Data Collection and Cleaning Coursework Report

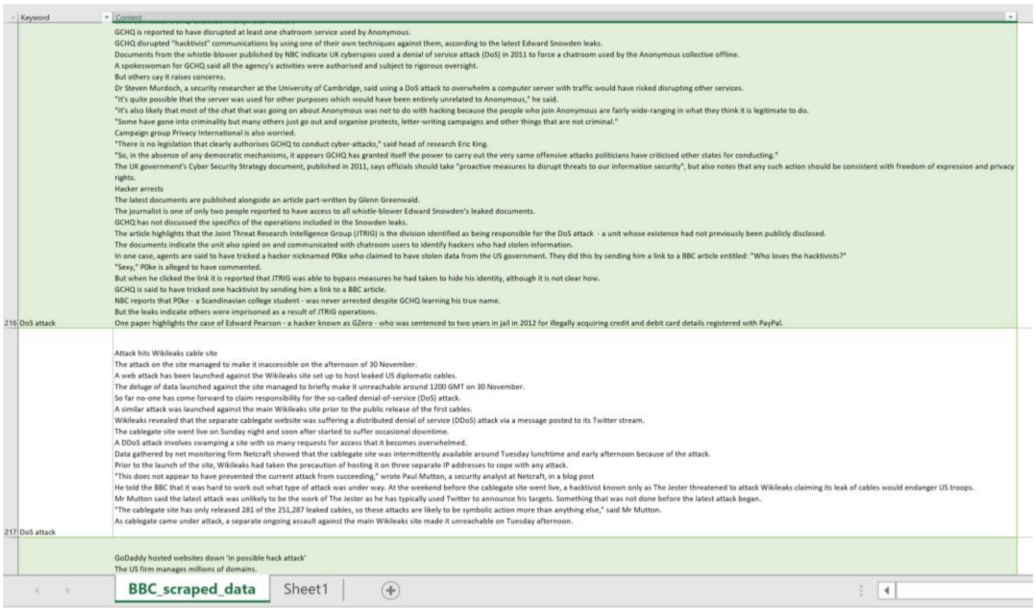
Problem 1 and 2- requests and web scraping:

For this problem it takes about 2 mins 40 seconds for my program to run.

I decided to code problem 1 and 2 together to increase efficiency, as once I had requested information from webpages it was easy to then store them in a suitable data structure.

After reading in keywords from keywords.xlsx I used the bbc search bar to find article links most relevant to each keyword. I took the first 100 (or as many as were present) relevant articles. I filtered links to make sure they were news articles and were actually links to the bbc website as opposed to a different site. I also filtered scraped articles, ensuring they were relevant by testing if they had the keyword or a variation of it. However, to stop a massive increase in runtime, I stopped filtering for the keyword if my program had scraped 5 pages of articles without finding a relevant one.

I created 5 different functions to scrape all relevant content (including things like figure captions but not links to other parts of the website) from articles of different time periods that had different formats, making sure to preserve the order of text as this was important for my algorithm in problem 3. I used multithreading to increase the speed of my program, as this allowed making requests (the main bottleneck of the program) in parallel. I also used try-except blocks to deal with errors and implement a set number of retries if a request was unsuccessful. I saved article content to a csv file, each row containing the keyword, and article content, including the article title.



	Keyword	Content
216	DoS attack	GCHQ is reported to have disrupted at least one chatroom service used by Anonymous. GCHQ disrupted "hacktivist" communications by using one of their own techniques against them, according to the latest Edward Snowden leaks. Documents from the whistle-blower published by NBC indicate UK cyberspies used a denial of service attack (DoS) in 2011 to force a chatroom used by the Anonymous collective offline. A spokeswoman for GCHQ said all the agency's activities were authorised and subject to rigorous oversight. But others say it raises concerns. Dr Steven Murdoch, a security researcher at the University of Cambridge, said using a DoS attack to overwhelm a computer server with traffic would have risked disrupting other services. "It's quite possible that the server was used for other purposes which would have been entirely unrelated to Anonymous," he said. "It's also likely that most of the chat that was going on about Anonymous was not to do with hacking because the people who join Anonymous are fairly wide-ranging in what they think it is legitimate to do. "Some have gone into criminality but many others just go out and organise protests, letter-writing campaigns and other things that are not criminal." Campaign group Privacy International is also worried. "There is no legislation that clearly authorises GCHQ to conduct cyber-attacks," said head of research Eric King. "So, in the absence of any democratic mechanisms, it appears GCHQ has granted itself the power to carry out the very same offensive attacks politicians have criticised other states for conducting." The UK government's Cyber Security Strategy document, published in 2011, says officials should take "proactive measures to disrupt threats to our information security", but also notes that any such action should be consistent with freedom of expression and privacy rights. Hacker arrests The latest documents are published alongside an article part-written by Glenn Greenwald. The journalist is one of only two people reported to have access to all whistle-blower Edward Snowden's leaked documents. GCHQ has not discussed the specifics of the operations included in the Snowden leaks. The article highlights that the Joint Threat Research Intelligence Group (JTRIG) is the division identified as being responsible for the DoS attack - a unit whose existence had not previously been publicly disclosed. The documents indicate the unit also spied on and communicated with chatroom users to identify hackers who had stolen information. In one case, agents are said to have tricked a hacker nicknamed P0ke who claimed to have stolen data from the US government. They did this by sending him a link to a BBC article entitled: "Who loves the hacktivists?" "Sexy," P0ke is alleged to have commented. But when he clicked the link it is reported that JTRIG was able to bypass measures he had taken to hide his identity, although it is not clear how. GCHQ is said to have tricked one hacktivist by sending him a link to a BBC article. NBC reports that P0ke - a Scandinavian college student - was never arrested despite GCHQ learning his true name. But the leaks indicate others were imprisoned as a result of JTRIG operations. One paper highlights the case of Edward Pearson - a hacker known as GZero - who was sentenced to two years in jail in 2012 for illegally acquiring credit and debit card details registered with PayPal.
217	DoS attack	Attack hits Wikileaks cable site The attack on the site managed to make it inaccessible on the afternoon of 30 November. A web attack has been launched against the Wikileaks site set up to host leaked US diplomatic cables. The deluge of data launched against the site managed to briefly make it unreachable around 1200 GMT on 30 November. So far no-one has come forward to claim responsibility for the so-called denial-of-service (DoS) attack. A similar attack was launched against the main Wikileaks site prior to the public release of the first cables. Wikileaks revealed that the separate cablegate website was suffering a distributed denial of service (DDoS) attack via a message posted to its Twitter stream. The cablegate site went live on Sunday night and soon after started to suffer occasional downtime. A DDoS attack involves swamping a site with so many requests for access that it becomes overwhelmed. Data gathered by net monitoring firm Netcraft showed that the cablegate site was intermittently available around Tuesday lunchtime and early afternoon because of the attack. Prior to the launch of the site, Wikileaks had taken the precaution of hosting it on three separate IP addresses to cope with any attack. "This does not appear to have prevented the current attack from succeeding," wrote Paul Mutton, a security analyst at Netcraft, in a blog post He told the BBC that it was hard to work out what type of attack was under way. At the weekend before the cablegate site went live, a hacktivist known only as The Jester threatened to attack Wikileaks claiming its leak of cables would endanger US troops. Mr Mutton said the latest attack was unlikely to be the work of The Jester as he has typically used Twitter to announce his targets. Something that was not done before the latest attack began. "The cablegate site has only released 281 of the 251,287 leaked cables, so these attacks are likely to be symbolic action more than anything else," said Mr Mutton. As cablegate came under attack, a separate ongoing assault against the main Wikileaks site made it unreachable on Tuesday afternoon.
		GoDaddy hosted websites down 'in possible hack attack' The US firm manages millions of domains.

BBC_scraped_data | Sheet1

Problem 3- semantic distance:

In my python file I have combined the code for problem 3 and 4, as to carry out dimensionality reduction to visualize vectors in 2d tSNE needs access to all the word

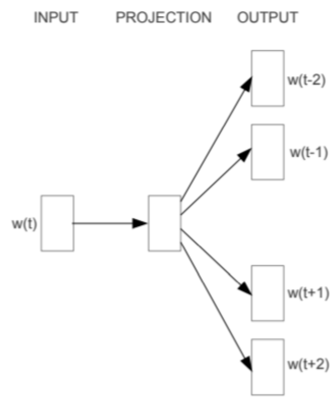
vectors created by Word2Vec. Combining code prevents me having to save a large amount of vectors in a file and read this in for problem 4. Together problem 3 and 4 take about 1 min 20 seconds to run.

To calculate semantic distances between keywords I decided to use word embedding with Word2Vec. In word embedding, words are mapped to vectors of real numbers with many dimensions. The Word2Vec python library uses a two layer neural network (1 input, 1 hidden and 1 output layer) to generate word embeddings. Unlike some word embedding approaches (e.g. bag of words and IF-IDF) Word2Vec retains the order of words and context information, and therefore their semantic meaning.

It is worth noting there are other word embeddings (e.g. GloVe) that work in a similar way to Word2Vec. However differences in accuracy vary per dataset, and all are usually very good at capturing semantics. I decided to use Word2Vec due to the large amount of literature pertaining to it, making it easier to understand and use.

To increase the amount of data for the model, I also used requests and beautiful soup to scrape the Wikipedia pages of each of the keywords. Before creating the Word2Vec model I pre-processed the data scraped from the bbc news and wikipedia articles. I convert articles to lowercase and substitute out any characters that aren't letters so I am left with only words. Word2Vec works on single words, so in the next pre-processing step I converted any multiword keywords to single words (e.g. targeted threat -> targeted_threat) so they are usable in the model. In this step I also convert any plural forms of keywords to singular, so these are also counted in the model (e.g. malicious bots -> malicious_bot). Finally, I remove stopwords, although Word2Vec down-samples frequent words automatically I did not think stopwords were relevant for finding keyword semantic distance.

I then convert these articles into lists of words and use these to train my Word2Vec model. I decided to use articles instead of individual sentences to train the model, as many articles were not clearly split up into sentences.



Skip-gram

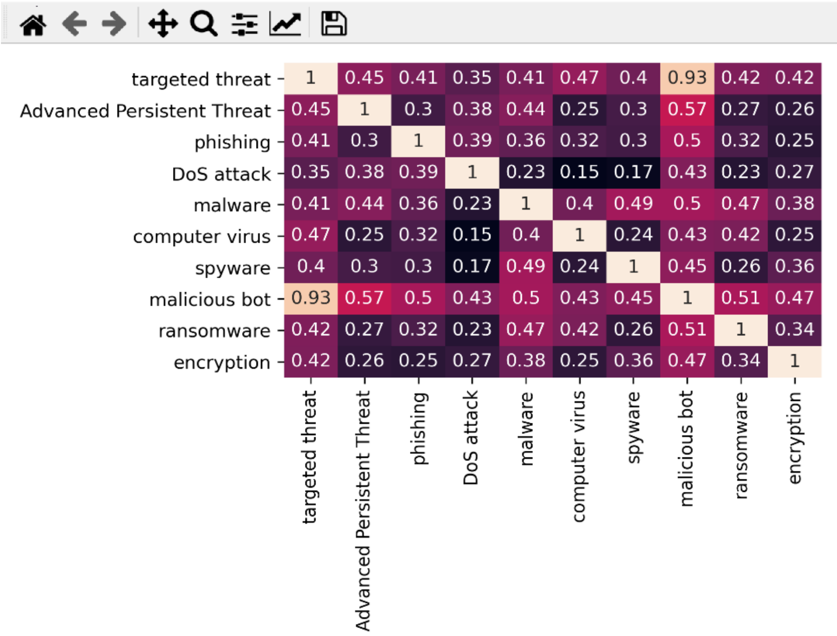
I used the skip-gram architecture in my model, which predicts surrounding context words in a specific window given an input word. Some of the keywords are not common in the articles, and skip-gram works better than CBOW architecture for rare words or phrases. I also use a vector size of 50, as this helps with dimensionality reduction for problem 4.

The Word2Vec model puts words in similar contexts closer to each other in the vector space, words with high semantic similarity will have more similar vectors. I used cosine similarity of keyword vectors (between 0 and 1) to measure how semantically similar they were. Cosine similarity works well even when inputs are of different sizes, it will still be an accurate measure even between keywords that occur at very different rates, unlike Euclidean distance.

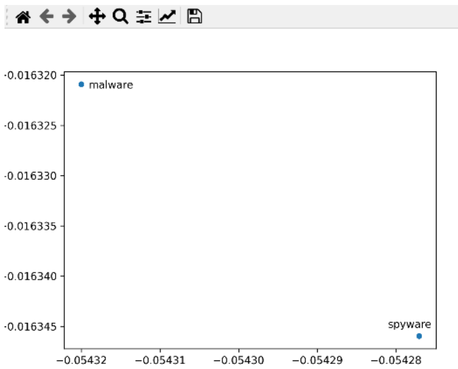
This part of the problem takes about 40 seconds to run.

Problem 4- visualization:

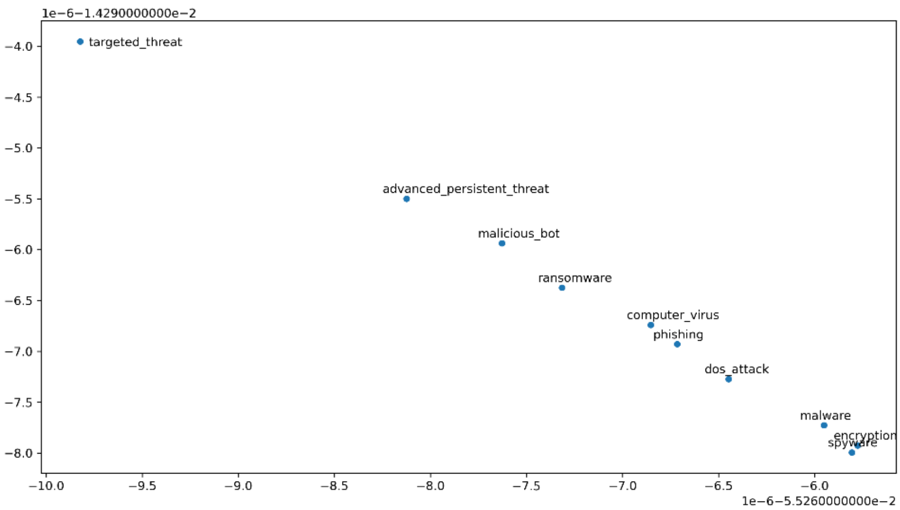
In problem 4, I first used seaborn to plot a heatmap with the cosine similarity of all keywords. This was an easy way to see how closely related all keywords were, with the cosine similarity values between 0 and 1 also on the heatmap:



To get a more visual representation of distances between vectors, I carried out dimensionality reduction with t-SNE. This compresses the 50 dimensional vectors created by the Word2Vec model into 2 dimensional vectors, whilst still keeping similar words close together, and dissimilar words further away. I make sure to set the metric = cosine, as this is the similarity measure I used in problem 3. This will prioritize preserving cosine similarity in the dimension reduction, not Euclidean similarity.



I first plot just 2 keywords to visualize distances on a smaller scale, before then creating a plot with all the keywords:



This part of the problem takes about 40 seconds to run.