## Pre-processing

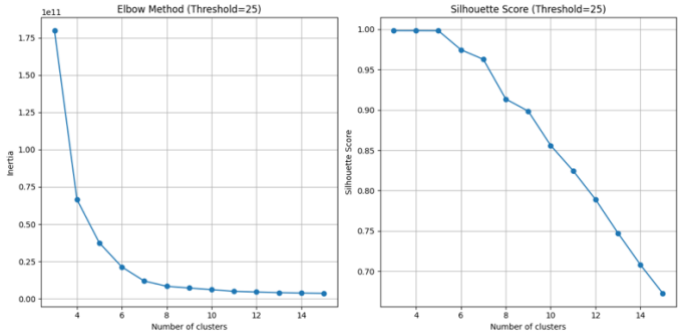
1) Remove words shorter than 2 characters. Short words don't add anything meaningful and only serve as a reading aid. This value is configurable in the provided code (default value is 2). 2) Remove stopwords[1]. These common words ('the', 'are', 'was' etc.) do not carry much meaning. 3) Remove words that appear less than 25 times. This threshold is also configurable in the provided code (default value is 25), and was found by testing values [10, 15, 20, 25, 40, 50, 75]. 4) Create cooccurrence matrix with window size 7. A window size of 7 seemed an appropriate balance between preserving the semantic meaning of the text and performance. This value is also configurable in the provided code (default value is 7). This reduced the dataset from 17005207 words to 9882206: a 41.89% reduction. The co-occurrence matrix was of shape (27350, 27350).

## K-means

I tested $3 \leq k \leq 15$. I used *j*-fold cross-validation[2] to thoroughly evaluate the model. For each *k* value, I created and trained a new *k*-means model (KMM) on the *j*-1 training folds from the pre-processed co-occurrence matrix. I predicted the clusters of the test fold and used this to calculate the inertia and silhouette score for each *k* value. The scores and inertias were averaged over the *j* folds. The range of *k*s (default is [3, 15]) and the number of folds (default is 5) are configurable in the provided code.

I used the Elbow Method which uses the KMM's inertia attribute, describing how "tightly" clustered data points are to their respective centroids, to evaluate performance. A lower inertia means better clustering. I also used the silhouette score metric to measure the model's performance which considers the intra- and inter-cluster distances. A higher silhouette score means better clustering.

While the graph below shows that the lowest inertia is at k=15, using the principle of Occam's Razor, we can see that the simplest with an adequate, low result is k=6. After this point, the inertia begins to drop slowly showing the diminishing returns from analysing additional clusters. The silhouette score for k=6 is also almost maximum (~0.975), which agrees with the Elbow Method result.




During this coursework, I tried various ways of vectorising the dataset for use with the KMM, including Word2Vec, TF-IDF and GloVe. While these likely would've increased the clustering performance, they massively increased the complexity of the model; so much so that when trying to run the model using these methods it either ran out of memory or took an extremely long amount of time. The decision to use a co-occurrence matrix as a method of vectorisation provided sufficient performance whilst being relatively simple. This trade-off between model complexity and performance is at the heart of the Occam's Razor principle.

In the context of machine learning, the principle of Occam's Razor suggests choosing the simplest model that yields a satisfactory explanation of the data. This can help us to prevent or reduce overfitting and produce a model that generalises and performs better on real-world data. The *simplest* model is where k=1. Obviously, this provides little to no insight into the data patterns, and we are looking for the simplest model (lowest k value) that provides an accurate explanation of the data. As shown above, the lowest value of k that provides satisfactory results is **k=6**. Without Occam's Razor, we might incorrectly choose k=15. While this does result in better inertia, a k this large needlessly complicates the model, resulting in diminishing returns and would be prone to overfitting.

[1] The NLTK Python library was used for stopword removal.

[2] K-fold cross-validation was renamed to j-fold to avoid confusion with the k value in k-means.