Honours Thesis

Clustering The Breakthrough Method

by

Gerard van Konijnenburg

(gkg410)

Supervisor: Sandjai Bhulai

August 9, 2023

Submitted for the Honours Project of the Bachelor Business Analytics

Clustering The Breakthrough Method

Gerard van Konijnenburg
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
G.van.Konijnenburg@student.vu.nl

ABSTRACT

Text-based clustering is a method that can be useful to group complex social care cases in the Netherlands. In this research, a bag-of-words approach is used with four different clustering methods to attempt to find a meaningful clustering method for the cases in the Breakthrough Method dataset. The clustering methods that are used are Affinity Propagation, Agglomerative Clustering, DBSCAN and Spectral Clustering.

1 INTRODUCTION

In the Netherlands, social care is the responsibility of the municipalities. Social care is a broad range of services designed to help people who need assistance in their daily lives due to physical, mental or emotional challenges. Some examples of these services are assistance with bathing or cleaning a house, transportation for those who are not able to use public transport, temporary use of a wheelchair, counseling on various topics like raising children, dealing with addictions, solving debts. Municipalities have delegated the responsibility of providing social care to various institutions that each have their own speciality or specialities. In addition to formalized social care, informal social care also exists, where support is given through a social network. In the Netherlands, an estimated 20% of all families ask for formalized social care[1].

Approximately 25% of the families that require formalized social care face more than 10 issues and have to deal with several[1] institutions. These institutions can have conflicting goals, an example is that a family councilor might advice a mother to spend more time with her children, while the unemployment benefit advisor requires her to search for a job[1]. In these cases a more customized approach can be required. The Instituut voor Publieke Waarden developed a method for giving this customized approach; the breakthrough method. This method specifies several questions that the professional(s) that assist the family can answer to better help the family[1].

One of the benefits of this method is that most questions have to be answered in text. This prevents a 'tick-the-box' mentality when answering the question and improves reflection on the situation and requirements of the family. However, this format makes it harder to aggregate data and learn from similar cases. As an example, the Insituut voor Publieke Waarden might want to generate a template for specific cases when they see that similar cases are frequent. By doing that, they can reduce the time spend on each case and also improve the solutions for the families, so that more families can be helped. Therefore, there is a use-case to group multiple cases.

In machine-learning, grouping items is the domain of clustering and classification. In clustering, we group items together based on some similarities and differences with the other items where classification there is prior knowledge about the classes and we can allocate a class label to each item based on the features. The items with the same labels then make up a group[2]. Because there is no prior knowledge on the labels this research uses clustering techniques to group the cases.

Earlier research on text-based clustering focuses on clustering in the English language[3–5]. Although Hendrix *et al.* [6] use text-based clustering in the Dutch language to group the complaints that are filed with the Landelijk Meldpunt Zorg (National Healthcare Reporting Center), they do not give insight into the algorithms that are used. Based on the limited literature research that was conducted, it appears that a research gap might exists on the area of text-based clustering in the Dutch language.

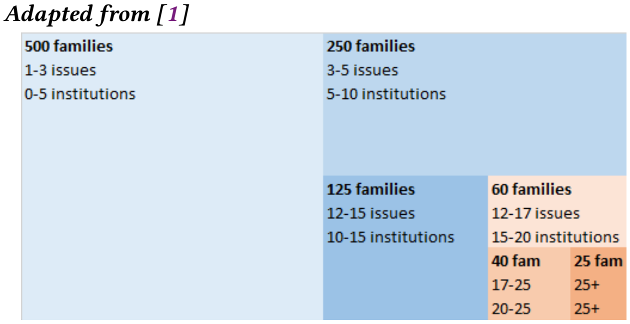
This research aims to reduce that research gap, by using several clustering techniques on the dataset for the breakthrough method that is developed by Instituut voor Publieke Waarden. The aim of this research is to find a clustering technique that finds meaningfull clusters for the cases in the breakthrough method. We start with a background on the Breakthrough Method after which the data is explored. We then turn to the algorithms that are used, the results and finally the conclusion and recommendations.

2 THE BREAKTHROUGH METHOD

The information in this section is adapted from [1]. While this section contains a high-level summary of that book, the book also contains several examples and is a recommended read for everyone who works in or is otherwise involved in social care in the Netherlands.

As stated in section 1, an estimated 20% of all families ask for assistance from formal social care. When we zoom in on the 20% and look at a neighbourhood or village with 5,000 families, then that means that there are 1,000 families with at least one issue who ask for assistance. These families are represented in Figure 1 with the number of issues that they face.

Figure 1: *One thousand families with social care issues. Adapted from [1]*



[1] More than 10

We observe that the more issues that a family faces, the more institutions are involved, and likewise, the more laws and regulations govern the way that assistance can be provided. With the increased number of laws and regulations come increased complexity and the probability of conflicting priorities also increases. It is likely difficult to maintain an overall view on the issues that a single family faces if there are multiple issues. It might also be necessary to customize a specific solution for the needs of the family. While a customized solution might fit the situation best, care must be taken that this is not an arbitrary solution and that grounds for the customization are reasonable and documented, so that involved parties can benefit from this customized solution. A benefit might be that this customization might also fit the issues of other families or that rules and regulations are revisited. Instituut voor Publieke Waarden developed the Breakthrough method; in six process steps the professional is led through a process to describe a situation, propose a solution and make future gains visible[2]. The process steps are documented in a database, bold items refer to the fields in that database:

(1) **Description** of the current situation, what are the issues that the family faces? What is the **future** situation if nothing is done.
(2) **Perspective** of the future situation, where does the family want to stand in the future?
(3) **Breakthrough**, what prevents the family from moving from the current situation to the situation where they want to be. What needs to be resolved for them to get where they want to be? What are alternative **scenarios** to achieve the perspective?
(4) From gaining perspective and breakthrough the family and professional gain overview, which is further improved by asking three subquestions:
  (a) What can the family **do self** to get to their perspective situation?
  (b) What long-term **support** does the family need to get to their perspective situation?
  (c) What short-term **treatment** does the family need to get to their perspective situation?
(5) Assesment; in this step the costs and benefits of the breakthrough are assessed using an assesment framework that consists of three values:
  (a) engagement: to what extend does the breakthrough scenario increase the family's engagement and willingness to address their own issues?
  (b) legitimacy: what are the rules and regulations that cover the issue(s) that the family face and are there **exception**s that need to be made to achieve the breakthrough?
  (c) return: what are the costs of doing nothing compared to the costs and benefits of the breakthrough scenario?
(6) What is the **solution**, what are the concrete actions that need to be taken?

The bold items in the previous list are fields in the breakthrough database. These fields are extracted for each case so that these cases can be clustered. In the next section we explore the data.

[2]Both from a social point of view, as from a monetary point of view.

3 EXPLORATORY DATA ANALYTICS

In this section the data for the breakthrough method that is explained in section 3 is explored. The dataset holds 3195 observations or cases. Each case has an attribute 'status', which is not filled for 2 cases and is either 'open' (682 cases), 'closed' (1.956 cases) or 'archived' (555 cases) for the other cases. Because the information in the open cases can be incomplete and the archived cases also contain test and demonstration cases, the focus in this report is on the 1.956 closed cases.

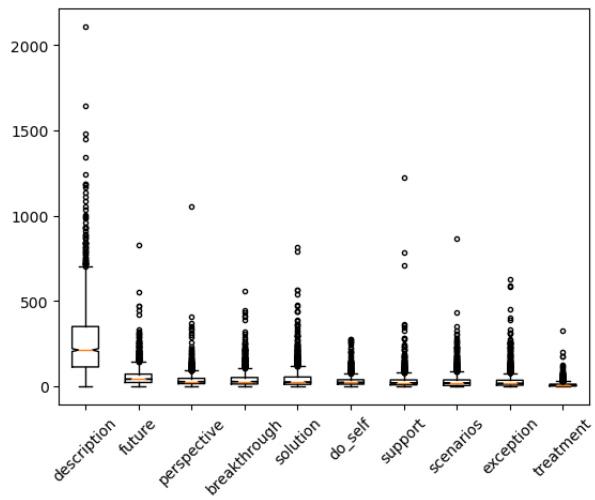
Each case has several features. In consultation with Instituut voor Publieke Waarden, the features that are considered to be the features of interest for clustering are 'breakthrough', 'description', 'do_self', 'exception', 'future', 'perspective', 'scenarios', 'solution' , 'support' and 'treatment'. All features contain text data in HTML format that can be empty. A summary of the data is included in Table 1, where the features are ordered in descending order on the number of cases that have text for that feature.

Table 1: ***Data summary***

Feature	***# with text***	***# words***		
		mean	***std. dev.***	***max***
description	1938	259	207	1938
future	1939	55	55	1939
solution	1939	45	64	1939
breakthrough	1944	41	47	1944
perspective	1939	37	43	1055
support	1938	30	50	1938
exception	1937	30	49	1937
do_self	1938	30	31	1938
scenarios	1939	29	44	1939
treatment	1938	10	17	1938

It is clear that 'description' is the feature that contains significantly more words than the other features. In Figure 2 it is even more obvious that the 'description' feature contains more words than the other features and is thus more likely to contain data that can be used to cluster the cases.

Figure 2: *Box plot for number of words per feature*



While the number of words in a text does not necessarily give information about the usability of that text for clustering, it seems reasonable to expect that longer texts are more usable for clustering than shorter texts. We therefore consider only the 'description' column. Furthermore, we are interested in the different topics in the cases and thus look at only the nouns in that feature. In Figure 3, a wordcloud with these nouns and their relative frequency is included. In the description feature 7.500 unique nouns[3] are identified.

Figure 3: *Wordcloud for nouns in description*

In Figure 3, we can identify several terms that are likely useful for clustering the cases; 'woning' (home), 'kinderen' (children), 'schulden' (debt) are some examples.

[3]Nouns with only one character and the noun 'xxx' are ignored. The noun 'xxx' is used as the only text, it seems that it is not possible to store an empty text and thus this string is used.

4 METHODS

The clustering of the cases considers the text features that are stored for each case, in Figure 2 it is obvious that the 'description' feature contains significantly more words than the other features and the focus is thus placed on this 'description' feature.

To analyse the texts, the bag-of-words approach for text analysis is used. In this approach the assumption is that the meaning of a word does not depend on it's place in a sentence or on other words that also appear in the sentence. As an example, the word 'dog' in the sentences 'The dog bit the child' and 'The child made a drawing of a dog' is treated in the same manner has exactly the same meaning in the bag-of-words method, while a human would recognize that the dog in the first sentence is an actual living dog and the second dog is a representation that might be more or less accurate. Because the clustering of cases is relevant if each cluster can be defined as a specific issue or set of issues that a family might face it seems reasonable to consider only the nouns in each text. Because the bag-of-words approach is used, no information is lost when the nouns are extracted.

In this research, four different clustering algorithms are used that are implemented in Python with the SciKit-learn package [7]. In the next subsections these algorithms are briefly explained and a motivation is given why these specific algorithms are selected.

4.1 Affinity Propagation

Affinity Propagation is a clustering algorithm that uses exemplar samples as the most representative sample for each cluster. Unlike most clustering algorithms, the number of clusters is determined by the algorithm and is not a hyperparameter. The algorithm uses an affinity metric instead of a distance metric, with higher affinity indicating that two samples are more similar than two samples with a lower affinity score. An drawback for this algorithm is that it is computationally complex [8]. Given the relatively low number of samples in this research the assumption is that that will not have an effect. The hyperparameter is $d = \{0.5 + 0.025i \mid i \in \{0, 1, 2 \ldots, 19\}\}$, the damping factor that prevents numerical oscillations.

This algorithm is selected because it not only clusters the samples, it also gives an exemplar sample for each cluster. Having this exemplar sample makes it easier to identify the common features in the cluster and also gives an actual example for each cluster. An added benefit is that the number of clusters is not a hyperparameter for Affinity Propagation, assumptions on the number of clusters are thus not required.

4.2 Agglomerative Clustering

In agglomerative clustering, each sample starts as a single cluster and clusters are merged until the desired number of clusters is reached. Clusters are merged based on the average distance between the samples in the respective clusters, starting with the lowest average distance between clusters[8]. The hyperparameter for this algorithm is n, the number of clusters. Results are determined for $N \in \{2, \ldots, 20\}$. The upper bound for the hyperparameter is set at 20 because the Instituut voor Publieke Waarden usually identifies 15 clusters[4] when doing manual clustering. An upper bound of 20

[4]Depending on the dataset and requirements of the research

allows for some additional clusters, while keeping computation time reasonable.

The agglomerative clustering algorithm is selected because it provides a bottom-up approach to clustering that is easy to understand. In addition, agglomerative clustering is able to capture nested clusters, where each cluster contains multiple sub-clusters. In the specific use-case for this research, the data can contain nested clusters.

4.3 DBSCAN

The DBSCAN[5] algorithm is based on the assumption that clusters are areas with a relative large number of observations that are separated by areas with less observations. The samples in the areas with higher densities are the 'core samples' for the clusters. The remaining samples are either 'non-core samples' if they are within a maximum distance ϵ of a core sample or are considered to be noise. The latter samples are not allocated to a cluster. Two hyperparameters determine the working for this algorithm. The first hyperparameter is m, the required number of core samples that should be within distance ϵ of a given sample for it to be a core sample itself. The other hyperparameter ϵ is the maximum distance that a sample should be from a core sample to be considered as either a core or a non-core sample. If there are no core samples within distance ϵ then the sample is considered to be noise [8]. In this research we consider all combinations for $m \in \{2, 3, \dots, 10\}$ and $\epsilon = \{0.01i \mid i \in \{1, 2, \dots, 24\}\}$. The maximum value $\epsilon = 0.25$, which seems reasonable given that all distances between points $d_{i,j} \in [0, 1]$ as documented in subsection 4.5.

DBSCAN is selected because it can handle noise. Given that our samples are texts that have been written by multiple people, it might be possible that the data contains outliers. Another benefit is that DBSCAN also does not require any assumption on the number of clusters. Finally, DBSCAN can detect clusters of arbitrary shapes because it uses density-based criteria.

4.4 Spectral Clustering

In the spectral clustering algorithm the affinity matrix is used. This algorithm uses the approach of minimizing the affinity between samples in different clusters in comparison to the affinity between samples in the same cluster[9]. The hyperparameter number of clusters $N \in \{2, \dots, 20\}$ determines the number of clusters. The reason for the upper bound of the number of clusters is the same as in subsection 4.2

The spectral clustering algorithm is selected because it is able to capture clusters of arbitrary shapes. It also is a clustering technique that reduces the dimensions in the data, because the eigenvectors of the similarity matrix are used instead of the similarity matrix itself.

4.5 Affinity and distance metrics

All clustering algorithms use either a distance metric $d_{i,j} \geq 0$ to measure how different two observations i and j are or an affinity metric $a_{i,j} \leq 1$ that measures how similar two observations i and j are. Note that the distance of a sample with itself $d_{i,i} = 0 \; \forall \; i$, while the affinity of a sample with itself $a_{i,i} = 1 \; \forall \; i$.

[5]Density-Based Spatial Clustering of Applications with Noise

To calculate the affinity and distance between two sets of nouns, each set of nouns is converted to a numeric vector $\boldsymbol{v}$ using the spaCy 'nl_core_news_lg' language model [10]. A common similarity metric for texts is the cosine similarity[11, 12]. However, the cosine similarity has the issue that the Schwartz inequality does not hold. An alternative is the angular cosine distance, which does not suffer from this issue, therefore the angular cosine distance and angular cosine affinity is used in this research.

The distance between vectors $\boldsymbol{v}(i)$ and $\boldsymbol{v}(j)$ is then

$$d_{i,j} = \frac{1}{\pi} \arccos\left(\frac{\boldsymbol{v}(i) \cdot \boldsymbol{v}(j)}{|\boldsymbol{v}(i)||\boldsymbol{v}(j)|}\right)$$

with $d_{i,j} \in [0, 1]$. From this distance metric the affinity metric is calculated: $a_{i,j} = 1 - d_{i,j}$.

4.6 Result testing

With the algorithms and the affinity and distance metrics in place the clustering can be done. However, there is still one element missing: how to test the results. For each clustering algorithm the 'best' hyperparameter set is selected and compared with the other algorithms. The 'best' hyperparameter set for a given algorithm is that set that has the highest average Silhouette Coefficient. This coefficient is a measure how good the clustering is and ranges from -1 to 1. For sample $i \in C_I$ where C_I is cluster I we calculate the mean distance a_i to all other samples in the cluster:

$$a_i = \frac{1}{\max(1, |C_I| - 1)} \sum_{j \in C_I, j \neq i} d_{i,j}.$$

The mean distance to the other samples in the cluster indicates how well sample i fits within the cluster. We also calculate b_i, the mean distance to all samples in the nearest cluster J:

$$b_i = \min_{J \neq I}\left(\frac{1}{|C_J|} \sum_{j \in C_J} d_{i,j}\right).$$

With these two values a_i and b_i we calculate s_i, the silhouette coefficient of sample i:

$$s_i = \begin{cases} 0 & |C_I| = 1 \\ \dfrac{b_i - a_i}{\max(a_i, b_i)} & |C_I| > 1. \end{cases}$$

A value close to 1 indicates that sample i is close to the other samples in the cluster, while a value close to -1 indicates that the sample is allocated to an incorrect cluster. Finally, a value of 0 either indicates that the sample i is in a cluster with only this sample or it is indifferent whether sample i should be allocated to this cluster or to the next nearest cluster.

From the individual silhouette coefficients per sample we calculate the mean silhouette coefficient for all samples:

$$\bar{s} = \frac{1}{n} \sum_{i=1}^{n} s_i$$

with n the number of samples.

Because the silhouette coefficient does not give information on the usability of the distance metric a qualitative reflection comparison between the different algorithms and a manual clustering of 50 samples is also done.

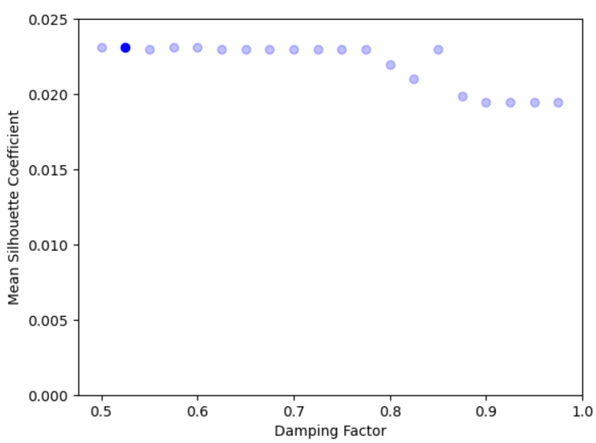
5 RESULTS

In this section the results are presented for each clustering algorithm. These results are discussed in section 6

5.1 Affinity Propagation

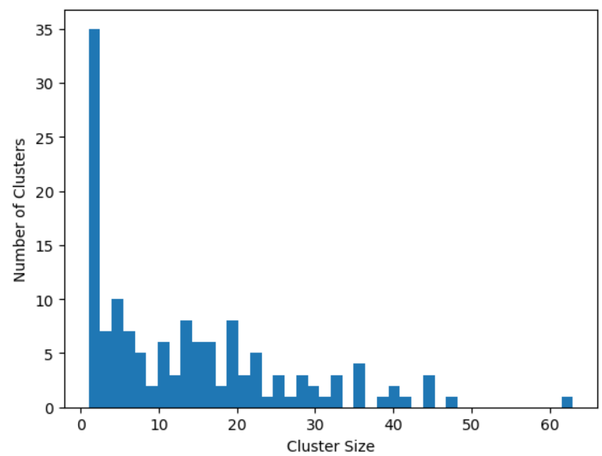
The results for the Affinity Propagation algorithm for each damping factor are presented in Figure 4. The maximum value for the mean silhouette coefficient is emphasized with a darker color.

Figure 4: ***Agglomerative Clustering Mean silhouette coefficient per damping factor***



The maximum mean silhouette coefficient is reached at damping factor $d = 0.525$. The algorithm returns 140 clusters at this damping factor. When we zoom in on these 140 clusters we observe that there is a wide variety in cluster sizes in Figure 5

Figure 5: ***Affinity Propagation histogram of number of clusters per cluster size***

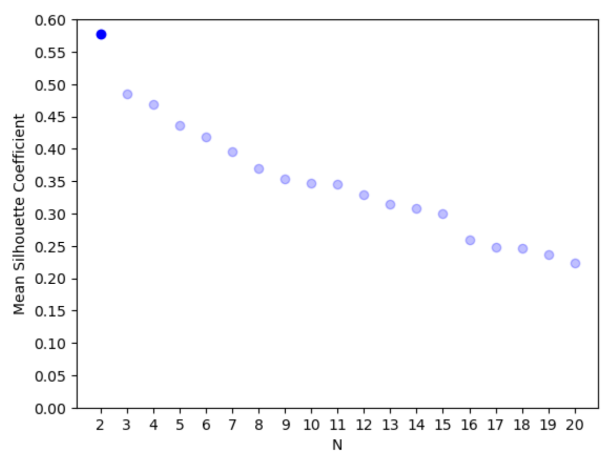


In this histogram we observe that the cluster sizes are right-skewed; there is a high number of clusters that contain only a limited number of samples. There are 29 clusters that contain only a single sample, and the median of the cluster size is 10.5 samples per cluster, while the mean is 13.7 samples per cluster.

5.2 Agglomerative Clustering

The results for the Agglomerative Clustering algorithm for each cluster size are presented in Figure 6. The maximum value for the mean silhouette coefficient is again emphasized with a darker color.

Figure 6: ***Agglomerative Clustering Mean silhouette coefficient for number of clusters***



The maximum value for the mean silhouette coefficient is reached when the samples are divided in 2 clusters. The size of each cluster is given in table Table 2, note that the cluster labels are only used to differentiate between the two clusters.

Table 2: ***Aglomerative Clustering Cluster size per cluster***

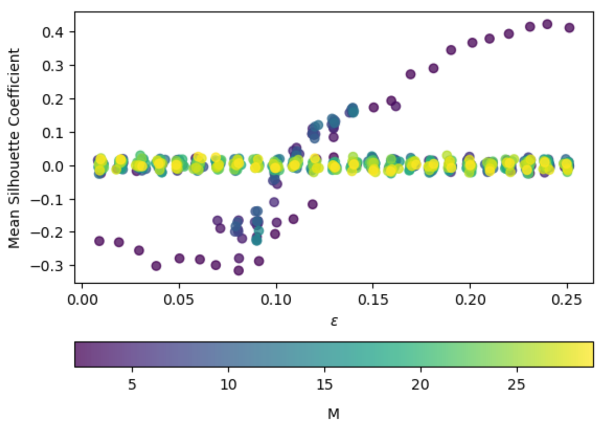
Cluster	*# samples*
A	1913
B	1

It is obvious that there is significant difference between the number of samples in the two clusters, which is discussed in section 6.

5.3 DBSCAN

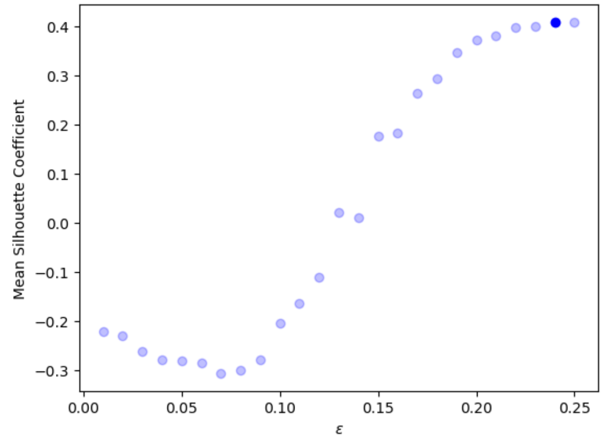
The DBSCAN algorithm uses two hyperparameters, M and ϵ. In Figure 7 the results for each ϵ and M combination are included. To avoid overlapping points some randomness has been added, instead of plotting ϵ we plot $\epsilon + X$ with $X \sim \mathrm{N}(0, 0.001)$ and for $\bar{s}$ we plot $\bar{s} + Y$ with $Y \sim \mathrm{N}(0, 0.01)$.

Figure 7: ***DBSCAN Mean silhouette coefficient for*** ϵ ***and*** M



In Figure 7 we observe that there is one value for M that results in some higher values for the mean silhouette coefficient than the other values for M. We therefore plot the mean silhouette coefficients for all ϵ for $M = 2$, without adding noise in Figure 8. The maximum value for the mean silhouette coefficient is again emphasized with a darker color.

Figure 8: ***DBSCAN Mean silhouette coefficient for*** ϵ ***at*** $M = 2$



For $M = 2$ and $\epsilon = 0.24$, the DBSCAN algorithm finds the highest value for the mean silhouette coefficient. At this combination there are two clusters also some samples that are considered outliers or noise. The size of each cluster is given in table Table 3.

Table 3: ***DBSCAN Cluster size per cluster***

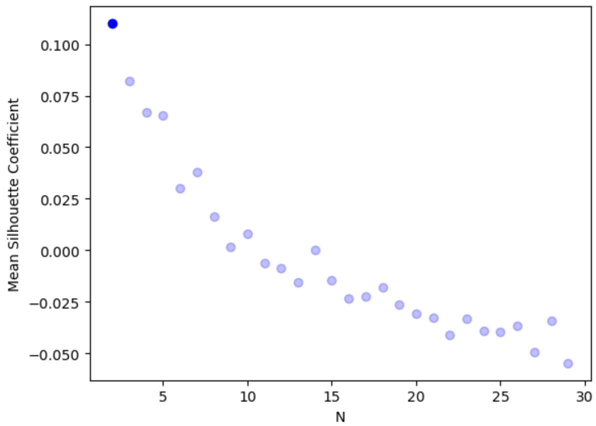
Cluster	*# samples*
A	1904
B	2
Noise	8

It is again obvious that there is significant difference between the number of samples in the two clusters, which is discussed in section 6.

5.4 Spectral Clustering

The results for the Spectral Clustering algorithm are included in Figure 9.

Figure 9: ***Spectral Clustering***
Mean silhouette coefficient for number of clusters



In this figure we see that the highest silhouette score results for $N = 2$. The cluster sizes per cluster are included in Table 4. In this table we see that the distribution of samples over the clusters is more even when compared to the other algorithms.

Table 4: ***Spectral Clustering***
Cluster size per cluster

Cluster	*# samples*
A	845
B	1069

5.5 Manual Clustering

In addition to the clustering algorithms, a manual clustering of 50 cases was also done. The results are included in Table 5, in this case the labels of the clusters give an indication on the subject of each cluster. This clustering is done using the information in the 'description' field for 50 cases.

Table 5: ***Manual Clustering of 50 samples***
Cluster size per cluster

Cluster	*# samples*
Childcare benefits scandal	3
Debt	5
Education	2
Education & Health	1
Health	5
Health & Housing	11
Health & Housing & Safety	1
Health & Housing & Trauma	2
Housing	11
Housing & Safety	7
None	2

One of the clusters in the manual clustering is the cluster 'None'. This cluster contains two cases. One of the cases had only the letter 'x' in the description, while the other case had no text at all in the description. In the four clustering algorithms, these cases are not included.

6 DISCUSSION

In section 5 the results for the four clustering algorithms plus a manual clustering are included. These results are summarized in Table 6.

Table 6: ***Results per method***

Method	*Number of Clusters*	*Mean Silhouette Coefficient*
Affinity Propagation	140	0.023
Agglomerative Clustering	2	0.578
DBSCAN	2	0.408
Spectral Clustering	2	0.110
Manual Clustering	11	-

Although both the Agglomerative Clustering and DBSCAN algorithms have the highest mean silhouette coefficients, they each have one huge cluster with the majority of the samples and a very small other cluster with either 1 or 2 samples. Therefore, the usefulness of these algorithms is considered to be low in this use-case.

When we assume that each cluster represents a combination of issues that a family can face, we can calculate a theoretical maximum for the number of clusters. In Figure 1 we see that a small number of families face 25 or more issues. When we set the maximum number of issues at 25 (and the minimum number at 1), we can calculate the theoretical maximum number of clusters O as follows:

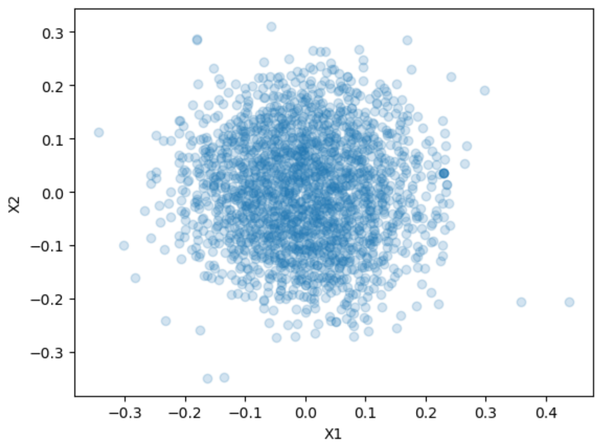
$$O = \sum_{i=1}^{25} \binom{25}{i} \approx 3.36 \times 10^7.$$

Given that manual clustering for only 50 cases already results in 11 clusters, it seems that spectral clustering is also not the best clustering method for the cases in the breakthrough dataset. That leaves us with one clustering method; affinity propagation. However, this algorithm has the issue that the mean silhouette coefficient is close to 0, which indicates that the assignment of cases to clusters is on average arbitrary.

Although the algorithms do not provide satisfactory results, these are all proven clustering methods. The data is also a real-live dataset and can not be adjusted to fit our clustering methods. However, in between the dataset and the clustering algorithms sits the transformation that was done to get from a text to a numeric vector and the distance definitions between the samples. In Figure 10 a two-dimensional representation of the distances between the points is included.

is currently stored in two tables following a redesign of the database.

Figure 10: *Two-dimensional representation of all samples*



Although the actual samples are represented in a vectors with 300 dimensions and the angular cosine distances between points are represented in this figure as euclidean distances, this figure gives an indication why the algorithms struggle with the data. There are no identifiable white areas with lower density that might represent borders between clusters, all data points seem to be in one big cluster, with some outliers.

7 CONCLUSION

In section 6 we find that none of the four clustering algorithms provides satisfactory results, which is likely the result of the data preparation that was done and the decision to use the angular cosine distance as a distance metric. Clustering in the social care field and especially in the breakthrough dataset is complex and more attention needs to be given to data preparation if clustering algorithms are to assist in clustering the data in a meaningful way.

8 RECOMMENDATIONS

Although none of the clustering methods returned statisfactory results, we can make some recommendations that might be helpfull in future research on this subject:

- It might be useful to transform the words in the texts into lemmas. In that case several synonyms can be aggregated into a single lemma, which reduces number of different terms used in the clustering models.
- Instead of the bag-of-words an alternative approach can be considered. An alternative that might work is the bag-of-ngrams approach, where combinations of words are considered. However, this would likely increase the number of different terms that are used in the models.
- This research uses the Dutch language model from spaCy [10] that is trained on a large number of articles from newpapers. Other language models might fit the samples from the breakthrough method better.
- The database for the breakthrough cases is currently not well normalized. Improving normalization would help in future data analytics. An example is that the data for ‘descriptions’

REFERENCES

[1] E. Westerveen, H. Kruiter, and M. Lüschen, *Het Grote Doorbraakboek.* Insituut voor Publieke Waarden, 2022.

[2] J. Goswami, “A comparative study on clustering and classification algorithms,” *International Journal of Scientific Engineering and Applied Science*, June 2015.

[3] A. Amine, Z. Elberrichi, and M. Simonet, “Evaluation of text clustering methods using wordnet,” *Int. Arab J. Inf. Technol.*, vol. 7, pp. 349–357, 12 2010.

[4] Y. Li, S. M. Chung, and J. D. Holt, “Text document clustering based on frequent word meaning sequences,” *Data & Knowledge Engineering*, vol. 64, no. 1, pp. 381–404, 2008. Fourth International Conference on Business Process Management (BPM 2006) 8th International Conference on Enterprise Information Systems (ICEIS’ 2006).

[5] Z. C., “Research on literature clustering algorithm for massive scientific and technical literature query service,” *Computational Intelligence and Neuroscience*, 2020.

[6] I. Hendrix, T. Voets, S. Ranke, and T. Kool, “Textmining in toezicht.” https://www.iqhealthcare.nl/media/124698/rapport_textmining_lmz_def.pdf, 2019.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn documentation - clustering.” https://scikit-learn.org/stable/modules/clustering.html, 2011.

[9] A. van Luxburg, “A tutorial on specral clustering,” *Statistics and Computing*, vol. 17, pp. 395–416, August 2007.

[10] M. Honnibal and I. Montani, “spaCy Natural Language Processing for Dutch - Models v3.6.” https://spacy.io/models/nl/, 2021. Accessed: 5 July 2023.

[11] N. E. Evangelopoulos, “Latent semantic analysis,” *WIREs Cognitive Science*, vol. 4, no. 6, pp. 683–692, 2013.

[12] W. H. Homaa and A. A. Fahmy, “A Survey of Text Similarity Approaches,” *International Journal of Computer Applications*, vol. 68, pp. 1792–1799, April 2013.