# Week 7 Learning Log

## Theory-based Challenges

* Research **graph-based approaches** to semantic representation.
* Consolidate understanding of **vector-based representations**.
* **Compare** and **contrast** both approaches; deduce which is preferred.

## Practical Challenges (Improve my NLP General Knowledge)

* Build **a Password classifier**, then update my how-to guide with steps for making this using Streamlit.
* Build an **Automated** **Text Summariser** using Gensim and/or Sumy, then **update my how-to guide** with steps to reproduce it.

## **Week 7**

This week, my learning log is divided into theory-based research by reading relevant papers and practical skills acquired through learning about new Python libraries and maths. I was researching how the LexRank and TextRank algorithms work and did not understand what an Eisenvector was initially, so I decided to investigate how Eisenvectors and Eisenvalues work as well.

I’ve decided to focus on graph-based representations of semantics this week because last week, all my research that is of value to the end goal of the topic was on vector-based represenations. My plan for this week describes how I found these papers. This document showcases what I learnt from reading them.

### Streamlit

Open source ML development framework for Python which lets me make a widget-based interactive application which can be launched locally. Widgets include text entry areas, buttons, subheaders. I chose to use this framework because of easy hosting/nice aesthetics.

### Automatic Text Summarisers

The ultimate goal of Automatic Text Summarisers is to generate abstracts for any document which address all its key points. There are two main types; extractive and abstractive. Extractive summarisers return a summary which only includes phrases from the source text. Abstractive summarisers may generate their own description of the source text. Each has pros and cons.

Pre-2017, abstractive summarisers coped well at summarising short documents but tended to produce repetitive and incoherent phrases. In 2017, Salesforce researchers published their deep learning reinforced model for abstractive summarisation. Its paper has been cited over 400 times, making it reliable.

### Exposure Bias, Availability Bias and Confirmation Bias – why do these have implications on representing the semantics of text?

Supervised learning runs the risk of causing exposure bias; the idea that the machine expects to be trained on the truth at each stage. Availability bias causes a decision maker to favour view A over view B because they have more information about view A, even if view B equally or more valid but the information is harder to obtain. Confirmation bias is when a decision maker favours view A over view B because view A proves a hypothesis/belief that they have held. Unless a machine is trained on an unbiased data set, the understanding of the training data may be inherently biased which will lead to the machine drawing unfair conclusions about a subject, then making a decision which is equally unfair. This is best described by Dr. Krishna Gummadi who spoke at the 2019 BCS Turing Talk in London on the importance of training unbiased AI in relation to Latent Text Representations. He identified a key issue with the existing Word2Vec; when you run “A is to X as B is to ?” tests on the vectors, sexist results like “Man is to computer as woman is to housekeeper” appeared in abundance. He also stated that the biases in these multidimensional representations are very difficult for humans to spot.

### LexRank vs TextRank

These ranking algorithms were developed by different teams at the same time and have a similar purpose. One key distinction between them is that TextRank was made for single-document summarisation whereas LexRank can also perform multi-document summarization.

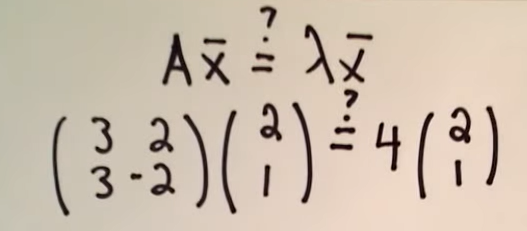
### Graph-based Survey Paper – 2013, Hansung University

This paper describes the format of a graph-based representation: nodes and edges where a node an be homogenous (represents one component of a sentence) or heterogenous (represents two or more components of a sentence). Unlike a vector-based model, the order of appearance of words can be considered. Nodes can also either be weighted or unweighted. These weights indicate importance within the graph.

### Eigenvectors and Eigenvalues

An Eigenvector is a non-zero vector which gives a scalar multiple of itself when an operation is performed on it. It does not change direction during a transformation. <http://setosa.io/ev/eigenvectors-and-eigenvalues/>

The website I’ve linked offers an interactive demonstration of the concept in terms of Adult and Child Amoeba. This fortifies my belief that interactive how-to guides are most effective. The best way of explaining Eigenvalues and Eigenvectors is through a simple Matrix Example.



In this question, A is the matrix, x bar is the Eisenvector and lambda is the Eisenvalue. The objective is to find an Eisenvalue such that Eisenvector \* Matrix = Eisenvector \* Eisenvalue.

Do Matrix multiplication:

(3\*2)+(2\*1) = 8

(3\*2)\*(-2\*1) = 4

Find a multiplier which changes the RHS into the simplified LHS. In this case, 4.