official uclouvain email address to register.

3 From n-grams to semantic similarity...

This section provides a step-by-step guide to extract the relevant counts from data and compute similarities between words.

3.1 Lexicon

The first step consists in defining the lexicon. In order to do that, we ask you to count the number of occurrences for each word type in the corpus (without any kind of preprocessing). Pay attention to take the n-gram frequency into account during this operation. Rank them in descending order according to their number of occurrences and discard the top 250 terms. The remaining terms will constitute the lexicon.

Report the top 20 terms in the lexicon as well as their frequency in a table. Could you think of another way, we could have filtered out less important words? Gain inspiration from the weighting scheme described in the previous section.

3.2 Bag-of-words representation of contexts

The context of a word will be computed based on the set of n-grams in which it appears. The bag-of-words representation of a term reports the co-occurrence counts of this term with the word types in the lexicon. Let's consider the example of *frozen* based on the excerpt presented in section 1.1. Its bag-of-words representation is

$$\{bag : 15,\ peas : 7,\ vegetables : 8\}$$

if we assume that *a* and *for* are not in the lexicon. This sparse representation of the vector implies that the entries corresponding to the other word types are null. Besides, note the effect of the n-gram frequency on the computation.

Report the bag-of-words representation of the word types *fireworks* and *furnace* (in a reader-friendly way).

happy	italy
jump	japan
plane	good
planes	october

Table 1: List of words

3.3 TF-IDF representation

Building on the results of the previous section, you are now asked to transform the bag-of-words representations of contexts to their corresponding tf-idf representation. Refer to the section 1.2 for the details of the weighting scheme.

Report this time the tf-idf representation of the word types *fireworks* and *furnace* (still in a reader-friendly way). What are the 10 most similar terms to the query terms listed in table 1? Report these words paired with their similarity score in tables. Don't forget to comment those results.

Bonus : Find the most similar pair of word types (excluding pairs of identical words).

3.4 Distribution of similarity scores

Plot the empirical distribution of similarity scores between the whole lexicon and the terms *christmas* and *gift*. The two plots should be histograms with log scale on the y-axis.

4 Practical information

You can choose to carry out this assignment with a partner or on your own. In practice, you are asked to submit a `PDF` report with all the elements mentionned in a frame in this document. It can be submitted here:

http://icampus.uclouvain.be/claroline/work/work_list.php?assigId=7&cidReset=true&cidReq=INGI2263.

The project starts on November 25 and must be submitted via iCampus not later than December 9 at 12.45 (penalty applies for late submission). A follow-up session is scheduled on December 2 after the lesson on machine translation.