

Predicting Mental Imagery Concreteness

Mark Angelo Gabriel

Advisors: Dr. Julie Ji, Dr. Wei Liu, Dr Francisco De Toni

Contents

1	Abstract	3
2	Review of Literature	3
2.1	Mental Imagery	3
2.2	Named Entity Recognition	4
2.3	Sentiment Analysis	5
2.4	Transfer Learning and XLNet	5
3	Data	6
3.1	Core dataset	6
3.2	Word concreteness dataset	7
3.3	Discretisation of Concreteness	7
4	Methodology	8
4.1	Inferential analysis	8
4.1.1	Baseline model	8
4.1.2	Word-level mined variables	9
4.1.3	One-vs-rest Classification	10
4.1.4	Decision Tree	10
4.2	Prediction with natural language processing techniques	10
4.2.1	BERT pretrained model	10
4.2.2	BiLSTM Classifier	11
4.2.3	Transformer model with XLNet	11
4.3	Prediction with classical techniques	11
4.3.1	SVM Classifier	11
4.3.2	Random Forest Classifier	11

4.3.3	XGBoost Classifier	11
4.4	Summary	11
5	Results and Discussion	11
5.1	Inferential analysis	11
5.1.1	Baseline model	11
5.1.2	Word-level mined variables	12
5.1.3	One-vs-rest Classification	12
5.1.3.1	Low Concreteness model	12
5.1.3.2	Medium Concreteness model	13
5.1.3.3	High Concreteness model	13
5.1.3.4	Concreteness Classifier	13
5.1.4	Decision Tree	14
5.2	Prediction with natural language processing techniques	15
5.2.1	BERT Pretrained models	15
5.2.2	BiLSTM Classifier	17
5.2.3	Transformer model with XLNet	17
5.3	Prediction with classical techniques	17
5.3.1	SVM Classifier	17
5.3.2	Random Forest Classifier	17
5.3.3	XGBoost Classifier	17
5.4	Summary	17

List of Figures

1	Annotation tools comparison	4
2	Pre-trained NLP models results on GLUE	6
3	Concreteness histogram	8
4	Discretised Concreteness histogram	9
5	Concreteness Decision Tree	15

List of Tables

1	Baseline Model	11
2	Word-level model	12
3	Low Concreteness Model	13

4	Medium Concreteness Model	13
5	High Concreteness Model	14
6	One-vs-rest Classifier	14
7	One-vs-rest Statistics by Class	14
8	Decision Tree	15
9	Decision Tree Statistics by Class	16
10	BERT Pretrained Model	16
11	BERT Pretrained Model with High Concreteness Word Count	16

1 Abstract

The COVID-19 pandemic has forced a sudden change in society that we weren't fully prepared for. Self-isolation and physical distancing has been imposed on everyone in varying degrees and the effect of it on people's mental wellbeing is an area of interest, especially if a second or third wave of the pandemic will require people to undergo isolation again. Certain members of the community are going to be affected more than others from this rapid disruption of the day-to-day because they are not getting the usual levels of social interaction they need or are used to to maintain a healthy mental wellbeing. It is in our interest to find which members of our community are at most risk with a method that is highly accessible given the physical restrictions that such a societal condition imposes.

This study aims to be able to predict a person's concreteness of mental imagery from their free-text response to a scenario using natural language processing techniques.

2 Review of Literature

2.1 Mental Imagery

Mental imagery is the simulation of perceptual experience [7] across sensory modalities. It has been strongly linked to depression [9] which is our main area of interest. As detailed in Kosslyn et al's 2001 paper, the current techniques for measurements of a person's mental imagery involve visual tests, classification tasks, comparison tasks, questionnaires and interviews.

The usage of machine learning and natural language processing has not yet been deeply explored in extracting mental imagery parameters, though theoretically, it would be most similar to the existing method of using interviews and analyzing a person's speech and selection of words to derive the wanted parameters, except with machine learning, the assessment would be automated rather than needing a trained professional to go through each case individually, which would be an expensive process that won't be easily scalable in an isolation scenario where potentially a huge percentage of people are affected.

System	Web-based project creation	Project monitoring	Curation feature	Document propagation	Class labels		
					Dynamic	Hierarchical	Multi-label
BRAT	✗	✗	✗	✗	✗	✓	✗
GATE	✓	✓	✓	✗	✗	✗	✗
SANTO	✓	✓	✗	✗	✗	✗	✗
SAWT	✓	✓	✗	✗	✗	✗	✗
YEDDA	✓	✓	✗	✗	✗	✗	✗
WebAnno	✓	✓	✓	✗	✗	✗	✗
Redcoat	✓	✓	✓	✓	✓	✓	✓

Figure 1: Annotation tools comparison

Dr. Julie Ji’s research on mental imagery, the experience of perception in the absence of external sensory input, shows that the person’s quality of mental imagery-based simulations has a significant link to maintaining and amplifying emotional states [6]. In one study on mental imagery as a motivational amplifier to promote activities as a key treatment in depression [10], the Motivational Imagery group reported higher levels of motivation, anticipated pleasure, and anticipated reward for the planned activities compared to the Activity Reminder control group and the No-Reminder control group.

2.2 Named Entity Recognition

We aim to be able to utilise the free-text responses of the survey to create a model that performs better than a regression model from the human-graded mental imagery parameters that predicts a person’s risk of loneliness. To achieve this, the collected free-text responses have to be annotated with respect to the domain of interest. In our particular research area, it will be important to tag self actions (meditate, jog, sleep in, watch TV, etc) and interpersonal actions (call friend, video call with parents, have dinner at restaurant with partner) separately, which most pre-trained models aren’t trained to do. Due to this, and also because we have a relatively small sample size, we have the luxury of using manual annotation to improve our data’s richness and context.

In the paper Multilingual Twitter Sentiment Classification: The Role of Human Annotators by Mozetic et al, [8], they find that the quality of classification models depends much more on the quality and size of training data than on the type of model trained. Human annotators can help us maximise our model’s performance by ensuring good annotation quality, which can be done by setting an overlap value (the number of times a single corpus is annotated) of at least 2.

There are a lot of tools for data annotation available. Among these options are BRAT, GATE Teamware, WebAnno, SAWT, Yedda, SANTO and UWA’s Redcoat. We can see a comparison of their features in Figure 1. With research assistants to help out with our data as annotators, an important feature we are looking for in an annotating tool is the ability to propagate documents easily and for the tool to scale the workload automatically based on the number of annotators. Redcoat [12] has been selected as the annotating tool for that reason, alongside being web-based and easy to set up and distribute.

2.3 Sentiment Analysis

One of the mental imagery parameters we're concerned with is a person's emotional tone when they're describing what they think or see when they're given a scenario and a goal of starting lonely in an isolated setting and ending in a brighter, happier mental state. A person's emotional tone can be approximated with sentiment analysis techniques. XLNet has been used to identify optimism and pessimism in twitter messages by Alshahrani et al [1], which is similar to what we're trying to achieve. XLNet models are able to model negations and other semantic relationships by paying attention to key words, leading to a model accuracy of 0.9645.

A point of discussion would be if our research problem would work better with basic sentiment analysis or aspect-based sentiment analysis. Normal sentiment analysis grades the emotional tone of the overall text, while aspect-based sentiment analysis goes through the text to identify various aspects and determines the corresponding emotional tone for each one. There isn't much available literature on when each of the techniques are relevant or when each of them should be used over the other, but the spirit of parsimony would lead us to towards using aspect-based sentiment analysis only if there are particular aspects in our research domain that we want to track individually.

For the lack of literature on comparison between the two techniques on different domains, there is merit in trying both individually, or creating a model that uses both techniques in tandem and empirically comparing which technique works best for our research problem.

2.4 Transfer Learning and XLNet

Language is complex. In order to get reasonably performing models from scratch, one would need an extremely large dataset in addition to the computing power needed to process it. This is often impossible for many researchers, and a solution to this problem is transfer learning. Transfer learning is a machine learning technique where a model is trained on a particular problem, then reused in a similar domain to carry over the knowledge of the neural network. Depending on how different the pre-trained model is to your problem, Additional training needs to be done to finetune the model to your specific domain. [11]

Popular pre-trained NLP models include Google's BERT, Google and Carnegie Mellon University's XLNet, and Facebook's RoBERTa. All of these models are transformers which are deep learning models designed to handle sequential data using the concept of attention mechanisms. Attention mechanisms allow the model to access and use any previous states and utilises them based on relevance to the current node. [13]

In particular, we want a pretrained model that is strong in sentiment analysis. Looking at Figure 2 in the SST-2 column which stands for the Stanford Sentiment Treebank v2, we can see that XLNet [15] outperforms both BERT and RoBERTa in sentiment analysis on the SST-2 dataset.

While XLNet, BERT and RoBERTa all come from the same family of models, XL-

Model	MNLI	QNLI	QQP	RTE	SST-2	MRPC	CoLA	STS-B	WNLI
<i>Single-task single models on dev</i>									
BERT [2]	86.6/-	92.3	91.3	70.4	93.2	88.0	60.6	90.0	-
RoBERTa [21]	90.2/90.2	94.7	92.2	86.6	96.4	90.9	68.0	92.4	-
XLNet	90.8/90.8	94.9	92.3	85.9	97.0	90.8	69.0	92.5	-
<i>Multi-task ensembles on test (from leaderboard as of Oct 28, 2019)</i>									
MT-DNN* [20]	87.9/87.4	96.0	89.9	86.3	96.5	92.7	68.4	91.1	89.0
RoBERTa* [21]	90.8/90.2	98.9	90.2	88.2	96.7	92.3	67.8	92.2	89.0
XLNet*	90.9/90.9[†]	99.0[†]	90.4[†]	88.5	97.1[†]	92.9	70.2	93.0	92.5

Figure 2: Pre-trained NLP models results on GLUE

Net differs by introducing permutation language modelling, where all tokens are predicted in a random, permuted order rather than the traditional sequential order. This helps XLNet learn non-adjacent and bidirectional relationships between the words in the corpus. Secondly, XLNet uses Transformer-XLs as the base architecture which enables learning dependency beyond a fixed length without disrupting temporal coherence. [3] Transformer-XLs, even without the permutation language modelling, has been shown to give performance increases in NLP tasks.

For the above reasons, and for the benchmarks set out in Figure 2, XLNet will be our pretrained model of choice. In addition, HuggingFace [14] provides a transformer library that wraps all of these transformers for use with PyTorch or TensorFlow in a neat package that allows us to utilise the base XLNet model and the large XLNet model, alongside any other pretrained transformer model if we wish to make model comparisons.

3 Data

3.1 Core dataset

The core dataset for this study is a conglomerate of surveys from volunteers (n=1269) including a baseline day zero survey, daily surveys, and weekly checkpoints up to the fourteenth day. The baseline survey asked for the volunteers’ mental health status, education, employment, age, coinhabitant count, isolation experience, the Hospital Anxiety and Depression Scale (HADS) scores, and other miscellaneous information. A loneliness score and social activity score have been assigned to each volunteer based on their answers to the relevant questions. In addition, a text response to the below instruction was requested.

The story begins with you feeling isolated and lonely.

The story ends with you feeling connected and closer to people.

Please come up with the steps you would take to achieve the story ending, providing as much detail as you can about your thoughts, feelings, and actions related to these problem-solving steps.

The responses are then manually rated by humans on four mental imagery parameters. Each response is assessed by two raters to improve reliability. The final score is calculated as the average of the two ratings. The four mental imagery parameters are as below:

- Number of relevant problem-solving steps: Discrete steps that helps the person to reach the desired goal or to overcome an obstacle along the way
- Solution Effectiveness: How much you think the described solution would maximise positive and minimise negative consequences in the short-term and long-term, both personally and socially
- Solution Concreteness: The degree to which the described solution reflects concrete plans, i.e. involving specific actions, times, places, and people
- Emotional Tone: The degree to which the participant sounds negative (downbeat/pessimistic/unsure) or positive (upbeat/optimistic/confident)

The daily surveys track the the volunteers' depression and anxiety levels, their Warwick-Edinburgh Mental Well-being Scale (WEMWBS) scores, physical activity levels, social activity levels, and if and how often they've been outside of their accommodation. The weekly surveys include the normal daily survey questions and adds HADS, loneliness, and optimism score assessments.

3.2 Word concreteness dataset

The paper *Predicting Word Concreteness and Imagery* [2] by Charbonnier and Wartena has outputted a dictionary of words and their concreteness ratings. These scores are obtained by training a regression model using precomputed word embeddings from GoogleNews and fastText, suffixes and the word's part of speech (POS) tag as the features and manually scored concreteness values as the target. The dataset contains 39955 unique singleton and bigram words.

3.3 Discretisation of Concreteness

Concreteness is recorded as a numerical number between 1 and 5 with half steps and some quarter steps in between due to the averaging done on the varying ratings on some of the observations. Given our dataset is small (n=1236, 1088 if you remove invalid responses), it is of interest to further decrease the number of bins for Concreteness. It will also help with the interpretation of the model, as in practice, we will be looking for people with generally low, medium or high concreteness values, not necessarily if they have a score of 1 or 1.5.

Three levels have been deemed important in the scope of this research: Low, Medium, and High. The cutoff points were chosen to minimise the relative class sizes, as if we were to trisect the number scale, we would end up with a high class

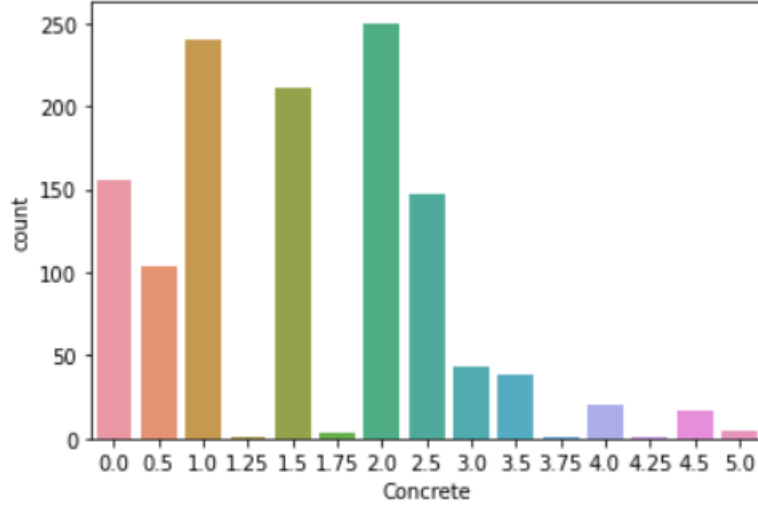


Figure 3: Concreteness histogram

imbalance particularly against High. See Figure 3 to see the raw concreteness distribution. Concreteness scores in the $[0, 1.5)$ range are classed into Low, $[1.5, 2.5)$ into Medium, and $[2.5, 5]$ into High. While the number of observations falling under High Concreteness is still fairly lower than those in Low or Medium as can be seen in Figure 4, any other cutoff combination for our dataset would make further imbalance the classes. This is just the nature of our data where concreteness scores are concentrated in the lower ranges.

Aside from the three levels, we will also explore models focusing on a boolean classification of Low Concreteness or not.

4 Methodology

4.1 Inferential analysis

The goal for this section is to discover key features in the data that can be easy identifiers for concreteness without resorting to unexplainable prediction models. It would be valuable to know particular features of low and high concreteness individuals as it may give us a better understanding on the subject matter.

4.1.1 Baseline model

The baseline model will be a predictor model for concreteness from non-textual information available from a candidate at day zero. Information such as loneliness and social activity scores available at future days will not be accessible to this model. Its purpose is to find out how accurate we can get our predictions to be of concreteness given a person's own profile, demographic, and self-assessment alone. The full list of parameters used for this baseline model is itemised below:

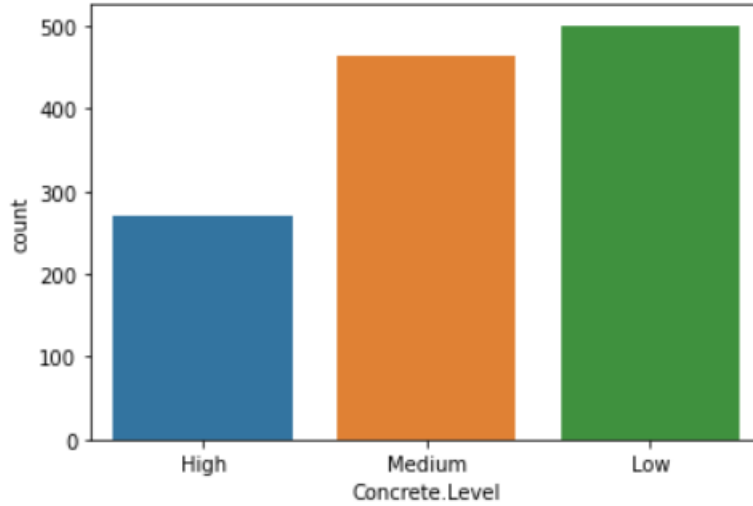


Figure 4: Discretised Concreteness histogram

- Demographical information: age, gender, education, employment, and working from home status
- Self-rated assessments of mental imagery in terms of actions, people, emotions, and vividness
- Number and type of cohabitants, including pets
- HADS scores for depression and anxiety calculated through a series of questions
- Number of days practicing social distancing to date
- Perceived risk of loneliness, worry, stress, family conflict, inactivity and listlessness
- If any current or past mental health diagnoses exist

Stepwise selection will be used to narrow down the predictive parameters to only the most significant few to maximise the AIC score of the model. Multiple linear regression will be the regression technique of choice in order to better understand the effect of each significant variable on concreteness as compared to less inferential techniques such as support vector machines.

4.1.2 Word-level mined variables

Word-level approaches will be applied to the response texts to gather high level information to see if non-NLP text approaches can bring an improvement in performance to our Concreteness prediction model. Primarily, the word concreteness dictionary from Charbonnier’s *Predicting Word Concreteness and Imagery* paper

[2] will be utilised to gather new features from the responses in our observations. Bigrams will also be included in the data mining.

Multiple linear regression will be used similar to the baseline model, with the difference being the addition of new data on word concreteness. Stepwise selection will be used to narrow down the predictive parameters, with the new fields being added to the model afterwards if not already selected by the algorithm.

4.1.3 One-vs-rest Classification

The discretisation of Concreteness will allow us to use the One-vs-rest logistic classification. Three individual models will be created for predicting if an observation is of Low, Medium, or High Concreteness. This will also make it possible to see if different factors count for different Concreteness levels which may come useful for inference of what variables to look out for for specific classes.

Stratified sampling will be done for the train test split as it is important to make sure that each class is relatively equally represented given the slight imbalance in class sizes.

4.1.4 Decision Tree

Decision Trees are highly explainable classification models which could give us heuristics for classifying a person's level of Concreteness. We will be using Conditional Inference Trees [5] using the `ctree` command from the `party` package in R.

4.2 Prediction with natural language processing techniques

4.2.1 BERT pretrained model

BERT (Bidirectional Encoder Representations from Transformers) [4] is a popular application of bidirectional training of transformers to language modelling. Transfer learning has been shown to be extremely valuable especially on small datasets such as in this study, and we are interested to see what kind of performance improvements we will see with using BERT pretrained models in a feed forward neural network ending with three nodes representing the three concreteness classes with a softmax layer applied at the end for classification.

In addition, we will try to augment the model above with the high concreteness word count variable to see if it will further push the performance upwards.

4.2.2 BiLSTM Classifier

4.2.3 Transformer model with XLNet

4.3 Prediction with classical techniques

4.3.1 SVM Classifier

4.3.2 Random Forest Classifier

4.3.3 XGBoost Classifier

4.4 Summary

5 Results and Discussion

5.1 Inferential analysis

5.1.1 Baseline model

Using the stepwise algorithm for model selection on linear regression has yielded the following model:

Coefficient	Estimate	Standard Error	p-value
Intercept	0.7309	0.4178	0.0806
Perceived Risk (Listlessness)	-0.1065	0.0298	0.0004
Self-rated Mental Imagery Score (Action)	0.0798	0.0289	0.0058
Coinhabitants (Adult, Other) ¹	-0.1901	0.0798	0.0175
Mental Health Diagnosis (none)	0.8957	0.4099	0.0291
Mental Health Diagnosis (unsure)	1.2529	0.4995	0.0123
Mental Health Diagnosis (yes - current)	0.9820	0.4133	0.0177
Mental Health Diagnosis (yes - past)	0.0485	0.0314	0.1226
Self-rated Mental Imagery Score (Vividness)	0.0485	0.0314	0.1226

Table 1: Baseline Model

This baseline model has an adjusted R^2 score of 0.0400 which means that the variables selected account only for approximately 4% of the variance of Concreteness, which is abysmally low and unusable for prediction purposes. This low score can mean two possible things. First is that the relationship between the predictors and the target variable may not be linear in nature. Second is that the baseline variables may simply not be strongly related to a person’s concreteness of mental imagery that is estimated by humans by assessing the response text.

¹Number of adult coinhabitants that aren’t your family, partner, or friend.

5.1.2 Word-level mined variables

A series of new fields have been generated using the word concreteness dictionary from Charbonnier’s *Predicting Word Concreteness and Imagery* paper [2]. Concreteness information from both individual words and bigrams have been mined. Raw counts and densities (raw count over total word count) were obtained for both high concreteness (defined by having a score of greater than three in an approximate five point scale) and extreme concreteness (defined by having a score of greater than four in an approximate five point scale) for individual words. Raw counts for high concreteness is the only metric obtained for bigrams as there are generally fewer tagged bigrams to score in the corpus. Raw word count has also been added.

Using the stepwise algorithm for model selection on linear regression has yielded the following model:

Coefficient	Estimate	Standard Error	p-value
Intercept	1.5834	0.1033	< 2e-16
High Concreteness Word Count ²	0.0191	0.0050	0.0001
Perceived Risk (Listlessness)	-0.0865	0.0268	0.0013
Self-rated Mental Imagery Score (Vivid)	0.0552	0.0273	0.0435
Outside Activity Score	-0.0550	0.0330	0.0961
Word Count	-0.0016	0.0011	0.1534

Table 2: Word-level model

This word-level model has an adjusted R^2 score of 0.0774 which means that the variables selected account only for approximately 7.74% of the variance of Concreteness. This is considerably higher than the baseline model’s adjusted R^2 score of 0.0400, but it is still inadequate for practical applications. It is valuable, however, to note that the contribution of word-level mined variables into the model nearly equals the effect of everything else in the model, which gives us further evidence to explore the text response of the observations using NLP techniques.

5.1.3 One-vs-rest Classification

A one-vs-rest classification approach has been applied to the data. This will also be useful to see if people with low levels of concreteness can be identified by different predictors to people with high levels of concreteness. A higher score in the individual models means that it is more likely that an observation belongs to the model’s designated class.

5.1.3.1 Low Concreteness model The model parameters can be found in Table 3. The low concreteness model has yielded a training accuracy of 0.6758 and a test accuracy of 0.72. This model is the simplest one we’ve encountered which is quite insightful. Both high concreteness word count and perceived risk of listlessness

²Number of words that have concreteness values of three or above in an approximate five point scale.

Coefficient	Estimate	Standard Error	p-value
Intercept	-0.0874	0.2159	0.6856
High Concreteness Word Count	-0.0469	0.0078	1.77e-09
Perceived Risk (Listlessness)	0.1847	0.0739	0.0125

Table 3: Low Concreteness Model

are consistently significant predictors for many of our models here, and the patterns are the same. More occurrences of high concreteness words generally lead to high concreteness scores (hence the negative relationship between high concreteness word counts and the probability of being a low concreteness observation) and a higher perceived risk of listlessness generally leads to low concreteness scores.

Coefficient	Estimate	Standard Error	p-value
Intercept	-1.0859	0.3971	0.0063
High Concreteness Bigram Count	-0.2209	0.1079	0.0407
Distancing Impact	0.0854	0.0512	0.0950
Isolation Days	-0.0145	0.0093	0.1216
Age Group (25-44)	0.5346	0.3739	0.1527
Age Group (45-64)	0.4569	0.3580	0.2019
Age Group (65-79)	0.9387	0.3858	0.0150
Age Group (80+)	1.3443	0.9836	0.1717
Physical Activity Score	0.0144	0.0094	0.1245

Table 4: Medium Concreteness Model

5.1.3.2 Medium Concreteness model The model parameters can be found in Table 4. The medium concreteness model has yielded a training accuracy of 0.5710 and a test accuracy of 0.5086. This model is the worst performer of the one-vs-rest models, sitting just a small bit better than guessing if an observation counts as a medium concreteness observation or not. Luckily, predicting low and high concreteness observations are more important in practical applications.

5.1.3.3 High Concreteness model The model parameters can be found in Table 5. The high concreteness model has yielded a training accuracy of 0.7762 and a test accuracy of 0.7771. From the relatively high accuracy scores, it seems like high concreteness observations are more easily predictable with our given variables. High concreteness word and bigram count appear as significant variables, with the negative estimate on word count working as a balancer to divide responses rich in high concreteness words as compared to responses that are just long. Perceived risk of listlessness is also present, staying consistent in its negative relationship to concreteness.

5.1.3.4 Concreteness Classifier A concreteness classifier that uses the outputs of the three individual models and selects the max has been created. A confusion

Coefficient	Estimate	Standard Error	p-value
Intercept	-1.6622	0.3240	2.88e-07
High Concreteness Word Count	0.0601	0.0157	0.0001
Word Count	-0.0067	0.0034	0.0513
High Concreteness Bigram Count	0.2446	0.1242	0.0488
Perceived Risk (Listlessness)	-0.2000	0.0875	0.0223
Self-rated Mental Imagery Score (Scenes)	0.1394	0.0776	0.0723
Working From Home (No)	-0.6766	0.2724	0.0130
Working From Home (Yes, some work days)	-0.1932	0.3338	0.5626
Working From Home (Yes, all work days)	0.1008	0.2330	0.6652
Coinhabitants (Children)	0.1707	0.1019	0.0939
Distancing Impact	-0.1031	0.0659	0.1176

Table 5: High Concreteness Model

matrix on the model’s predictions on the test dataset can be found on Table 6. Its overall accuracy is 0.4571 which still is low for practical usage. The statistics by class can be found in Table 7.

		True level			Total
		Low	Medium	High	
Predicted level	Low	18	15	3	36
	Medium	34	52	29	115
	High	5	9	10	24
Total		57	76	42	175

Table 6: One-vs-rest Classifier

Statistic	Low	Medium	High
Sensitivity	0.3158	0.6842	0.2381
Specificity	0.8475	0.3636	0.8947
Prevalence	0.3257	0.4343	0.2400
Detection Rate	0.1029	0.2971	0.0571
Detection Prevalence	0.2057	0.6571	0.1371
Balanced Accuracy	0.5816	0.5239	0.5664

Table 7: One-vs-rest Statistics by Class

5.1.4 Decision Tree

A conditional inference tree as specified in Hothorn, Hornik, and Zeileis’s paper [5] has yielded the decision tree in Figure 5. A confusion matrix on the model’s predictions on the test dataset can be found on Table 8. Its overall accuracy is 0.4663 which still is low for practical usage, but slightly better than the one-vs-rest classifier which is impressive considering the tree is minimalistic in its choice of nodes. The statistics by class can be found in Table 9.

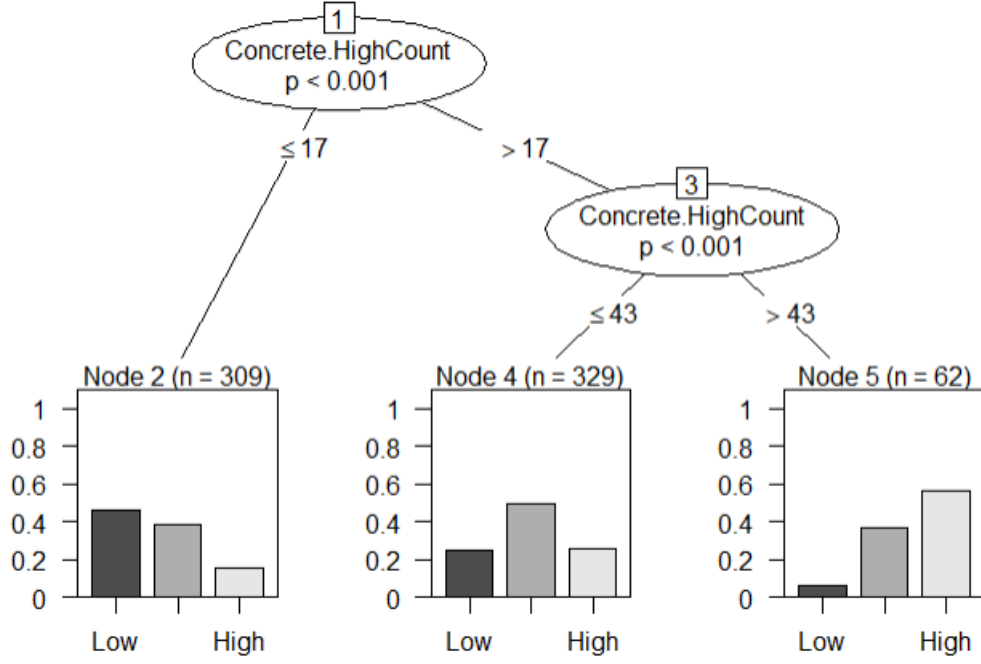


Figure 5: Concreteness Decision Tree

		True level			Total
		Low	Medium	High	
Predicted level	Low	28	30	9	67
	Medium	28	46	25	99
	High	2	1	9	12
Total		58	77	43	178

Table 8: Decision Tree

The importance of the count of high concreteness words have repeatedly shown up in many of our tested models which gives a strong indication that using NLP techniques on our response might yield rich information not usually attainable with other methods.

5.2 Prediction with natural language processing techniques

5.2.1 BERT Pretrained models

The bert-base-cased pretrained model has been selected for this study. The training time has been set to 15 epochs. The first feed forward neural network created has the following layers:

1. Input layer as tokenised document using the BertTokenizer ran through the BERT pretrained model
2. A dropout layer with probability 0.3

Statistic	Low	Medium	High
Sensitivity	0.4828	0.5974	0.2093
Specificity	0.6750	0.4752	0.9778
Prevalence	0.3258	0.4326	0.2416
Detection Rate	0.1573	0.2584	0.0506
Detection Prevalence	0.3764	0.5562	0.0674
Balanced Accuracy	0.5789	0.5363	0.5935

Table 9: Decision Tree Statistics by Class

3. A linear layer diluting the 768 features of the BERT layer into 3 nodes

The second feed forward neural network is similar but adds a step in between that appends the high concreteness word count as an extra feature. An argument can be made for removing the dropout layer given that we have a small dataset and not making use of as much learning as we could from it would not be ideal, however, removing the dropout layer has shown decreased performance..

The first network has achieved a test accuracy of 0.4404, while the second network with the added high concreteness word count variables has achieved a test accuracy of 0.4679.

		True level			Total
		Low	Medium	High	
Predicted level	Low	15	12	4	31
	Medium	17	30	20	67
	High	4	4	3	11
Total		36	46	27	109

Table 10: BERT Pretrained Model

		True level			Total
		Low	Medium	High	
Predicted level	Low	11	7	1	19
	Medium	24	32	18	74
	High	1	7	8	16
Total		36	46	27	109

Table 11: BERT Pretrained Model with High Concreteness Word Count

5.2.2 BiLSTM Classifier

5.2.3 Transformer model with XLNet

5.3 Prediction with classical techniques

5.3.1 SVM Classifier

5.3.2 Random Forest Classifier

5.3.3 XGBoost Classifier

5.4 Summary

References

- [1] A. Alshahrani, M. Ghaffari, K. Amirizirtol, and X. Liu. Identifying optimism and pessimism in twitter messages using xlnet and deep consensus. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.
- [2] Jean Charbonnier and Christian Wartena. Predicting word concreteness and imagery. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 176–187, Gothenburg, Sweden, May 2019. Association for Computational Linguistics.
- [3] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.
- [5] T. Hothorn, K. Hornik, and A. Zeileis. ctree: Conditional inference trees. 2015.
- [6] Julie L. Ji, David J. Kavanagh, Emily A. Holmes, Colin Macleod, and Martina Di Simplicio. Mental imagery in psychiatry: Conceptual & clinical implications. *CNS Spectrums: the international journal of neuropsychiatric medicine*, 24(1):114–126, February 2019.
- [7] Stephen M. Kosslyn, Giorgio Ganis, and William L. Thompson. Neural foundations of imagery. *Nature Reviews Neuroscience*, 2(9):635–642, Sep 2001.
- [8] Igor Mozetič, Miha Grčar, and Jasmina Smailović. Multilingual twitter sentiment classification: The role of human annotators. *PLOS ONE*, 11(5):e0155036, May 2016.
- [9] T. Patel, C.R. Brewin, J. Wheatley, A. Wells, P. Fisher, and S. Myers. Intrusive images and memories in major depression. *Behavior Research and Therapy*, 45(11):2573–2580, Nov 2007.

- [10] Fritz Renner, Fionnuala C. Murphy, Julie L. Ji, Tom Manly, and Emily A. Holmes. Mental imagery as a “motivational amplifier” to promote activities. *Behaviour Research & Therapy*, 114:51–59, March 2019.
- [11] Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, 2019.
- [12] Michael Stewart, Wei Liu, and Rachel Cardell-Oliver. Redcoat: A collaborative annotation tool for hierarchical entity typing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 193–198, Hong Kong, China, November 2019. Association for Computational Linguistics.
- [13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- [14] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.
- [15] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.