University of Pisa
Department of Informatics
Master's Degree Program of Data Science and Business Informatics

---

DATA MINING: FOUNDATIONS
A.Y. 2021-22

Knowledge Discovery from Dataset
Glasgow Norms

Mariami Narchemashvili 634789
Umberto Lamia 517110
Stefania Occhiuto 544326
Daniel Vigilio 645166

The following report is analyzing two datasets, one representing 4682 English words and the other one specifying 379 ambiguous words from the first dataset which have several meanings. In total both sets are composed of 5553 English words, which are characterized by the following psycholinguistic dimensions: length of the words, arousal, valence, dominance, concreteness, imageability, familiarity, age of acquisition, semantic size, gender, and web corpus frequency. To transform the raw input data into an appropriate format for subsequent analysis we go through following steps.

1 DATA UNDERSTANDING AND PREPARATION

1.1 DATA SEMANTICS

This section of the report represents two tables for both datasets – (1) Words Glasgow & (2) Words Polysemy which are provided as an input data. We can say that both datasets are high-dimensional as the number of attributes is quite high.

Through the variable "Name" we indicate the name of the attribute/variable we are considering. In total we have 13 attributes (in dataset (1)), with the following table we demonstrate each feature with its description, type, and domain.

The first three dimensions (arousal, valence, dominance) are used to measure the emotional and psycological empact of a word. All the other variables indicate the grade of knowing the word and the use of it.

NAME	Description	TYPE	DOMAIN
Word	English words	Categorical - Nominal (string)	4683 (number of records)
Length	Word length	Numerical - Discrete (integer)	[2; 16]
Arousal (AROU)	Measure of excitement (excitement, calmness)	Numerical - Contininuous (float)	[2.057, 8.177]
Valence (VAL)	Measure of value or worth (positive, negative)	Numerical - Contininuous (float)	[1.03, 8.647]
Dominance (DOM)	Measure of the degree of control (dominant, controlled)	Numerical - Contininuous (float)	[1.941, 8.371]
Concreteness (CNC)	Measure of how concrete or abstract something is (concrete, abstract)	Numerical - Contininuous (float)	[1.636, 6.938]
Imageability (IMAG)	Measure of generating a mental image of something (imageable, unimageable)	Numerical - Contininuous (float)	[1.737, 6.941]
Familiarity (FAM)	Measure of how familiar a word is (familiar, unfamiliar)	Numerical - Contininuous (float)	[1.647, 6.939]
Age of acquisition (AOA)	Measure of the age at which a word was initially acquired	Numerical - Contininuous (float)	[1.219, 6.971]
Semsize (SIZE)	Measure of magnitude (big, small)	Numerical - Contininuous (float)	[1.375, 6.912]
Gender (GEND)	Measure of a word considered to be associated with male or female behavior (masculine, feminine)	Numerical - Contininuous (float)	[1.0, 6.971]
Polysemy	Measure of semantically ambiguous words which convey multiple meanings (homographs)	Categorical - Binary (integer)	{0,1}
Web corpus frequency (WCF)	Measure of frequency of a word in Google Newspapers Corpus	Numerical - Discrete (integer)	[12770, 2022459848]

Out of all features we distinguish 9 variables as 9 dimensions of each record. Those nine psycholinguistics variables are continuous: Arousal, Valence, Dominance, Concreteness, Imageability, Familiarity, Age of Acquisition, Size, and Gender.

The following table represents dataset (2) – 'Words Polysemy', where in total there are 11 attributes (columns).

NAME	Description	TYPE	DOMAIN
Word	English words	Categorical - Nominal (string)	872 (number of records)
Length	Word length	Numerical - Discrete (integer)	[2; 16]
Arousal (AROU)	Measure of excitement (excitement, calmness)	Numerical - Contininuous (float)	[2.057, 8.177]
Valence (VAL)	Measure of value or worth (positive, negative)	Numerical - Contininuous (float)	[1.03, 8.647]
Dominance (DOM)	Measure of the degree of control (dominant, controlled)	Numerical - Contininuous (float)	[1.941, 8.371]
Concreteness (CNC)	Measure of how concrete or abstract something is (concrete, abstract)	Numerical - Contininuous (float)	[1.636, 6.938]
Imageability (IMAG)	Measure of generating a mental image of something (imageable, unimageable)	Numerical - Contininuous (float)	[1.737, 6.941]
Familiarity (FAM)	Measure of how familiar a word is (familiar, unfamiliar)	Numerical - Contininuous (float)	[1.647, 6.939]
Age of acquisition (AOA)	Measure of the age at which a word was initially acquired	Numerical - Contininuous (float)	[1.219, 6.971]
Semsize (SIZE)	Measure of magnitude (big, small)	Numerical - Contininuous (float)	[1.375, 6.912]
Gender (GEND)	Measure of a word considered to be associated with male or female behavior (masculine, feminine)	Numerical - Contininuous (float)	[1.0, 6.971]

1.2 DISTRIBUTION OF THE VARIABLES AND STATISTICS

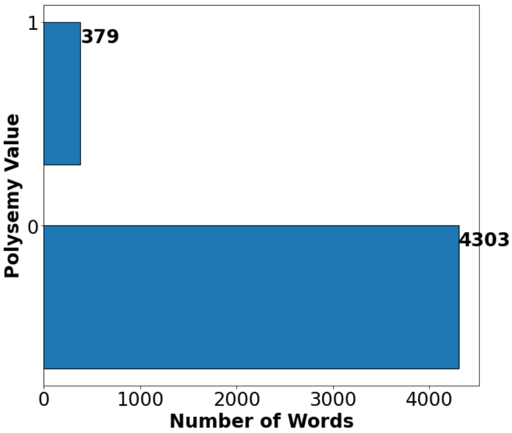


Figure 1: Distribution of Polysemy Variable

In the following section the distribution of variables are represented with the help of different kinds of vizualization tools.
The first variable represented is 'Polysemy', that shows the ambiguouty of the word and that we consider as a target or also known as dependent variable. It is represented as a boolean variable giving two values 0 and 1 (word has one meaning or several meanings respectively). In total there are 4303 words that has polymesy equal to 0, and 379 equal to 1.

Out of ambiguous words there are 289 words with two different meanings; 69 – three; 19 – four; 2 – five. ***Figure 1*** and ***figure 2*** represent the distribution of words in dataset 1 with respect to 'Polysemy' attribute. From figures it can be assumed that the dataset is quite unbalanced with respect to the target variable.

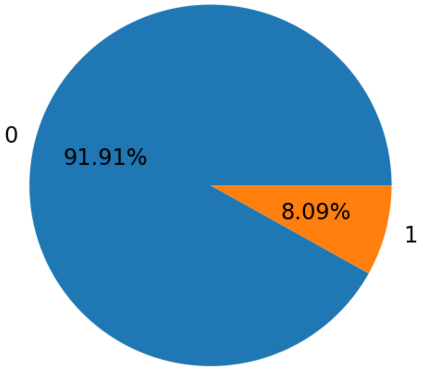


Figure 2: Distribution (%) of Polysemy

The following are the visualisation of density plots in order to learn the statistics and distribution of variables with respect to our target variable ‘Polysemy’. The dotted lines represent the mean values in both ambiguous and not ambiguous words’ sets (respectively same colors).

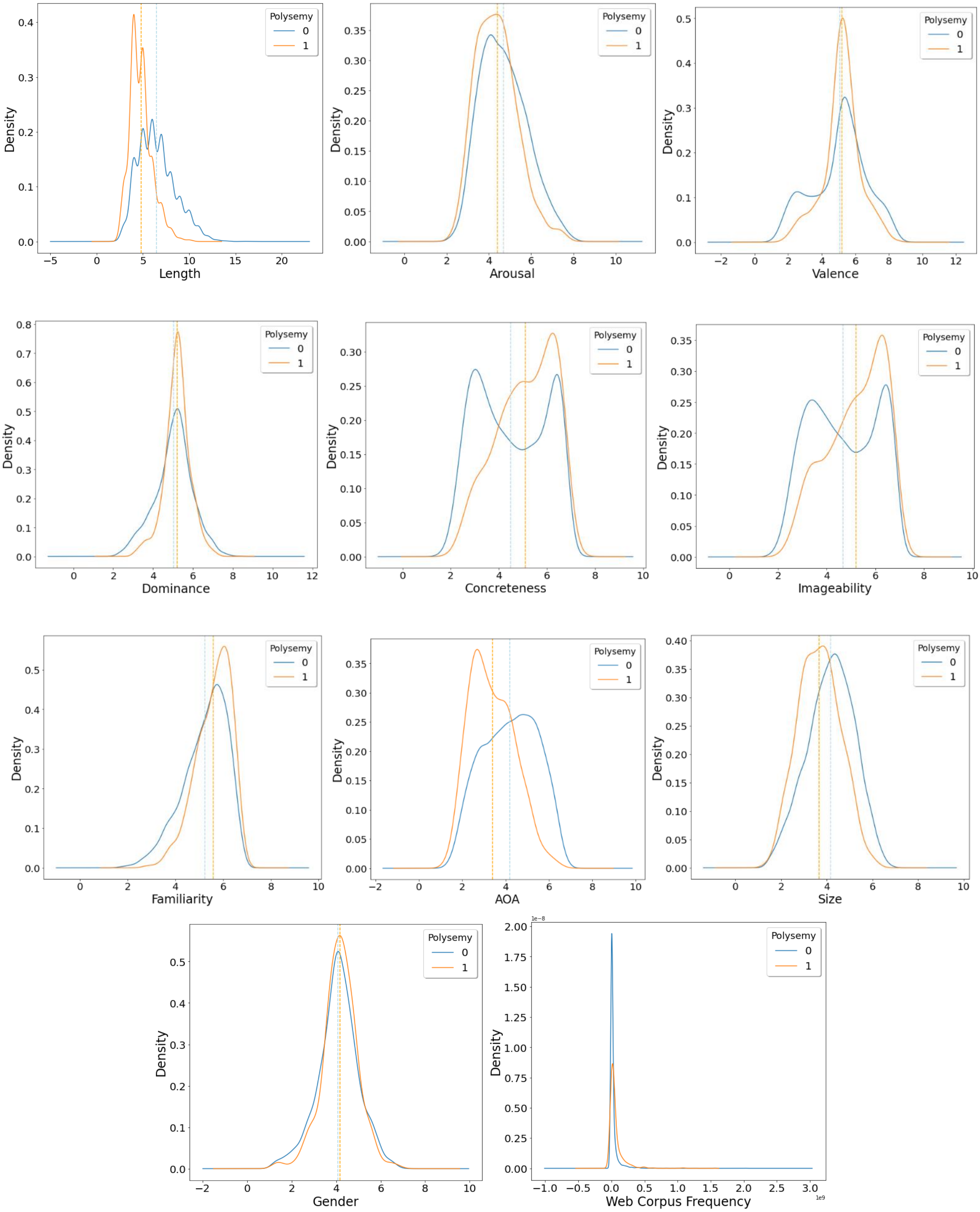


Figure 3: Distribution of all attributes w.r.t Polysemy variable

According to ***figure 3***, it is quite obvious that some of the attributes, such as: Concreteness, Imageability, Web Corpus Frequency are not well-disctributed across the dataset 1.

1.3 ASSESSING DATA QUALITY

The following section studies and evaluates data quality. In order to assess the quality of the dataset it explores and handles missing values, outliers, and any other semantic errors or inconsistencies.

Duplicate Data

First, we investigated duplicate records and we found out that in our data we do not face the issues regarding to it.

Syntactic Accuracy

Another thing that we checked was accuracy of the variable 'length', making sure the closeness of measurement to the true value. We counted actual length of the words' strings. We compared string lengths to our attribute 'length' in order to make sure that the data represented with this attribute does not contain any inaccuracies. Finally, we found zero errors in these variable.

Semantic Accuracy

After that, we checked the number of ambiguous (polysemous) words. For this part we used both datasets. In dataset (2) we separated the description part from the actual word and counted the unique values. Finally, we compared the words from dataset (1) with the polysemy value equal to 1 and unique words from the dataset (2). On the first phase we faced the difference between the total number of words. However, we found out that the difference was caused only because of the word 'apple' which was represented in the second dataset in two differenc ways (with capital letter representing the brand - Apple).

With the help of previous checks of nominal, discrete and binary variables, we can say that in the dataset (1) we do not face any major semantic errors.

As for continuos attributes we checked for missing values and outliers, which is represented in the following sections.

Missing values

For this part we checked all the continuous attributes and detected some missing values. We found them in only one attribute which represents 'web corpus frequency'. The number of those missing values was 14, which is not significant with respect to the number of all records. As there are several strategies (elimination or substitution) to tolerate poor data quality caused by missing values, in the following sections we show how we handled them.

Outliers

The outliers are anomalous objects that have different characteristics from all the others in the dataset. They have an unusal value of an attribute from the usual values of that attribute. In order to find those anomalous values we applied several methods.

In order to detect outliers in our dataset we used boxplots and z-score normalization for all the attributes. In total we found 325 outliers with respect to all the attributes. Only five of them were the same as missing values from web corpus frequency attribute. We decided to remove all of the outliers from the original dataset for further improvements.

1.4 VARIABLE TRANSFORMATIONS

This section represents some transformations done on dataset 1 in order to represent data in a scale considered more suitable and well-distributed.

First transformation that was applied is logarithm transform of the attribute 'web corpus frequency' in order to stabilize the variance. As this variable is represented as high continuous numbers and is poorly distributed, it was not convinient to analyze it with other attributes. In order to visualize data with histograms we used Sturge's rule to define the number of bins (k = dlog2(n)+1e).

After logarithm transformation and elimination of outliers, we filled remained missing values with mean. As it is visualized on ***figure 4***, there is no significant difference in the two ways of substitution (with mean or with median), we made the decision easily.

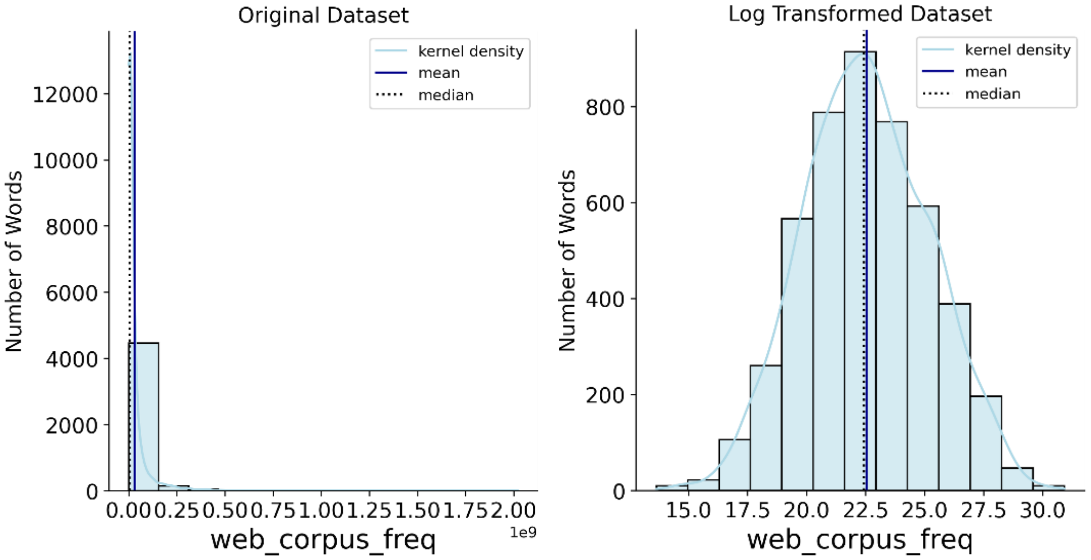


Figure 4: Distribution of Web Corpus Frequency before and after transformation

Other types of transformations have been tried, such as square root and reciprocal. Square root transformation has been applied to all the attributes seperately and the significance was found in the attribute 'arousal', as it improves the distribution of variable. ***Figure 5*** is the demonstration of square root transformed attribute. Even though, reciprocal transformation has been applied to the attributes, it did not seem helpful for further improvements so we decided to not use it.

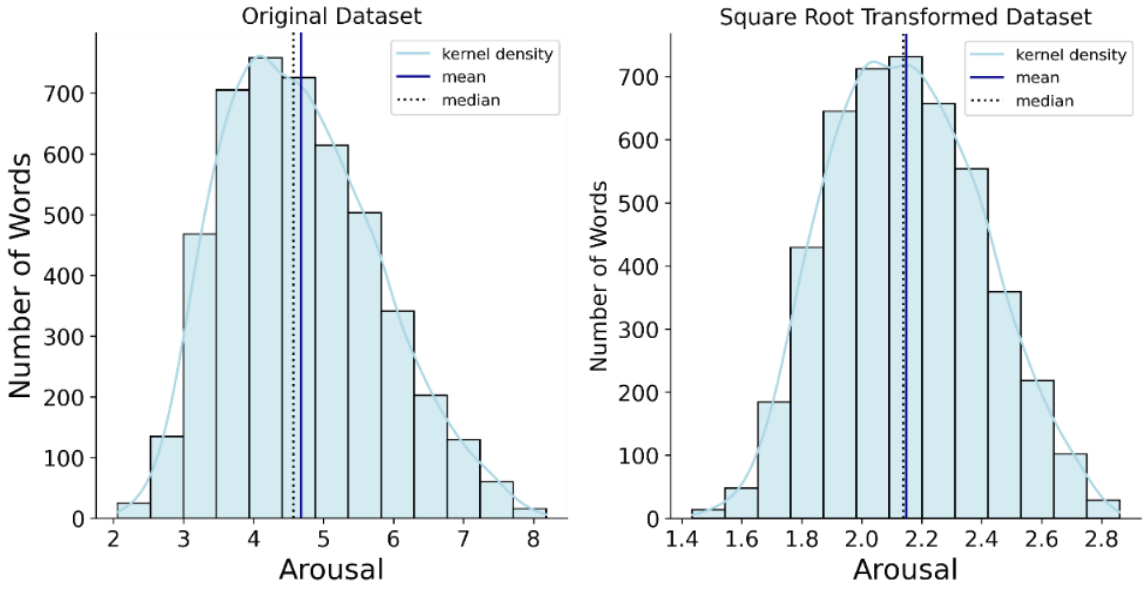


Figure 5: Distribution of Arousal variable before and after transformation

1.5 PAIRWISE CORRELATIONS AND EVENTUAL ELIMINATION OF VARIABLES

It is possible to compute measures of correlation between attributes to confirm expected dependencies or to discover unexpected correlations between attributes.

For this part of the report Pearson's Correlation Coefficient was used, which is a measure of a linear relationship between two normally distributed variables. When the variables are not normally distributed or the relationship between the variables is not linear, it may be more recommended to use the Spearman rank correlation method. Spearman Rank Correlation Coefficient has also been examined which intend to measure monotonous correlation between attributes where the function does not have to be linear.

After applying both correlation methods to the dataset 1, just a little difference has been found between the results of these two. It was quite expected because after the elimination of semantic inacuracies and transformation of variables, the data became better distributed. ***Figure 6*** demonstrates the colleration between all the attributes.

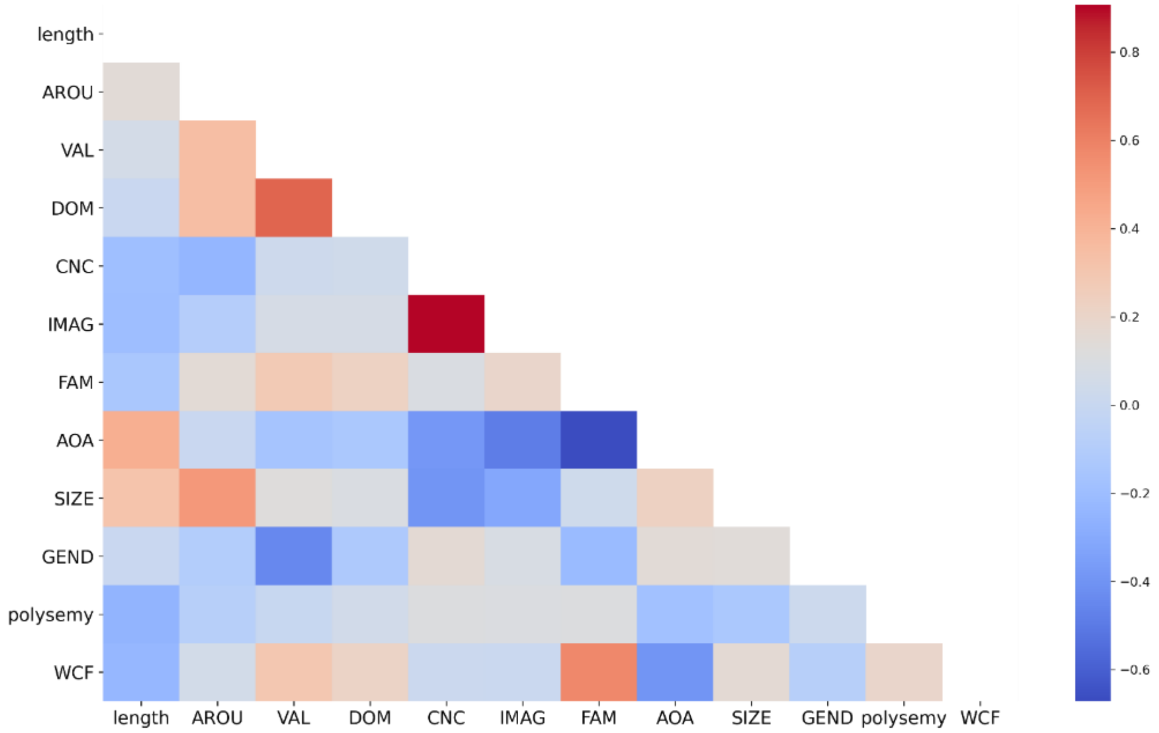


Figure 6: Correlation between all Attributes

Figure 7 represents the attribute pairs which were highly correlated to each other.

Pairs of highly correlated attributes are the following: [CNC & IMAG] r = 0.91; the more concrete a word is, the easier it is to imagine; [VAL & DOM] r = 0.69; the more positive a word is, the more it provokes feelings of dominance; [FAM & AOA] r = – 0.67; the more familiar a word is, the earlier that word was learned; [SIZE & AROU] r = 0.51; the bigger the object or concept is to which a word refers, the more arousing the word is; [FAM & WCF]; We decided to eliminate one from each paired attribute (the ones of which variable distribution was wors e).

Redundant Features

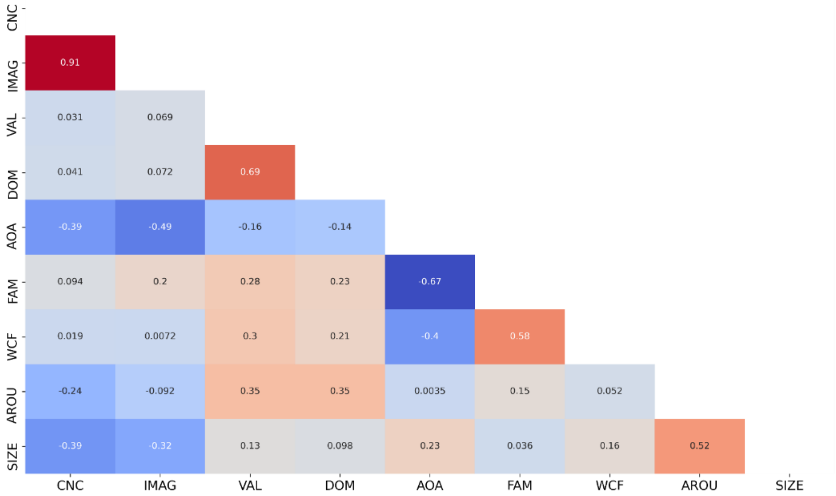


Figure 7: Highly correlated pairs

2 CLUSTERING

Clustering exploits similarities between the data to be analyzed, similarities that can be of various nature but which are essentially a distance between the dataset points. Different algorithms can be used for clustering analysis, but the following sections will examine K-Means, Density-based and Hierarchical clustering. Before applying any of those, it is required to go through some preprocessing steps.

Choice of Attributes

For the clustering algorithms the categorical and discrete attributes (such as 'word', 'polysemy', 'length') are dropped from the set and the further analysis are carried out with other variables. To increase the performance of the clustering algorithms it has been decided to normalize the ranges of attributes. In order to complete this step multiple kinds of scalers can be used, but in this case only the most widely used ones (Standard, Minmax, Robust) are discussed. **Standard Scaler -** Scaled between std ranges; **Minmax Scaler -** Scaled between the range [0, 1]; **Robust Scaler -** Works good on outliers and consideres interquartile ranges instead of std. ***Figure 8*** represents standard scaled attributes in dataset 1. Even though all three of the scalers have been examined, the following analysis have been carried out with standard scaled attributes.

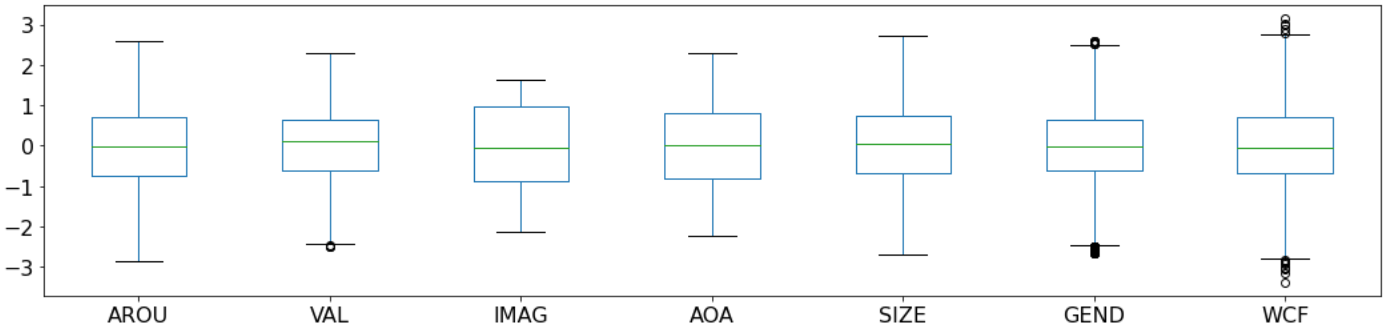


Figure 8: Boxblot of Standard Scaled Attributes

2.1 CLUSTER ANALYSIS BY K-MEANS

K-means is one of the most widespread and best performing clustering algorithms. Its is prototype-based, partitional clustering technique that attemps to find user-specified number of clusters (k), which are repsresented by their centroids.

Centroids are mean of the points of clusters, not real ones. Initial centroids are often chosen randomly. As an initial phase 3 number for clusters have been chosen. In order to find optimal number of clusters there are two different approaches.

Elbow Method helps to plot the WCSS (Within Clusters Summed Squares) values and selects the point where the parameter value falls more than the previous value. For each point, the Sum of Squared Error (SSE) is the distance to the nearest cluster. To get SSE, we square these errors and sum them. Finally we choose the optimal number of clusters considering the lowest error with respect to lower number of clusters. The lower the SSE the better.

Another method that can help to choose the number of clusters is **Silhouette method**. In this procedure the silhouette coefficient is plotted and the maximum value is selected. The higher the Silhouette Coefficient the better.

The range for the algorithm have been defined as follows: $2 \leq \ k \leq \ \sqrt{n}$, where k is the number of clusters, and n is the the number of records in the dataset. So k's highest value has been computed, which is ~ 66. ***Figure 9*** visualizes the results of SSE and Silhouette Score achieved on standard scaled dataset.

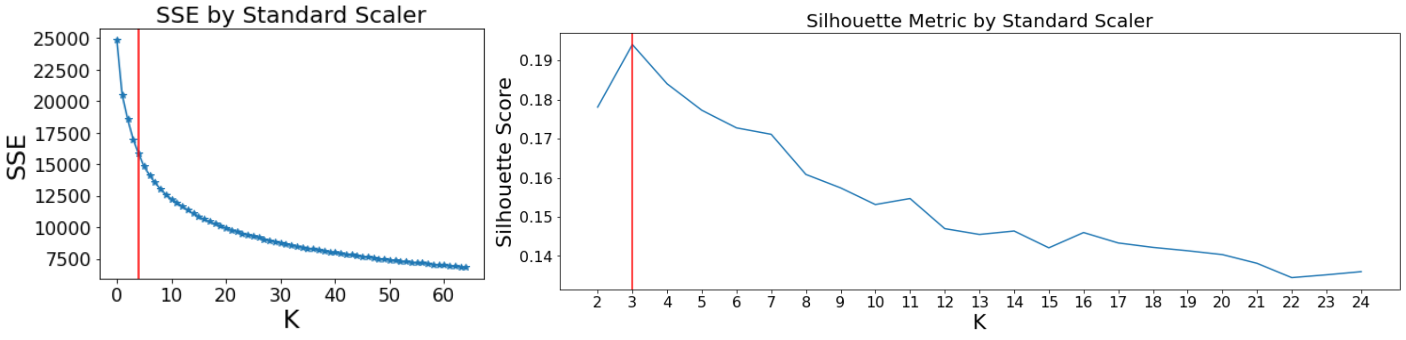
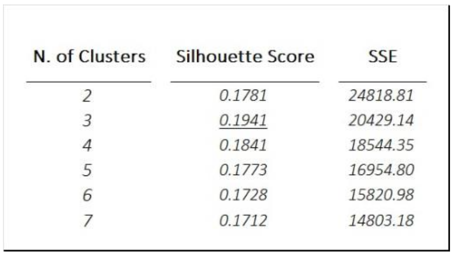


Figure 9: SSE and Silhouette indexes for Standard Scaled Data

Finally, SSE was quite high considering the high-dimensionality of the data. For clustering analysis we get optimal SSE with respect to clusters' number equal to 6 and Silhouette Coefficient with respect to clusters' number equal to 3.
Figure 10 is a demonstration of Silhouette Score and SSE with respect to different number of clusters.



N. of Clusters	Silhouette Score	SSE
2	*0.1781*	*24818.81*
3	*0.1941*	*20429.14*
4	*0.1841*	*18544.35*
5	*0.1773*	*16954.80*
6	*0.1728*	*15820.98*
7	*0.1712*	*14803.18*

Figure 10: Silhouette Score and SSE for different number of clusters

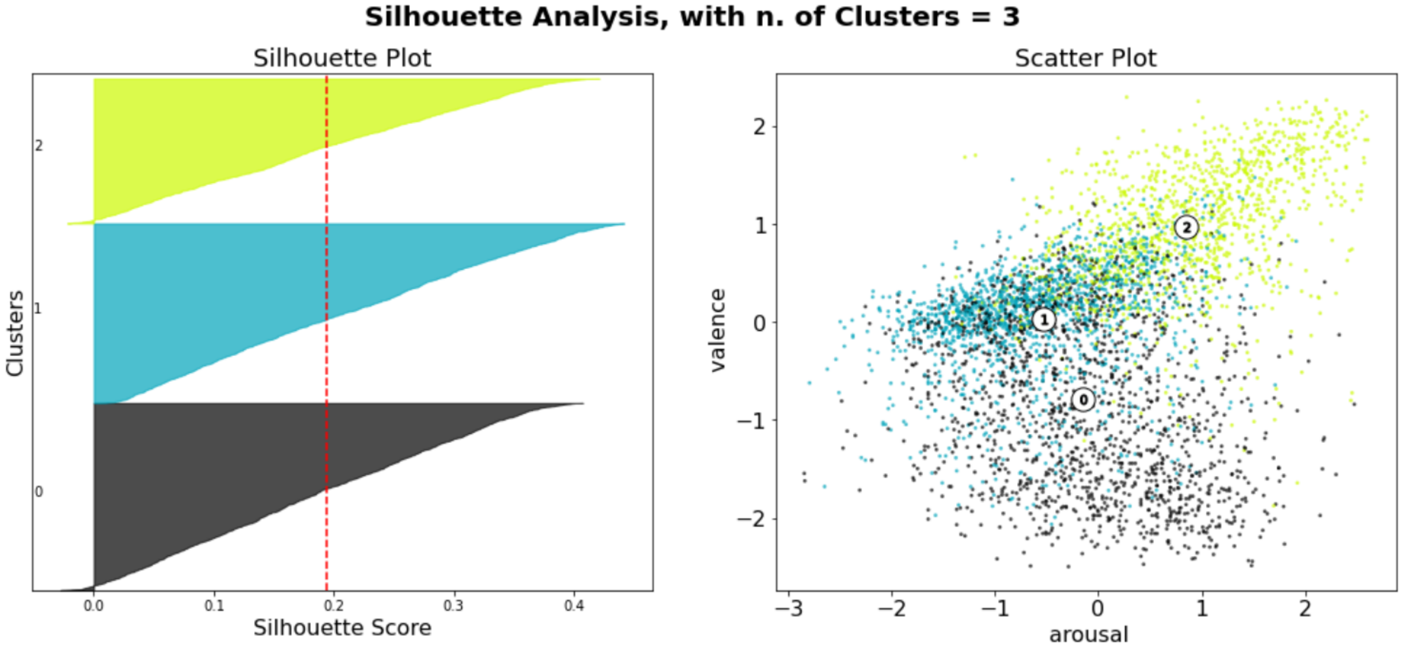


Figure 11: Sillhouette Plot and Scatter Plot for 3 Clusters

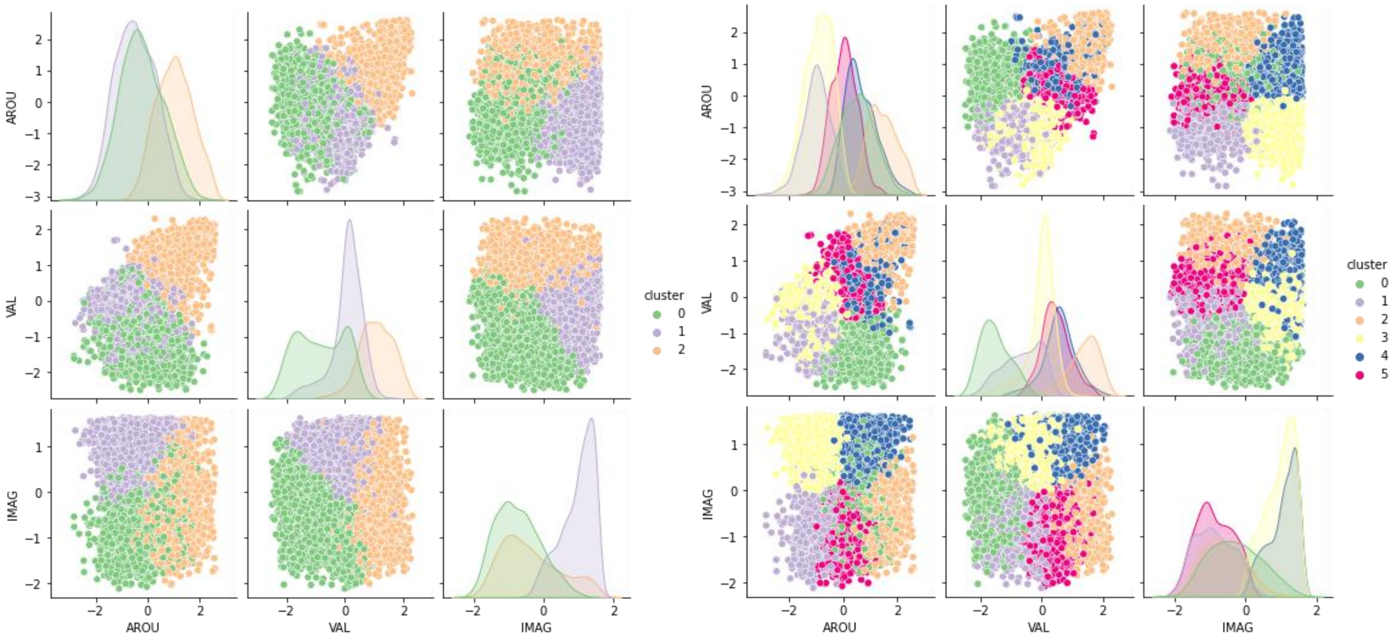
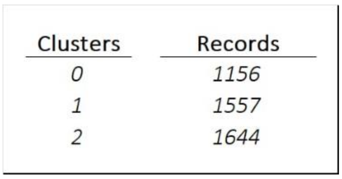


Figure 12: Scatter plots with 3 and 6 number of clusters for 3 chosen attributes

The records were almost equally distributed in each cluster, which states that the k-means algorithm worked well on the dataset 1 and separated the records in a balanced way.



Clusters	Records
0	*1156*
1	*1557*
2	*1644*

Figure 13: Distribution in clusters

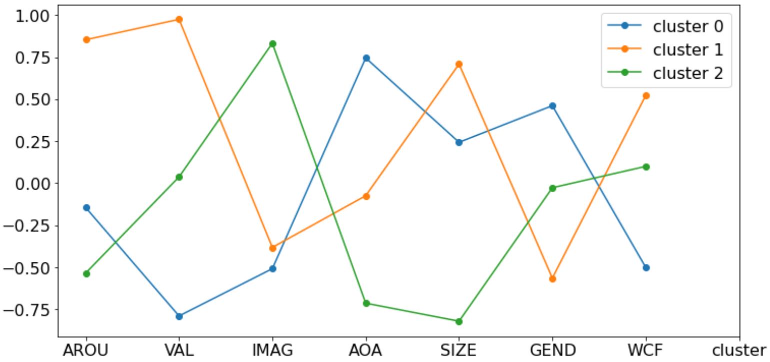


Figure 14: Parallel-Coordinate Plot for 3 clusters

Parallel-Coordinate Plot

In order to see the separation of mean values of each cluster with respect to each attribute, clustered data is visualized with parallel-coordinate plot. Mostly all the attributes' mean values are well-separated in the clusters, except the one for 'Imageability'. *Figure 14* shows the parallel-cordinate plot for the data in 3 clusters.

Barchart

In order to see the distribution of variables with respect to clusters data have been visualized on barcharts. *Figure 15* shows three different barcharts: first two represent polysemy attribute values distribution in three different clusters; whereas the thirds one represents values of 'length' attribute distributed in three clusters.

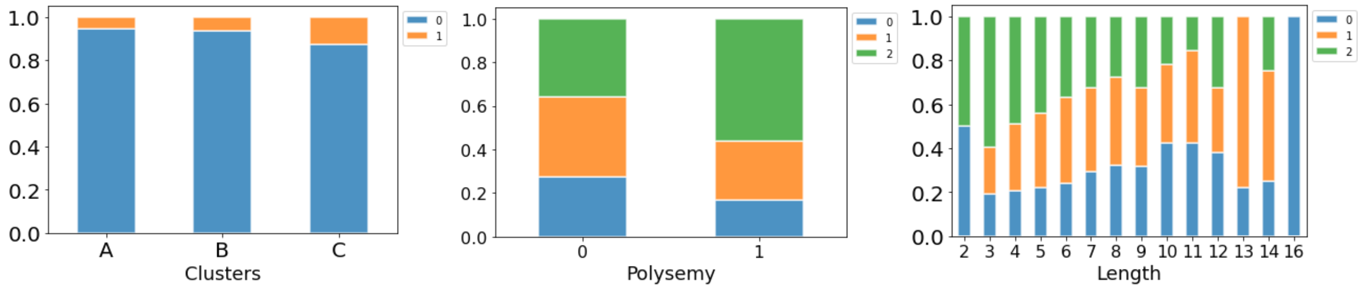


Figure 15: Distribution of polysemy and length attributes in 3 clusters

2.2 ANALYSIS BY DENSITY-BASED CLUSTERING

In the analysis by density-based clustering number of clusters is not defined by us, but two other main parameters which are ε (eps), which defines the radius of neighbourhood around a point and min_samples which defines the minimum number of neigbhours of the point within the radius. With DBscan the actual clusters and the noise points can be visualized. The larger the dataset, the larger min_sample value is needed. It has been mostly used in practice to choose minpts = 2* dimensions, therefore it has been decided to set minpts = 2 * 7 = 14.

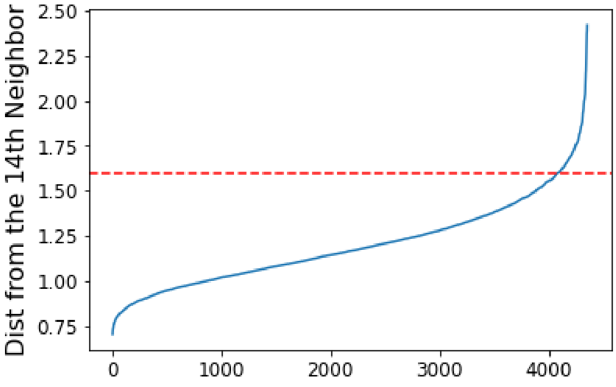


Figure 16: eps distance plot

After setting minimum sample size, the optimal value of eps has been calculated and found in the range [1.5, 1.75], as shown on the *figure 16*. The algorithm with different values of the parameters have been examined for several times and the information have been collected. However, different values of eps, in the range stated previously, did not impact the clusters, as the achieved number of clusters was always one labeled as 0, and noise points labeled as -1. Even though the little change in eps have been efffecting the number of noise points as predicted, it has been decided to keep eps = 1.6

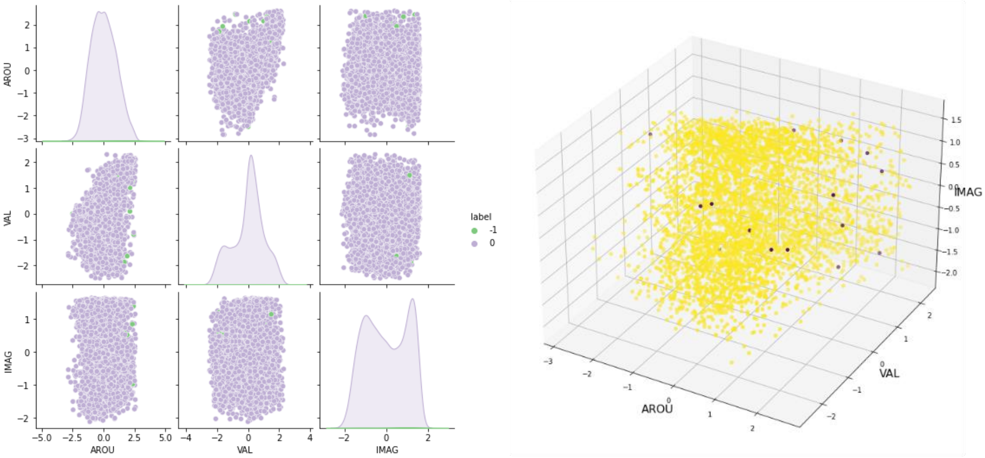


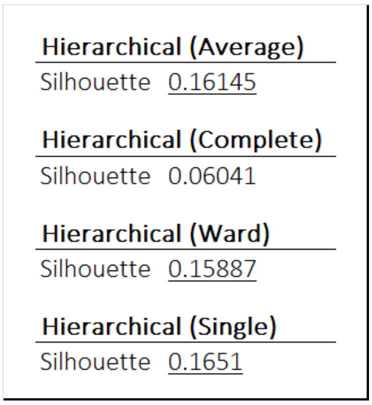
Figure 17: Scatter plots of DBscan

Figure 17 shows 1 cluster and the number of noise points.

2.3 ANALYSIS BY HIERARCHICAL CLUSTERING

Hierarchical Clustering method builds clusters step by step. The final result is visualized as a dendrogram. The following analysis has been conducted on an agglomerative clustering method.

There are four different methods to visualize dendogram: Average (good against noise and outliers, bad with clusters of different number of elements); Complete; Single; and Ward (good against noise and outliers, good for spherical or rounded clusters). Even though all of them have been examined, the best visualizaiton was achieved by the 'single' method. However, single method did not work well on the dataset in terms of cluster distribution. Therefore, it has been decided to chose 'ward method which performed really good in terms of balanced cluster distribution. ***Figure 19*** is the demonstration of best performing hierarchical method on our dataset. It is a visualization of dendrogram using 'ward' method.



Hierarchical (Average)
Silhouette 0.16145

Hierarchical (Complete)
Silhouette 0.06041

Hierarchical (Ward)
Silhouette 0.15887

Hierarchical (Single)
Silhouette 0.1651

Figure 18: Silhouette Score for different hierarchical methods

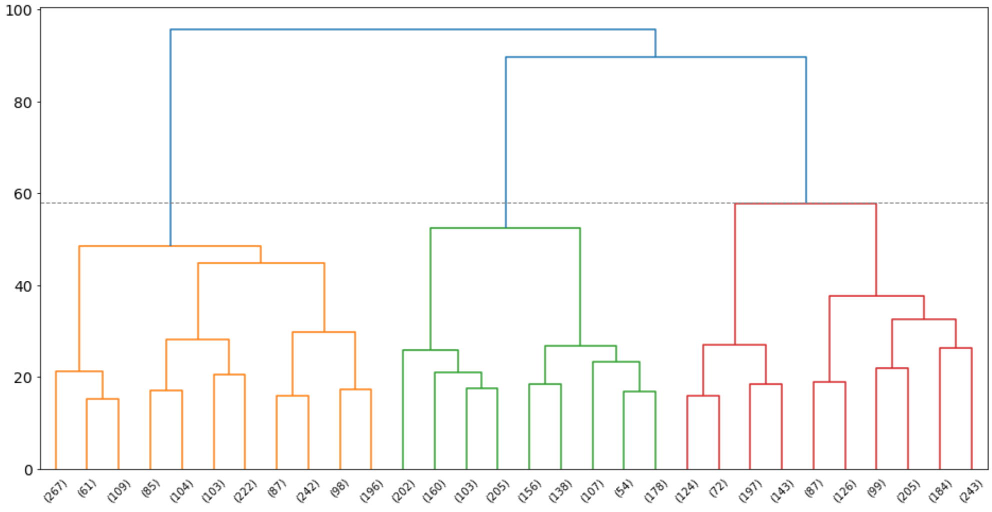
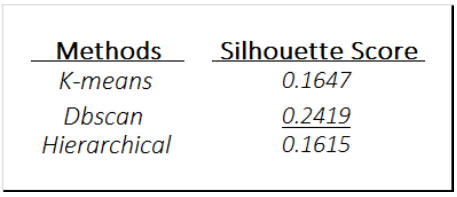


Figure 19: Hierarchical Cluster Plot with ward method

2.4 FINAL DISCUSSION

In order to evaluate the best clustering approach, it has been decided to consider the silhouette score for all clustering algorithms applied in the previous sections. ***Figure 20*** represents the silhouette score for K-means, DBScan and Hierarchical algorithms. The values of the parameters used for each algorithm are: In K-Means - K = 3 used; In DBScan - eps = 1.60 and min_samples = 14 used; In Hierarchical – 'ward method' - Euclidean metric with 3 clusters;



Methods	Silhouette Score
K-means	*0.1647*
Dbscan	*0.2419*
Hierarchical	*0.1615*

Figure 20: Silhouette Score for all three Clustering Algorithms

To conclude, given the results obtained from the previous sections, it is feasible that the clustering method that fits best our dataset is DBscan. As the examination on DBscan identified only one cluster in the whole dataset, it ephesizes the fact that the dataset is hard and unfeasible in terms of applying clustering, due to its high-dimenstionality and low correlation of most of the attributes.

3 CLASSIFICATION

Data for classification task consists of collection of instances/records and each of them is characterized by the tuple (x, y). X – representing attribute/predictor and Y- representing class/response.

Classification model serves two important roles: (1) It is used as a **Predictive Model** to classify unbalanced instances; (2) It is used as a **Descriptive Model** to identify characteristics that distinguish instances from different classes. Below are the steps that will be carried out in the following sections.

- ✓ Learning algorithm: systematic approach to learn classification model on training set.
- ✓ Induction: by using learning algorithm, building classification model.
- ✓ Deduction: applying classification model on test set (unseen test instances).

DATA PREPARATION

Choice Of The Attributes

In order to start working on classification we have to do some preprocessing, such as, dealing with missing values (*section 1.3*) and removing/dropping useless variables (*section 1.5*).

Also, two categorical attributes have been dropped, 'word' and 'polysemy' (target attribute). Final choice of the attributes is the following: Length, Arousal, Valence, Imageability, Age of Acquisition, Size, Gender, and Web Corpus Frequency.

Additionally, the dataset should be prepared, meaning separating training and test sets from the original one. As the dataset is not balanced with respect to the target attribute ('polysemy'), random oversampling is applied, in order to have equal number of records for both, positive and negative classes (respectively polysemy = 1 and polysemy = 0).

After oversampling, data is splitted into training and test sets, and also training set is splitted into train (D.tr) and validation (D.val) sets. D.tr takes 2/3 of a training set and is used to build a model, whereas D.val is used to estimate generalization error.

Even thought oversampling technique has been applied to balance the data, in the following sections the results obtained from both balanced and unbalanced data are presented.

3.1 CLASSIFICATION BY DESICION TREE

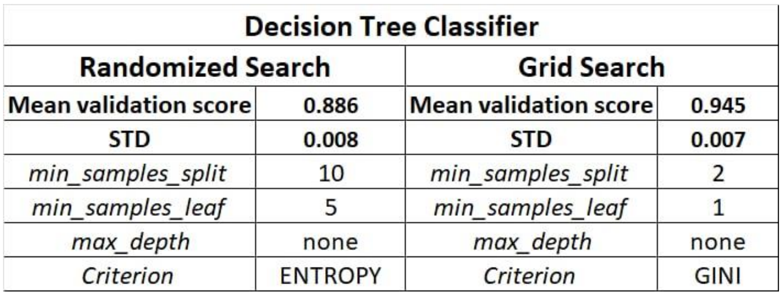
Desicion tree is a clasiffication technique which is structured hierarchically representing organized series of questions and their possible answers.

Identify The Best Parameter Configuration

In order to identify the best parameter configuration before modeling decision tree classifier, it has been decided to apply parameter tuning. Gridsearch and RandomizedSearch methods have been used with an f1-scoring criterion, as f1-score describes the harmonic mean of precision and recall. Parameters are the following:

- **criterion**: The function to measure the quality of a split. Available: GINI, Entropy.
- **max_depth**: The maximum depth of the tree. Range: None + (2, 20). If None, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
- **min_samples_split**: The minimum number of samples required to split an internal node. Range: {2, 5, 10, 20, 30, 50, 100}.
- **min_samples_leaf**: The minimum number of samples required to be at a leaf node. Range: {1, 5, 10, 20, 30, 50, 100}.

Figure 21 represents the suggested criterions to use for Decision Tree classifier on the balanced data. The best combination of maximum depth, minimum sample split and minimum sample leaf has been chosen, in order to avoid underfitting or overfitting in classification model. For building a decision tree the results provided by Randomized Search have been chosen.



Decision Tree Classifier			
Randomized Search		**Grid Search**	
Mean validation score	**0.886**	**Mean validation score**	**0.945**
STD	**0.008**	**STD**	**0.007**
min_samples_split	10	*min_samples_split*	2
min_samples_leaf	5	*min_samples_leaf*	1
max_depth	none	*max_depth*	none
Criterion	ENTROPY	*Criterion*	GINI

Figure 21: DT Classifier parameters

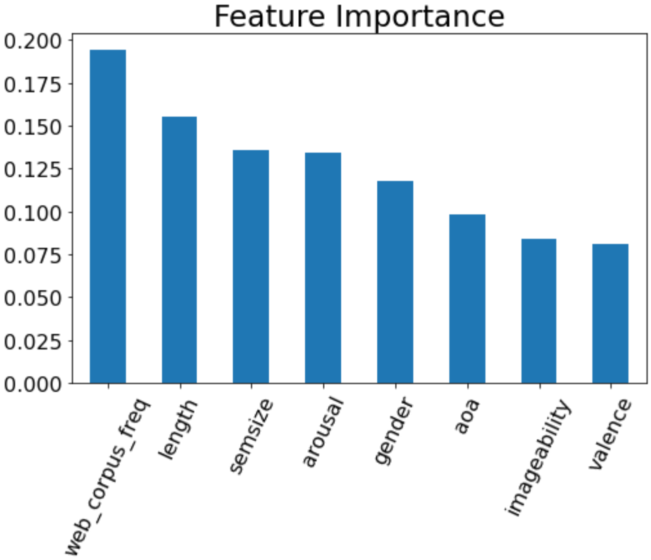


Figure 22: Feature Importance w.r.t balanced data

Feature importance have been measured for splitting. ***Figure 22*** demonstrates feature importance in an ordered way. In this graphical representation we can see which attributes play the most important role for the further analysis. Most important features can also be identified by simply looking at the decision tree provided below built on balanced data.

Figure 23 represents the first levels of the Decision Tree considering the previous features and parameters. Each node of the tree represents: test condition attribute, the index of Entropy (impurity measure of target class distribution), total number of samples and the numbers of each target instances (positive and negative) distributed for the child nodes. For example, the first node's entropy measure is equal to 1, which means that the polysemy values (0, 1) are equally distributed in the child nodes. In the next nodes, we can see that criterion 'class' represents the boolean values correspondingly to target attribute, the one which is superior to the other considering the number of records.

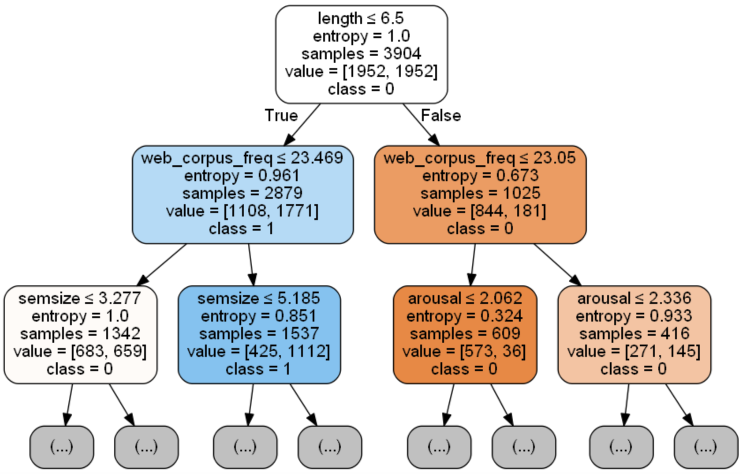


Figure 23: Decision Tree on balanced data

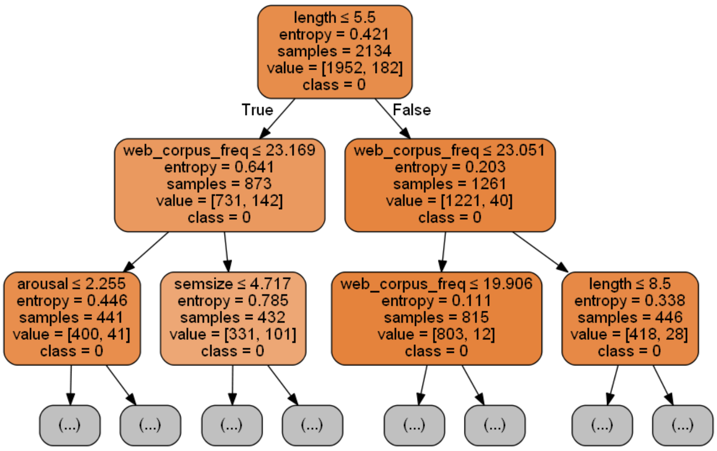


Figure 24: Decision Tree on unbalanced data

For the unbalanced data same steps have been repeated, using the same algorithms that were used to select the best hyperparameters, measuring the importance of the features, and finally implementing Decision Tree model shown on *figure 24*. The main difference that should be noticed is the value of impurity mesure, which is lower than in the previous case as the data is unbalanced w.r.t. target class. Therefore there is less-balanced distribution in the child nodes.

3.1.2 PERFORMANCE EVALUATION OF THE ALGORITHM

For the evaluation phase we discuss methods for estimating its generalization performance on unseen instances outside of D.tr (train set).

SETS			Precision	Recall	f1-score	Support	Accuracy
Balanced	D.tr (Train)	0	*0.99*	*0.96*	*0.97*	*1952*	***0.97***
		1	*0.96*	*0.99*	*0.97*	*1952*	
	D.val (Validation)	0	*0.97*	*0.86*	*0.92*	*837*	***0.92***
		1	*0.88*	*0.97*	*0.92*	*837*	
Unbalanced	D.tr (Train)	0	*0.96*	*0.98*	*0.97*	*1952*	***0.94***
		1	*0.74*	*0.53*	*0.62*	*182*	
	D.val (Validation)	0	*0.93*	*0.96*	*0.94*	*837*	***0.90***
		1	*0.31*	*0.18*	*0.23*	*78*	

Figure 25: DT evaluation on balanced and unbalanced data

Figure 25 represents cost-sensitive measures such as: Precision, Recall, F-measure, and Accuracy, on both balanced and unbalanced data, in order to make a comparison between them.
More in detail, to establish which is the model that performs better, it is necessary to compare the evaluating measures as Accuracy, F1 Score, Precision and Recall of both sets. Looking at the *figure 25*, the model built on the balanced train set feels to perform better. It shows in the validation set an accuracy value (0.92) greater than the classifier built on the unbalanced training set (0.90). Going more into deep details, the difference between the accuracy of these two sets is quite small, and we can say that even the unbalanced model is quite perfect in predicting polysemy values. However, it has poor score in classifying ambiguous words as the recall value is really low.

Another evidence, concerning the preferability of a balanced model is given by the Roc curves (***figure 26***). Balanced model presents a higher level of sensibility at a higher level of specificity. Due to this evidence, it was decided to continue the analysis and the comparison with the other models using the balanced classifier.

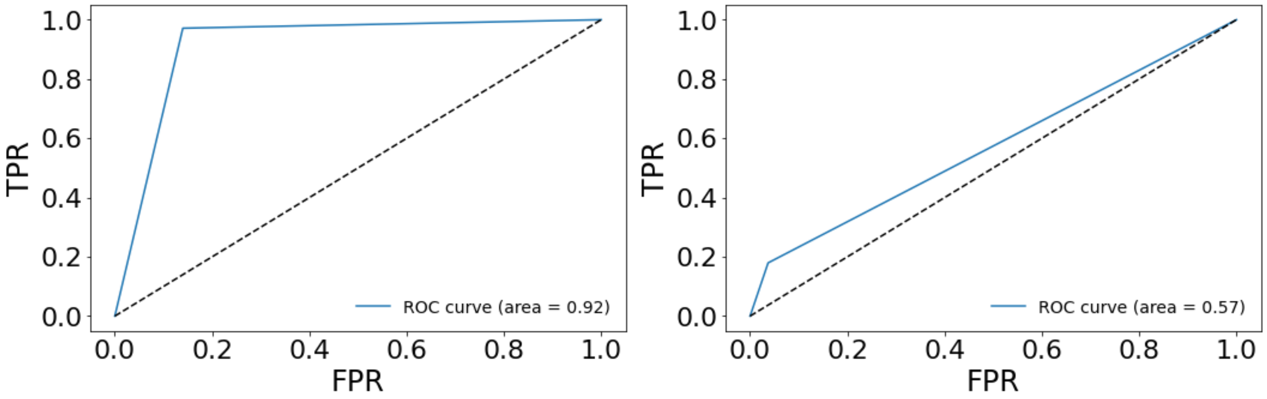


Figure 26: ROC curve of validation set in balanced and unbalanced data

Confusion matrix have been used for the performance evaluation, where the focus was on the predictive capability of a model. ***Figure 27*** shows confusion matrix done on D.val (validation set). The values provided are normalized. 0.97 represents the normalized number of True Positive instances, and 0.86 represents the normalized number of True Negative instances.

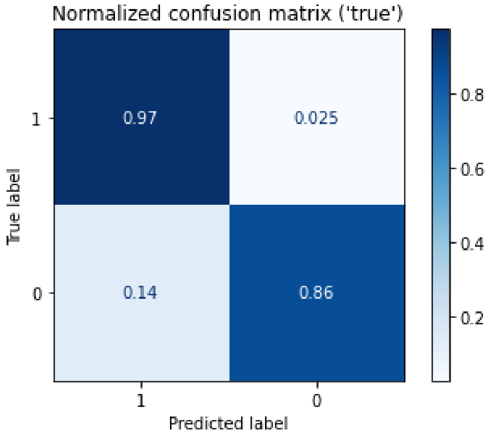


Figure 27: Confusion Matrix of DT on validation set

3.2 CLASSIFICATION BY OTHER ALGORITHMS

3.2.1 KNN CLASSIFIER

KNN classifier is one of the techniques that are known as lazy learner algorithms and which are different from Decision Tree classifier tehcniques. The algorithm is the following, given a test instance, we compute its proximity to the training instances according to one of the proximity measures. In order to find the best number of neighbours, as initial step we took the K range of (1, 40) and used accuracy scoring. ***Figure 28*** represents computed optimal number of K with respect to misclassification error rate.

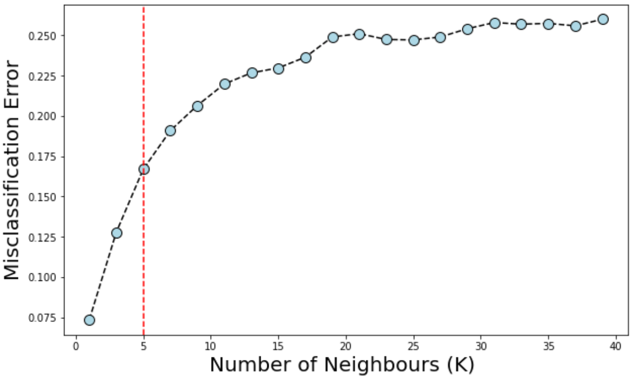
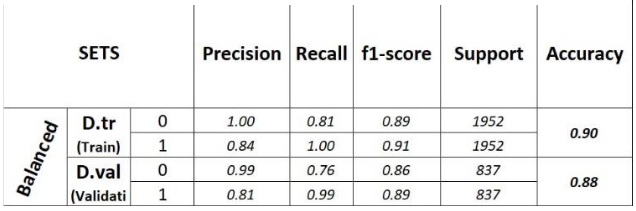


Figure 28: Accuracy scoring for KNN

According to the results, the red dotted line shows the chosen number of K (neighbours), and the relative performance are summarized on ***figure 29***.



	SETS		Precision	Recall	f1-score	Support	Accuracy
Balanced	D.tr (Train)	0	1.00	0.81	0.89	1952	0.90
		1	0.84	1.00	0.91	1952	
	D.val (Validati	0	0.99	0.76	0.86	837	0.88
		1	0.81	0.99	0.89	837	

Figure 29: KNN evaluation on balanced data

Provided confusion matrix (***figure 30***) represents the results of KNN classifier done on D.val (validation set). All the results are normalized. TP = 0.99, FP = 0.24, TN = 0.76, FN = 0.006.

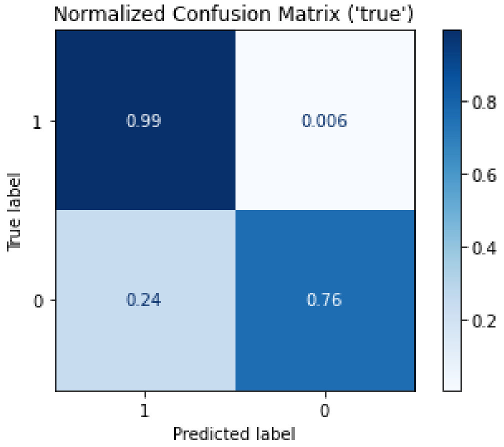


Figure 30: Confusion Matix of KNN on validation set

3.2.2 RANDOM FOREST CLASSIFIER

The Random Forest is an ensemble algorithm which was used in this case to combine different decision tree models. Randomized Search technique was used to determine the best parameters to be used into the implementation. The parameters are: min_samples_split: 10, min_samples_leaf: 5, max_depth: 46, criterion: 'entropy', with a mean validation score of 0.93 (std: 0.010). The evaluation was made computing the accuracy of training and validation sets as before. ***Figure 31*** represents the confusion matrix on validation set (D.tr) where we can see that the classifier predicted well 98% of ambiguous records as polysemy = 1 (True Positive) and it misclassified 8% as False Positive; on the other hand, it well classified 92% of non-ambiguous records as Polysemy = 0 (True Negative) and misclassified only 1.7% as False Negative.

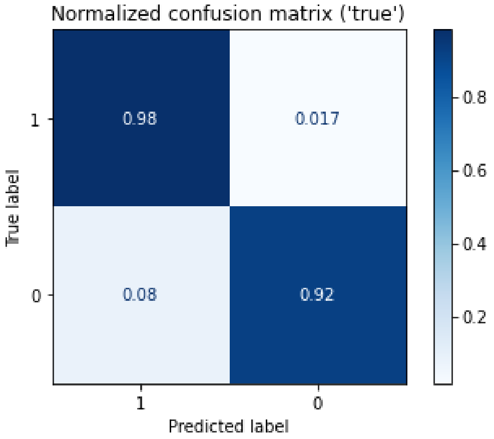
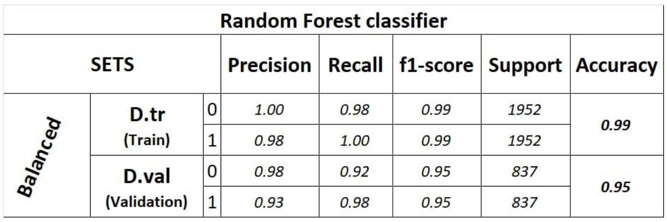


Figure 31: Confusion Matrix of RF on val. set

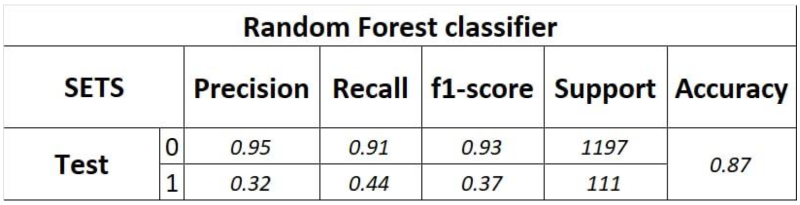


Random Forest classifier							
SETS			Precision	Recall	f1-score	Support	Accuracy
Balanced	D.tr (Train)	0	*1.00*	*0.98*	*0.99*	*1952*	***0.99***
		1	*0.98*	*1.00*	*0.99*	*1952*	
	D.val (Validation)	0	*0.98*	*0.92*	*0.95*	*837*	***0.95***
		1	*0.93*	*0.98*	*0.95*	*837*	

Figure 32: RF evaluation on balanced data

3.3 FINAL DISCUSSION

In this section of the report the evaluation is done on all three classifiers presented above (Decision Tree, KNN, Random Forest). A comparison was made for all the models. In order to choose the best classifier and confirm its efficiency in classifying the test set, evaluations of validation sets should be taken into consideration. The accuracy and the recall are the performance metrics that should be most valued, in such a way to have a model that correctly predicts when a word is ambiguous (polysemy 1). The model that best performed according to these criterions is the Random Forest Classifier, as previously shown the evaluation metrics: accuracy equals to 0.95 and f1-score [1] equals to 0.95. On the test set the following metrics were scored by the model, the results are shown on ***figure 34*** and confusion matrix ***figure 35***. This model, applied to a set of data never seen before, performs well enough.



Random Forest classifier

SETS		Precision	Recall	f1-score	Support	Accuracy
Test	0	*0.95*	*0.91*	*0.93*	*1197*	*0.87*
	1	*0.32*	*0.44*	*0.37*	*111*	

Figure 34: Final evaluation on test set with RF classifier

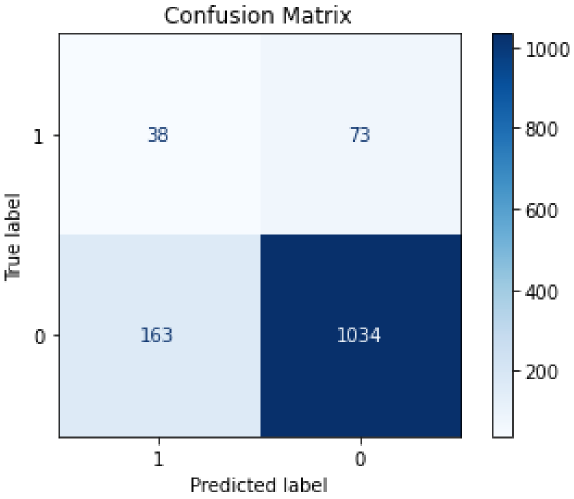


Figure 33: Confusion Matrix on test set with RF

4 PATTERN MINING

This section presents a methodology of Association Analysis, which is useful for discovering interesting relationships hidden in large data sets. The uncovered relationships can be represented in the form of sets of item, which are known as Frequent Itemsets, and Association Rules, that represent relationships between two itemsets.

Therefore, the strategy is to decompose the problem into two major subtasks:

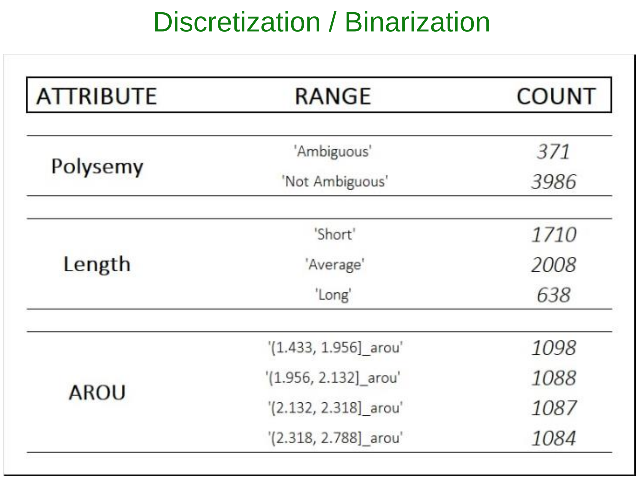
- Frequent Itemset Generation, where objective is to find all the itemsets that satisfy the minimum support threshold.
- Rule Generation, where objective is to extract all the high confidence rules from the frequent itemsets found in the previous step.

In the following sections will we see, at first, some preprocessing steps in order to prepare dataset for the further analysis. After we will extract frequent patterns from the data and then discuss those patterns with qualitative and quantitavice analysis. As a next step we will extract association rules using different values of confidence, followed by the discussion of explored rules. As a final step, we will see the summerized evaluation of all the activities carried out through this section.

DATA PREPARATION

To make correct association analysis, it has been decided to: firstly, split continuous attributes into intervals; secondly, transform the categorical attributes; lastly, for a better understanding, add abbreviations at the end of each value to recognize the attribute at which they refer.

Figure 35 represents the final result of the transformed attributes. On the figure we cans see only two categorical and one continuous attribute, only because all other continuous attributes have been first treated and then visualized in the same way.



Discretization / Binarization

ATTRIBUTE	RANGE	COUNT
Polysemy	'Ambiguous'	*371*
	'Not Ambiguous'	*3986*
Length	'Short'	*1710*
	'Average'	*2008*
	'Long'	*638*
AROU	'[1.433, 1.956]_arou'	*1098*
	'(1.956, 2.132]_arou'	*1088*
	'(2.132, 2.318]_arou'	*1087*
	'(2.318, 2.788]_arou'	*1084*

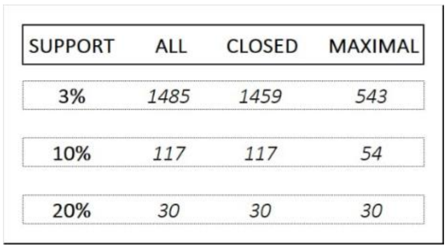
Figure 35: Attribute preparation

4.1 FREQUENT PATTERN EXTRACTION

Frequent itemsets extraction has been carried out by Apriori program from Pyfim library.

APRIORI ALGORITHM

Apriori algorithm helps to find association rules and frequent items. The implementation is pretty fast. The minimum support and minimum confidence are set by us, and are parameters of the Apriori algorithm for association rule generation. These parameters are used to exclude rules in the result that have a support or a confidence lower than the minimum support and minimum confidence respectively.



SUPPORT	ALL	CLOSED	MAXIMAL
3%	1485	1459	543
10%	117	117	54
20%	30	30	30

Figure 36: Different representation of frequent itemsets with different support threshold

During the process of frequent itemsets extraction different values of support have been tested (3%, 10% and 20%) considering different types of itemsets (all, closed, and maximal). ***Figure 36*** shows the number itemsets of each type by different support values. Considering the relationship between all presented types and looking at ***figure 37*** we can say that 'closed' and 'maximal' represent subsets of itemsets extracted by 'all'. Nevertheless, increasing the support, the number of itemsets decreases, but at the same time converges to the same value (sup = 20%, itemsets extracted = 30 for each type);

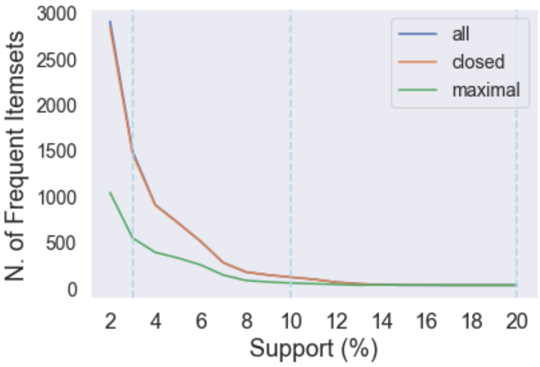


Figure 37: Number of frequent itemsets w.r.t support threshold

4.2 FREQUENT PATTERN DISCUSSION

As we already mentioned in classification analysis our dataset records are not balanced with respect to target class, polysemy. As the number of records of ambiguous words (polysemy = 1) is really low, it has been decided to set all the parameters for frequent pattern extraction algorithm in a way that we could get patterns including both, positive and negative class. After examining, we decided to set k=2 as a minimum number of items in each itemset, in order to get some significant patterns of our target class. ***Figure 38*** represents minimum number of items w.r.t different values of support.

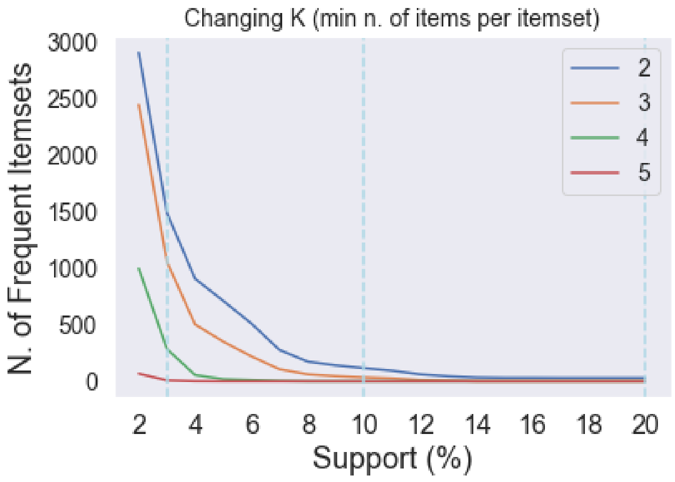


Figure 38: Number of frequent itemsets w.r.t support thresholds with changing k

Even though we mentiond imbalance of our target class in extracted patterns, we can clearly see on ***figure 39*** the difference between positive and negative class with respect to changing support values.

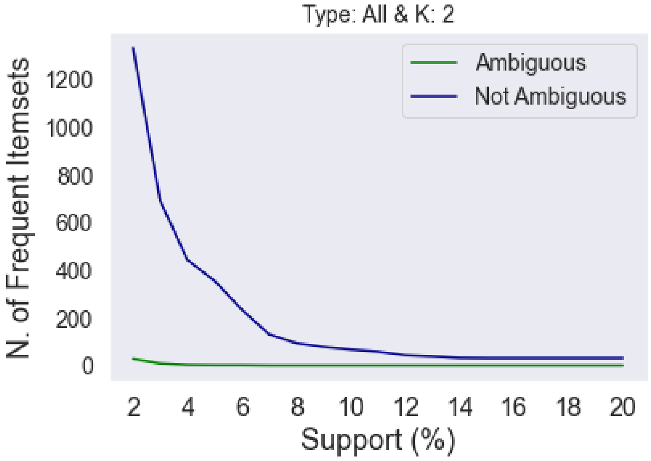
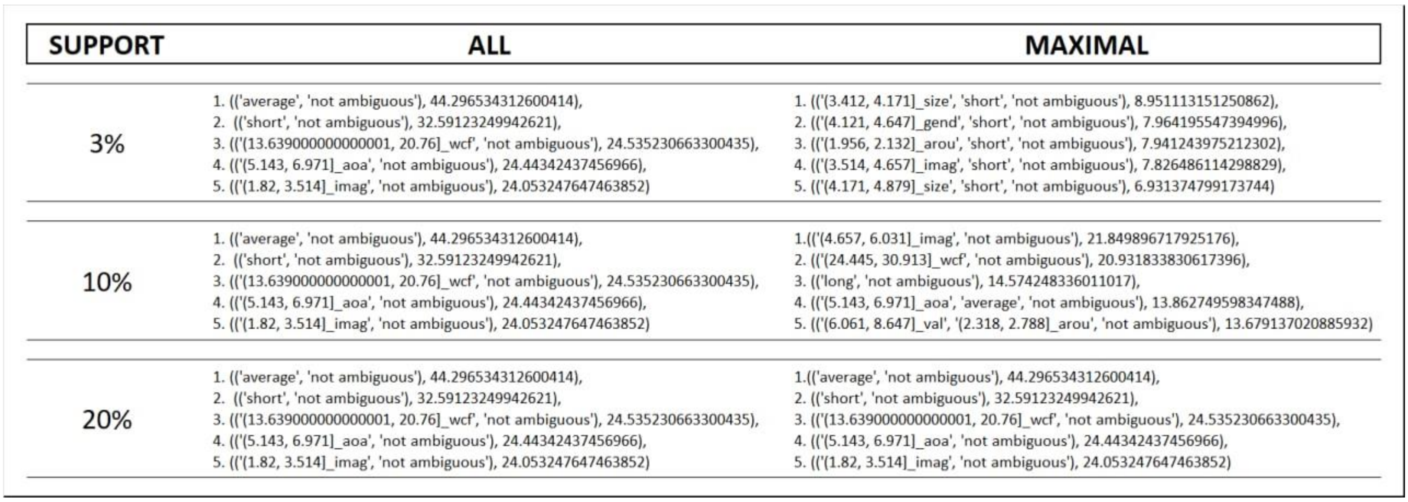


Figure 39: Number of frequent itemsets w.r.t suppot thresholds for target class

In order to evaluate the performance of previously presented different types, the itemsets extracted have been sorted by support value and presented first five (concequently the most frequent patterns) on ***figure 40.*** As 'closed' and 'all' frequent itemsets have shown almost the same results, only 'all' and 'maximal' have been analized with respect to changing support values. As we showed previously on ***figure 38*** as support value increases the different types of frequent itemsets converges to the same sets. For example, on the following figure we see that for minimum support = 0.03 (3%), we have different set of first five most frequent itemsets with different support values.

As a maximal frequent itemset is a frequent itemset for which none of its immediate supersets are frequent, therefore it is obvious to have highest support equal to 8.95, while in the other case the highest support is 44.29. Additionally, it is natural that the first highest support itemset in 'All' does not change per value of support, while in 'Maximal' we have differences. Finally, as support increases both types of itemsets converges to the same variables and max support value as mentioned earlier, which is represented on the figure as a last row corresponding to the minimum support value equal to 20%.



SUPPORT	ALL	MAXIMAL
3%	1. (('average', 'not ambiguous'), 44.296534312600414), 2. (('short', 'not ambiguous'), 32.59123249942621), 3. (('(13.639000000000001, 20.76]_wcf', 'not ambiguous'), 24.535230663300435), 4. (('(5.143, 6.971]_aoa', 'not ambiguous'), 24.44342437456966), 5. (('(1.82, 3.514]_imag', 'not ambiguous'), 24.053247647463852)	1. (('(3.412, 4.171]_size', 'short', 'not ambiguous'), 8.951113151250862), 2. (('(4.121, 4.647]_gend', 'short', 'not ambiguous'), 7.964195547394996), 3. (('(1.956, 2.132]_arou', 'short', 'not ambiguous'), 7.941243975212302), 4. (('(3.514, 4.657]_imag', 'short', 'not ambiguous'), 7.826486114298829), 5. (('(4.171, 4.879]_size', 'short', 'not ambiguous'), 6.931374799173744)
10%	1. (('average', 'not ambiguous'), 44.296534312600414), 2. (('short', 'not ambiguous'), 32.59123249942621), 3. (('(13.639000000000001, 20.76]_wcf', 'not ambiguous'), 24.535230663300435), 4. (('(5.143, 6.971]_aoa', 'not ambiguous'), 24.44342437456966), 5. (('(1.82, 3.514]_imag', 'not ambiguous'), 24.053247647463852)	1.(('(4.657, 6.031]_imag', 'not ambiguous'), 21.849896717925176), 2. (('(24.445, 30.913]_wcf', 'not ambiguous'), 20.931833830617396), 3. (('long', 'not ambiguous'), 14.574248336011017), 4. (('(5.143, 6.971]_aoa', 'average', 'not ambiguous'), 13.862749598347488), 5. (('(6.061, 8.647]_val', '(2.318, 2.788]_arou', 'not ambiguous'), 13.679137020885932)
20%	1. (('average', 'not ambiguous'), 44.296534312600414), 2. (('short', 'not ambiguous'), 32.59123249942621), 3. (('(13.639000000000001, 20.76]_wcf', 'not ambiguous'), 24.535230663300435), 4. (('(5.143, 6.971]_aoa', 'not ambiguous'), 24.44342437456966), 5. (('(1.82, 3.514]_imag', 'not ambiguous'), 24.053247647463852)	1.(('average', 'not ambiguous'), 44.296534312600414), 2. (('short', 'not ambiguous'), 32.59123249942621), 3. (('(13.639000000000001, 20.76]_wcf', 'not ambiguous'), 24.535230663300435), 4. (('(5.143, 6.971]_aoa', 'not ambiguous'), 24.44342437456966), 5. (('(1.82, 3.514]_imag', 'not ambiguous'), 24.053247647463852)

Figure 40: All and maximal first five (highest support) frequent itemsets by support thresholds

4.3 ASSOCIATION RULES EXTRACTION

This part of the report extracts and analyses association rules between previously examined frequent itemsets. ***Figure 41*** shows the relationship between value of confidence and number of rules with a minimum support of 3%. The flow demonstrates how the number of rules decreses with respect to the incresing number of confidence.

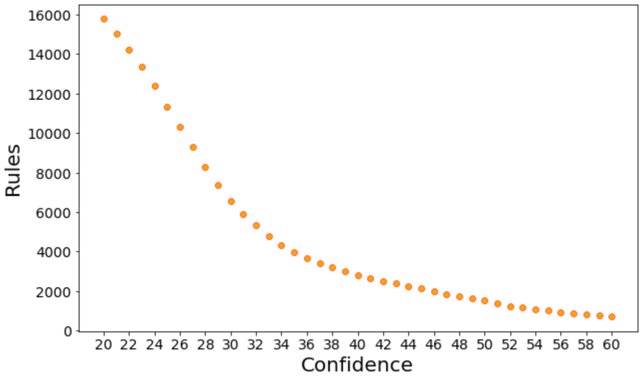


Figure 41: Number of rules w.r.t different confidence threshold

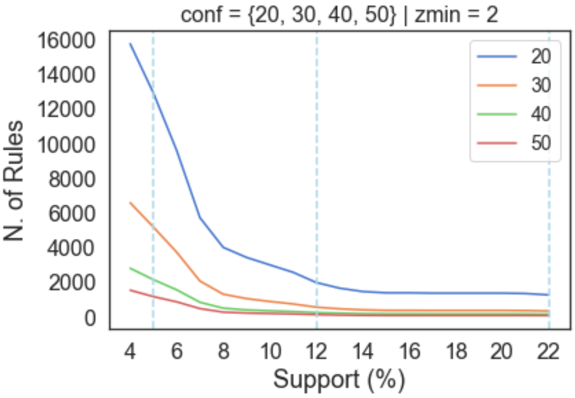
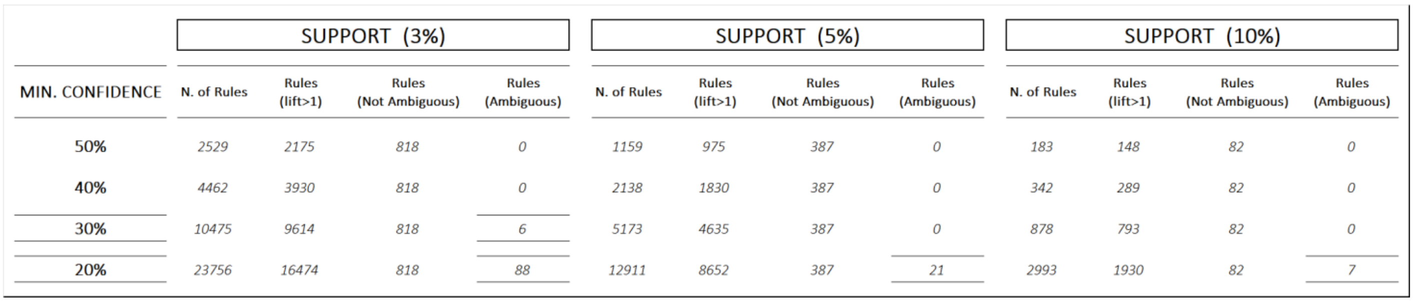


Figure 42: Number of rules w.r.t support and confidence thresholds

Figure 42 represents different values of confidence {20%, 30%, 40%, 50%} with respect to the relationship between support and number of itemsets. As the confidence and support increases the number of itemsets decreases.

4.4 ASSOCIATION RULES DISCUSSION

This section provides qualitative and quantitative analysis on how the number of rules changes with respect to minimum confidence and support values. ***Figure 43*** represents the different categories of rules with respect to threshold value of confidence of [20%, 30%, 40%, 50%] and support of [3%, 5%, 10%]. The different categories provided are total number of rules, rules where lift is higher than 1, rules where negative and positive target classes are predicted.



	SUPPORT (3%)				SUPPORT (5%)				SUPPORT (10%)			
MIN. CONFIDENCE	N. of Rules	Rules (lift>1)	Rules (Not Ambiguous)	Rules (Ambiguous)	N. of Rules	Rules (lift>1)	Rules (Not Ambiguous)	Rules (Ambiguous)	N. of Rules	Rules (lift>1)	Rules (Not Ambiguous)	Rules (Ambiguous)
50%	2529	2175	818	0	1159	975	387	0	183	148	82	0
40%	4462	3930	818	0	2138	1830	387	0	342	289	82	0
30%	10475	9614	818	6	5173	4635	387	0	878	793	82	0
20%	23756	16474	818	88	12911	8652	387	21	2993	1930	82	7

Figure 43: Numbers of rules w.r.t. different support and confidence threshold

As the dataset is unbalance with respect to the target class, the number of positive class (ambiguous words) are very low, therefore increasing min confidence or support value was decreasing the chance of getting association rules predicting positive class. Finally, the analysis of association rules are provided only for these low values of min confidence and support.

4.5 FINAL DISCUSSION

Final discussion about the association rules are carried out by taking into consideration the accuracy and preciseness of predicting target variable. In order to get the association rules that perform on target class, it has been decided to choose the parameter values presented in the previous section.

As a first phase, min_sup = 3% and min_conf = 30% has been examined on the whole dataset. Association rules predicting target class as positive was six in total. The rule with the highest lift value (3.9709), implying antecedents as word's 'Length' = short, 'Imageability' = (4.657, 6.031], and 'Web Corpus Frequency' = (24.445, 30.913] and consequent as 'polysemy' = ambiguous word. All six rules have been analysed and tested.

On the whole dataset, total prediction on target positive class was 415, where **TP** = 112, **FP** = 303, **TN** = 3683, and **FN** = 259.

Accuracy = 0.8710, **Precision** = 0.2699, **Recall** = 0.3019, **F1-score** = 0.2849.

On the second phase, the association rules have been extracted from the Training set built during classification analysis and the most interesting rules has been tested in predicting a target class on Test set. Only two association rules have been extracted from training set, the one with the highest lift value provided the same actecedents as previously, the other one implied antecedents as word's 'Size' = (1.374, 3.412], and 'Gender' = (4.121, 4.647], and consequent as 'polysemy' = ambiguous word.

On the test dataset, total prediction on target positive class was 226, where **TP** = 83, **FP** = 183, **TN** = 3803, and **FN** = 288.

Accuracy = 0.8919, **Precision** = 0.3120, **Recall** = 0.2237, **F1-score** = 0.2606.

It is significant for this dataset to know how many true instances have been correctly predicted, therefore the main performance indicators that have been taken into consideration are recall and F1-score, which is also a harmonic mean of precision and recall.

In conclusion, the strongest association rule selected with min_sup = 3% and min_conf = 30% threshold values, have been the one implying antecedents as word's 'Length' = short and 'Web Corpus Frequency' = (24.445, 30.913] (the highest quarter range). However, the extracted rules in general are unsuitable to predict the entire dataset since they have very low level of support and confidecence. Therefore, it has to be taken into consideration that the extracted rules are not good enough to apply rule-based classifier.