Linear Algebra Structure for Word Senses

Fangzhou Zhai

2017.12

Motivation

- Polysemous words yield a huge headache for word embeddings. Essentially, different senses of a polysemy should predict different neighbours.

Motivation

- Polysemous words yield a huge headache for word embeddings. Essentially, different senses of a polysemy should predict different neighbours.

e.g. 'Tie' could mean either 'a piece of clothes' or 'an undecided match'.

Motivation

How elegant would it be if we could decompose word senses linearly. Say,

$$v_{tie} = \alpha v_{tie1} + \beta v_{tie2}$$

Are There Really Linear Structures

Do the word senses really yield a linear structure?

Are There Really Linear Structures

Do the word senses really yield a linear structure? Let's have some fun ;).

Are There Really Linear Structures

Do the word senses really yield a linear structure? Let's have some fun ;).

1. Take Mister Johnson's favourite pre-trained word embeddings.

Are There Really Linear Structures

Do the word senses really yield a linear structure? Let's have some fun ;).

1. Take Mister Johnson's favourite pre-trained word embeddings.
2. Randomly pick two words w_1, w_2 and create a new word w to represent a polysemy with two senses w_1, w_2.

Are There Really Linear Structures

Do the word senses really yield a linear structure? Let's have some fun ;).

1. Take Mister Johnson's favourite pre-trained word embeddings.
2. Randomly pick two words w_1, w_2 and create a new word w to represent a polysemy with two senses w_1, w_2.
3. Delete embeddings v_{w1}, v_{w2} and train an embedding v_w.

Are There Really Linear Structures

Do the word senses really yield a linear structure? Let's have some fun ;).

1. Take Mister Johnson's favourite pre-trained word embeddings.
2. Randomly pick two words w_1, w_2 and create a new word w to represent a polysemy with two senses w_1, w_2.
3. Delete embeddings v_{w1}, v_{w2} and train an embedding v_w.

Now the fun part. We have a look at the cosine distance between v_w and the span of $\{v_{w1}, v_{w2}\}$. The average is 0.97, with an SD of 0.02. Wow.

A Closer Look at the Linear Structure

A Closer Look at the Linear Structure

Now we can confidently assume that for polysemous w,

$$v_w = \alpha v_{w1} + \beta v_{w2}$$

but what are these coefficients *?

A Closer Look at the Linear Structure

Now we can confidently assume that for polysemous w,

$$v_w = \alpha v_{w1} + \beta v_{w2}$$

but what are these coefficients *?Statistics show that the Pearson correlation

$$\rho(\beta, freq(w_1)/fre1(w_2)) = 0.67$$

and we have

$$\beta \approx 1 - clg(freq(w_1)/fre1(w_2))$$

A Closer Look at the Linear Structure II

Now we have

$$v_w = \alpha v_{w1} + \beta v_{w2}$$

and a plausible interpretation of the coefficients. If we are faced with a potentially polysemous word w, how to obtain the senses v_{w1}, v_{w2}? What are they after all* ?

The Word Senses

- In the context of word embeddings, senses are distinguished by their behaviours of predicting context words, regardless of their frequencies.

The Word Senses

- In the context of word embeddings, senses are distinguished by their behaviours of predicting context words, regardless of their frequencies.
- These senses are considered as vectors on the **unit sphere**, i.e. directions in the embedding space, termed *discourses*.

The Word Senses

- In the context of word embeddings, senses are distinguished by their behaviours of predicting context words, regardless of their frequencies.
- These senses are considered as vectors on the **unit sphere**, i.e. directions in the embedding space, termed *discourses*.
- Behind the scene there is a corpus generation model. Given a discourse c, the probability of generating a word w

$$pr[w|c] \propto exp(c \cdot v_w)$$

Model

We now have

$$v_w = \sum_{j \leq m} \alpha_{w,j} A_j + \eta_w$$

where A_js are the discourse atoms and η_w is a noise vector, s.t. at most k of the coefficients are non-zero.

Model

We now have

$$v_w = \sum_{j \leq m} \alpha_{w,j} A_j + \eta_w$$

where A_js are the discourse atoms and η_w is a noise vector, s.t. at most k of the coefficients are non-zero.
The models yields two hyper-parameters k (max number of senses per word) and m (number of different discourses). This is a standard sparse coding task. Best performance is achieved with $k = 5$ and $m = 2000$.

Hierarchy

Using $m = 2, k = 200$ yields a less fine-grained decomposition. Which is interesting.

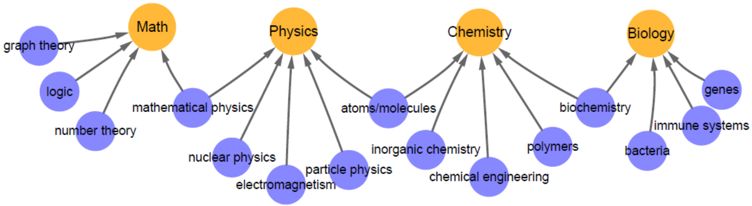

Figure: Discourses in different hierarchies.

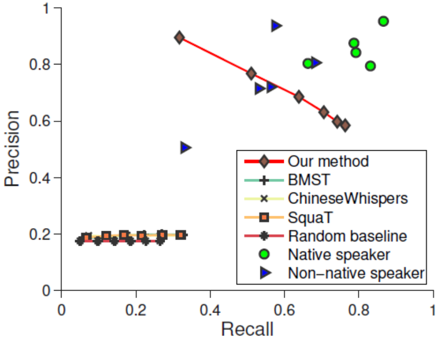

Figure: Performance on word sense disambiguation. The algorithm is comparable with non-native speakers (those living in the states and have been learning English for at least 10 years).

Summary

Summary

- We see an approach of decomposing word embedding vectors into a linear combination of their senses.

Summary

- We see an approach of decomposing word embedding vectors into a linear combination of their senses.
- The senses are represented as vectors on the unit sphere.

Summary

- We see an approach of decomposing word embedding vectors into a linear combination of their senses.
- The senses are represented as vectors on the unit sphere.
- The decomposition result supports a word sense decomposition algorithm whose performance is comparable to that of non-native speakers.

Summary

- We see an approach of decomposing word embedding vectors into a linear combination of their senses.
- The senses are represented as vectors on the unit sphere.
- The decomposition result supports a word sense decomposition algorithm whose performance is comparable to that of non-native speakers.
- Unless this lame for-fun presentation, the paper elegantly and convincingly illustrated the procedure of the research. Would make an excellent read.