Operations on word vectors - v2

July 20, 2019

1 Operations on word vectors

Welcome to your first assignment of this week!

Because word embeddings are very computionally expensive to train, most ML practitioners will load a pre-trained set of embeddings.

After this assignment you will be able to:

- Load pre-trained word vectors, and measure similarity using cosine similarity
- Use word embeddings to solve word analogy problems such as Man is to Woman as King is to _____.
- Modify word embeddings to reduce their gender bias

Let's get started! Run the following cell to load the packages you will need.

```
In [1]: import numpy as np
        from w2v_utils import *
```

```
Using TensorFlow backend.
```

Next, lets load the word vectors. For this assignment, we will use 50-dimensional GloVe vectors to represent words. Run the following cell to load the `word_to_vec_map`.

```
In [2]: words, word_to_vec_map = read_glove_vecs('../../readonly/glove.6B.50d.txt')
```

You've loaded: - `words`: set of words in the vocabulary. - `word_to_vec_map`: dictionary mapping words to their GloVe vector representation.

You've seen that one-hot vectors do not do a good job capturing what words are similar. GloVe vectors provide much more useful information about the meaning of individual words. Lets now see how you can use GloVe vectors to decide how similar two words are.

2 1 - Cosine similarity

To measure how similar two words are, we need a way to measure the degree of similarity between two embedding vectors for the two words. Given two vectors u and v, cosine similarity is defined as follows: