# Natural Language Understanding: Assignment 1

Christopher Sipola (s1667278)

17th February 2017

## Question 1

### Part *c*

The similarity scores are below.

| word 1 | word 2 | similarity |
|---|---|---|
| *house.n* | *home.n* | 0.813 |
| *house.n* | *time.n* | 0.824 |
| *home.n* | *time.n* | 0.818 |

No, the fact that *house.n* and *home.n* have the lowest similarity score does not reflect our intuition that they are the most similar of the three comparisons.

### Part *e*

The similarity scores for the *tf-idf* space are below.

| word 1 | word 2 | similarity |
|---|---|---|
| *house.n* | *home.n* | 0.644 |
| *house.n* | *time.n* | 0.578 |
| *home.n* | *time.n* | 0.544 |

Yes, this captures similarity better in that *house.n* and *home.n* are deemed most similar, reflecting our intuition.

## Part *f*

I performed a grid search over the parameters `learningRate` $\in \{0.01, 0.03, 0.05\}$, `downsampleRate` $\in \{0, 10^{-5}, 10^{-3}, 10^{-1}\}$, and `negSampling` $\in \{0, 5, 10\}$. The combination `learningRate = 0.05`, `downsampleRate = 0.001` and `negSampling = 10` performed best, with an accuracy of 0.037.

The second-best accuracy was quite a bit lower at 0.028. Of the 36 combinations, 19 had an accuracy of exactly zero. Run times were fairly consistent, with a range of 78 to 97.

`negSampling` seems to be the best predictor of performance: the average accuracy of all runs where `negSampling = 10` was 0.0118, while for `negSampling = 5` it was 0.0094 and for `negSampling = 0` it was exactly zero.[1]

## Part *g*

The similarity scores for the word2vec model are below.

| word 1 | word 2 | similarity |
|---|---|---|
| *house.n* | *home.n* | 0.576 |
| *house.n* | *time.n* | 0.192 |
| *home.n* | *time.n* | 0.380 |

The word2vec model seems to capture the similarities better then the *tf-idf* space. In particular, word2vec gives a similarity score for *house.n* and *home.n* that is much higher than the other two word comparisons. In contrast, with *tf-idf*, all the scores were much closer.

## Part *i*

The similarity scores for the LDA model are below.

| word 1 | word 2 | similarity |
|---|---|---|
| *house.n* | *home.n* | 0.428 |
| *house.n* | *time.n* | 0.058 |
| *home.n* | *time.n* | 0.348 |

The LDA model seems to capture lexical similarity better than *tf-idf*, but it's fair to say that it doesn't capture similarity as well as word2vec does. In particular, LDA gives a *house.n*-*home.n* similarity score that is just 23% higher than the *home.n*-*time.n* score, whereas the word2vec gives a *house.n*-*home.n* score that is 52% higher.[2]

[1] That is, for all runs where `negSampling = 0`, there were no successful predictions.

[2] I don't believe it is useful to compare scores across models. Rather, it seems more useful to looks at other comparisons made by the model/space.

### Part *j*

Most of the topics contain the same uninformative, generic words. For example, one topic that is representative of many other topics has the top 10 topic terms *have*, *do*, *other*, *make*, *will*, *get*, *go*, *would*, *say*, and *can*.

However, there are some topics that stand out. Here are some of those topics and their top words, along with my attempt at labeling them (in **bold**).

- **Energy:** *have*, *water*, *will*, *can*, *air*, *energy*, *do*, *light*, *time*, *get*
- **Education:** *have*, *do*, *social*, *school*, *education*, *art*, *course*, *study*, *science*, *year*
- **Government/politics:** *minister*, *have*, *secretary*, *affair*, *will*, *prime*, *ministry*, *government*, *foreign*, *deputy*
- **Prices/economics/policy:** *have*, *per*, *will*, *rate*, *would*, *pay*, *price*, *cost*, *tax*, *can*
- **War:** *united*, *have*, *nuclear*, *do*, *new*, *will*, *blood*, *waste*, *would*, *use*
- **Directions:** *have*, *south*, *north*, *west*, *east*, *country*, *england*, *go*, *road*, *other*

## Question 2

### Part *e*

The precision and recall scores are below.

| model | attempted | precision | recall |
|---|---|---|---|
| *tf-idf*, addition | 199 | 8.85 | 8.81 |
| *tf-idf*, multiplication | 199 | 13.96 | 13.89 |
| word2vec, addition | 198 | 12.14 | 12.02 |
| word2vec, multiplication | 167 | 5.95 | 4.97 |
| LDA | 199 | 8.03 | 7.99 |

### Part *f*

The best-performing model/space overall is *tf-idf* with multiplication, which has a precision of 13.96 and a recall of 13.89.

Interestingly, multiplication performs better than addition for *tf-idf*, but the reverse is true for word2vec. Although word2vec is a more sophisticated model than *tf-idf*, we chose the hyperparameters based on its accuracy on a different task—that is, one where the model had to predict the best analogy—and therefore may not perform as well as expected on the substitution task. The reason word2vec with multiplication may perform so poorly is that it may introduce *too much* sparsity, as evidenced by the fact that only 167 of the substitutions were attempted.[3]

LDA performs second-worst, after word2vec with multiplication. Perhaps the performance could be improved by choosing the hyperparameters using some form of validation.

[3] Although, of course, this could be due to a bug or error.