Open Day Handout

March 18, 2017

1 Matching Harry Potter Spells to Their Definition

1.1 Set Up

```
In [3]: from IPython.display import Image
        import hp_spells as hp
        import numpy as np
        import seaborn as sns
        import pandas as pd
        import matplotlib.pyplot as plt
        from tabulate import tabulate
        sns.set(style="white", color_codes=True)
        %matplotlib inline
        im_path = "../graphs/poster/"

In [11]: model = hp.load_vectors("../../vectors/GoogleNews-vectors-negative300.bin'

Loading:  ../../vectors/GoogleNews-vectors-negative300.bin
Loaded:  ../../vectors/GoogleNews-vectors-negative300.bin
```

1.2 General Overview With Spell Examples

Key Points about the program:

- Unless the "–verbose" parameter is specified when executing, the program will not print out spell names.
- The program itself doesn't have a mode which allows the user to enter their own spell name.
  - If the user wants to do this, then they will have to import the package hp_spells and call generateSpell(definition,model).
  - This program runs off a csv list of spell definitions, taken from existing spells.

- The program allows the user to choose whether the analyse one vector model, or whether it analyses both.

Generating A Spell With Word2Vec

```
In [18]: hp.generateSpell("close the door quietly", model)

Out[18]: ([u'claudere', 'spell', u'close'], 2)
```

This function returns the following:

1. Spell Name
2. Spell Type
3. The word generated from the model
4. Number of Bogus Words

1.3 Results & Analysis

1.3.1 Table

```
In [5]: headers=["Metric (Avg.)","Word2Vec", "GloVe"]
        table=[
            ["Accuracy (%)", 68.48,80.0],
            ["Similarity (-1 to 1)", 0.697, 0.600],
            ["Synonyms", 7.124, 4.89],
            ["Gibberish Words", 32.194, 0.672]]
        print tabulate(table, headers, tablefmt="grid")
```

```
+----------------------+------------+---------+
| Metric (Avg.)        |   Word2Vec |   GloVe |
+======================+============+=========+
| Accuracy (%)         |     68.48  |  80     |
+----------------------+------------+---------+
| Similarity (-1 to 1) |      0.697 |   0.6   |
+----------------------+------------+---------+
| Synonyms             |      7.124 |   4.89  |
+----------------------+------------+---------+
| Gibberish Words      |     32.194 |   0.672 |
+----------------------+------------+---------+
```

Analysis

- This displays the average score for each vector on each metric.
- Allows for a more exact comparison of averages compared to the violin graphs.
- As you can see GloVe outperforms Word2Vec on every metric.
  - It has a higher accuracy.
  - It has a lower cosine similarity meaning that the angle between the new word vector and the original is larger.
  - GloVe produces less synonyms, which means that the new words are more "novel".
  - GloVe barely produced gibberish words, whereas Word2Vec produced a high amount, a typical gibberish word might be "j_123".

1.3.2 Scores

```
In [19]: Image(filename=path+"accuracy.png")
```

Out[19]:

Graph Explained

- The score on the y-axis is measured in percentage, and is the percentage of new words that do not exist in the supplied definition.
- On the x-axis is the two different models.
- Each score for each model is plotted on the diagram, the wider the violin at a given point, the higher the density of points there.

Analysis

- GloVe scored more on average than Word2Vec, this is shown as the white dot in both diagram represents the average, and GloVe's is located higher up on the y-axis
- Word2Vec had more results which were at the higher end of the range than the lower end. Shown by the narrow tail endof the Word2Vec violin.
- GloVe's distribution of scores are more dense around the median with only a few outliers, shown by wide bulge in the middle, and then the narrow head and tail.

1.3.3 Cosine Similarity

```
In [20]: Image(filename=path+"similarity.png")
```

```
Out[20]:
```

Graph Explained

- The y-axis displays the cosine similarity between the new spell word vector and the one word definition. The cosine similarity ranges between -1 to 1 where an angle of 0 degrees is 1 and 180 degree as -1.
- On the x-axis is the two different models.
- Each score for each model is plotted on the diagram, the wider the violin at a given point, the higher the density of points there.

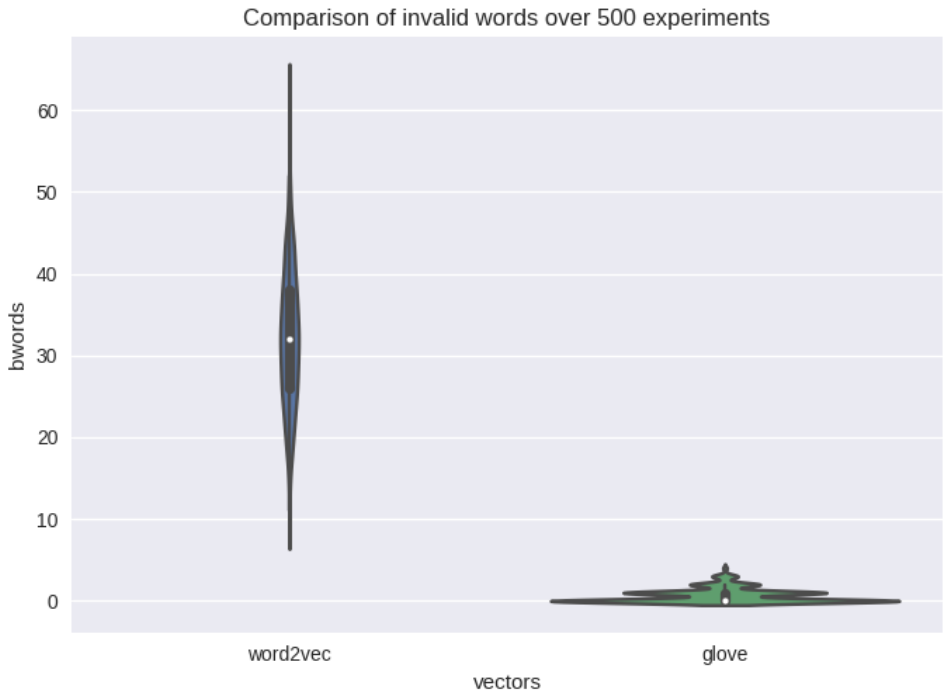
Analysis

- From the graph, you can see that the GloVe vectors have a lower cosine similarity with an average of ~0.60, this means that the new word vectors generated by GloVe are less similar than the vectors generated by Word2Vec.
- Both distributions are very wide and very short meaning that the cosine similarities generated by different experiments were very close together.

1.3.4 Gibberish Words

```
In [22]: Image(filename=path+"invalid.png")
```

```
Out[22]:
```



Graph Explained

- The score on the y-axis is the number of gibberish (or bogus) words.
- On the x-axis is the two different models.
- Each score for each model is plotted on the diagram, the wider the violin at a given point, the higher the density of points there.

Analysis

- The distribution for these two vector sets are very different when it comes to gibberish words.
- Word2Vec has a very long and narrow distribution which means that very experiments returned the same number of gibberish words. With the most amount of occurences ranging in the region of 20-45 bogus words.
- GloVe on the other hand has very little variation of words, with the highest density result being extremely low. There is a much smaller range as demonstrated by how short the graph is.

1.3.5 Synonyms

```
In [21]: Image(filename=path+"synonyms.png")
```

```
Out[21]:
```

Graph Explained

- The score on the y-axis is the number of synonyms generated from an experiment.
- On the x-axis is the two different models.
- Each score for each model is plotted on the diagram, the wider the violin at a given point, the higher the density of points there.
- To check whether a word was a synonym or not, a list of synoynms was generated using NLTK's WordNet.

Analysis

- Synonyms was the metric where there was the smallest difference between the two vector sets.
- GloVe still scored less than Word2Vec on average.
- Word2Vec's distribution was a lot more equal on both ends of the scale. Shown by the symetrical shape on the horizontal axis.
- GloVe had more synonyms closer or below it's average, demonstrated by a wide middle and bottom of the violin and the narrower head.