School of Computing and Information Systems
The University of Melbourne
COMP90042 WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2017)

Workshop exercises: Week 6

Discussion

1. For the following term co-occurrence matrix (suitably interpreted):

	cup	not(cup)
world	55	225
not(world)	315	1405

(a) Find the Point-wise Mutual Information (PMI) between these two terms in this collection.

- To evaluate PMI, we need the joint and prior probabilities of the two event (in this case, probably w: document contains `world` and c: document contains `cup`.
- We estimate these based on their appearance out of the total number of instances in the collection (2000), and then substitute:

$$\begin{aligned} P(w) &= 280/2000 = 0.14 \\ P(c) &= 370/2000 = 0.185 \\ P(w,c) &= 55/200 = 0.0275 \\ PMI(w,c) &= \log_2 \frac{P(w,c)}{P(w)P(c)} \\ &= \log_2 \frac{0.0275}{0.14 \times 0.185} \\ &\approx 0.0865 \end{aligned}$$

(b) What does the value from (a) tell us about **distributional similarity**?

- This value is slightly positive, which means that the two events occur together (in documents) slightly more commonly than would occur purely by chance. There is some possibility that `world` and `cup` occurring together is somehow meaningful for documents in this collection.

2. In the `WSTA_N9_distributional_semantics` iPython notebook, a document–term matrix is built for the purposes of IR-style document retrieval.

(a) What is the Singular Value Decomposition (SVD) method used for here? Why is this helpful?

- We are using the SVD method to build a representation of our matrix which can use to identify the most important characteristics of documents.
- By throwing away the less important characteristics, we can have a smaller representation of the collection, which will save us (potentially a great deal of) time when evaluating the cosine similarities between the documents and the query.

(b) What is the significance of the `transform_query()` function?

- To find the cosine sensibly, we need the query and the documents to have the same number of dimensions — in this case, that means transforming the query so that it is in the same **vector space** as the document collection.
- In brief, for a (truncated) SVD: $M = U_k \Sigma_k V_k^T$, our document collection is represented as $U_k \Sigma_k$, and then the transformed query can be found as: $q_k = qV_K$ (note the transposition is gone).

3. What is a **word embedding** and how does it relate to **distributional similarity**?

   - We're going to have a representation of words (based on their contexts) in a **vector space**, such that other words "nearby" in the space are similar
   - This is broadly the same what we expect in distributional similarity, e.g. "you shall know a word by the company it keeps."
   - Using a dimensionality-reduction method like SVD helps keep this to a manageable size, and, if we're lucky, allows us to emphasise the more meaningful contexts (and de-emphasise meaningless contexts, like *the*).
   - The row corresponding to the word in the relevant (target/context) matrix is known as the "embedding".

   (a) What is the difference between a **skip-gram** model and a **CBOW** model?

   - In short — the element in the condition of the posterior probability: skip–gram models analyse the probability of the context words **given** the target word; CBOW models analyse the probability of the target word **given** the context words.
   - Another way of looking at this is how we lay out the term–term matrix (before, say, SVD): do we label the target words on the row, and contextual words on the columns, or *vice versa*? (Which one is which?)

   (b) How are the above models trained?

   - The probabilities here are more complicated than just counting some events in a collection; they are based around taking the dot product of the relevant vectors (or average of vectors, in the case of CBOW), and then **marginalising**.
   - More complicated methods for this are beyond the scope of this subject.