

Matching Harry Potter Spells to Their Definition

Set Up

```
In [10]: 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
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
```

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 to 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

Results & Analysis

Table?

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.

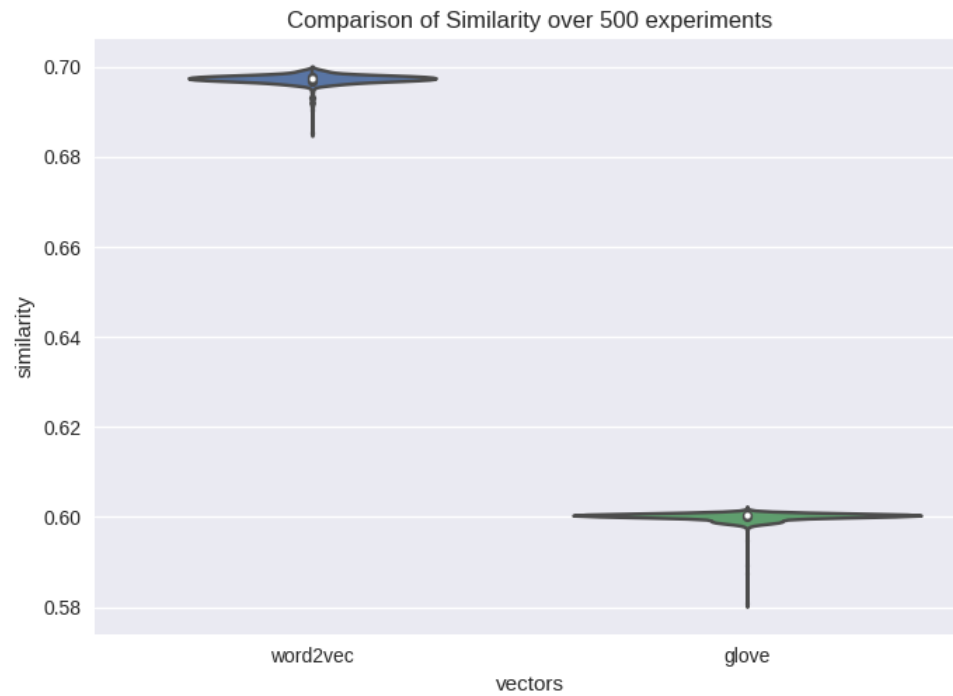
Analysis

- GloVe vector 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 end of the Word2Vec Violin.

Cosine Similarity

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

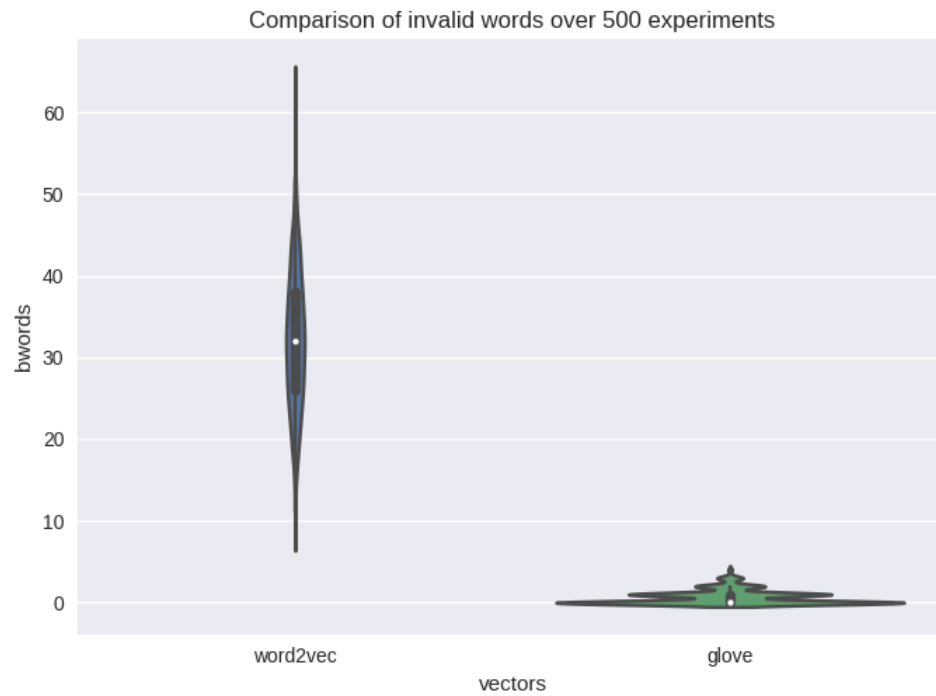
```
Out[20]:
```



Gibberish Words

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

```
Out[22]:
```



Synonyms

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

```
Out[21]:
```

