Report for Voiceline clulstering

Leong Eu Jinn

September 2023

1 Introduction

This report provides a summary of the work conducted on implementing a clustering method of Voiceline snippets

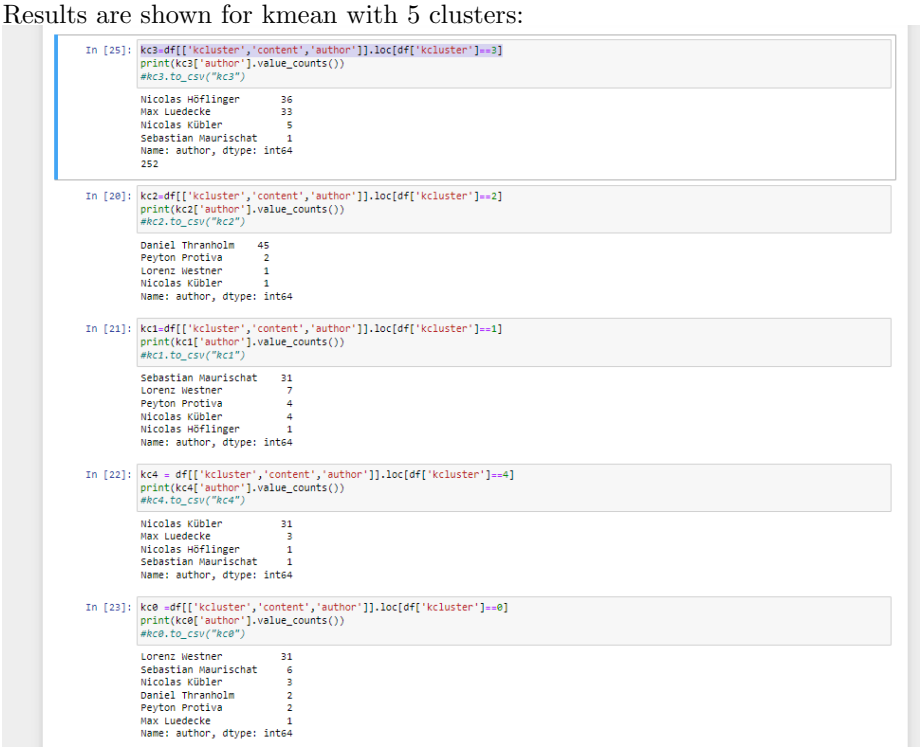
- Problem Description: tasked to program and train an unsupervised model to cluster voiceline snippets
- Outline of solution: Initial embedding of voiceline snippet contents using various models such as Llama-7 or openAI's ada-002. Following it various methods are used to cluster voiceline snippets. DBScan and k-means clustering was used initially to test which clustering would make more sense and can be expressed visually with clarity.

2 Methods

The process of embedding the voiceline snippets will be made with state-of-the-art models such as openai's ada-007 or llama-7b.
Initial testing with huggingface and open source llama models result in failure due to hardware failure. After that, openai's ada-007 API was used. Embeddings with dimension 1536 were created for each sentence.
Clustering methods were then applied to the data. Due to the embedding dimensions being too high, dimensionality reduction methods were employed to enable visualisation.

3 Results and Discussion

Results are shown for kmean with 5 clusters:



```
In [25]: kc3=df[['kcluster','content','author']].loc[df['kcluster']==3]
         print(kc3['author'].value_counts())
         #kc3.to_csv("kc3")

         Nicolas Höflinger       36
         Max Luedecke            33
         Nicolas Kübler           5
         Sebastian Maurischat     1
         Name: author, dtype: int64
         252

In [20]: kc2=df[['kcluster','content','author']].loc[df['kcluster']==2]
         print(kc2['author'].value_counts())
         #kc2.to_csv("kc2")

         Daniel Thranholm    45
         Peyton Protiva       2
         Lorenz Westner       1
         Nicolas Kübler       1
         Name: author, dtype: int64

In [21]: kc1=df[['kcluster','content','author']].loc[df['kcluster']==1]
         print(kc1['author'].value_counts())
         #kc1.to_csv("kc1")

         Sebastian Maurischat    31
         Lorenz Westner           7
         Peyton Protiva           4
         Nicolas Kübler           4
         Nicolas Höflinger        1
         Name: author, dtype: int64

In [22]: kc4 = df[['kcluster','content','author']].loc[df['kcluster']==4]
         print(kc4['author'].value_counts())
         #kc4.to_csv("kc4")

         Nicolas Kübler          31
         Max Luedecke             3
         Nicolas Höflinger        1
         Sebastian Maurischat     1
         Name: author, dtype: int64

In [23]: kc0 =df[['kcluster','content','author']].loc[df['kcluster']==0]
         print(kc0['author'].value_counts())
         #kc0.to_csv("kc0")

         Lorenz Westner          31
         Sebastian Maurischat     6
         Nicolas Kübler           3
         Daniel Thranholm         2
         Peyton Protiva           2
         Max Luedecke             1
         Name: author, dtype: int64
```

I could not find any similarity topicwise between the machine's clustering but it did cluster the messages that correlates with the data's author.

4 Final notes

- Resources used: pandas,json,torch,openai,numpy,sklearn and initial given data
- Suggestions: A better method for unsupericsed learning to model topics would be to do it with LDA or with a supervised learning method with a labelled data set.