MA5851 A3 Assessment Report Part Three (NLP Tasks)

Student: Sacha Schwab

Location:Zurich, Switzerland

Date: 6 December 2021

Github link to repository: https://github.com/sachaschwab/NLP-Clustering (https://github.com/sachaschwab/NLP-Clustering)

Note:

Figures (images) are not correctly loaded by github file; please use the documents submitted on JCU Learn in the assessment submission folder

1. Preliminary thoughts

Basic task: Group articles into events they report about.

- Extract keyword and named enttiies features (Li et al., 2021, p.1).
- Clustering

Challenges:

- Titles often do not fully represent the event
- Side events in same article
- Different classes of articles e.g. regulatory, acquisitions etc.
- Verb-argument patterns expected to be not applicable (Rusu et al., 2014)

2. Literature Review

The literature pertaining to event clustering is diverse insofar as it reflects a variety of

- definitions of "event", e.g. "Actor-Action-Object-Time" (e.g. in Xing et al., 2018), or a "who, when, where, what, why, how" syntax (see Xiang and Wang, 2019), or syntax schemes according to the event category under investigation (e.g. "attacker-target-instrument-time-place", see Li et al., 2021, p.2)
- 'event extraction' emerged to a separate study field with a diverse range of approaches (see below; overview see see Xing et al., 2018).;

Li et al. (2021) investigated the approaches applied by the literature, and divided them into 3 groups, which are: (1) pattern-matching, (2) machine learning, and (3) deep learning. They also state that the recent work focuses on combinations of deep learning techniques.

As to clustering, Goya et al. (2018, p. 24) find that a combination of named entities and keywords improves the clustering quality. In particular, named entity recognition has shown a remarkable improvement for clustering.

Capdevila et al. (2016, p. 1 f) and further resources describe DBSCAN constitute a well-known approach to event detection due its noise resilience capability.

Cao et al., 2012 conclude from their research that a weighted combination of named entities and keywords are significant to clustering quality.

3. Approach

- Identify named entities and their importance;
- Identify keywords using word embedding and taking the words with x top weights;
- Sentiment analysis;
- Normalise numerical features from the 3 steps above (i.e. entities weights, keyword weigths, sentiment scores)
- Apply clustering technique;
- Measure the performance of the model variations.

Algorithms applied:

- Named entity extraction and relations: NLTK
- Embeddings: TF-IDF
- For clustering: DBSCAN (see indications in the literature).

4. Data Wrangling

Approach

In this part the harvested data generated by the web crawler is preprocessed and tokenised for further use for NLP tasks.
Data preprecessing follows a rather standard approach as per literature indications, learning materials in MA5851, and online tutorials. It includes lower-casing, erasure of one-character words and symbols as well lemmatization. The latter is applied knowing that it is slower, however appears to make more sense since it better reflects natural speech than stemming (Jabeen, 2018).

Summary and visuallisation of the data

Reference is made to section (k) of the submitted report for part 2 of this assessment.

5. NLP Tasks

Embedding / Vectorisation

Since TF-IDF computes the importance of a term in a document (taking all other documents in the corpus into account), it appears appropriate for finding keywords (see e.g. Ellis, 2019). The TfidfVectorizer from NLTK package was employed for the purpose of producing TF-IDF vectors.

Keyword extraction

For each document - every word in the tokenised text is looked up in the TF-IDF matrix, - its weight collected and put into a dataframe for sorting, - evaluate the weight value of the (top) x'st word, and - drop all other words. The two arrays (top x words and their weights) are returned for their use in the clustering task below.
Since the project has cryptocurrencies in its focus, the list of cryptocurrency names fetched by crawling with Selenium was leveraged for this purpose, providing an 'artificial' important weight (0.5) to these when occurring in the text.

Named Entities

For this rather complex task a number of pre-trained models can be employed. The Spacy framework was applied with satisfactory results. The "select top x" approach as per above was employed.
Spacy unfortunately does not recognise cryptocurrencies as 'currency' entity. This gap was filled by leveraging the web-crawled list of cryptocurrenccies.

Sentiment Analysis

For this NLP task a rather simplistic approach was selected for efficiency reasons, i.e. application of the VADER sentiment analysis (SentimentIntensityAnalyzer from NLTK), which was run for every document.
The resulting compound score is the sum of positive, negative and neutral scores, adn these are normalized between -1 (highly negative) and +1 (highly positive).

NLP Task Output

Output from keyword extraction

HDBSCAN output

Figure: Frequency distribution of extracted entities

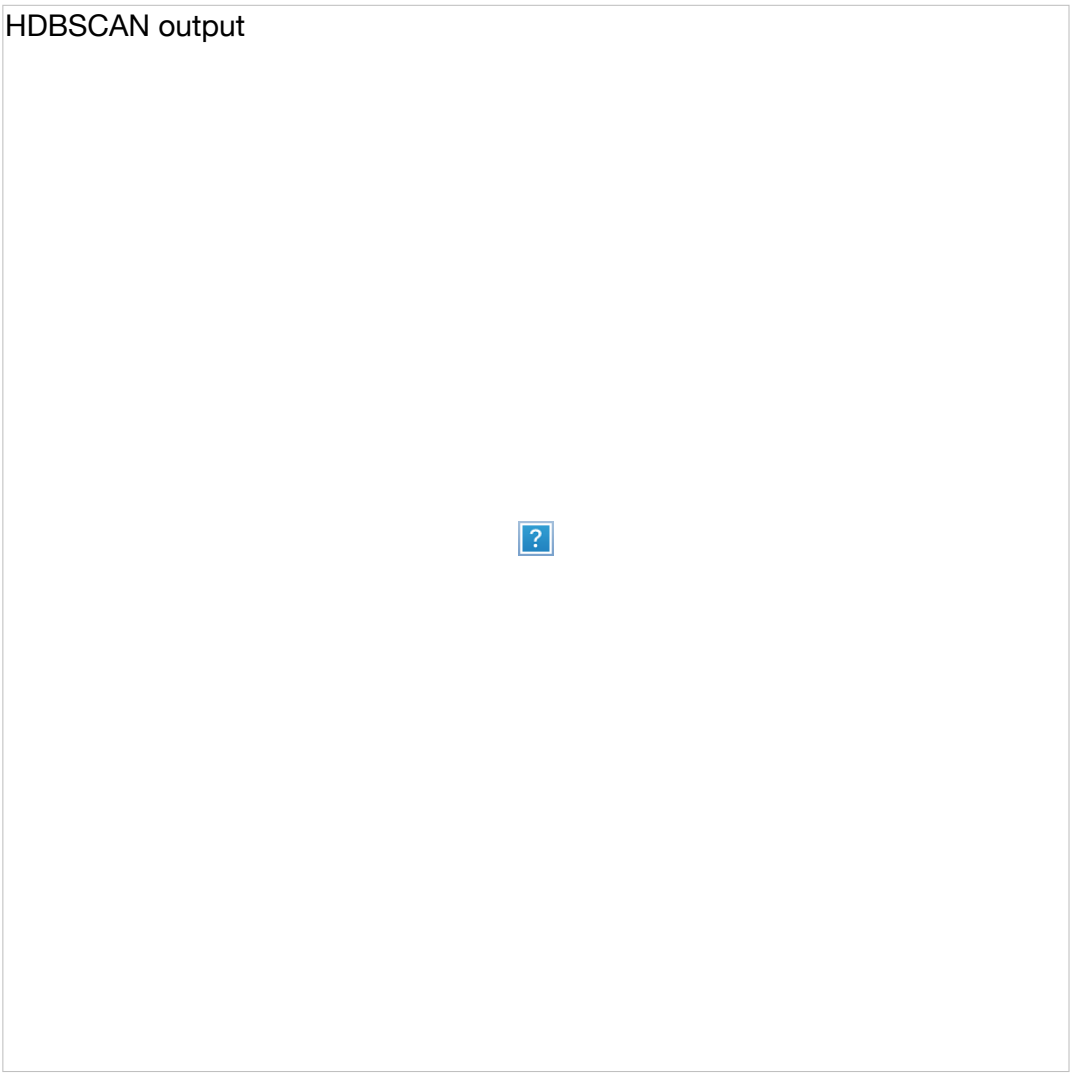


Figure: Frequency distribution of extracted entities

More work needs to be done: (a) non-english stopwords ("el", "de" etc.) and (b) frequent unwanted concatenations happened

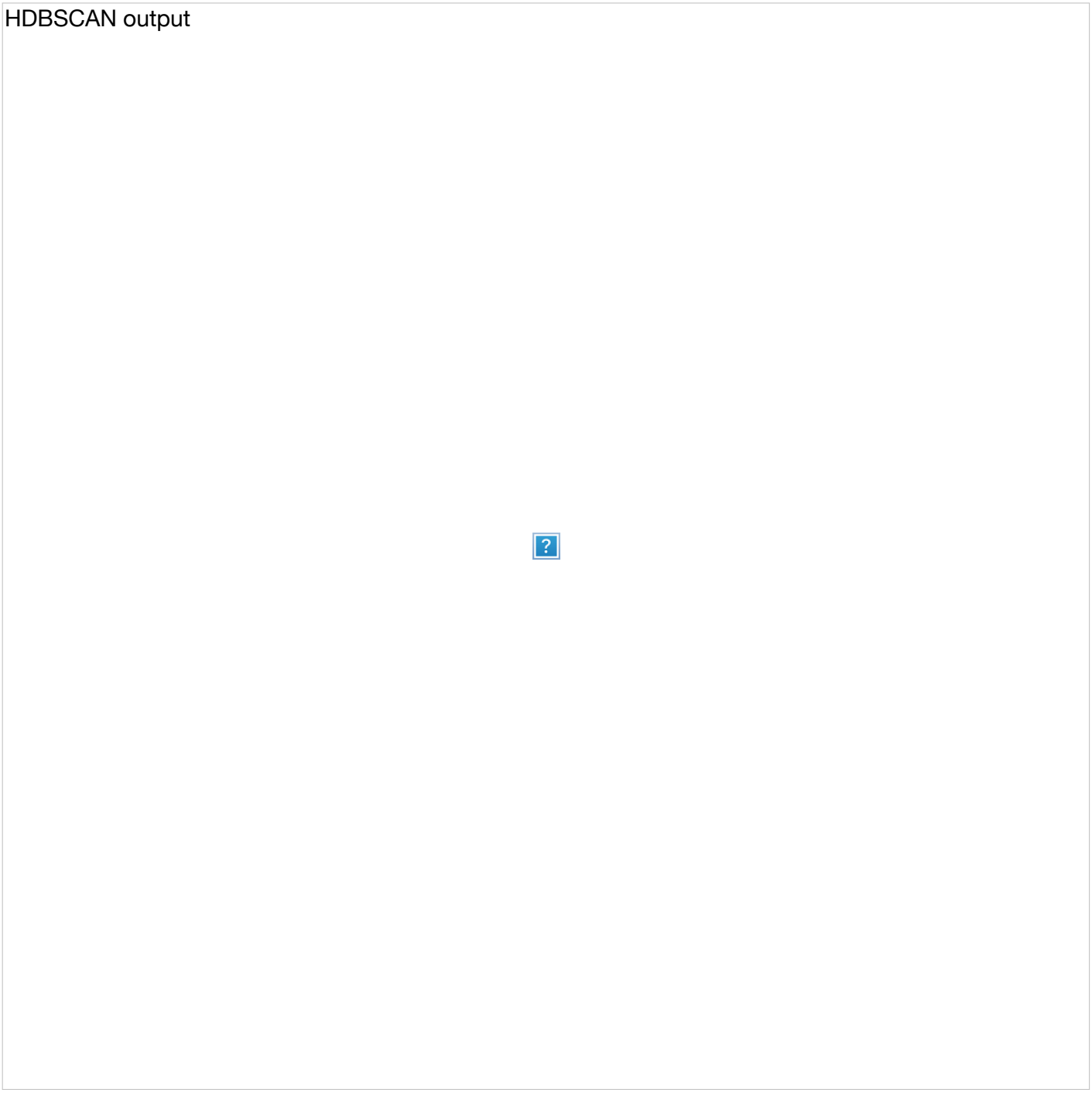
Output from entity extraction

Also here the output could be more meaningful. The distribution of weights indicates that the entities' importance is generally rather inferior, i.e. the entities, as to be observed above, are not represented among the keywords as it might be expected.

HDBSCAN output

Figure: Frequency distribution of extracted entities

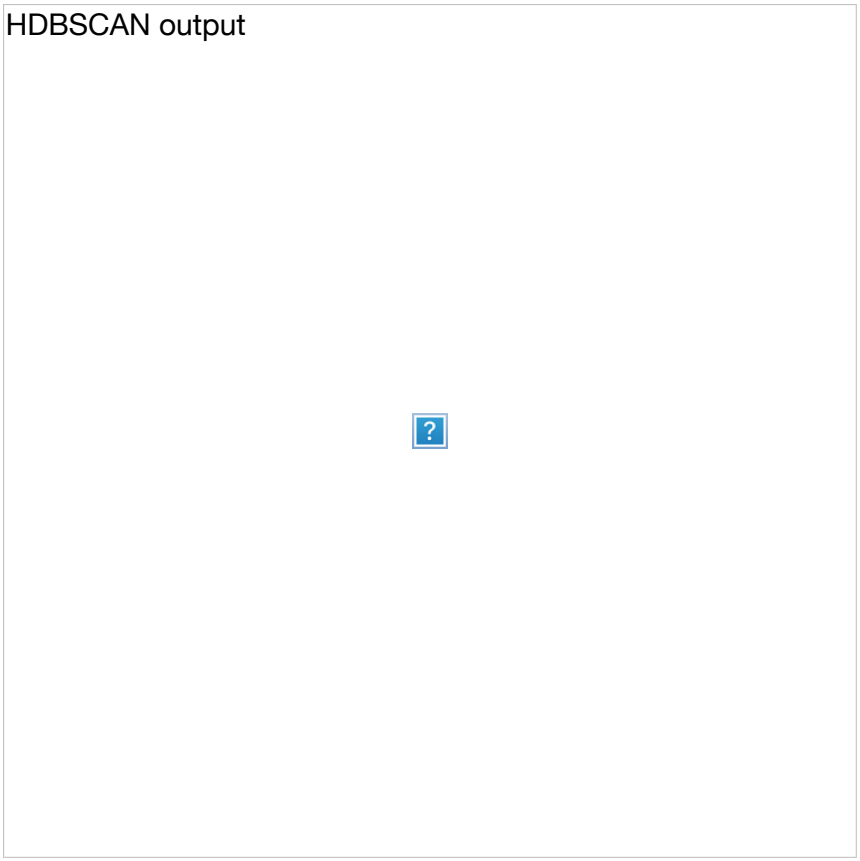
HDBSCAN output



</img>

Output from sentimeent analysis

Provided sentiment scores are unequally distributed: The sentiment representations of titles and body texts do not match at all.



7. Machine Learning

DBSCAN

Variations of eps and the min_samples parameters were run, with the normalised 10 top keywords and entities, and the sentiment scores served as input.
Result: DBSCAN would not present meaningful results: The algorithm would recognise either a cluster label for every article, or find only noise and one label (0).

Conclusion: DBSCAN cannot be employed for the task at hand.

HDBSCAN

Parameter variations for HDBSCAN provide results that are more in the space of expectations (see above considerations), i.e. a rather high level of noise and a rather small number of actual clusters as shown in the visualisation below.
Based on preliminary checks, variations were applied to parameters leaf_size (20 and 80) and min_samples (none and 1).

HDBSCAN output

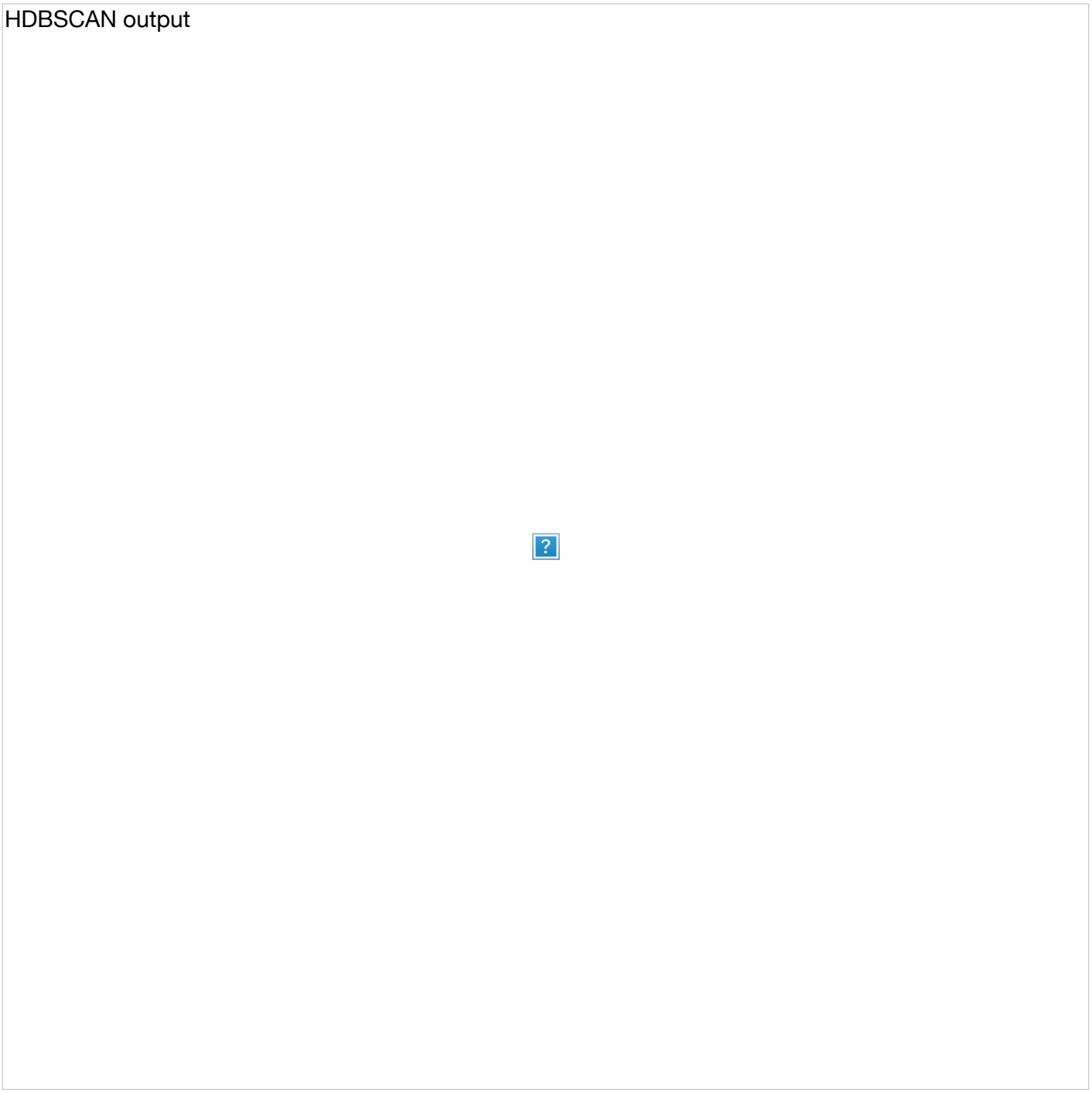


Figure: Resulting cluster distribution from HDBSCAN

Also, the silhouette scores are not in a satisfactory range:

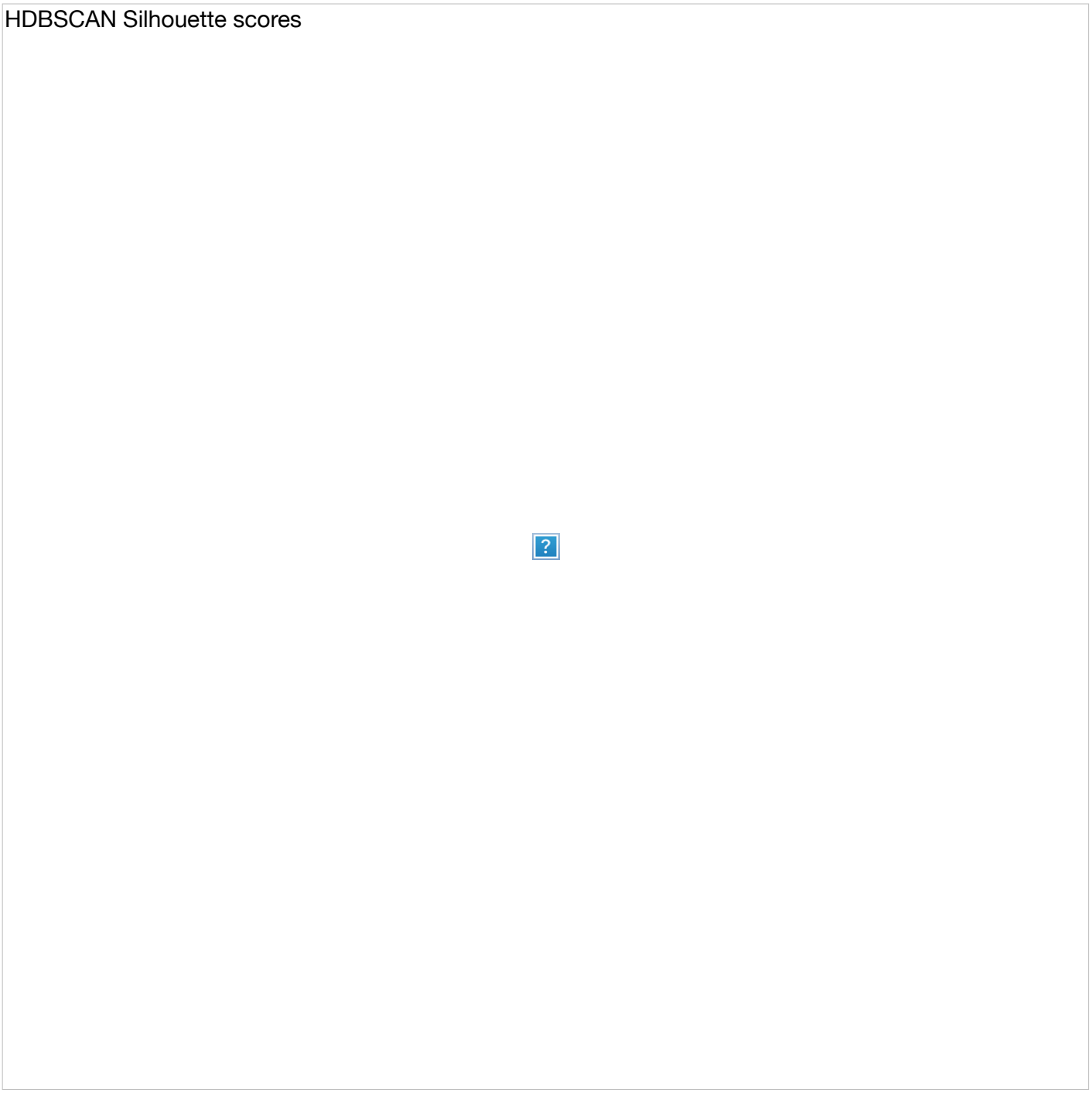


Figure: Silhouette scores obtained

Predictions ("Show case")

The following shows the result for a sample taken from the above used data themselves. It is not very promising (only 2 of 4 results match), however a valid start for further iterations.

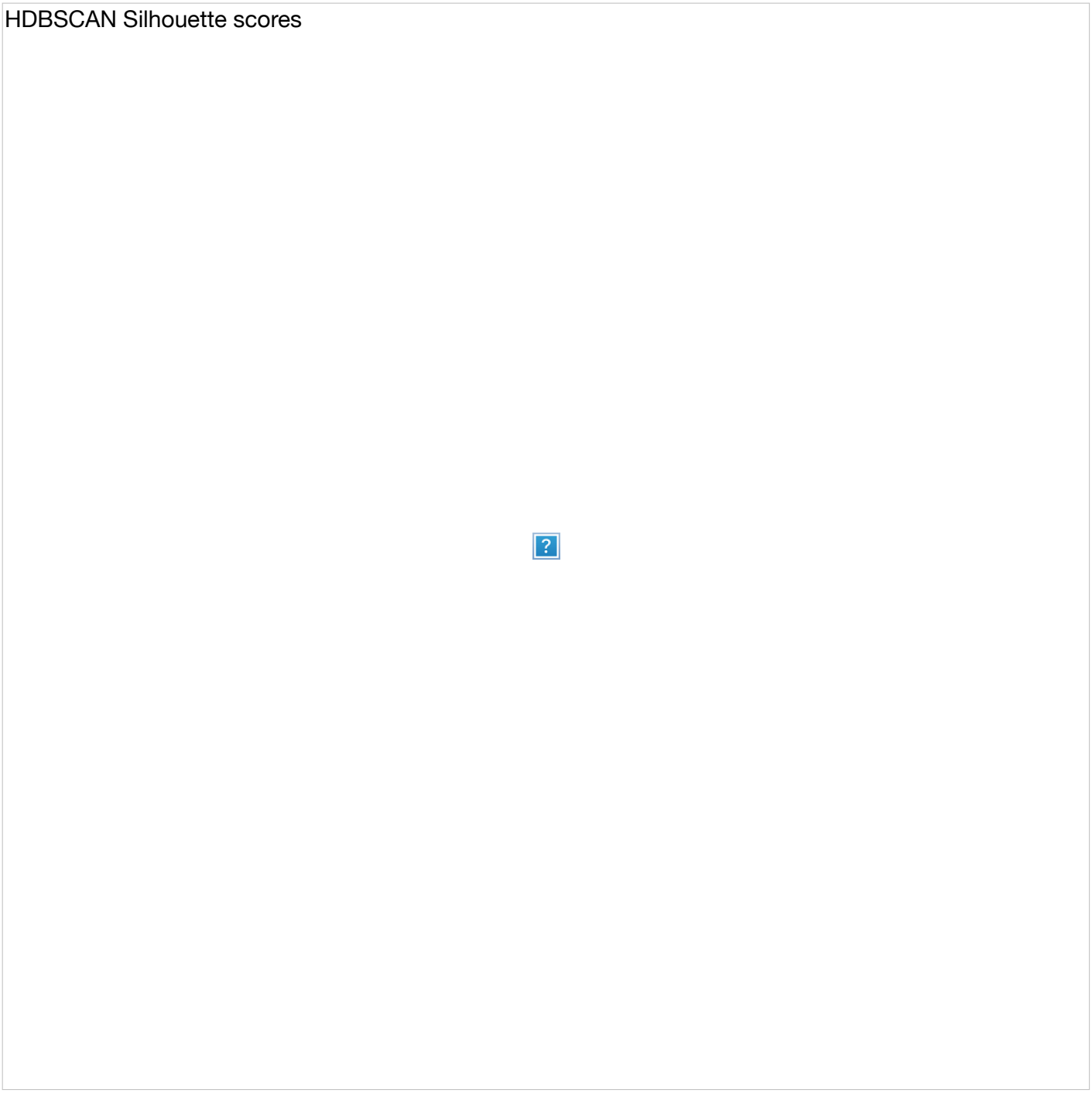


Figure: HDBSCAN cluster match result

Conclusion

More work needs to be invested into extraction and handling of features. For clustering, an alternative approach needs to be considered.

Code

The code for this part of the assessment is available in the seperate file "A3_DocumentNumber_2_Code_sacha_schwab.ipynb".