Exercises on vector semantics and meaning composition

November 21, 2016

1 Vector Semantics

Exercise 1. Figure 1 shows co-occurrence vectors for four words, computed from the Brown corpus, showing only six of the dimensions (hand-picked for pedagogical purposes). The vector for the word digital is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser. Compute the corresponding *pointwise mutual information* vectors of the given

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Figure 1: Co-occurrence vectors

co-occurrence vectors.
Give an advantage of pointwise mutual information measure over raw frequency measure.
Think of a scenario in which pointwise mutual information values tend to be unreliable and give a solution.

Exercise 2. Which of the words *apricot* or *digital* is closer in meaning to *information*? Use Cosine similarity and vectors of raw frequency as in the following table.

	large	data	computer
apricot	2	0	0
digital	0	1	2
information	1	6	1

2 Meaning Composition

Exercise 3. In this exercise, we practice building simple distributional meaning composition for adjective-noun (AN) constructions, and learning binary classifier for entailment relations of adjective-noun and noun (N).

AN $\models$ N dataset: (Baroni et al., 2012) introduces a dataset for AN $\models$ N entailment including 2490 pairs of adjective-first noun and second noun ($A - N_1, N_2$). Each pair is labeled as positive if $A - N_1$ entails N_2, or negative if $A - N_1$ does not entail N_2. In this task, $A - N_1$ entails N_2 if whenever something is $A - N_1$, it must also be N_2. For example, *(big cat, cat)* is a positive instance because a *big cat* is a *cat*. In contrast, *(big cat, dog)* is a negative instance because a *big cat* is not a *dog*. Our goal is to build a binary classifier that classify each pair of $(A - N_1, N_2)$ to positive class or negative class.

Building the representation for $A - N_1$ and N_2:

Given 32-dimensional Skip-Gram word embeddings computed by Word2Vec tool (see file sg32.txt):

- Write a function to extract vector representation for a single word. It then can be used to extract vector representations for each of A, N_1 and N_2.
- Think of methods to represent meaning composition for $A - N_1$ constructions. For each of composition methods, write a function to compute vector representation for $A - N_1$ using vector representations of A and N_1.

Building entailment classifier:

Evaluate the effectiveness of your meaning composition methods on the AN $\models$ N dataset: Use different machine learning classifiers (e.g., Logistic Regression) provided by WEKA tool `http://www.cs.waikato.ac.nz/ml/weka/` to learn a function classifying instances in AN $\models$ N dataset to positive and negative classes. Cross-validation 10 folds can be used for evaluation.

Guidelines:

- Download WEKA tool from `http://www.cs.waikato.ac.nz/ml/weka/downloading.html`
- You can use any programming language that you are familiar with. But we recommend Python 3. The library "numpy" may be needed.
- WEKA input formats:
  - ARFF: `http://www.cs.waikato.ac.nz/ml/weka/arff.html`
  - CSV