Data Mining Project Report: Glasgow Norms

Ludovico Lemma

January 4, 2022

1 Introduction

The Glasgow Norms are a set of normative ratings for English words on nine psycholinguistic dimensions: arousal, valence, dominance, concreteness, imageability, familiarity, age of acquisition, semantic size, and gender association. Three other variables are included here, namely length, polysemy and the frequency in the Google Newspapers Corpus. There were originally two sets of words, one of 808 words, the other of 4800 words, which then were merged for this study. The experiment was run online (via the university of Glasgow's platform), each participant rated a list of either 101 (from 8 possible lists of the 808 word set) or 150 words (from 32 possible lists of the 4,800 word set).

2 Data Understanding and Preparation

2.1 Data Semantics and Data Description

The provided data-set is composed of 4682 words. The length of the words span from 2 ("up" and "TV") to 16 ("intercontinental") with a median value of 6. Polysemy is a dummy variable which takes the value of 0 if the word is non-polysemous (4303 occurrences) and 1 if it is polysemous (379 occurrences). Web_corpus_freq corresponds to the frequency of the word in the Google Newspaper Corpus, it exhibits an heavy-tailed distribution, the minimum value is a frequency of 12,770 for the word "enthral", while the most frequent word is "all" which appears in the corpus 2,022,460,000 (more than two billion times) distancing itself from the next most frequent words ("have", "new" and "home") by about half a billion occurrences.
The nine other dimensions were evaluated in the experiment by the participants, they expressed individually for each word a number on a scale representing their perception grades of each of the nine features. The evaluation scale was from 1 to 9 for arousal, valence and dominance and from 1 to 7 for the other 6 dimensions. In the case of Age of Acquisition ("aoa") the scale from 1 to 7 corresponds respectively to the intervals 0-2, 3-4, 5-6, 7-8, 9-10, 11-12, 13+, with a value of 1 indicating a word learned in the first two years after birth (estimated), and 7 a word learned from 13 years old on. For the case of gender the scale was 1 for a very feminine word and 7 for a very masculine word. The other dimensions were ordered from least to most of the respective feature, e.g. Imageability goes from least easy to imagine, to most easy to imagine. Semantic Size goes from lowest perceived magnitude, to highest perceived magnitude, and so on.

2.2 Distribution of the variables and Statistics

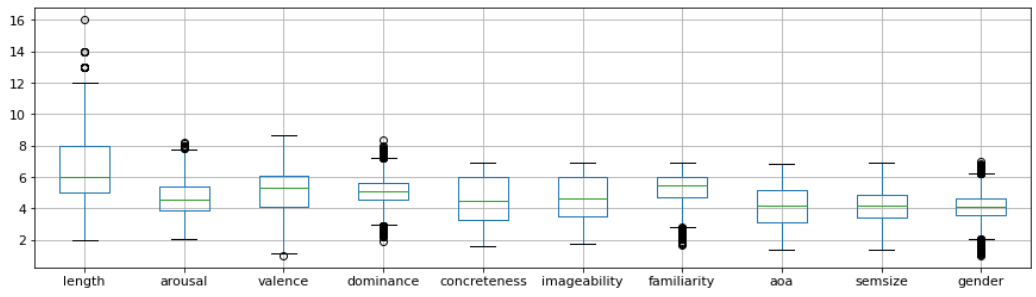

Figure 1: Box-plot, distribution of the variables

As a first step I looked at the distribution of the variables. As it can be seen from Figure 1 the data are mostly distributed across the variables without many distant outliers. I excluded from the box-plot polysemy, because it is a binary variable, and web corpus frequency, because it is heavy-tailed distributed on a larger scale and it would disrupt the visualization of the other variables.

Variable	non-polysemous	polysemous	abs. diff.
aoa	0.603926	0.486008	0.117918
length	0.405371	0.299142	0.106229
concreteness	0.651192	0.737200	0.086009
imageability	0.674377	0.749428	0.075051
semsize	0.604274	0.532178	0.072096
familiarity	0.755504	0.806947	0.051443
arousal	0.575540	0.533141	0.042400
dominance	0.600953	0.622147	0.021194
web_corpus_freq	0.013520	0.029022	0.015502
valence	0.587142	0.601110	0.013967
gender	0.587181	0.599044	0.011863

Table 1: Absolute differences of the means by Polysemy (Normalized)

Then, I decided to study the relation of Polysemy and Length with other variables, these two being direct measures of the single observations can clearly help in the understanding of the data. I started by looking at divergent means of the distributions of the variables in the case of polysemous words. I did it by normalizing the data (because of the different scales) by the maximum absolute values, then I measured and sorted the divergence of the means as it can be seen from Table 1.
The highest divergences of the mean were age of acquisition and length. In Figure 2 we can see the bar plot of polysemy and then the histogram and density plot of polysemy across the length and the age of acquisition.

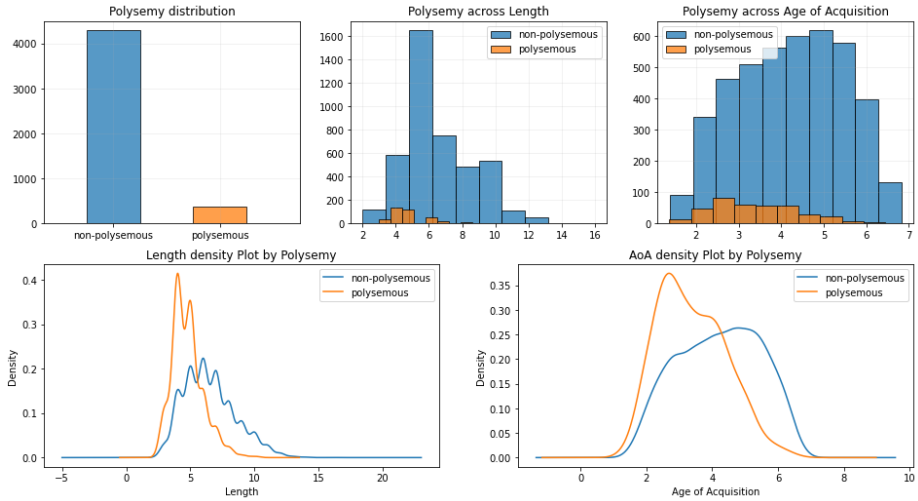

Figure 2: Polysemy distribution and density plot across Length and AoA

It appears polysemous words tend to be shorter than non-polysemous ones and there is evidence of them to be learned earlier. I decided to understand better the relation of length with other variables, also because I suspected that the lowest value of aoa may be shorter words, thus the divergence of the mean in that case may be explained by the length of the word. I decided to plot the median values of the normalized 9 psycholinguistic dimensions across the length as it can be seen from Figure 3. As I imagined, the most eye catching variable is Age of Acquisition, it shows immediately that the earlier a word is learnt the shorter it could be. This made it clear that polysemy across aoa may be explained partially by a positive correlation with length. From the line plot it appears also that Imageability and Concreteness are correlated between themselves, as they follow an identical path along the length of the word and negatively correlated with length. It's also clear from the plot a very small negative correlation of familiarity and length, with an interesting secondary peak for long words between 12 and 14 characters, before dropping as expected, this range seems interesting also because of the strange behaviour of the other variables, the cause it is probably the drop in the

number of observed words (Figure 2, the number of words in the range [12, 14] is only 54, only half of those with 11 characters).

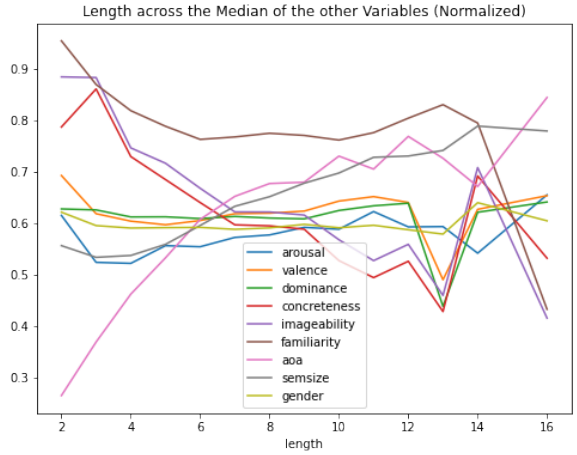

Figure 3: Median of the variables by the length of the word

I decided to finish the preliminary analysis by looking at the distribution of variables (not normalized) which didn't emerge from previous plots, I used a scatter-plot and I plotted also polysemy to deepen the visualization. From Figure 4 it generally appears polysemy tend to be more frequent with small and calm inspiring words while for both gender and dominance it tends to stay around the center of the distribution. From the scatter-plot there also seem to be some positive correlation between semantic size and arousal.

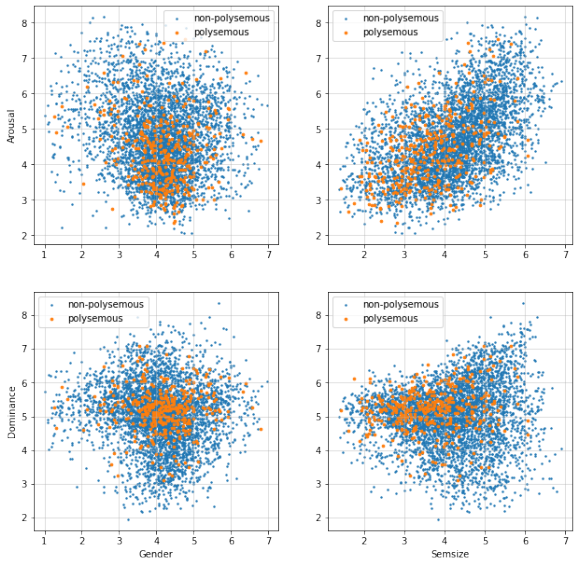

Figure 4: Scatter-plot, polysemy across gender, semsize, arousal and dominance

2.3 Data Quality and Variable Transformations

The dataset provided is clear from major problems, there are no semantic inconsistencies, all the lengths of words correspond to the values in the length column. As stated in the introduction the variables do not have the same scales, I will deal with this problem later. There are 14 missing values in web_corpus_freq and I analyzed some of them, the values "TRUE" and "FALSE" (all uppercase) may have been deleted by mistake considering their large use in databases, "Facebook" and "Twitter" may have been filtered out by some algorithm, also different colloquial words for mother and father were missing frequencies which they may appear more in private conversations. I also noticed most missing values showed at least an uppercase first letter (10). I decided first to correct all the

uppercase letters in the dataset to avoid issues if I resetted the indexes, as they would appear first if re-sorted by the column "words", then I moved to fill these 14 missing values: I wrote a function which sorted the dataset by "familiarity", which is monotonically correlated with web_corpus_freq, as it emerged from a quick check at the correlation matrix with the spearman method, and it calculated the mean of the previous and next "web_corpus_freq" values by the "familiarity" order and it filled the corresponding missing value with it.

Variables	Inliers	Outliers
length	4666	16
arousal	4671	11
valence	4680	2
dominance	4549	133
concreteness	4682	0
imageability	4682	0
familiarity	4627	55
aoa	4682	0
semsize	4682	0
gender	4518	164
(A) Total	4307	375
(B) web_corpus_freq	4070	612
(A+B)	3723	959
(C) log_web_corpus_freq	4668	14
(A+C)	4297	385

Table 2: Outliers Table in three different cases

Table 2 is the output of another function I wrote to count the outliers in the box-plot, the Totals are smaller than the sum of the outliers indicated by the single variable as they appear in multiple columns and therefore are counted only once in the Total. I evidenced three different cases of outliers. This is due to the fact I also decided to perform a logarithmic transformation on web_corpus_freq, in order to reduce the skeweness of the heavy-tailed distribution I noticed earlier and also to permit a cleaner visualization. I recalculated the outliers in the transformed variable, from Table 2 it can be seen the largest portion of outliers were mainly from the skewed distribution of web_corpus_freq, now outliers greatly reduced from 959 total to 385. There are no problematic outliers, the majority of them is concentrated in gender and dominance, while those from the log transformed web_corpus_freq are now around the rest of the distribution as it is the case of the other variables. From the previous boxplot (Figure 1) the only distant outliers were those with length higher than 12 which are only 16. Considering the nature of the variables I decided not to delete outliers from the dataset as they can deliver more information later.

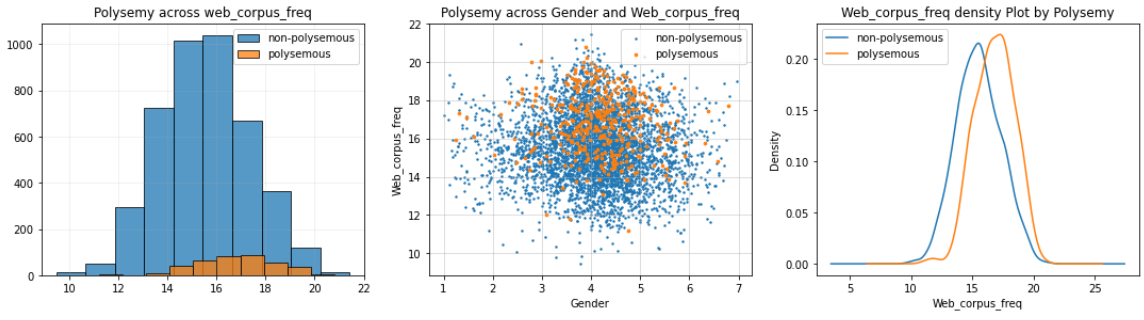

Figure 5: Log transformation of web corpus frequency

Now that it is meaningful, in Figure 5 I visualized the new distribution of web_corpus_freq, I also studied the polysemy across it, as I already noticed before in Table 1 polysemous words are more frequent than their counterpart even if slightly. I also visualized web corpus with gender to see if the resulting graph would be globular or elliptical as I expected from two normal distributions.

2.4 Highly Correlated Variables

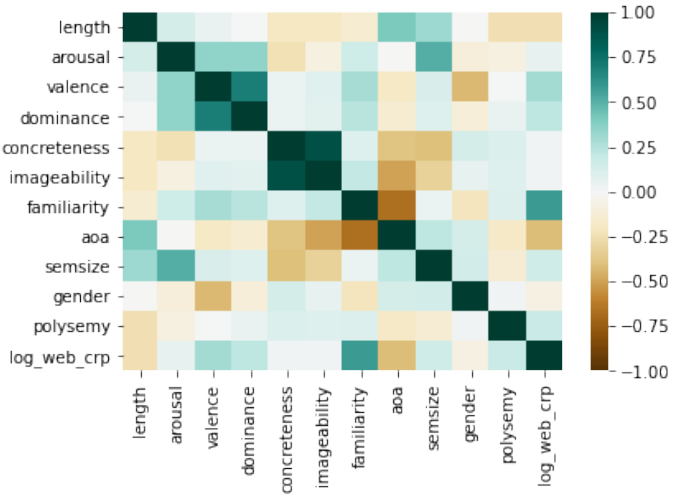

Figure 6: Correlation matrix heatmap (Spearman)

Here we'll analyze highly correlated variables. I noticed there were a lot of medium level correlations in the correlation matrix heatmap in Figure 6, and three highly correlated ones. Before analyzing the 3 most correlated, I also created a function which extracted in a smaller matrix the correlation in an input range in absolute values with an input method, I used it to look at correlations smaller than 0.5 as they wouldn't emerge as clearly with other methods. I noticed some correlations I already observed from previous charts such as imageability-concreteness, length-aoa and arousal-semsize, and some other isolated ones such as valence-gender and familiarity-web_corpus_length which I also used for filling missing values. But the 3 most correlated ones have a 0.6 or higher correlation, those are: the imageability-concreteness pair with 0.9 correlation, the dominance-valence pair with 0.72 correlation and familiarity-aoa pair with -0.67 correlation.

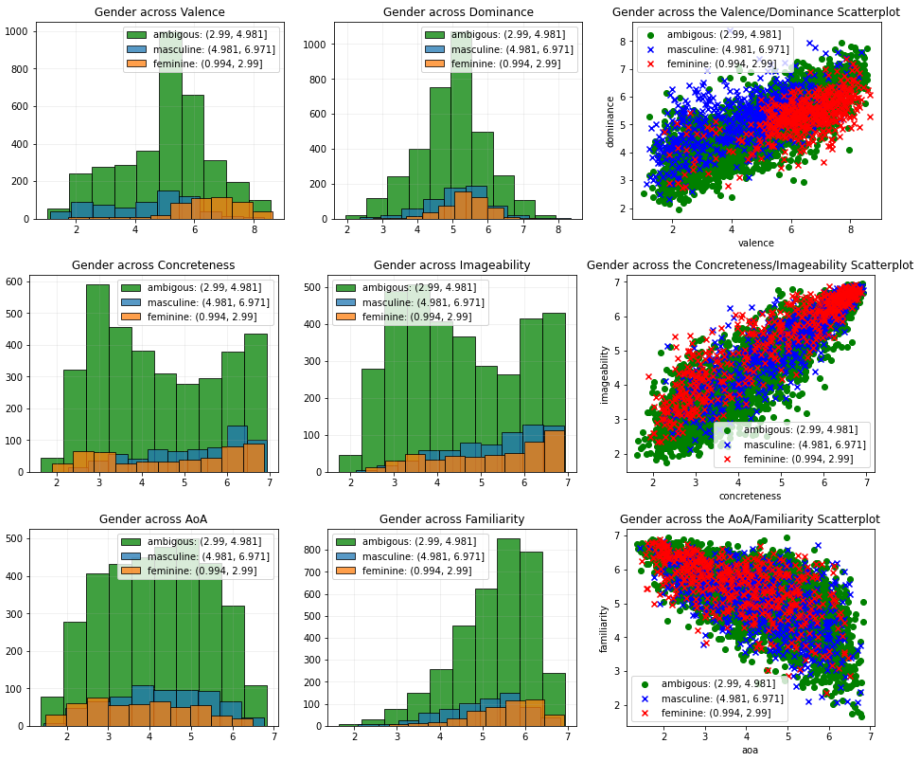

Figure 7: Gender distribution across highly correlated variables

To visualize them I decided to plot with them gender having took notice of the relation between valence and gender which was absent for dominance. In order to do that I discretized the variable gender, I created three bins to evidence only the extreme values of its distribution, I called the center of the distribution "ambiguous" and the extremes, which I was looking for, "feminine" on the lower end and "masculine" on the higher end of the scale, then I created histograms and scatterplots to analyze the pairs in relation with the discretized variable. As it can be seen from Figure 7 the distribution of the extreme of gender varies a lot for valence while in dominance they almost overlap, in the histogram it can be clearly visualized how the most feminine words seems to have a higher perceived worth.

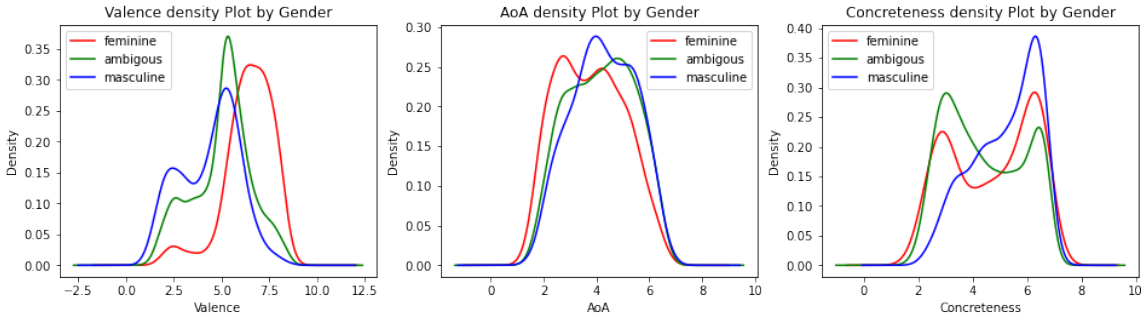

Figure 8: Gender density plot

In Figure 8, I also represented the density plot of the seemingly most divergent masculine and feminine perceived valence, age of acquisition and concreteness, here we can notice something interesting also for concreteness, the second peak of the bimodal distribution here seems to be in part explained by the concreteness of extreme femininity and extreme masculinity of some words. From the histograms and scatterplots it seemed valence, aoa and concreteness contained more information than the correlated pair but I preferred to have a measure of this intuitive reasoning.

2.5 Feature Selection

To select the most important features to analyze further I used two other methods, first I reversed the function used to analyze the most correlated variables and I looked at the lowest correlated pair, which was gender-length, but I was conflicted about analyzing the length further as it seemed correlated with lots of other variables and it was not continuous. The second least correlated pair was arousal-aoa which seemed more interesting with an absolute correlation lower than 0.01.

Variable	VIF_{ALL}	$VIF_{<5}$	$VIF_{<5}$	$VIF_{<10}$	$VIF_{<10}$	$VIF_{<10}$
arousal	35.74					
valence	33.82			8.74	9.99	8.00
dominance	75.97					
concreteness	89.32	4.44			6.80	6.94
imageability	101.06		4.43	7.57		
familiarity	70.41					
aoa	14.60	4.44	4.43	5.63		5.75
semsize	33.11				8.28	
gender	32.35					
log_web_corpus_freq	106.23					

Table 3: Feature selection (VIF), variable combinations

But considering the high number of low-to-middle correlated variables I needed a measure of possible multicollinearity, thus I decided to calculate the variance inflation factor. I excluded polysemy and length from the process of feature selection altogether, because one is binary and the other not continuous. In Table 3, imageability and the transformed web_corpus_freq variable immediately emerged with clear high VIFs in the case I calculated it without dropping any other variable, it also immediately emerged aoa would probably be the least affected variable if chosen. To avoid a

random choice of variable to measure the VIF I decided to calculate how many combinations there were with a VIF lower than 5, in this case there were only 2 possible combinations which included aoa with either concreteness or imageability. Then I decided to check also if there were a higher number of variables with a VIF lower than 10, in this case there were only 3 combinations of triplet which included also valence and semsize as it can be seen from the table. Considering the number of times the variables appeared in the combinations I could select the pair aoa-concreteness for $VIF_{<5}$ or the triplet valence-aoa-concreteness for $VIF_{<10}$.

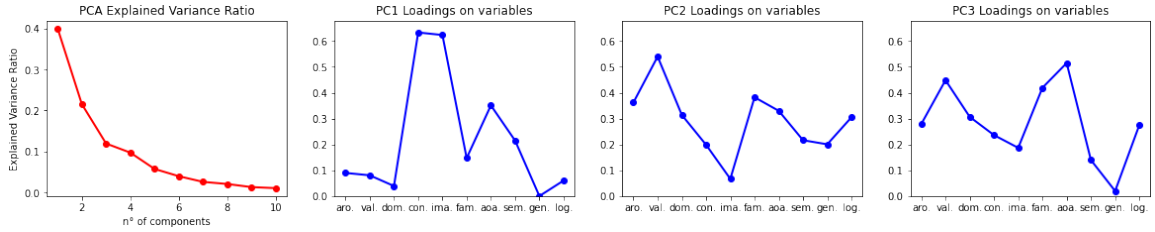

Figure 9: PCA Explained Variance and Loadings of the first three PC

Even though I could select these 2 or 3 features to create a model I needed to cut out too much of the original data, so I decided it was better to perform a PCA on the dataset, first I applied on the preprocessed data a MinMax scaling because I know the scales of the original domains. Previously I normalized with max absolute values but I considered to be more precise in this phase to move the data on a 0-1 range. Then I plotted the variance ratios of the components to find the proper number of principal components to select. Through the elbow method I determined 3 elements would be enough, as they would cover 73,52% of the variance of the original data. As it can be seen from Figure 9 I also plotted the loadings of the said first three principal components, the first PC load imageability and concreteness the most with also a focus on aoa as I predicted previously analyzing the combinations of VIF lower than five, as it can be seen excluding valence and semsize which appeared in the combinations of VIF lower than 10 through the PCA I included more variance from the original variables, especially also the variance from familiarity.
Here in Figure 10 I plotted the result of the PCA (in particular for the first two components) on a scatterplot, here I reduced the size of the points to enhance both the visibility of the arrows and the shape and density of the plot. In the labels I indicated the explained variance of the components in the round brackets. As it can be seen PC1 and PC2 represent more than 60% of the original data their shape is that of a fat crescent moon (more generally a globular shape), with a slight increase of point caused by "aoa", but the distribution is almost uniform, with no clear visually identifiable clusters (if not for slightly higher density on the left caused by concreteness and imageability).

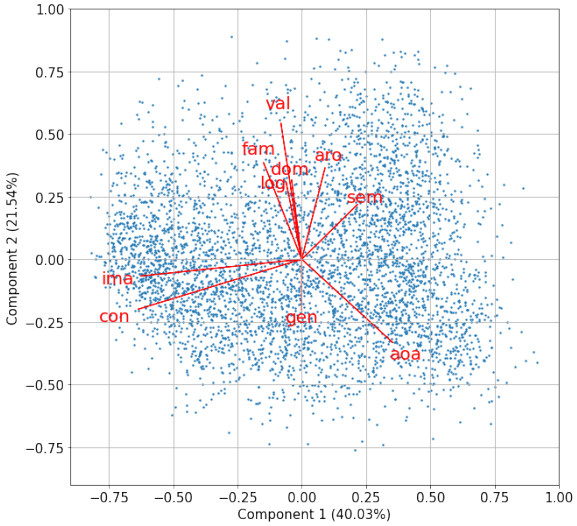

Figure 10: Score and Loading plot of PC1 and PC2

3 Clustering

3.1 Clustering analysis by K-Means

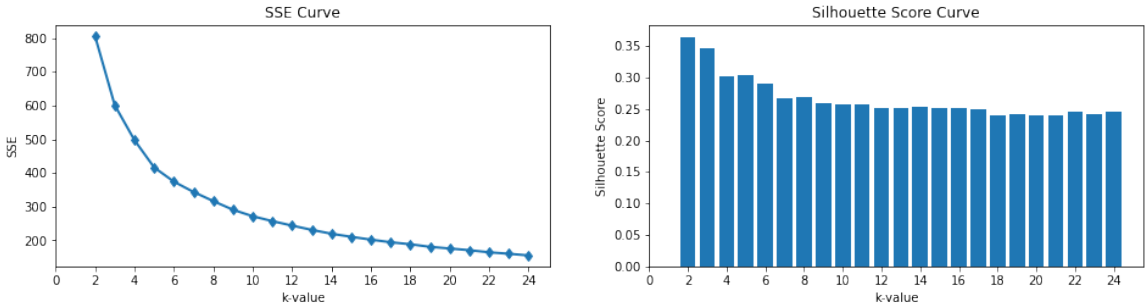

Figure 11: SSE and Silhouette score plots (k-value selection)

As I said I opted for using the 3 components of the PCA as a base for further analysis, thus I didn't use non-continuous variables such as length and polysemy for k-means, also because it wouldn't perform well with those variables. To start I set the hyperparameters, I set 10 as the number of initialization of the K-Means algorithm with different centroid seeds. Then, in Figure 11 as it can be seen I looked for the best value of k by looping through different value of SSE and Silhouette scores, as it can be seen from the figure there is not a strong curvature in the SSE plot to apply the elbow method perfectly even though at 5 k there is a slight sign of an elbow, also by looking to the Silhouette scores' plot I noticed that with a k-value of 5 there is a slight increase of the silhouette score. Thus, I applied K-Means with a k-value of 5 and I visualized how it clustered the first two principal components as it can be seen in Figure 12, as before I reduced the point sizes but I also evidenced the centroids as large red dots.

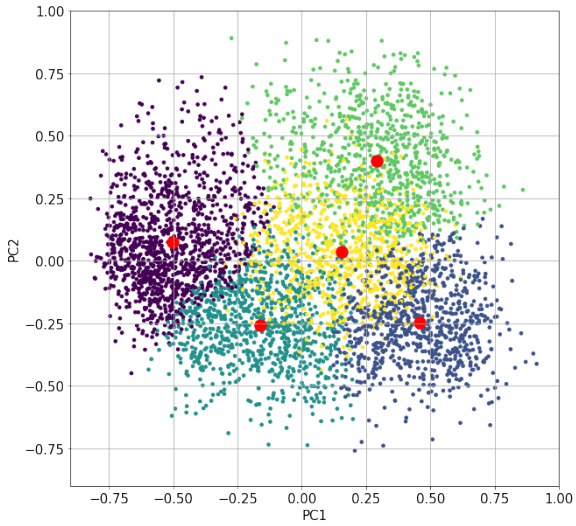

Figure 12: K-means on a PCA with 3 components (PC1/PC2 plot)

Even though I visualized the clustering on the first two components of the PCA it is not evident how the clustering would appear on the original distributions, thus I decided to take 3 combinations of variables I divided by the weight of their loadings in PCA to see how that reflects on variables which loaded less on the components. As it can be seen from Figure 13 I selected semsize/gender as a low loading pair, familiarity/semsize as a medium loading pair and concreteness/aoa as a high loading pair. I also evidenced again the centroids as red dots for which I reversed the fitting I performed previously (the PCA first and MinMax scaling then) to represent with the unfitted data. As it could be expected the algorithm subdivided better the variables which loaded the most the components, even though the green cluster seemed not to get nearer the other centroids in low loading pairs it is clear the subdivision gets blurry with some pairs of variables.

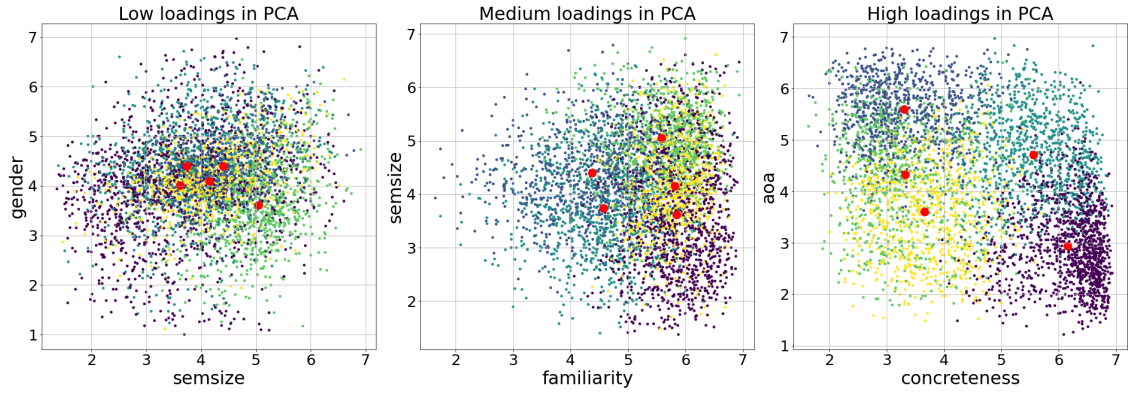

Figure 13: K-means results with three pairs of the original data with different loadings in the PCA

To understand also how the 5 clusters would behave with the variables not used for PCA and K-Means I also designed two barplots for length and polysemy. As it can be seen there is one cluster with generally shorter and polysemous words, namely B, with C following it.

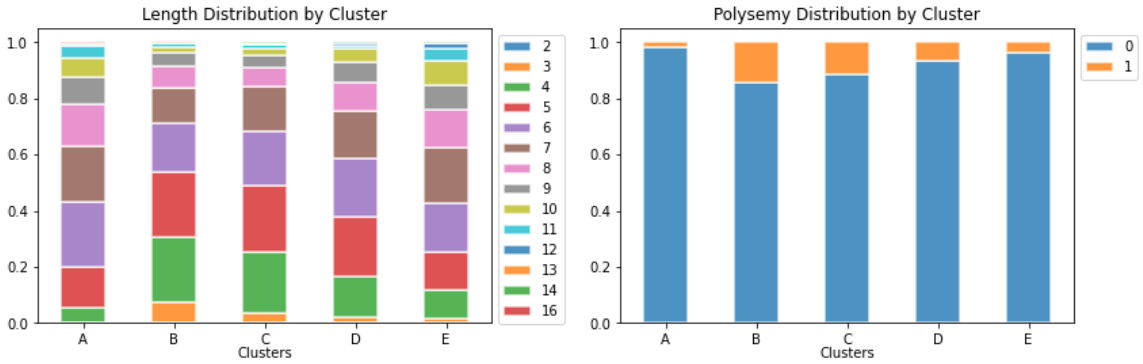

Figure 14: Length and Polysemy distribution by each cluster

We'll also provide some numbers and characterizing connections to the previous plots: Cluster A (in dark blue) has 852 elements, it has been formed mainly by words with high evaluations of AoA (see loadings and clustering in Figure 10 and 12) and as such it has the lowest number of short words. Cluster B (in violet) is the most numerous with 1219 elements, it has the highest number of highly concrete and familiar words that tend to be shorter, learned earlier and with higher chances of being polysemous. Cluster C (in yellow) is at 835 with the least number of elements, it follows B in terms of the presence of short and polysemous words. Cluster D (in light blue) has 940 elements, it is the second most numerous one. Finally, Cluster E (in light green) has 836 members and it has words that tend to be less concrete which give a high sense of magnitude and familiarity.

To be sure there weren't errors with the PCA and the clustering I performed, I tried to apply K-Means also only to the triplet valence-concreteness-aoa, first normalizing the 3 variables in a min-max scale, but as I suspected there didn't seem to be much difference in how the algorithm performed. I also considered the possibility that the outliers may have created too much noise for K-Means, thus before trying a scaler robust to outliers, I tried removing them before I performed again a PCA on them with a subsequent run of the K-Means algorithm, but even in this case I found that the results were approximately the same, thus I started suspecting outliers may not be that anomalous or that they didn't distort the data that much.
Instead, using a robust scaler before performing the PCA it first changes drastically the loading composition by heightening the presence of gender and dominance which had a discrete number of outliers, it also requires one more principal component by applying the elbow method to the scree plot of the explained variance, and it lowers the explained variance ratio of all the components. As a result we get that the best number of k to run K-Means is 3, there is less confusion with some pairs, but that is probably caused by the low number of clusters. With these considerations I guessed that, having not that high number of outliers, the Robust Scaler was not appropriate for my objectives, thus I decided to keep the model I presented prior.

3.2 Analysis by density based clustering

To cluster the points by density I used DBSCAN, starting again from the PCA I performed as the reduced form of my original data. Then, I needed to set the hyperparameters: for the number of minimum samples, with which the algorithm should decide if there are enough points in the range to consider it a cluster, I decided to set it as 8 by taking the middle value between twice the number of principal components and the number of input features of the PCA such as $(2 * N_{PC} + 10)/2 = 8$. After that, with this value set, I computed the average distances between each point and its 8 neighbours, I sorted the list by ascending order and I plotted the points as it can be seen in Figure 15.

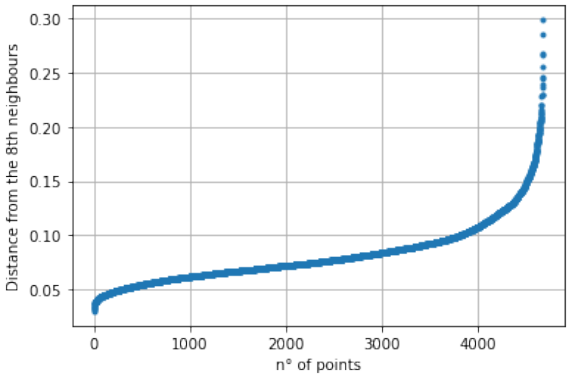

Figure 15: Average distances of points from their 8th neighbours

As it can already be seen from the plot most points are near each other and their distances increases only for 1000 of them, from the plot the curvature seem to be at its maximum in the range [0.10, 0.15], I approximated it at the middle value of $\varepsilon = 0.125$ and I computed how many points were nearer than this distance, the result was that 4306 points were at this average distance or less and the other 376 points had a higher average distance. Of this 376 points 95 were identified as outliers and the others were included into a cluster with all the other points. I listed these 95 outlier points identified by DBSCAN, I noticed some of them had affinity with extreme levels of arousal and valence, so I plotted the results of DBSCAN on that pair in the following scatterplot (Figure 16). No other cluster appeared with these parameters. With other parameters it is possible to get some very small clusters of correlated words, but I didn't found any reason to consider those parameters and results more precise than the one I represented. Also, of the 95 outliers identified through DBSCAN, only 33 were in common with the previous 385 I counted, 1.5 times the IQR beyond the first and third quartiles of each feature distribution, in brief a third of a fourth was also recognized as outlier by DBSCAN. With other parameters I found similar proportions in the number of identified outliers, and only with higher ε where the numbers of outliers greatly reduced I reached a proportion of 1/2 of outliers identified also by DBSCAN, but I was satisfied with representation I have given so far.

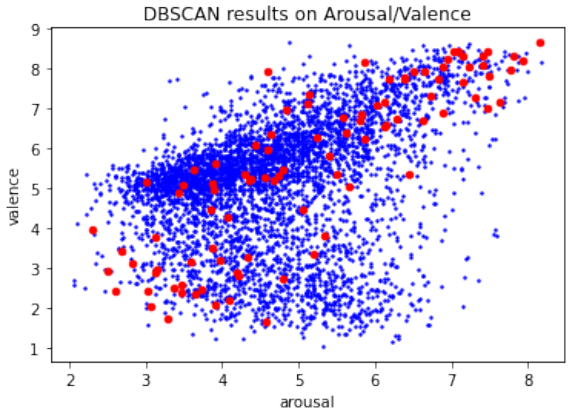

Figure 16: Outliers across Arousal/Valence in the DBSCAN clustering results

3.3 Analysis by Hierarchical Clustering

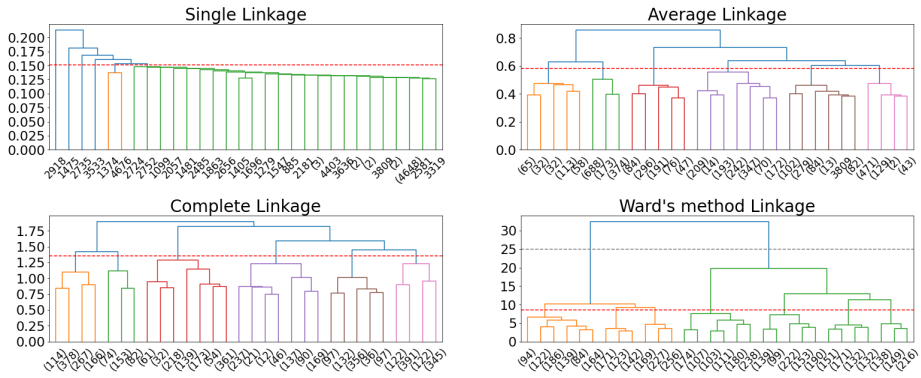

Figure 17: Dendrograms with their respective clusters

Starting from the same PCA, focusing only on the three most important PCs, I applied four different linkage criterions of the agglomerative hierarchical clustering algorithm. As it can be seen in Figure 17, considering also potential outliers, six distinct groupings emerged with each criterion. I evidenced this cut on the dendrograms with a dotted red line and the exception, in the case of the ward criterion with also two possible clusters, with a dotted gray line.
By working with the single linkage criterion, the points from the proximity matrix tend to aggregate into a single large cluster, by cutting the clustering at a distance of 0.151 we get a single large cluster with most points, a smaller cluster with two points, namely the words "edifice" and "zephyr", and finally 4 other isolated points. These six isolated words were all identified as outliers by the previous configuration of DBSCAN, with the exception of "music" and "romanticize" which appeared though with slightly more restrictive parameters.

Cluster (color)	AVERAGE		COMPLETE		WARD'S METHOD	
	Tot.	n° of polysemic	Tot.	n° of polysemic	Tot.	n° of polysemic
BLUE	1235	134	925	133	1088	123
RED	1247	125	718	54	803	56
GREEN	694	14	1038	61	913	39
CYAN	561	64	980	79	503	39
YELLOW	645	11	712	14	586	9
MAGENTA	300	31	309	38	789	113

Table 4: Clustering results with three linkage criterions

To compare how the different algorithms aggregated the data I referred to the first PC (which accounts for 40% of the variance) and I analyzed how the clusters would distribute back on the high loading pair, concreteness-age of acquisition, as it can be seen in Figure 18. In order to better compare the differences between the linkage criterions I assigned a color to the most similar identified clusters, to select the colors to assign I computed the centroids of each algorithm and I clustered them. The 18 centroids were well separated and grouped together with the concreteness-aoa pair, for practicality I clustered them with the complete linkage criterion and then I mapped the color of the centroids back to the original clusters.
As it can be seen from Table 4 and Figure 18, of the three linkage criterions the Average and

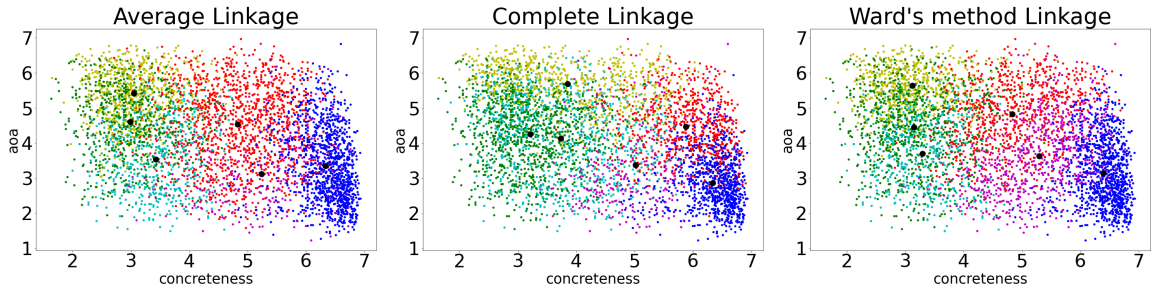

Figure 18: Hierarchical clustering on the concreteness-aoa pair

Ward's method get similar results on the six clusters level. The blue cluster, with high level of

concreteness seems to be the one the hierarchical algorithm identifies better, even though with the complete linkage we get different results. In particular by analyzing the output of the Complete linkage on another pair, valence-gender, the "Cyan" cluster, which seems to be more dispersed on the concreteness-aoa pair, capture more points in the low valence spectrum, but this criterion gets lower results both with polysemy and with high levels of valence (the "Green" cluster) which gets way more dispersed with respect to the Average Linkage criterion. Regarding polysemy, which wasn't an input, it seems clear that the Average Linkage criterion seems to capture it better, the ward's method is similar but it tends to break the "Red" Cluster and it doesn't capture as much polysemic words in the "Cyan" cluster as the other two criterions.
Finally, by looking at the dendrograms in Figure 17 the euclidean distance I decided to cut to determine the number of clusters were respectively, 0.115 for the single, 0.58 for the average, 1.35 for the complete, and 8.5 for the ward, but in the case of the Ward's method criterion I also checked how the algorithm would behave with 2 clusters at the distance of 25, but it didn't give any insight. For these reasons, it seems the average linkage criterion is the best one with this dataset.

3.4 Comparison of the three Clustering Algorithms

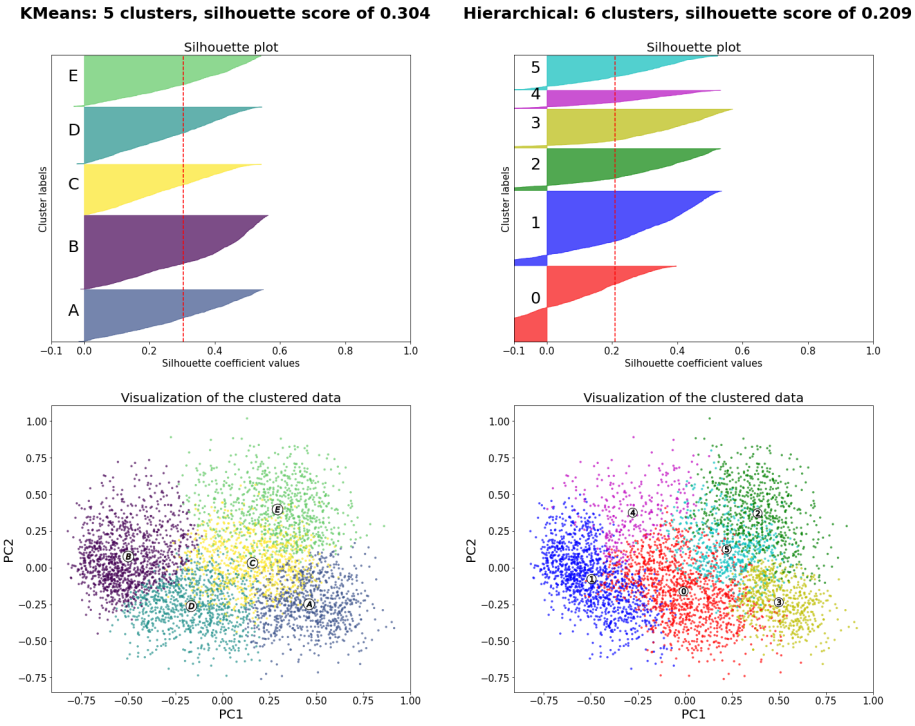

Figure 19: Silhouette Analysis with K-Means and the Hierarchical algorithms

Considering the results I got, I chose to compare the output of KMeans and the Hierarchical clustering with the average linkage criterion. I excluded DBSCAN because its results identified only a core single shape while aggregating the too distant points as outliers. DBSCAN produced anyway interesting results with different hyperparameters, namely it identified some very small cluster of some words in specific semantic areas, but to capture some of them it required some heavy tinkering with the parameters. The analysis I performed focused on one internal validation method, namely the silhouette score in Figure 19, and one external validation method, or rather how much the model could group together the polysemic words, in Table 5, there I also computed another metric, the SSE of each cluster (not on PCA's data), to compare the percentages of the two methods with the SSE of the clusters.
I performed my analysis on the range of possible clusters between 2 and 6, the results allowed us to show here only the clusters in continuity with what I already identified for each algorithm, as the comparison with the same number of clusters doesn't change that much, and the best number of clusters was anyway 5 for KMeans and 6 for the Hierarchical.
From these measures I concluded that, of the three algorithms, KMeans worked better on the dataset. By looking at the silhouette scores, even though I am working with average low scores, I got 0.10

more points with KMeans where the clusters tended to include fewer outliers (in Hierarchical there are a lot in the Red (0) one). By looking at specific clusters it's also clear how the almost corresponding Violet (B) of Kmeans and Blue (1) of the Hierarchical, which is the area that tend to capture most polysemic words, we get that KMeans grouped 10% more while keeping a lower SSE, and by looking at the second cluster by the number of polysemic words between the Yellow (C) of KMeans and the Red (0) of the Hierarchical it is clear KMeans has created a more compact cluster (almost half of the SSE).

KMEANS						HIERARCHICAL (AVERAGE LINKAGE CRITERION)					
Cluster	Tot.	poly.	% of cluster size	% of polysemic	**SSE**	**Cluster**	Tot.	poly.	% of cluster size	% of polysemic	**SSE**
BLUE (A)	854	14	1,64%	3,69%	**7.470,85**	**RED (0)**	1247	125	10,02%	32,98%	**13.994,50**
VIOLET (B)	1219	172	14,11%	45,38%	**10.354,35**	**BLUE (1)**	1235	134	10,85%	35,36%	**11.265,90**
YELLOW (C)	835	97	11,62%	25,59%	**7.850,34**	**GREEN (2)**	694	14	2,02%	3,69%	**5.607,06**
CYAN (D)	938	65	6,93%	17,15%	**8.269,57**	**YELLOW (3)**	645	11	1,71%	2,90%	**5.417,91**
GREEN (E)	836	31	3,71%	8,18%	**7.211,40**	**MAGENTA (4)**	300	31	10,33%	8,18%	**2.708,46**
						CYAN (5)	561	64	11,41%	16,89%	**4.913,71**
Tot.	**4682**	**379**	**8,09%**	**100,00%**	**41.156,51**	**Tot.**	**4682**	**379**	**8,09%**	**100,00%**	**43.907,53**

Table 5: Various metrics compute to compare the KMeans and Hierarchical algorithms

4 Classification

For classification I decided to avoid the PCA which would make it difficult to interpret the results. In continuity with what I have done for the clustering task, I decided to focus on "polysemy" as the target variable of the following two models. Then, I split the dataset in a training set and a test set, with the test set size ratio of 30%, stratifying the choice of the random samples on "polysemy", the resulting training set contained 3277 elements (with 265 polysemous words), the test set 1405 (with 114 polysemous words). I considered the imbalance of the "polysemy" class to be a problem, to solve it I decided to perform a random undersampling of the training set. The undersampling was done this way: I split the training set into two, one with and one without polysemous words. Then I split the training set without polysemous words (the majority class), into four smaller sets of 66 elements (one with 67) performing a random undersampling stratified on the length quartiles, to avoid losing the ratios of the length distribution, the most important for polysemy as I showed in the data understanding part. After the undersampling the two classes were perfectly balanced (265 elements each) I merged the 4 smaller datasets with the part of the training set with polysemous words I set aside. Thus, for the two following algorithms I used a balanced training set of 530 elements and test set of 1405 elements.

4.1 Decision Tree Algorithm

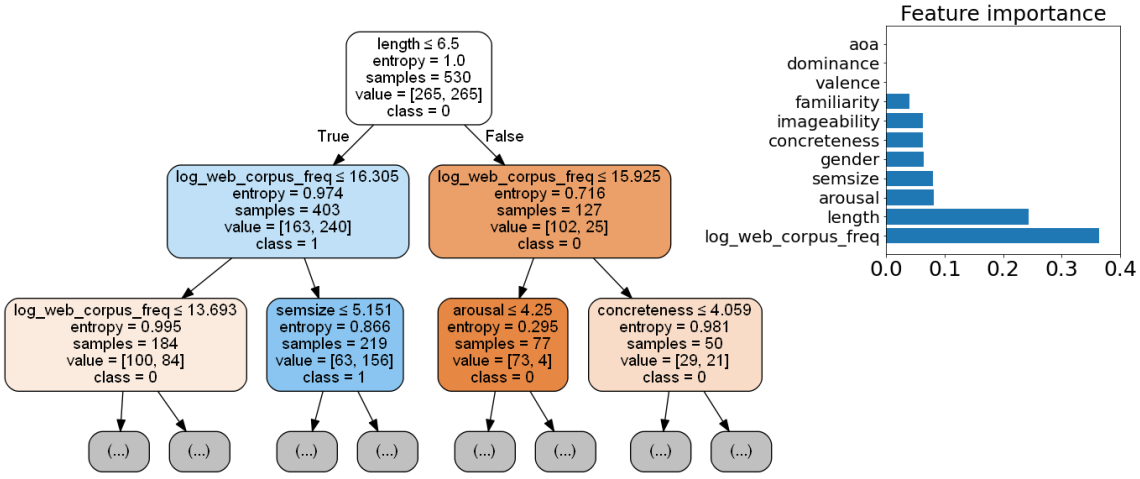

Figure 20: Decision Tree and Feature Importance plot

The first step before running the algorithm was to determine the parameters, to this end I performed a gridsearch with a 5 fold cross validation while shuffling the training set (otherwise it would also pick subsections with a homogeneous length distribution and without polysemous words from the undersampled training set). The gridsearch looked for the best f1 score, the resulting best classificator

had a mean validation score of 0,676 (with 0,046 standard deviation), the criterion was determined to be "entropy", the max depth of the tree 5, the minimum sample of a resulting leaf also 5, and the minimum number of samples to split further 2. By looking at the other classificators by mean validation score, in general it appeared that entropy was the better criterion to split and that the decision tree shouldn't have much depth. Thus, I ran the decision tree with these parameters, as it can be seen from Figure 20, the frequency of the word was the most important feature with almost 0.4 importance, followed by length with 0.24 importance.
Then I moved to test the model. Before presenting the results and comparing them I prefer to present how the results would be with the best threshold I computed, to have a measure how the standard model predictions came close to best possible predictions it could give. I computed this threshold by running the classification algorithm and then by minimizing an objective function to calculate the threshold in the upper-left corner of the ROC curve (the plot can be seen in Figure 22), I considered that this threshold would be the one used in the equation $TPR_{th} = 1 - FPR_{th}$, thus the nearest possible threshold in the data to make the true positive rate and the true negative rate equals, then it's minimized by $argmin_{th}(|FPR_{th} + TPR_{th} - 1|) \ \forall \ th \in Th$, with Th as the set of all possible thresholds, in particular the minimum threshold is $0,58$ which is satisfied for $|FPR_{0,58} + TPR_{0,58} - 1| = 0,254$, in the following Figure 21 I plotted in red this point on a plot which shows all the points of $|FPR_{th} + TPR_{th} - 1|$ on the y axis and the various thresholds on the x axis, then I also showed the confusion matrix corresponding to the minimized threshold.

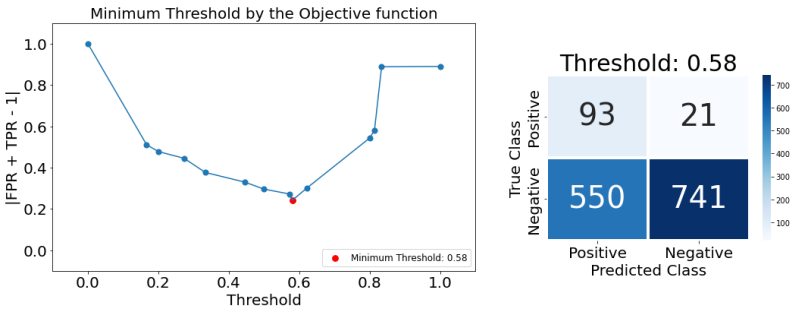

Figure 21: Decision Tree: best threshold by the objective function

Then, hereunder I also plotted the ROC curve and the Precision-Recall curve to measure how much close I got to the best possible prediction. As it can be seen the threshold 0,58 is very close to threshold 0,50 in the ROC Curve, both are around a recall of 0,80 and a false positive rate of 0,40. we can see clearly that if we sacrifice around 0,15 of Recall we can maximize the precision but only gaining 0,04, half of the baseline. Other thresholds are shown as a comparison. The area under the ROC curve is not very high, but if I wanted to maximize it through the gridsearch it would be practically identical, and by using the non-undersampled training set I would lower drastically precision and recall at the 0,50 threshold.

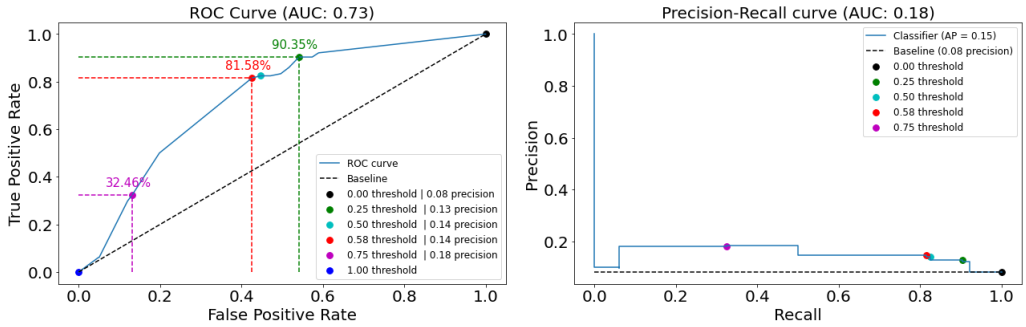

Figure 22: Decision Tree: ROC Curve and PR Curve

Indeed, following the same procedure without undersampling we can reach similar results only by looking for the highest area under the ROC curve in the gridsearch, with a model which does make similar predictions only from a threshold of 0.10, but that doesn't even make one correct prediction at a threshold of 0.50 which in particular produces predictions identical to threshold 1, or rather the dummy constant classifier I used also later for comparison. I didn't present that model even

though it reaches slightly better results in terms of precisions (but always around twice the baseline on average) and recall at the said threshold of 0,10, in particular it reaches 0,825 of recall and 0,17 of precision, with an AUC of 0,78, but it's not worth it for how unbalanced the model is.

TRAINING SET (0.50 THRESHOLD)		
	Predicted	
	Positive	Negative
Actual Positive	246	19
Actual Negative	108	157
	Polysemous	
Support	265	
Recall	92,83%	
Precision	69,49%	
F1 Score	79,48%	
Accuracy	76,04%	

TEST SET (0.50 THRESHOLD)		
	Predicted	
	Positive	Negative
Actual Positive	94	20
Actual Negative	578	713
	Polysemous	
Support	114	
Recall	82,46%	
Precision	13,99%	
F1 Score	23,92%	
Accuracy	57,44%	

TEST SET (0.58 THRESHOLD)		
	Predicted	
	Positive	Negative
Actual Positive	93	21
Actual Negative	550	741
	Polysemous	
Support	114	
Recall	81,58%	
Precision	14,46%	
F1 Score	24,57%	
Accuracy	59,36%	

TEST SET (0.00 THRESHOLD)		
	Predicted	
	Positive	Negative
Actual Positive	114	0
Actual Negative	1291	0
	Polysemous	
Support	114	
Recall	100,00%	
Precision	8,11%	
F1 Score	15,01%	
Accuracy	8,11%	

Table 6: Decision Tree: Classification Report comparison

Finally in Table 6, there are the confusion matrices, and the classification reports referred to the polysemous class, with accuracy obviously referred also to the other. It is clear this model is as good as it could be, as it produces similar results to its best possible threshold with slightly higher precision and a much higher accuracy in respect to the Dummy classifier (indicated as 0.0 threshold).

4.2 KNN Algorithm

Then I also tried the K Nearest Neighbours algorithm. The preprocessing steps are the same I indicated at the beginning of the Classification section, but considering this algorithm works with distances I used the Min-Max scaler on the continuous variables (thus keeping length and polysemy out of the rescaling).
Then, as before I performed a gridsearch looking for the best value of the f1 score with K neighbours from 1 to 25, I did not search for the area under the curve because I want to find first and foremost a balanced model which doesn't require adjusting the threshold as I also did before. Through the gridsearch I performed also a 5 fold cross validation while shuffling the training set as before, the resulting best parameter was k = 15 with a mean validation score of 0,701 (with a standard deviation of 0,305), slightly better results with respect to the Decision Tree.

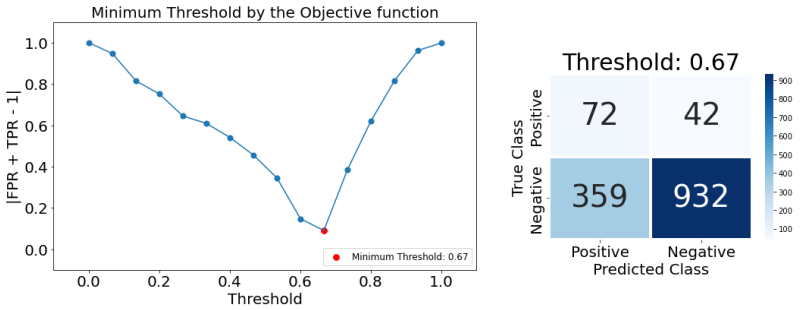

Figure 23: KNN: best threshold by the objective function

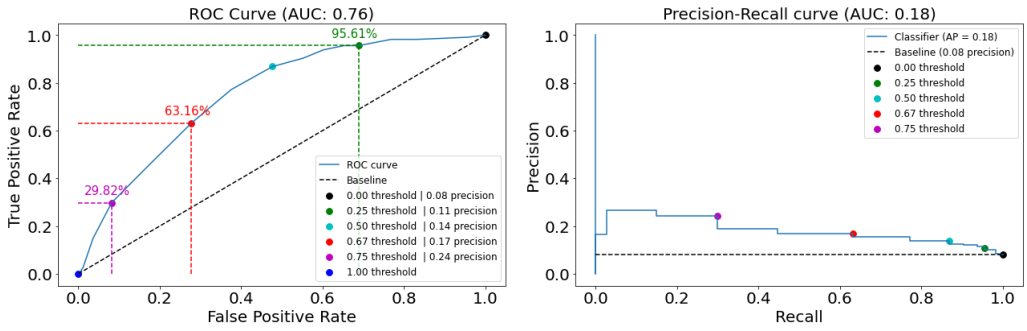

Figure 24: KNN: ROC Curve and PR Curve

Then, just as before, I trained the model and I moved to testing it. Here if I minimized through the same kind of objective function the threshold, the resulting best one would be at 0,67 as it can

be seen from Figure 23, a little far from the default threshold and higher than the one I found with the Decision Tree. Then, in Figure 24 I plotted the ROC curve and the Precision-Recall curve, the model performance are slightly better than those of the decision tree. The cost of lower recall of the best threshold doesn't seem to improve much the precision though.
Finally, herunto I built the table comparing again the confusion matrices and the derived performance metrics of the training set, the 0,50 threshold and the 0,67 threshold and finally again a constant dummy classifier corresponding to the 0,00 threshold. This particular model seems to have not only higher AUC but also it seems to get more accurate for the 0,67 threshold.

TRAINING SET (0.50 THRESHOLD)		
	Predicted	
	Positive	Negative
Actual Positive	236	29
Actual Negative	145	120
	Polysemous	
Support	265	
Recall	89,06%	
Precision	61,94%	
F1 Score	73,07%	
Accuracy	67,17%	

TEST SET (0.50 THRESHOLD)		
	Predicted	
	Positive	Negative
Actual Positive	99	15
Actual Negative	616	675
	Polysemous	
Support	114	
Recall	86,84%	
Precision	13,85%	
F1 Score	23,88%	
Accuracy	55,09%	

TEST SET (0.67 THRESHOLD)		
	Predicted	
	Positive	Negative
Actual Positive	72	42
Actual Negative	359	932
	Polysemous	
Support	114	
Recall	63,16%	
Precision	16,71%	
F1 Score	26,42%	
Accuracy	71,46%	

TEST SET (0.00 THRESHOLD)		
	Predicted	
	Positive	Negative
Actual Positive	114	0
Actual Negative	1291	0
	Polysemous	
Support	114	
Recall	100,00%	
Precision	8,11%	
F1 Score	15,01%	
Accuracy	8,11%	

Table 7: KNN: Classification Report comparison

4.3 Performance Comparison of the two models

By looking at Table 6 and Table 7 we can notice how the KNN model at threshold 0,50, the standard split of the predictions, presents a slightly higher recall while the decision tree a even more slight higher precision and accuracy. Though the average performances are similar, by looking also at the areas under the ROC Curves and the PR curves, the results are slightly in favor of KNN. To have a clearer view of how much the two models differ I plotted here the comparison for the three curves I plotted before. The blue lines represent the Decision Tree while the orange ones KNN. I compared also the best thresholds together with the curves, the decision tree's best threshold as a red dot while the KNN as a green one.

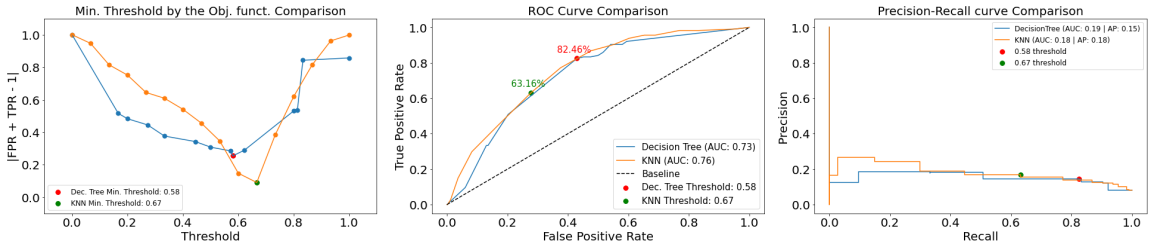

Figure 25: Comparison of the two models

As it can be seen these two models are almost equivalent, I only get that the best threshold of KNN minimizes more the objective functions and it captures in a more accurate way the polysemous and non-polysemous words than the Decision Tree and the KNN default threshold, in contrast it brings a lower recall than the decision tree's best threshold and its 0,50 one, while also being farther apart from the default threshold of the model.
From this we can get that the best predictions of KNN are not represented well by the 0.50 threshold, even though it depends if Recall or Precision is more important as the Recall is higher for KNN's 0.50 threshold with respect to the showed configurations of the Decision Tree. Also the average precision and the areas under the ROC and PR curves are higher for KNN. For this reason I concluded KNN worked better on average, even though the two models are comparable.

5 Pattern Mining

The first step to search for frequent patterns was to make categorical variables out of the continuous ones. For this purpose I decided to divide the continuous variables into 3 bins I cut each based on the 0,33 percentile of the distribution, to interpret the pattern easily I changed the scale in "low", "medium" and "high". I reported in Table 8 the intervals and their new denominations.

In continuity with what I did in the data understanding section I made and exception for gender to evidence the extreme of the distribution and I cut not on the distribution quantiles but in three equally distant interval edges, this way I avoided too large intervals for the extremes (that would capture more "ambiguous" terms), as I considered the most interesting patterns of gender would be connected to its extremes. Thus all the variable are perfectly balanced between each interval, while gender has 3498 words of ambigous gender, 681 male and 503 female perceived terms.

	arousal	valence	dominance	concreteness	imageability	familiarity	aoa	semsize	log_web_freq		gender
Low	(2.06, 4.09]	(1.03, 4.73]	(1.94, 4.77]	(1.64, 3.61]	(1.74, 3.88]	(1.65, 5.00]	(1.22, 3.48]	(1.37, 3.71]	(9.45, 14.78]	Feminine	(0.99, 2.99]
Medium	(4.09, 5.11]	(4.73, 5.77]	(4.77, 5.42]	(3.61, 5.50]	(3.88, 5.61]	(5.00, 5.79]	(3.48, 4.82]	(3.71, 4.63]	(14.78, 16.42]	Ambiguous	(2.99, 4.98]
High	(5.11, 8.18]	(5.77, 8.65]	(5.42, 8.37]	(5.50, 6.99]	(5.61, 6.94]	(5.79, 6.94]	(4.82, 6.97]	(4.63, 6.91]	(16.42, 21.43]	Male	(4.98, 6.97]

Table 8: Discretized Variables

5.1 Frequent Pattern extraction

After I preprocessed the dataset for the task I extracted a few frequent patterns, I decided to which support considering a pattern frequent based on the plots presented in Figure 26. I focused my attention on Maximal Frequent itemsets as it can be seen from the first figure, as it is more compact, then I plotted further the distribution of polysemous words in maximal frequent itemsets on the percentage of support.
As it can be see the best support to look for frequent itemsets would be around 2% or 3%, as it removes most noise and it starts to decrease more slowly with higher supports, it's not a surprise polysemous words do not appear in most transactions as they represent only 8,09% of words, the same argument applies for Feminine words as their percentage is 10,74%.

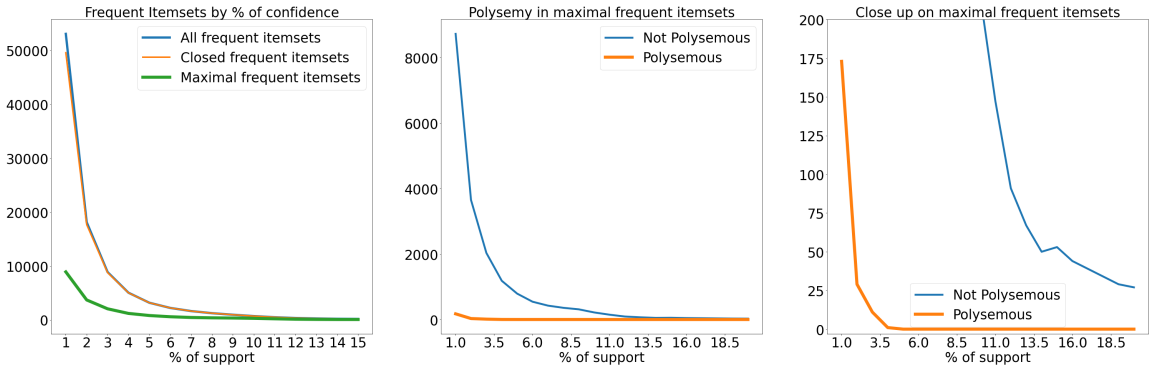

Figure 26: Frequent Itemsets and Polysemous itemsets

Starting from a 2% support threshold (minimum of 94 words in absolute terms) I looked for some interesting patterns, focusing only on the maximal frequent itemsets to avoid having their repetitive subsets. I found in general 3693 maximal frequent itemsets, here I presented just 7 of the most interesting ones presenting "Polysemous" in the transaction, there were 33 of them in total (all around 2% as expected by looking at the previous plot):

- {Polysemous, 4_length, high_frequency} 2,01% support
- {Polysemous, 4_length, low_aoa} 2,01% support
- {Polysemous, high_familiarity, high_frequency, low_aoa, Ambiguous Gender} 2,03% support
- {Polysemous, high_concreteness, low_aoa, low_semsize} 2,05% support
- {Polysemous, high_imageability, low_aoa, low_semsize} 2,05% support
- {Polysemous, high_concreteness, high_imageability, low_aoa} 2,46% support
- {Polysemous, high_concreteness, high_imageability, low_semsize} 2,46% support

5.2 Association Rules extraction

Before extracting some association rules I decided to have an overview of how many association rules I would get from different levels of confidence and support. For this scope I set 8 as the minimum number of items to consider the rules valid, a lower number would result in too high number of rules that would make further anlysis difficult to interpret. Then I plotted in Figure 27 an overlay imshow with a grey color-map. Considering I already noticed the support should be kept low to have a reasonable number of frequent itemsets, in order to find the association rules I decided to look only at supports between around 1% and 3%, which would be in absolute terms from around 50 to 150.

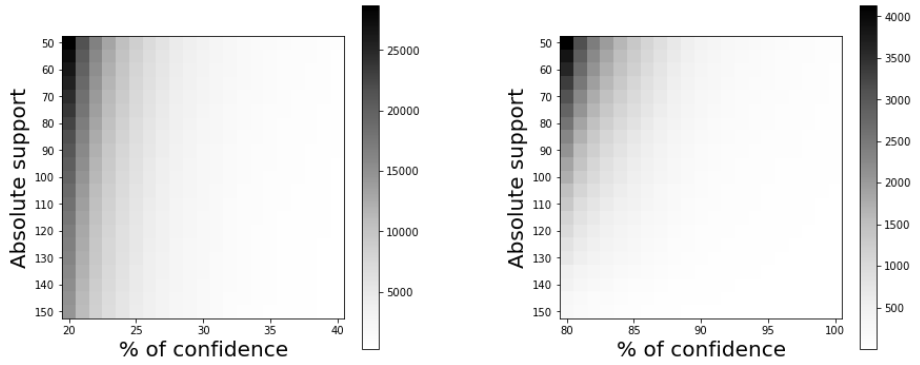

Figure 27: Number of association rules with different levels of confidence and support

Then I looked for the right range of confidence to examine, the highest the better, but Polysemous words appeared to be quite rare as they didn't seem to create a strong logical implication with elements with which they appeared together. I plotted the number of association rules implying various classes with a support of 2% with respect to level of confidence from 1 to 100, in Figure 28, again with a condition of 8 as the minimum number of items to have a valid rule.

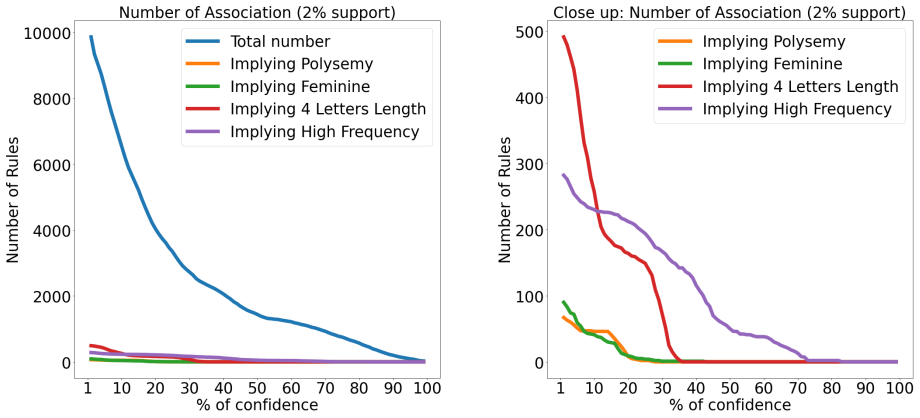

Figure 28: Number of association rules implying the "Polysemous" class

Indeed the maximum percentage of confidence I found for polysemous words was around 20% to 30% but at the same time there also seemed to be yet a high number of association rules in the confidence range beyond 80% as it can be seen frome the overlay imshow plotted in Figure 27. In particular there are rules with 80% of confidence falling under the "high frequency" class.
Then I discarded the minimum number of items to compute the whole set of rules with a minimum confidence of 25% and a minimum support of 2%, I extracted rules that would imply polysemy in the transaction.
Here the most interesting; respectively: the first rule has the highest number of items, the second (on which I will focus later on) has the highest possible confidence and lift.

- {'high_familiarity', 'high_concreteness', 'high_imageability', 'low_aoa', 'high_frequency', 'low_semsize', 'Ambiguous Gender'} → {'Polysemous'} with a confidence of 0,27 and a lift of 3,38

- {'4_length', 'medium_dominance', 'high_frequency', 'Ambiguous Gender'} → {'Polysemous'} with a confidence of 0,37 and a lift of 4,55

As it can be seen I do not have high confidence, but with the high lift we've found high positive correlation (> 1) between the itemset and the target variable.

Then I extracted also another rule which implied "high frequency" to test later. From all the rules which implied "high frequency" this was the one with the highest confidence and lift:

- {'high_familiarity', 'high_valence', 'medium_concreteness', 'Ambiguous Gender'} → {'high_frequency'} with a confidence of 0,86 and a lift of 2,57

5.3 Application of the extracted rules

Out of the 3 rules presented, I tested the one with the highest confidence and lift for polysemy, to compare it with the work done during the classification task. In the case I try to predict polysemy on the test set I used, during the classification task, with the aforementioned rule, I would have very poor results. With 6% precision and 2% recall (2 true positives, 29 false positives and 112 false negatives), it's not strange though that it guessed right almost all the true negatives with a total accuracy at 90%. It doesn't need to be stated that it's worse than both the classification models I presented.

Then if instead I try to predict the class "high frequency", to have better means of comparison, I built briefly a model from scratch, with KNN without undersampling, based around the new variable. The test set as before was 30% of the original data and the K Neighbours I set through the grisearch was 5.
Then, I compared the results of the rule and the model on the same test set, which can be seen in Figure 29. The "High Frequency" association rule (in a red confusion matrix) gets 4% recall and 40% of precision, which is good but nothing compared to the goodness of the KNN model I built, which predict right the "High Frequency" class 285 times, with a recall of 61% and a precision of 63% at the 0,5 threshold. It would even get to a recall of 78,63% with the best threshold 0,4 (minimized with the aforementioned objective function in section 4.1).

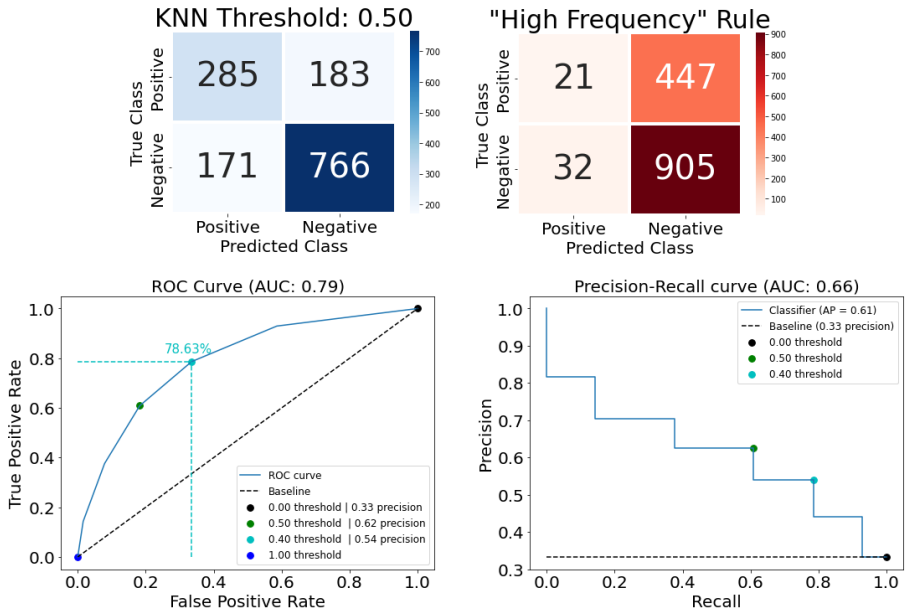

Figure 29: Confusion matrices comparison between the KNN new model and association rule)

Thus, I concluded the rules I extracted and presented cannot predict, as well as the classification models I built, the possible target variables.