Fixation and Machine Learning

A new method for measuring fixation in internet users using machine learning and natural language processing

Katarina Rivera Ahlin
Abstract

Fixation — the state of mind where an individual is intensely fixated on one individual, activity or idea — can be an early warning sign of an individual that will later go on to commit violent acts. This paper details the development of a method that measures fixation from texts written on internet forums, a method that could (when expanded) be a part of threat assessment and prevention of violence.

Text documents gathered from Flashback Forum were pre-processed, analysed, clustered and scored using a new fixation scoring algorithm.

Three different clustering algorithms were tested — k-means, agglomerative hierarchical and DBSCAN. After evaluation with both ARI and AMI, DBSCAN was found to be the best fit.

The method was tested on a set of 100 users from Flashback Forum and gave satisfactory results.
10 Future Work

A Test Data: Clustering Algorithm

B Test Data: Scoring Algorithm
1 Introduction

At the end of the 20th century there was a shift in how law enforcement investigations of violent crime were conducted. The investigations were usually made after a crime had occurred, but later as a result of new types of crime — for instance stalking and violence in schools — there was an increasing need for methods to prevent violent behaviour. Threat assessment is one way to approach this problem — to analyse behaviours and patterns in order to make an assessment of how big a threat a subject poses to a target (Borum et al. 1999).

Meloy, Hoffmann, Guldimann, et al. (2012) uses the term “warning behaviours” to indicate factors which constitute change in patterns of behaviour and which indicate an increasing threat. The warning behaviours contain dynamic variables that can change over time. They suggest a typology of eight warning behaviours; pathway, fixation, identification, novel aggression, energy burst, leakage, directly communicated threat and last resort warning behaviours. This can be useful in threat assessment.

Meloy, Hoffmann, Roshdi, et al. (2014) performed a study with a sample of nine German school shooters and 31 students of concern where they tested the typology of eight warning behaviours to see if there was any significant difference in warning behaviours between the two groups. The results showed that among the school shooters there were a significantly higher frequency in five of the warning behaviours; pathway, fixation, identification, novel aggression and last resort.

This thesis will focus on fixation with the purpose of finding a way to estimate a level of fixation in a person on the internet.

1.1 Thesis Goals

A method will be developed using Python that takes a text as input and produces a real number, representing a fixation level, as output. The method will be automated, using only the input data and other models trained with machine-learning algorithms.

The thesis goals are the following:

- produce a method that can automatically estimate a level of fixation in written text and
- evaluate the method on real data.
1.2 Limitations

- The method will only work with text written in Swedish.
- The real data used will come from an Internet forum.

2 Background

2.1 Fixation

Fixation is a state of mind that describes an intense preoccupation with an individual, activity, or idea. To be fixated is a part of life and it can be about anything — romantic love, intense loyalty, hobbies — but it can also be pathological if it grows into an abnormally intense degree (Johansson, Kaati, and Sahlgren 2016). The pathologically fixated spend a lot of time thinking about the object of fixation, they can spend time gathering information from different sources and try to communicate with the object (Mullen et al. 2009).

As the fixation becomes more intense, it begins to affect the life of the fixated — both socially and financially — who becomes more and more isolated (Meloy, Hoffmann, Guldimann, et al. 2012; Mullen et al. 2009). If the object of fixation is a cause it can often transform into a fixation on a person as it grows more intense. The fixated is one of the biggest threats to public figures in today’s world (Mullen et al. 2009).

Meloy, Hoffmann, Guldimann, et al. (2012) defines fixation as “any behavior that indicates an increasingly pathological preoccupation with a person or a cause”. According to Meloy, Hoffmann, Guldimann, et al. (ibid.) fixation can be measured by:

- increasing perseveration on the person or cause,
- increasingly strident opinion,
- increasingly negative characterization of the object of fixation,
- impact on the family or other associates of the object of fixation (if present and aware),
- angry emotional undertone

and is typically accompanied by social or occupational deterioration.
2.2 Fixation on the Internet

The behaviour preceding an attack of a violent actor that acts on their own can take place in real life but also on the Internet by communicating opinions, values and actual intent. Cohen et al. (2014) believes that the warning behaviours that are most easily detectable in the subject’s written communication in social media are leakage, fixation and identification. Certain expressions of attitude or mindset that might be detectable are referred to as linguistic markers for radical violence. The linguistic markers can be used as input to computer algorithms that may be able to recognize signs of radical violence (ibid.).

Cohen et al. (ibid.) states the following as linguistic markers for fixation:

- One person, group, or issue is mentioned by the subject with a significantly higher frequency than it is mentioned by other discussants.

- Frequent combinations of certain key terms, for instance “Jew” and “communism,” can reveal a fixation with a certain idea.

To find this kind of fixation in text Cohen et al. (ibid.) propose counting relative frequency of key terms relating to named entities such as persons, organisations, etc. It is also important to consider the subject’s changes in behaviour over time — both when it comes to perseveration and negative characterisation (Johansson, Kaati, and Sahlgren 2016).

According to Johansson, Kaati, and Sahlgren (ibid.) vocabulary variation — when different people use different terms to refer to the same thing — is the most difficult problem when dealing with natural language in online data, specifically when using keyword-based approaches. One way to approach the problem is to use unsupervised machine learning techniques that can identify semantically similar terms in the data by reading lots of text — generally known as “distributional semantic models”.

2.3 Flashback

Flashback is a Swedish Internet forum where the users can discuss various categories and subcategories that cover a wide range of topics from family life, food and relation problems to drugs, ongoing investigations etc. A user can start a new thread about a topic or write a comment on an existing one.

Flashback claims to be a place where you can write about almost anything.

The purpose of Flashback Forum is to protect free speech and actively defend the freedom of speech and opinion. This includes
defending those who wish to abolish the freedom of speech. The goal is a society where every member can make their voice heard, even those with dissenting opinions. (Flashback n.d.[a])

Flashback is the fourth most visited social media in Sweden. 32 % of Swedes over 16 years of age said that they used Flashback at least once during 2019 and 1 % said that they used Flashback daily. By comparison, 24 % of Swedes over 12 years of age used Twitter at least once during 2019 and 7 % used Twitter daily. Most of the people using Flashback are men in the age of 36-45 (Internetstiftelsen 2019).

2.3.1 History

Flashback started as a newspaper in 1983 and was launched on the internet in 1995, Flashback Forum was created in 2000. During this period of time there were many other controversial websites launched in Sweden, many of them were forced to shut down because the internet providers decided that they were morally questionable. As a result, Flashback opened a web hosting service, claiming to be a safe haven for all kinds of opinions, and sought out the websites that were closed to offer them a place on their servers. Flashback was forced to shut down in 2000-2001 due to allegations of directly supporting a nazi organisation. In 2002 Flashback Forum was forbidden in Sweden and therefore forced to shut down, they changed their location to London and has been open since 2003. In January 2010 the legal responsibility for Flashback Forum moved to Flashback International Inc., based in the United States (Flashback n.d.[b]).

2.4 Familjeliv

Familjeliv is a Swedish Internet forum that started in 2003. Familjeliv also covers a wide range of topics but focuses more on questions about pregnancy, children and parenting. On their website Familjeliv writes “It should be nice to hang out at FamiljeLiv.se” and they talk about their site as “Sweden’s best, and nicest, family site” (Familjeliv n.d.).

2.5 Previous Work

In their study, Curiskis et al. (2020) used K-means and hierarchical agglomerative clustering (among other methods) using the Euclidean metric when performing document clustering in Twitter and Reddit. K-means was one of the clustering algorithms used by Godfrey et al. (2014) in order to cluster tweets. Before clustering they used DBSCAN to remove noise from the
tweets. Throughout their research they used the cosine distance as metric distance (Godfrey et al. 2014).

Romano et al. (2016) propose using ARI when the reference clustering has large equal sized clusters and AMI when the reference clustering is unbalanced and there exist small clusters. Furthermore they state that NMI is not a suitable clustering comparison measure as it does not show constant baseline value equal to 0 when partitions are random. In the experimental study by J. Xu et al. (2015) NMI was one measure used for evaluating their proposed Short Text Clustering method. Curiskis et al. (2020) used ARI, AMI and NMI as evaluation measures when evaluating different methods for document clustering and topic modelling in online social networks. Furthermore, ARI was used by Rui, Xing, and Jia (2016) as evaluation method in their study.

3 Theory

To construct a method for finding fixation in text I had to determine how much a user talks about one topic. To find this, I needed to somehow categorise all the meaningful words to see if there was one category that was bigger than the others. For this purpose I chose to use a clustering algorithm. To be able to do the clustering I needed to prepare the data and I also did some experiments to see which clustering algorithm works the best for my purpose. In this section I will describe the necessary theoretical background.

3.1 Text Preprocessing

Written communication is easy for most humans to understand but hard for a computer to read and analyse. Text written on social media is often even harder since it usually contains numbers, links to other websites, punctuation, misspellings, etc. To make the data more computer friendly it needs to be preprocessed.

Numbers, punctuation and letters in upper or lower case all carry different meanings in a text and can be useful for different purposes. The more variation, the more complex it gets for an algorithm to work with the text. Therefore it can be useful to remove or convert these elements. There are different ways to deal with this — e.g. numbers can be converted to text or simply removed.

A document is interpreted as one single string by a computer. Tokenization means splitting a string into smaller pieces, called tokens, e.g. a paragraph can be split into sentences or a sentence into words.
Stop words are common words that do not contribute much to the analysis and can be removed to make the algorithm more effective, e.g. “a”, “the”, “on”, “is” (Beysolow 2018).

An example of a sentence that has not been preprocessed:

“I walked, through 30 degree heat, all the way to the British Museum.”

The same sentence after a preprocessing:

["walked", "through", "degree", "heat", "all", "way", "british", "museum"]

3.2 TF-IDF

Term frequency-inverse document frequency (TF-IDF) is a statistical concept meant to measure the importance of a term to a document in the corpus. The idea is that the importance of a term cannot only be computed by how often that term appears in the document, it is also important to measure how common it is among other documents.

Term frequency (TF) is the ratio between how many times a word appears in the document and the total number of words in the entire document.

\[
TF = \frac{\text{Number of times the term appears in a document}}{\text{Total number of terms in the document}}
\]

Inverse document frequency (IDF) measures how common the term is among all documents. A term that appears in many documents will yield a lower IDF.

\[
IDF = \log_{10}\left(\frac{\text{Total number of documents}}{\text{Number of documents with the term in the document}}\right)
\]

TF-IDF is the combination of the two.

\[
TF-IDF = TF \cdot IDF
\]

A term with a high TF-IDF value is a term that appears a lot in one document and is absent from most other documents. It thus helps to weight terms that appear often in one document but are very common among all other documents lower than other less common terms that do not appear as often among other documents (Thanaki 2017).
3.3 Word2vec

Word2vec is a word embedding technique — i.e. it converts words into vectors of real numbers. The result is a high dimensional word space where words that are similar to each other are represented by vectors close together while words that are not similar are represented by vectors far apart from each other (Mikolov et al. 2013; Thanaki 2017).

A vector space makes it possible to compute the similarity between two words using a similarity function, one commonly used similarity function is the cosine of the angle between the vectors (Goldberg 2017).

3.4 Clustering Algorithms

Grouping objects into categories, e.g. categorising both dogs and cats into the category “animals”, is something humans do in order to learn and better understand new objects. When analysing data a common task is to classify or group data into categories, this can either be done by classification or clustering. In classification class labels exists and are assigned to new data; in clustering there are no known labels, the algorithm finds hidden patterns in the data and uses this to group the data into a finite set of unlabelled clusters (R. Xu and Wunsch 2009).

To use the metaphor of animals, consider a scenario where a large new batch of different new animals were discovered. If the new animals were categorised into existing animal categories, that would be considered classification. If, instead, the features of the new animals were identified and used by themselves to induce a new set of groups, that would be considered clustering.

I am going to use three clustering algorithms in my experiments: agglomerative clustering, K-means and DBSCAN. The clustering algorithms will be briefly described below.
3.4.1 Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering begins by assigning each data point a unique cluster. Then, in each step the two closest clusters are grouped together into one new cluster. The algorithm is finished when all data points are in the same cluster. The results of the clustering can be represented by a dendrogram where the root represents the whole data set, and the leaves are the data points. Different numbers of clusters can be obtained by cutting the dendrogram at different levels (R. Xu and Wunsch 2009). See figure 1 for an example.

Figure 1: An example of an agglomerative hierarchical clustering. The result can be shown as a dendrogram and cutting the dendrogram at the dashed line yields three clusters: one containing A and B, one containing D and E and one containing only C.

3.4.2 K-means

K-means partitions the data points into $k$ clusters, $k$ is chosen by the user. In the first step the algorithm randomly chooses $k$ distinct data points to be the centroids of the initial clusters. Then for each data point, the algorithm measures the euclidean distance between the data point and each centroid. The data point will belong to the cluster with closest centroid. When all data points have been assigned a cluster, the centroid of each cluster is moved to the mean of the respective cluster. The last two steps are repeated until the clustering doesn’t change with a new iteration (Thanaki 2017). See figure 2 for an example.
Figure 2: An example of a K-means clustering with $k = 3$. In step 3 and 5 the centroids are moved to the new mean of the clusters. In step 6 the clusters are the same as in step 4, therefore the clustering is finished.

3.4.3 DBSCAN

DBSCAN (Density Based Spatial Clustering of Applications with Noise) is a density-based clustering method. Density-based clustering methods groups data points into clusters based on the density between the points. In a DBSCAN clustering there exists three types of data points — core points, border points and noise points. A core point is any data point that has at least $n$ neighbours within $\varepsilon$ distance, where $n$ and $\varepsilon$ are values specified by the user. A border point does not contain enough data points in its neighbourhood to be considered a core point itself, but it lies in the neighbourhood of another core point. Noise points, also called outliers, are points that are neither core points nor border points. Given two points $x$ and $y$, $x$ is density-reachable from $y$ if $y$ is a core point and there is a sequence of core points that leads to $x$ where every core point in the sequence are within $\varepsilon$ from the previous core point. Two points are density connected if both points are density reachable from a common point. A cluster is a set of core points and border points where any two points are density connected to each other (R. Xu and Wunsch 2009). See figure 3 for an example.
Figure 3: An example of a DBSCAN clustering with \( n = 2 \). The neighbourhood of a data point is drawn as a circle with radius \( \varepsilon \). Orange points are border points, green points are core points and black points are outliers. The result of this clustering is two clusters and three outliers.

### 3.5 Performance Evaluation

Different performance evaluation methods can be used to evaluate the results of different clustering algorithms. Two popular such methods are the Adjusted Rand Index (ARI) and the Adjusted Mutual Information (AMI) methods.

ARI evaluates the clustering by looking at pairs of points and whether they are in the same or different clusters — it generally works better when the categories are large and of equal size.

AMI is a more complicated algorithm that uses probabilities to measure whether the clustering and the true categories contain the same information — it is preferable when categories are unbalanced with some large and some small clusters (Romano et al. 2016).

A thorough description of both methods can be read below.
3.5.1 Adjusted Rand Index

ARI requires a ground truth class assignment, a set of pre-defined true labels, and an assignment from a clustering algorithm on the same data. ARI measures the similarity of the two assignments, ignoring permutations and with chance normalization — i.e. random label assignments will have a score close to 0. The ARI score is in the range \([-1, 1]\) — negative values mean that the assignments are completely different from each other while 1 means they are equal (Pedregosa et al. 2011).

The raw, unadjusted, Rand index (RI) and ARI are given by:

\[
RI = \frac{a + b}{C_2^{n_{samples}}} \quad \text{ARI} = \frac{RI - E[RI]}{\max(RI) - E[RI]}
\]

Where, if \(K\) is a clustering,

- \(C\) is a ground truth class assignment,
- \(a\) is the number of pairs of elements that are in the same set in \(C\) and in the same set in \(K\),
- \(b\) is the number of pairs of elements that are in different sets in \(C\) and in different sets in \(K\),
- \(C_2^{n_{samples}}\) is the total number of possible pairs in the dataset (without ordering),
- \(E[RI]\) is the expected RI of random labelings.

3.5.2 Adjusted Mutual Information

AMI is a mutual information-based score. The mutual information between two class assignments measures their agreement ignoring permutations. AMI measures the performance of a clustering based on the mutual information between the clustering and a ground truth class assignment. Normalized mutual information score (NMI) is another mutual information based score. Whereas AMI is normalized against chance, that is not the case with NMI. The AMI and NMI scores has an upper bound of 1 — values close to 0 indicate that the two label assignments are largely independent and values close to 1 indicate significant agreement (ibid.).
The mutual information (MI) is given by:

$$\text{MI}(U, V) = \sum_{i=1}^{\left| U \right|} \sum_{j=1}^{\left| V \right|} P(i, j) \log \left( \frac{P(i, j)}{P(i)P(j)} \right)$$

Where

- $U$ and $V$ are two label assignments,
- $P(i)$ is the probability that a random element is assigned to the $i$th category in $U$.
- $P'(j)$ is the probability that a random element is assigned to the $j$th category in $V$.
- $P(i, j)$ is the probability that a random element is assigned both to the $i$th category in $U$ and the $j$th category in $V$.

The NMI and AMI scores can then be given by:

$$\text{NMI}(U, V) = \frac{\text{MI}(U, V)}{\text{mean}(H(U), H(V))}$$

$$\text{AMI}(U, V) = \frac{\text{MI} - E[\text{MI}]}{\text{mean}(H(U), H(V)) - E[\text{MI}]}$$

Where

- $U$ and $V$ are two label assignments,
- $H(U)$ and $H(V)$ is the entropy — the amount of uncertainty — of $U$ and $V$,
- MI is the mutual information between $U$ and $V$,
- $E[\text{MI}]$ is the expected mutual information.
4 Requirements

The algorithm should be able to read a text and, based on that text, calculate a level of fixation represented by a float $f \in [0, 1)$ where a higher score indicates a higher level of fixation. The algorithm should also be able to return the top $n$ tf-idf terms with their labels produced by a clustering algorithm.

5 Experiment Setup

The corpus used consisted of written text from Flashback Forum — 400 documents that each contained every post written by a single user before the year 2020. Those texts were preprocessed, features were extracted and translated into vectors. The vectors were clustered and the result of the clustering was used in a scoring algorithm to get a number representing a grade of fixation of the user.

Two side-experiments were made — one to decide which clustering algorithm to use and one to create and decide on a scoring algorithm.

In figure 4 the steps are shown as a flowchart. The main experiment is coloured purple and the side experiments are coloured green and blue respectively. Below, some of the steps will be described more thoroughly.
5.1 Text Preprocessing

The documents in the corpus were preprocessed by making all text lowercase and removing numbers, punctuation and stop words. Two collections of Swedish stop words were used to remove stop words, one provided by Natural Language Toolkit (Bird, Klein, and Loper 2009) and one provided by Svensk Text (Dahlgren 2018). Additional stop words were added during the implementation — e.g. words that were not in the vector space.

The package `re` (Van Rossum 2020) was used to remove numbers, HTML and words that match some specific criteria — e.g. every string beginning with “haha”.

Figure 4: The different steps taken in the development of the fixation method.
5.2 Feature Extraction

As described in section 2.2, one suggested method for finding fixation is to count the frequency of key terms. Tf-idf was chosen to be able to identify the words that carry the most information.

The top \( n \) tf-idf terms of a document, were \( n = 100 \) as default but can be chosen by the user, were used as key terms to use in the method. In order to calculate the tf-idf weights, scikit-learn (Pedregosa et al. 2011) was used to train a TfidfTransformer and CountVectorizer in advance. When training these models, I used a corpus that consisted of 400 text documents. Each document represented one user from Flashback Forum and contained all text that the user had written on the forum before the year 2020. The users were chosen such that each of them had a minimum of 500 posts — 500 posts was chosen because a fixated user would need to have written a lot.

5.3 Vector Spaces

Two pre-trained 100-dimensional vector spaces produced by Word2vec were used, one trained on texts from Flashback Forum and one trained on texts from Familjeliv. The vector spaces were supplied by FOI.

For the experiments where different clustering algorithms were tested, both vector spaces were used. The Familjeliv vector space was added to see if there was any difference in the results between that and one trained on Flashback since both Flashback Forum and Familjeliv are two major Swedish Internet forums. For the other experiments only the vector space over Flashback Forum was used.

5.4 Clustering Algorithm

To choose which clustering algorithm to use in the method for finding fixation some experiments were made.

First, a set of test data was manually created. The data consisted of 10 categories with 20 words each. The data is meant to represent key terms from the text to be clustered later. Figure 5 is a two-dimensional visualisation of the ten categories.
As can be seen in the figure some categories are well separated — e.g. exercise and drugs — while others are closer together and overlapping in the vector space. The test data in its entirety can be found in appendix A.

Three clustering algorithms were tested — K-means, agglomerative clustering and DBSCAN, and two evaluation methods were used in the experiments:

- Adjusted rand index (ARI) since the test data contained 200 words where each cluster is of equal size — 20 words,

- Adjusted mutual information (AMI) since the Flashback Forum data was not so large for one user and each cluster was expected to be of different size with some smaller clusters.

In the experiments, the coordinates for every word in the test data set were fetched from the two vector spaces and saved for later use. The Python library Gensim (Rehurek and Sojka 2011) was used for this purpose. In the first experiments all test data was used — 200 words with 20 words per category, i.e. balanced data. As a second experiment a subset of the test data was used — every category had a different amount of words ranging
from 3 to 20 words, i.e. unbalanced data, to make the data more similar to real data.

The experiments and evaluations were conducted as follows:

1. A subset was picked of the set of all categories and the words from all categories in the subset were combined.
2. The clustering algorithms were run on the combined set of words.
3. Steps 1 and 2 were repeated until every subset of the set of categories had been tested.
4. The results of all subsets of $n$ number of categories were evaluated using two evaluation measures.
5. For each evaluation measure, the mean value of the evaluation scores from step 4 was calculated.
6. Steps 4 and 5 were repeated for every $n \in [2, 10]$.

5.5 Scoring Algorithm

When measuring fixation two criteria were considered. The first was how many data points there were in the biggest cluster, $C_{\text{max}}$. The ratio between the number of data points in $C_{\text{max}}$ and the total number of data points, $N$, in the clustering is called $r$,

$$r = \frac{|C_{\text{max}}|}{N}.$$ $N = 100$ in this research. The logic that was followed here was that the more data points there is in one cluster, the more that user has talked about one topic and that user should therefore get a higher fixation score.

The second criteria to consider was how close the data points in $C_{\text{max}}$ are in the vector space. The more dense the cluster is, the more focused the topic and the higher chance that the cluster actually represents a single topic.

Two different density values were calculated and used: the first value, $D_1$, was calculated by first computing the mean vector of all vectors in $C_{\text{max}}$, then for each data point, the cosine distance was calculated between the data point and the mean vector. $D_1$ is the mean value of all those distances.

The second value, $D_2$, was calculated by calculating the pairwise cosine distances of all vectors in $C_{\text{max}}$. $D_2$ is the mean value of all pairwise distances.

$D_1$ and $D_2$ can have values from 0 to 1, 0 indicates that the vectors are orthogonal and 1 indicates that they are the same.

By multiplying the two values, two candidates for the final scoring algorithm were achieved.
\[ s_1 = D_1 \cdot r \]
\[ s_2 = D_2 \cdot r \]

To improve the odds of good results, two more candidate scoring algorithms that use the error function were added.

The error function, \( \text{erf} \), of \( r \) lowers the importance of one additional point in \( C_{\text{max}} \) when \( |C_{\text{max}}| \) is large or small. The error function is given by:

\[
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt
\]

Where \( e^{-t^2} \) is the normal distribution. See figure 6 for a graph over the error function.

![Figure 6: The error function, \( y = \text{erf}(x) \).](image)

The idea was that if \( |C_{\text{max}}| \) is large the density of \( C_{\text{max}} \) should be given more weight and if \( |C_{\text{max}}| \) is small it should keep the score low until more data points are in \( C_{\text{max}} \).

\[ s_3 = D_1 \cdot \frac{\text{erf}(3 \cdot (r - \frac{1}{2})) + 1}{2} \]
\[ s_4 = D_2 \cdot \frac{\text{erf}(3 \cdot (r - \frac{1}{2})) + 1}{2} \]

The error function of \( D \) was added for completeness.

\[ s_5 = \frac{\text{erf}(3 \cdot (D_1 - \frac{1}{2})) + 1}{2} \cdot r \]
\[ s_6 = \frac{\text{erf}(3 \cdot (D_2 - \frac{1}{2})) + 1}{2} \cdot r \]
To choose which of the six scoring algorithms to use in the finished method, they were tested on test data.

The set of test data was manually created similarly to the side experiments to determine a clustering algorithm earlier, 10 categories with 20 words each that represent key terms from the text we will later score. Figure 7 is a two-dimensional visualisation of the ten categories. The test data in its entirety can be found in appendix B.

The data was clustered, scored and manually evaluated. First, all ten categories of the test data were used, then the experiments were done a second time but with seven categories. The three categories with worst score were removed to see how the scoring algorithms worked with more well separated categories.

The experiments were conducted as follows:

1. $n$ words from one main category and $20 - n$ words from the other categories evenly distributed and randomly chosen within each category were combined to one set.

2. The set of words were clustered.

Figure 7: t-SNE representation of the test data used in the experiments for choosing scoring algorithm. Every dot represents a word and each color represents a category. The visualisation is done by transforming the data from strings into word embeddings in a vector space trained on data from Flashback. Dots close together in the graph are also close in the 100-dimensional vector space.
3. A fixation score was calculated from the result of step 2 using six different scoring algorithms.

4. Step 1-3 was repeated ten times, such that each category got to be the main category, and for every even \( n \in [2, 20] \).

5. Step 1-4 were repeated ten times to gather more information since many words were randomly chosen.

6. For each scoring algorithm one boxplot was plotted, all results from \( n \) clustered words were put together in one box.

6 Experiment Results

For all three clustering algorithms (K-means, DBSCAN and agglomerative), the ARI score and the AMI score were significantly higher when using the vector space trained on Flashback Forum. The results are shown in figure 8.

For balanced data, K-means and agglomerative clustering scores higher than DBSCAN. When the data is unbalanced, DBSCAN scores slightly higher than the other algorithms. These results are shown in figure 9.

![Figure 8: Results of the performance evaluation methods AMI and ARI when using K-means, DBSCAN and agglomerative clustering on the test data with vector coordinates from the vector spaces trained on Flashback Forum and Familjeliv. The results show that using vector coordinates from Flashback Forum vector space performs better than using vector coordinates from Familjeliv vector space.](image)
Figure 9: Results of the performance evaluation methods AMI and ARI when applying K-means, DBSCAN and agglomerative clustering on the test data with vector coordinates from the vector space trained on Flashback Forum. The two graphs on the first row are the results of clustering all words from all categories — 20 words per category giving 200 words in total. The two graphs on the second row are the results of clustering a subset of the words from all categories — every category is of different size ranging from 3 to 20 words. The results show that for balanced data K-means and agglomerative clustering scores higher than DBSCAN and with the unbalanced data DBSCAN scores slightly higher than the others.

Figure 10 and figure 11 shows the results of the experiments testing different fixation scoring algorithms. They each contain six boxplots, one for each scoring algorithm. Figure 10 shows the results from using ten categories and figure 11 shows the results from using seven categories.

The variance in most boxes is high but gets smaller when only seven categories were included. In general, the median gets lower as the number of words in the primary category decreases.
Figure 10: Boxplots of the results from the six scoring algorithms detailed in section 5.5. For these results all test data were included.

Figure 11: Boxplots of the results from the six scoring algorithms detailed in section 5.5. For these results only seven categories from the test data were included.
7 Experiment Conclusions

The initial run comparing the vector spaces trained on Flashback Forum and Familjeliv showed better performance for Flashback Forum, see figure 8. This shows that the vector space supplied is usable, fits the data and is meaningfully different from a vector space trained on data from a different source.

Figure 9 shows the results of the performance when applying K-means, DBSCAN and agglomerative clustering on the test data with vector coordinates from the vector space trained on Flashback Forum. The results when using ARI and AMI as evaluation measures show that for the balanced data, K-means and agglomerative clustering performs better than DBSCAN while DBSCAN performs slightly better when applied to the unbalanced data. Furthermore we can see that for the unbalanced data and for a few number of clusters, DBSCAN performs better while DBSCAN and K-means produce similar results for a larger number of clusters, using ARI and AMI evaluation measures.

Curiskis et al. (2020) obtained AMI average results up to 0.684 and ARI average results up to 0.708 when evaluating different clustering methods on Twitter data and Reddit data. A quick comparison with my results shows that the algorithms performed comparatively well.

In the Flashback Forum user data we can expect the data to be unbalanced — i.e. look more like the unbalanced test data. Furthermore, when finding fixation, the most interesting users are the ones who talk a lot about one topic and less about other topics — i.e. unbalanced data with a lower number of clusters. Another aspect is that K-means and agglomerative clustering require the user to specify a number of clusters as input, which is not the case with DBSCAN — in the test data, the number of categories was known, but in the wild the number of categories is unknown. Therefore, DBSCAN was concluded to be a good choice to use for the automated method — it produced the best results for unbalanced data with a few clusters while providing satisfactory results overall.

For all scoring algorithms, when having only two data points in the biggest cluster — i.e. two words per category — the majority of the scores were zero. This is positive since this implies that choosing a few random words from many categories gives a low score. Figure 11 shows that this is not the case when the experiment was run with seven categories, which is reasonable since the experiment still uses 20 words even if there only is seven categories, i.e. some of the categories will have more than two words in it. Therefore the results of seven categories can not be completely trusted when the number of words in the primary category are low.
In figure 10, in cases with all 20 words in the primary category, the scores are high, which is the expected outcome. However, the variance is also very high, with some examples still given a very low score. This is because of an overlap in the test data, see figure 7. In figure 11 three categories that gave a bad clustering result, i.e. the clusters did not correspond well to the predefined categories, were removed to improve the separation in the test data and in the results we can see that a lot of the variance disappeared.

Comparing the different density metrics described in section 5.5, the top row uses $D_1$ and the bottom row $D_2$. The boxplots in the top row of figure 10 have slightly bigger variance than the ones in the bottom, while they also have much higher scores, the median value is also higher. For a good scoring algorithm, there should be a big difference in score between the number of words — this was better achieved by the top row so the bottom row was discarded: scoring algorithms 2, 4 and 6. Looking at algorithm 1, 3 and 5 we can see that both algorithms that use the error function produce higher top results. Scoring algorithm 3 produces the desired effect of giving high scores when having a big main cluster and low scores when having very small clusters. Scoring algorithm 5 gave higher or the same scores for all number of words compared with scoring algorithm 1, while algorithm 3 raised scores for big clusters and lowered for small clusters — note the differences in the results for number 4. For bigger clusters, the variance in algorithm 3 is high but the median is closer to the top value, e.g. half of the scores for 20 words have higher score than 0.6 even though the lowest score is around 0.1.

Because of the reasons detailed above, scoring algorithm 3 was chosen to be the scoring algorithm for the automated method for finding fixation.

8 In The Wild

When finished, the algorithm was tested in the wild on documents that contain written text from 100 Flashback Forum users, where each document contains all text written on flashback before the year 2020 by one user. The results show fixation scores from 0.0285 to 0.06932. Most users got a lower score and a few ones got higher. The results are shown below, in figure 12 as a histogram.
9 Discussion

Fixation is one aspect in threat assessment of individuals. A fixated person is focused exclusively on one topic or object, on a level where it impacts everyday life. Alongside other behaviours, a high level of fixation can be an indicator that someone will perform violent acts in the future (Meloy, Hoffmann, Guldimann, et al. 2012; Mullen et al. 2009).

Natural language processing, vector spaces and clustering algorithms were all used to construct a method that can measure fixation. The method takes text as input and outputs a value between 0 and 1, where a higher value...
indicates higher signs of fixation. See figure 4 for a step by step picture.

The results in the wild shows satisfactory results. The results of 100 users shows that most of them are not fixated and a few ones got a higher fixation score. The five users who got the highest fixation score were studied more closely. The 100 tf-idf terms that were used by the algorithm and clustered into different clusters were manually studied, including the outliers. For these five users the majority of the words were in the same cluster — a “main cluster”. There were a few words that could have been in a different cluster and some words that could have been sorted out in the preprocess state with a more accurate preprocessing. Despite this, it is clear that these five users have one subject that they talk about more than other subjects. When taking a closer look at the five users with lowest fixation score it can be seen that their tf-idf terms got clustered into multiple clusters (seven to 12) and many terms, more than half of them, were outliers. Even though there are many nonsense words and the clusters could have been grouped together in a different way, there is no clear main subject to find throughout the 100 terms, why the low score seems accurate.

The method could be used as a rough detection system to find users on Flashback Forum that shows signs of fixation, or determine levels of fixation on a specific user. If the models are trained with data from other online spaces and/or languages the method should work the same way, though this is not tested in this work. It should not be used on its own for finding fixation in threat assessment as it is not always correct and does not take into consideration what the object of fixation is.

10 Future Work

In this thesis a first attempt to measure fixation on the internet has been constructed and presented, there is much that can still be done and improved. These are some ways to expand on my work:

1. One could look for more specific combinations of key terms and named entities. This could help distinguish between fixation that could be a threat to others and “innocent” fixations.

2. Do a more thorough preprocessing for a cleaner clustering.

3. Analyse emotions in the text. Is there an angry emotional undertone?, since this is one way to measure fixation according to Meloy, Hoffmann, Guldimann, et al. (2012).
4. Take time into consideration — does a user talk more and more about a topic or subject over time? Does it become more negative and/or more strident over time? If yes, it can be a sign of higher fixation (Meloy, Hoffmann, Guldimann, et al. [2012]).

5. Test the method on other online spaces and other languages. E.g. use a vector space trained on data from Twitter and analyse users on Twitter.

References


Bird, Steven, Ewan Klein, and Edward Loper (2009). Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc.


Meloy, John, Jens Hoffmann, Karoline Roshdi, et al. (Sept. 2014). “Some Warning Behaviors Discriminate between School Shooters and other students of concern”. In: *Journal of Threat Assessment and Management* 1, pp. 203–211. DOI: 10.1037/tam0000020


Rehurek, Radim and Petr Sojka (2011). “Gensim–python framework for vector space modelling”. In: *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic 3.2.*


Appendix

A Test Data: Clustering Algorithm

1. **Weapon**: automatvapen, beretta, gevär, glock, hagelbōssa, hagelgevär, jaktgevär, jaktvapen, kaliber, karbin, kulspruta, luftgevär, pistol, revolver, skjutvapen, spjut, svärd, uzi, vapen, ālgstudsare

2. **Military**: civilförsvar, flotta, försvarsmakten, general, gmu, hemvärnet, marinen, menig, militären, militärmakt, militärtjänst, mönstring, officerare, sergeant, soldat, underrättelsetjänst, veteran, värnplikt, värnpliktiga, yrkesmilitär

3. **Exercise**: axelpress, benpress, bänkpress, chins, crossfit, crosstrainer, dips, gym, hantelrodd, hantlar, kardio, kettlebells, knäböj, löpning, marklyft, muskelgrupper, styrketräning, träning, tyngder, vikter

4. **War**: anfalla, anfallskrig, angrepp, annektera, bomba, bombning, inbördeskrig, invadera, invasion, krig, kråga, motangrepp, närkamp, närstrid, ockupation, spränga, storkrig, strid, strida, världskrig

5. **Nazi**: antisemitism, arisk, auschwitz, etnicitet, förintelsen, hitler, holocaust, hudfärg, högerextremism, judar, nasse, nationalism, nazism, nazist, nynazist, ras, rasbiologi, rasgrupp, skinnskalle, treblinka

6. **Drugs**: benso, cannabis, centralstimulerande, droger, ecstasy, fentanyl, hallucinogen, hasch, heroin, knark, kokain, lsd, marijuana, mdma, metamfetamin, meth, narkotika, opiat, rohypnol, spice

7. **Politics**: alliansen, anarkist, borgerlig, centerpartiet, fascist, feminist, kommunist, liberal, löfven, marknad, marx, miljöpartiet, moderat, parti, regering, riksdag, socialdemokrat, sossarna, val, vänsterpartiet

8. **Religion**: bibeln, buddhism, gud, hinduism, islam, jesus, judendom, katolik, koranen, kristen, kristendom, kyrka, moské, muhammed, muslim, protestant, påven, rabbin, synagoga, torahn

9. **Immigration**: anhöriginvandring, assimilering, asyl, asylsökande, bidragsparasiter, ensamkommande, flyktning, flyktingar, flyktingström, gränser, integration, invandrar, invandring, migration, mångkultur, nysvenskar, papperslös, skäggbarn, syrien, uppehållstillstånd
10. **Computers**: bredband, cache, cpu, dator, gb, hacker, hårddisk, internet, ip, kryptering, mac, nätverk, operativsystem, pc, processor, programmering, ram, router, wifi, windows

**B Test Data: Scoring Algorithm**

1. **Medicine**: vårdcentral, läkare, sjukhus, sjukdom, halsont, akuten, influensa, hjärtinfarkt, stroke, cancer, blodtryck, puls, hjärtfrekvens, förkylning, hosta, snuva, smärta, blodpropp, andningssvårigheter, operation

2. **Religion**: kyrkan, kristna, ortodoxa, muslimer, moské, tempel, gud, ateism, böön, präst, allah, eid, jesus, sunni, shia, vishnu, buddha, zen, taoism, hinduism

3. **Environment**: växthusgaser, havsnivå, miljö, klimatförändringar, koldioxid, syre, utsläpp, kärnkraft, växthuseffekten, klimat, uppvärmning, naturkatastrofer, thunberg, kolkraftverk, förnybar, olja, ekosystem, övergödning, klimatavtal, miljövänlig

4. **University**: vetenskap, forskare, universitet, professor, forskning, rapporter, avhandling, lektor, undervisning, föreläsning, tenta, examen, munta, seminarium, godkänd, blanka, plugga, tentaplugg, exjobb, handledare

5. **History**: stormaktstiden, romarriket, antiken, medeltiden, forntiden, kolonier, germaner, istiden, hellenistisk, franker, förhistorisk, bronsåldern, vasa, sturar, kalmarunionen, fälttåg, hansan, dynastin, världskriget, kartago

6. **Crime and punishment**: fängelse, cell, straff, dödsstraff, rån, mishandel, livstid, åklagare, bōter, rättvisa, dagsböter, domstol, häktad, övervakning, stenkastning, polis, anhållen, polisutredning, mord, dräp

7. **Football**: allsvenskan, zlatan, fotboll, elfsborg, aik, liverpool, juventus, barcelona, ronaldo, arsenal, chelsea, ifk, straffläggning, offside, linjedomare, hörna, övertid, vm, em, ligan

8. **Economy**: aktier, avkastning, sparande, inkasso, räkningar, kronofogden, konto, amortering, bitcoin, valuta, fonder, bokföring, pension, skatt, bolån, skuldbesparing, moms, lön, ränta

33
9. **Drug**: medicin, värktablett, smärtstilleande, ssri, benzo, lugnande, antidepressiva, neuroleptika, antiepileptika, biverkningar, centralstimulerande, paracetamol, ibuprofen, receptbelagt, receptfritt, bipacksedel, fass, janus, apoteket, antibiotika

10. **Video games**: playstation, xbox, nintendo, handkontroll, mario, sonic, halo, doom, steam, spel, gamer, dota, cod, esport, twitch, mmorpg, wii, fps, pvp, rts