RorschIA: NLP and help with coding

Supervised Project : 2023-2024

Rachel ATHERLY, Alberto LORENTE GALE, Aubin MEDJAED, Mina OULHEN
Institut des sciences du Digital, Management & Cognition, Université de Lorraine
Supervised by : **Renaud EVRARD** (MCF-HDR, project leader)
InterPsy
psychology help and other project members :
Mélanie LAURENT (MCF), Antoine FRIGAUX (Psychologist, Associate Researcher)
Côme PIERRE (Psychologist), Eva RAMORINO (student M2 psychology).
+Yassine BENAÏD (computer science)
RorschIA Github Page

1 Abstract

The widely known Rorschach inkblot test has proved difficult to use as a common clinical and institutional tool due to its long scoring process (sometimes taking up to 4 hours). This project focused on automatizing this process using the well known BERT transformers and other natural language processing and classification tools in the hope of significantly reducing the time required for the completion of the test.

Models were split for canonical and grouped labels, with the grouped labels averaging a 9% increase in f1 score. Content models achieved a promising 0.82 to 0.89 on f1 score, but determinants only achieved 0.47 and 0.59 for the canonical and macro label respectively.

2 Introduction

The Rorschach inkblot test, designed by psychiatrist and psychoanalyst Hermann Rorschach in 1921, is a projective psychological test where the subject is presented with ten symmetrical inkblot cards, some of which sporting colors. For each card, the subject must respond and explain what they see in the blots, on what area. Cards may be rotated.

All responses are recorded and then interpreted by following precise guidelines. They are categorized by their content (what the subject sees in the blots), the location (where it sees this content) and then the determinants (which qualifies the content and location regarding the cards). Additional qualitative elements can be added to denote the banality of the responses, or any reaction of the subject.

Once those four qualifiers have been determined for all responses, the psychoanalyst or psychiatrist can group them and calculate a “psychogram”, a sum of various indices and compare them with normative data. This will help the practitioner come up with a qualitative assessment of the subject personality.

As of 2024, no major research has been conducted with the goal of combining machine learning models and the Rorschach test. (Camati et al., 2021) worked with another projective test in a similar project, achieving promising results. The interest was mostly coming from NLP team, who applied the test to multiple models (namely Deep dream model). Additionally, no Rorschach dataset has yet to be built to facilitate natural language processing usage.

Rorschach coding format lends itself to a multi-labeled classification task where each response would be associated with four labels (determinant, content, location and quality). We will not focus on the location since, typically, it is encoded by a set of special characters easily parseable with regex. We will also ignore the additional qualitative elements as they lack sufficient data, have a more semantic focus, and are not needed for the psychogram.

To do so, we fine-tuned the well known BERT model to codify contents and determinants. The uneven nature of the data and high number of labels is not well suited for a standard classification task, so we also tried to regroup the labels, prompting the creation of four models, two for each category.

3 State of the art

3.1 BERT

To do so, we used a widely known transformer model named BERT (Devlin et al., 2019). This deep learning architecture follows a bidirectional encoding structure that leverages the attention mechanism to capture both the left and right context of the text. This differs from previous models by abandoning recurrence and convolution usage.

BERT (Bidirectional Encoder Representations from Transformers. (Devlin et al., 2019)) differs from other transformer models as it is pre-trained to condition itself from left and right. This allows the fine-tuning of the model with only one output layer.

In BERT, the definition of the number of layers, attention heads and learning rate are hyperparameters defined prior to training process. Base BERT uses 12 separate attention mechanisms at each of its twelve layers, letting each token from the initial sequence focus on other tokens.

With 110 million parameters and state-of-the-art results, BERT is one of the best compromises for a potent yet light-enough large language model.

3.2 Neural networks

Neural networks are a method of information processing where the data is spread throughout 'neurons' that are formatted in layers that mimic the general computation structure of a human brain. The use of neural networks for coding and understanding natural language in therapeutic settings is not yet common, as seen by the lack of studies that relate to the realm of this project. However it is a technique that is seeing more exploration through various studies, such as the recursive neural networks used to encode patient and therapist conversations during motivational interviewing sessions (Tanana et al., 2015).

Neural networks are increasingly popular tools for natural language processing. Multiple NLP specific frameworks have been developed. Pretrained models such as GPT and BERT have wide popularity and a significant range of uses (Zhou et al., 2020).

Despite their complex structures and detailed data modeling capabilities, a neural network is only as good as the data that it is given. The quantity and quality change the way the network functions and incorrect handling can create problems and biases within the results. Overfitting is a common issue that occurs when the model becomes too specific to the details of the training data. This can be seen by low training error but high test error. The result is a model that initially seems very well trained but is unsuitable for actual use on any different data. On the opposite end, underfitting comes from a model that fails to learn any trends or patterns within the dataset, also rendering it useless outside of the training set (Jabbar and Khan, 2015). It is critical to carefully select the data to find the sweet spot where it is not too simple but also does not contain too much excessive noise. From there, metrics and graphs can be analyzed to identify the trends if one of these issues does occur.

3.3 Loss Functions

Loss functions are used to calculate the difference between predicted and target outcomes. It is essentially a mechanism with which to penalize predictions that are off the mark. Many different loss functions are used in model training such as the Cross Entropy and Binary Cross Entropy functions, the latter of which proves ideal for data such as ours which is multi-label classification (Mao et al., 2023).

4 Objectives

The main objectives of the realization of the project were the following:

- To verify the lack of literature connecting the Rorschach test and natural language processing.
- To collect information about popular techniques applied to other research studies concerning the fields of psychology and machine learning.
- To lay the theoretical background to conceptualize the Rorschach test through an NLP and machine learning lens.
- To define the challenges inherent to solving the coding of the test and possible strategies to overcome those challenges.
- To apply those trends observed during the literature review to develop machine learning models that classify the determinants and contents of the test.
- To conduct the model development phase of the research in an environmentally friendly

manner, especially the more demanding tasks like grid search.

- To assess the applicability of those machine learning models and evaluate their strengths and weaknesses.

5 Data

One of the first challenges that the project presented was the sparsity of the data. Encoded Rorschach protocols are not a type of data commonly shared. Additionally, the variations in encoding processes, from one country to another, or even from one practitioner to another, make it complicated to have one sufficiently large dataset of standardized protocols. We considered two datasets of transcribed and encoded protocols.

5.1 First dataset

The first one, the COVID-19 Rorschach test dataset by Virtual Psychology, initially contains over 500,000 coded responses to the Rorschach test, originating from several countries worldwide and collected from 2017-01-01 to 2020-09-15. Although the sources for the data are not explicitly stated, the purpose of the data was to study how the frequency of certain labels changed after the COVID pandemic.

One response corresponds to the information given by the patient about one specific element of an inkblot.

The text value for the answers of this first dataset are generally reported as short noun phrases, with rather rare further explanation.

Due to the high number of countries and practitioners this dataset assembles, the quality of the reported answers was inconsistent, with many answers having up to four different annotations. However, the qualification level of the psychologist was indicated in the features. After an overall selection of the featured most qualitative answers as well as a removal of duplicates and features superfluous to our task, the dataset we obtained included 747 responses. The cleaning process then mainly targeted the content and determinant labels.

To achieve uniformity over all our data, the labels needed a translation, since the labels in the original dataset differed from the ones defined by the project's psychology team.

The final 25 content and 12 determinant labels were reorganized as one hot encoded features with the associated text value of each answer. After a variety of attempts at developing models based on this dataset, this dataset was proved to be not usable.

5.2 Second dataset

The second dataset was provided by the psychology department and includes ten Rorschach protocols, which equates to a number of 380 responses.

The French protocols were translated into English with the DeepL translator. The text value's answer length was generally much higher than in the first dataset, given that the reported text consisted of entire sentences of transcribed speech and responses were manually segmented.

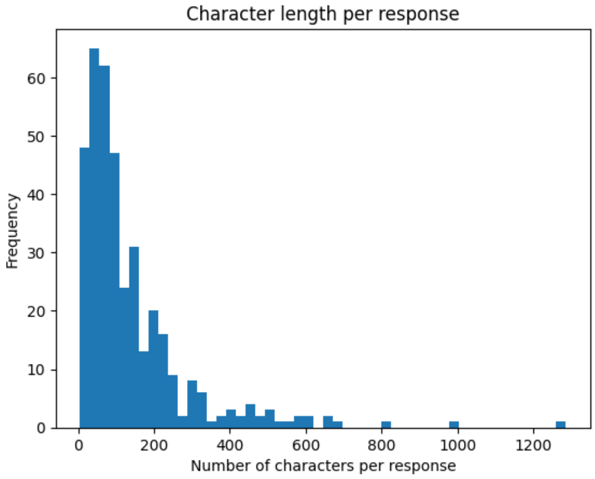


Figure 1: Character Length Distribution

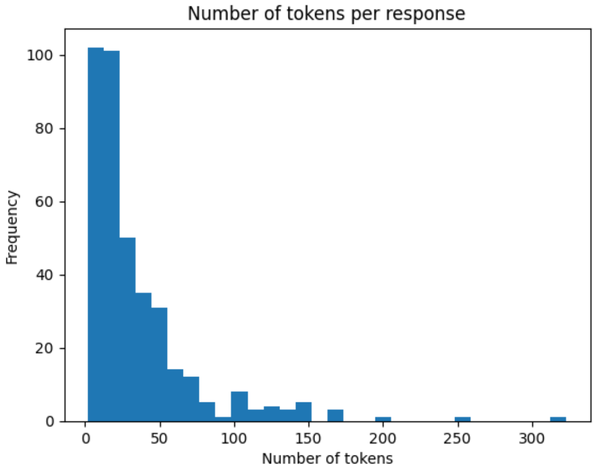


Figure 2: Token Length Distribution

The data then underwent a similar process of standardization and one hot encoding of the content and determinant labels.

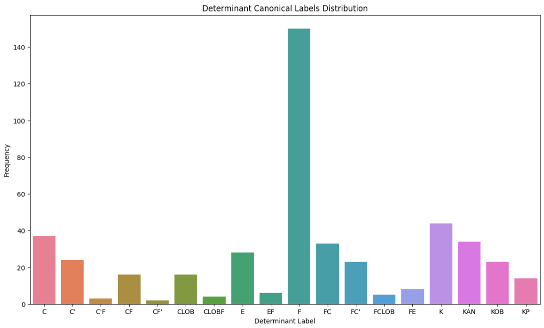


Figure 3: Determinant Canonical Labels Distribution

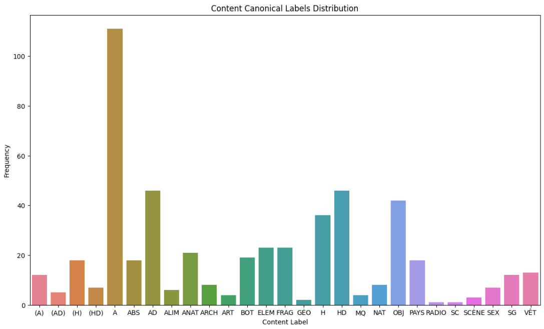


Figure 4: Content Canonical Labels Distribution

As shown in the graphs above, there is a definite unevenness in the original classes. Hence another comparative approach was designed, regrouping content and determinant labels into broader categories. The question then was how to devise these macro-labels for optimum representativeness, which was realized as effectively as it could be considering some of the labels did not have positive classes. That is, there was no example. As such this renders the evaluation metrics less representative of the performance of the model.

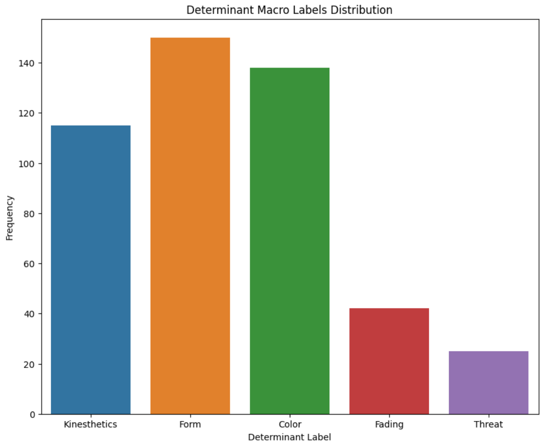


Figure 5: Determinant Macro Labels Distribution

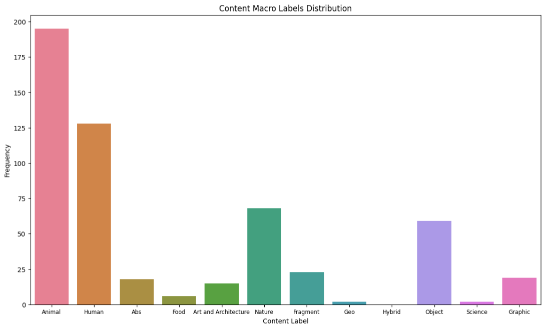


Figure 6: Content Macro Labels Distribution

Of the 380 responses of this dataset, 324 were used for training the model, 36 for validating and 20 for testing.

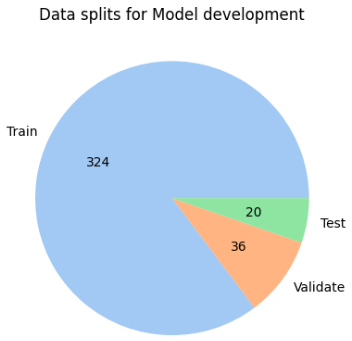


Figure 7: Data splits for Model Development

As we can observe in subsection C.1, which contains the distribution of labels for each split for each model, we had to contend with the fact that we do not have enough examples to include in every data split. Given the situation, we chose to prioritize the quality of the data split in the training data, firstly, and the validation data, secondly. Again, we have to reiterate that because of this setback the testing of the models may not be fully representative of their performance

6 Methodology

The literature review previously conducted proved the novelty of the intersection between the Rorschach test and natural language processing methods. In the wake of this situation, we thought it beneficial to approach the task in a hierarchical way, attempting simpler and more explainable solutions first before moving on to more complex, back-box strategies.

Given the scarce nature of Rorschach related data and the heavy class imbalances present in it, it initially seemed plausible that training models with the original labels would not produce a satisfying result. In an attempt to remedy this would-be set back, we developed two sets of models parallelly: one set whose determinant and content labels are the canonical ones provided originally and a second set of models whose labels would be reduced by grouping. The grouping was carried out by semantic proximity while simultaneously aiming to balance the frequency of the classes.

Once these two data approaches were defined, the next step was deciding which machine learning techniques to apply. After the initial failure of the traditional algorithms implemented, as explained in the Traditional Machine Learning Models section and the impossibility to acquire enough data to train a neural network from scratch, we pivoted to leveraging fine-tuning techniques. Fine-tuning pre-trained, task-agnostic BERT models, more specifically, the bert-base-uncase (Devlin et al., 2018) model, remedied the shortage of domain-specific data while achieving much better results than the first batch of models.

After deciding on this model architecture, we conducted a grid search to find the optimal hyper-parameters for the final models. Because of the computing-devouring nature of such a process, the search was organized in three stages to minimize its environmental impact. An initial exploration was made to get familiar with the way the Bert model worked with our data. After this, a second search was conducted where the number of hyper-parameter options was significantly cut down before the final stage where the best two to three models for each data representation were trained longer. With the grid search concluded, a model was picked for each combination of data approach and label type. This gave us four ideal models, one for the individual determinant labels, one for the individual content labels, one for the grouped determinant labels and one for the grouped content labels. These models were compared for a final evaluation.

7 Models Development

7.1 Traditional Machine Learning Models

Simpler models were developed prior to the transformer models. The small size of the data allowed for the exploration of a wide variety of algorithms without having to worry about computing consumption. These first experiments consisted of the following machine learning algorithms: logistic regression, support vector machine, stochastic gradient descent, k-nearest neighbors, random forest and gradient boosting classifier.

The initial models followed the two data approaches previously defined. As far as the text vectorization techniques implemented, one set of models used tf-idf after pre-processing the text removing stopwords and another set of models used the centroid of the array resulting from the tokenization of the text using the all-MiniLM-L6-v2 (Reimers and Gurevych, 2019) language model.

- Let $label_{determinants}$,$label_{contents} \in L$ be the set of labels type that are to be classified.
- Let d_{macro},$d_{individual} \in D$ be the set of possible data approaches.
- Let a_{logreg},a_{svc},a_{sgd},a_{knn},a_{rf},$a_{gdc} \in A$ be the set of algorithms used for the model development.
- Let $vect_{tf-idf}$,$vect_{token} \in V$ be the set of text vectorization techniques applied.

Then the set of models developed is defined as $M = L \times D \times A \times V$.

In general, the results for the models were underwhelming with a few exceptions. The analysis of the results will be split by the vectorization technique used for each set of models, the main focus will be observing the trends by data approach rather than centering on individual models and their potential use and the f1 score will be the main metric used for the evaluation.

7.1.1 Tf-idf Models

The average f1 score for the tf-idf models was 0.28 with a standard deviation of 0.18. 75% of the models performed under 0.41. However, a significant division can be observed when the performance is grouped by the type of data approach, as can be observed in table 1.

Table 1: Mean F1 score per data approach

Data approach	F1 score
Individual labels	0.15
Macro labels	0.41

Moreover, this distinction can be verified examining the f1 score distribution for each type of data approach[8]. The distribution for the performance of the individual labels models is much more right-skewed while the one for the macro labels models looks much more normal.

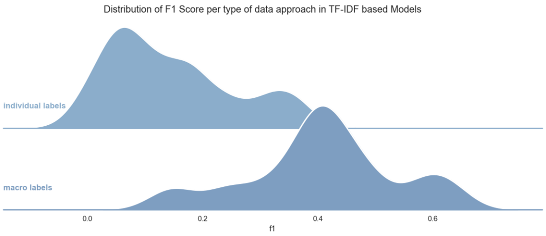


Figure 8

Zooming in to the performance by data approach plus the label the models classify[2], the trend continues. The macro classifiers clearly outperform their individual labels counterparts by a wide margin.

Table 2: Mean F1 score per data approach and label

Data approach and label	F1 score
Individual content labels	0.18
Individual determinant labels	0.11
Macro content labels	0.46
Macro determinant labels	0.35

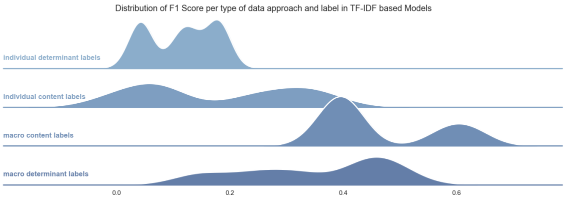


Figure 9

7.1.2 all-MiniLM-L6-v2 Models

Shifting the focus towards the Sentence Transformers all-MiniLM-L6-v2 (Reimers and Gurevych, 2019) models, the trends observed in the tf-idf models continue, albeit in a weaker way[3]. The mean f1 score for these models is 0.15 with a standard deviation of 0.16. Overall, the performance of these models was worse than the tf-idf ones regardless of data approach and label to classify.

Table 3: Mean F1 score per data approach

Data approach	F1 score
individual labels	0.08
macro labels	0.21

The distribution for the models looks rather different; the f1 score is much more polarized and is much lower.

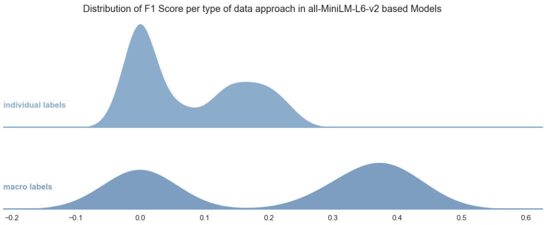


Figure 10

At a more granular level the trend still continues and macro label representations continue to outperform their counterparts by a wide margin.

Table 4: Mean F1 score per data approach and label

Data approach and label	F1 score
Individual content labels	0.08
Individual determinant labels	0.08
Macro content labels	0.23
Macro determinant labels	0.20

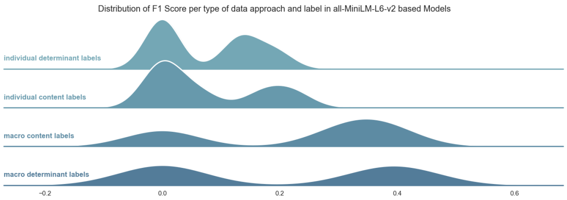


Figure 11

7.1.3 Conclusion

Even though the performance is underwhelming, it is worthy to notice that there is an increase when grouping the individual labels into the defined macros. The expectation going forward would be that this trend translates across different machine learning architectures.

7.2 Transformers Models Development

The machine learning models are based on a standard configuration of bert-base-uncased (Devlin et al., 2018) for multi-label classification with a maximum token length of 128 tokens, a cross-entropy loss function, an Adam optimizer, a dropout layer with a probability of 0.3 and a sigmoid activation function with a threshold of 0.5 as well as an argmax algorithm serving as a safety in the rare case that the model does not output a high enough probability for any class. This configuration is common to all models irrespective of the data modelling and to achieve stronger results, the models' hyper-parameters will be fine-tuned separately for each data approach.

7.2.1 Hyper-parameter Tuning

The hyperparameters for the models were fine-tuned via a grid search. This allowed us to find the best combinations of learning rate, batch size, and training epochs for our models.

Initially, we began our parameter tuning limiting ourselves to a single data representation, utilizing a larger range of batch sizes with more epochs. The time requirement for the processing was high and the benefits were not significant, so we pivoted to a more limited set of parameters for the second grid search, where we started looking for specific hyper parameters for every data representation.

After our second grid search, we employed a number of metrics to evaluate which were our best model candidates. The f1 scores and accuracy were analyzed along with the training loss, validation loss, and the hamming score. In short, hamming loss is the proportion of incorrectly classified classes. It is different from other scoring metrics in that it not only considers whether the classes predicted as positive are correct or not but also it rewards the number of negative classes that were predicted as negative. In this multi-label setting where the number of classes is so high, hamming loss throws light at the fact that a model is not outputting a high proportion of classes. This score was used together with the f1 score. Given the small amount of examples for certain classes in the the dataset and the low prediction values that we tended to work with, the hamming and f1 scores were not always reliable on their own and could paint a biased picture of the data. Together, however, they gave a more complete view of the performance. Loss was calculated with the pytorch binary cross entropy with logits loss function (BCEWithLogitsLoss). All of these metrics were compared in data frames and the best preforming models were then graphed for better visualization.

After both our grid search tests were compete, we took the best candidates, and we trained them for more epochs to see how far we could push them before they would start overfitting. These top models for each category were graphed and analyzed. Figures 12 and 13 depict the best performing models for the individual contents and individual determinants data representation, respectively.

The same metrics were selected for both variations of the individual models (batch size 8 and learning rate 5-5e).

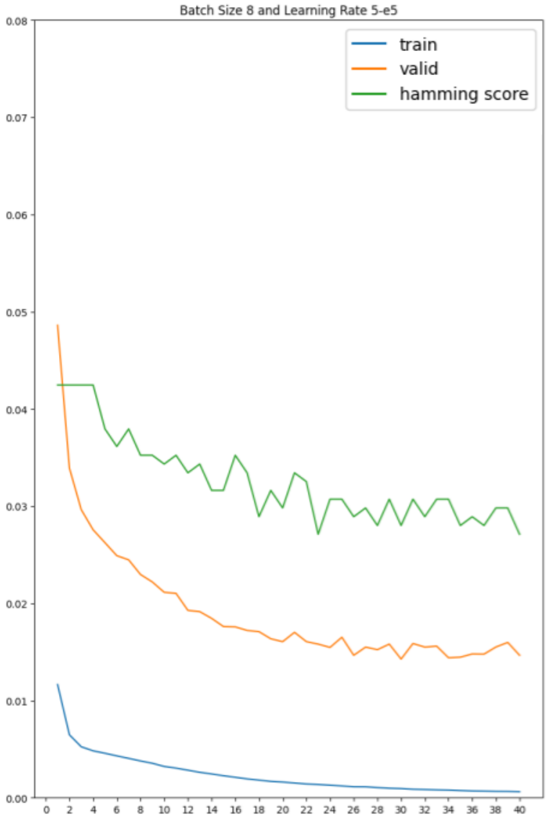


Figure 12: Individual Content

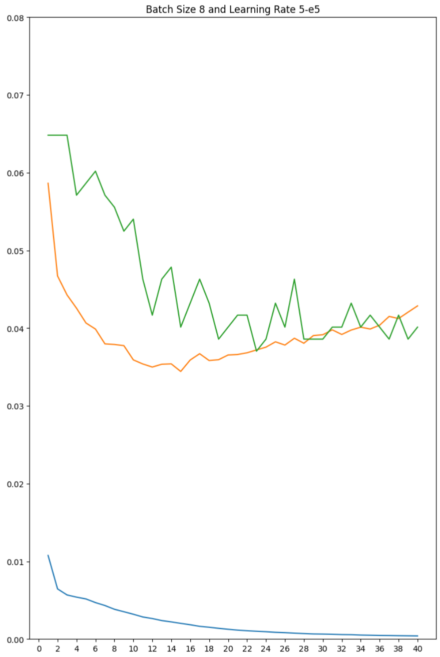


Figure 13: Individual Determinants

Following this, we conducted a similar analysis on the macro determinant models and macro content models seen in Figures 14 and 15. Once again, the best performing candidates fell into batch size 8, although there were better results from slightly lower learning rates on the macro models than the individual ones, with the best running on a learning rate of 2-e5 versus the 5-e5 of the individual models.

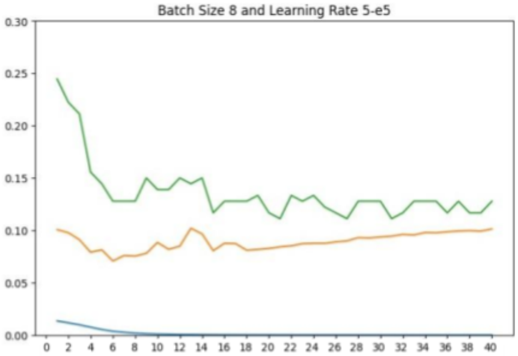
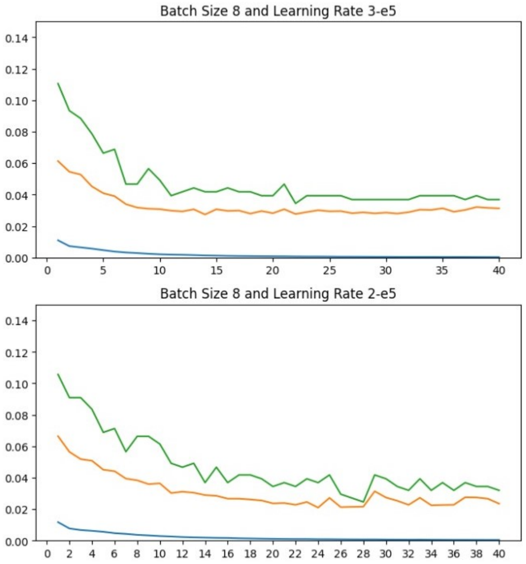

Batch Size 8 and Learning Rate 5-e5

Figure 14: Macro Determinants


Batch Size 8 and Learning Rate 3-e5

Batch Size 8 and Learning Rate 2-e5

Figure 15: Macro Content

7.2.2 Final Transformer Models

After the final grid search, one model was selected per type of label representation and the optimal epochs for each model was determined by examining their learning curve. As Table 5 indicates, all the final models had a batch size of 8, their learning rates varied between $2e-5$ to $5e-5$ and the epochs they were trained for ranged from 15 to 32.

Data Model	Learning rate	Batch size	N. epochs
Canon. Determinants	3e-5	8	15
Canon. Contents	3e-5	8	32
Macro Contents	2e-5	8	28
Macro Determinants	5e-5	8	8

Table 5: Final Models

8 Evaluation

8.1 General Results

Overall, the performance of the models is markedly different depending on the aspect they classify. This fact was to be expected, as the nature of the aspects is vastly different; the Determinants are much more ambiguous while the Contents latch onto more concrete concepts.

One tendency that can be observed is that the performance for the models where the labels were grouped is consistently better across most metrics, although not by a big margin. In terms of f1 score, the content and determinants macro representations were around 7% and 12% higher than their canonical counterparts.

Model	F1	Accuracy	Precision	Recall	Hamming loss
Canon. Content	0.82	0.74	0.80	0.83	0.01
Canon. Determinant	0.47	0.40	0.50	0.43	0.06
Macro Content	0.89	0.86	0.91	0.87	0.021
Macro Determinant	0.59	0.44	0.62	0.57	0.20

Table 6: Performance of the Models in the Test Split

Among all the models, both the macro and canonical contents models stand out with an impressive performance while the determinant models' performance is mild. The fact that the models with individual forms of representation performed at the level they did is worthy of mention too.

8.2 Trade-off between Specificity and Performance

Table 7 recalls the label grouping performed for the macro determinant groups, condensing 18 possible classes into 5, while Table 8 recalls the transformations for the content labels, by which 27 classes were grouped as 11.

Macro label	Canonical labels
Kinesthetics	K, KAN, KOB, KP
Form	F
Color	FC, CF, C, FC’, CF’, C’F, F’C, C’
Fading	FE, EF, E
Threat	FCLOB, CLOBF, CLOB

Table 7: Determinant Labels Equivalence

Macro label	Canonical labels
Animal	(A), (AD), A, AD, ANAT
Human	(H), (HD), ANAT, H, HD
Abs	ABS, SYMB
Food	ALIM
Art and Architecture	ARCH, ART, SCÈNE
Nature	BOT, ELEM, PAYS, NAT
Fragment	FRAG
Geo	GÉO
Hybrid	H/A, H/AD
Object	MQ, OBJ, VÊT
Science	RADIO, SC
Graphic	SEX, SG

Table 8: Content Labels Equivalence

Taking into account the significant amount of labels deduced for the macro representations, $13/18$ and $16/27$ respectively, and the improvement in class balance that it entailed, the faint improvement in performance falls short of expectations.

Having said that, the lack of training examples for certain classes lingers over all of the models and puts a metaphorical asterisk next to the performance metrics.

8.3 Carbon Emissions and Energy Consumption

The total estimated time for training and tuning the transformer based models was around 12 hours of Google Colab’s T4 GPU. Taking into account the specifications of 16 GB of memory and inputting our geolocation, total amount of memory available and runtime into Green Algorithms Calculator (Lannelongue et al., 2021), our energy consumption comes to a total of 1.01 kWh and our carbon footprint reaches 51.88g CO2e. According to Green Algorithms’ calculation, that consumption would translate into driving around 0.3km by car and would be just about enough to take us from the IDMC to Porte Désilles.

9 Future Work

A common theme discussed throughout this research is that of data quantity and quality. In order to produce more robust and reliable models, regardless of the architecture chosen to develop them, a corpus building project needs to be carried out for more than just the collection of anonymized Rorschach protocols. Additionally, it must focus on the development of a consistent annotation guideline to handle the ambiguity and subjectivity of the test. Given the technical nature of the coding process, it is crucial that trained psychologists with Rorschach experience are at the center of this corpus building process.

Regarding potential continuation of this research, a logical next step could be the potential merging of the two approaches by first assigning a macro-category, then returning a canonical label for each macro-categories detected, based on output. This would leverage the higher reliability of the grouped-labels approach without the need to sacrifice the specificity of the second approach.

While this research handled the Rorschach coding as a multi-label classification problem, other strategies that fulfill the objectives, coding the test swiftly and reliably, should be acknowledged as well. Just a few days before the writing of this paper,OpenAI presented its latest flagship model, GPT-4o, leaving the general public in awe over the smooth human-computer interaction experience its demo provided. There is also a rising trend of retrieval augmented generation (Lewis et al., 2021), a framework to incorporate one’s personal documents into LLMs. With this, it is no longer hard to imagine an assistant-like agent that collaborates with a domain expert (Yuhan et al., 2023) or provides suggestions based on ground truth, minimizing possible hallucinations.

All in all, whichever direction the relationship between the Rorschach and NLP may take, the continued collaboration between psychology and NLP practitioners will be crucial to the development of satisfying solutions.

10 Conclusion

To recapitulate, the coding of the Rorschach test was conceptualized as a multi-label classification problem where two sets of labels, one for the Determinants and one for the Contents, had to be predicted. To handle the high number of labels for each aspect and the scarcity of data, there was an aim to explore the grouping of the individual canonical labels for each aspect into bigger, macro categories where the grouping criteria would be semantic similarity. Although this approach partly relieved the class imbalances, it did come with a trade-off between better model performance and the use of more precise labels that are already familiar to the psychology domain.

The potential of this approach was verified by initial experiments with traditional machine learning algorithms where the macro representation of the labels clearly outperformed the canonical one.

Given that the overall performance of these initial models was weaker than desired, we moved on to developing transformer-based models that would take advantage of Bert's fine-tuning power. After a thorough grid search phase to select the optimal hyper-parameters, one model was selected for each combination of aspects (Determinant or Content) and label grouping (individual canonical labels or the proposed macro labels).

Overall, the models based on macro labels averaged a performance 9% higher in terms of f1 score and the quality of both Content models has to be highlighted. One hypothesis that could explain why the Determinant models performed below expectations is difference in abstractness. While the Content aspect measures more concrete concepts. The Determinant is a more flexible notion, potentially leading to conflicting annotations in the training data which are difficult to pinpoint if one is not a trained psychologist. Moreover, the uneven class distribution of labels in the training, validating and testing datasets, highlights the need for a data collection and validation effort. A final point to consider when comparing the models is the trade-off between using the canonical or macro labels and the increase in model performance that the macro labels can entail. This trade-off would be valuable in cases where the difference in performance is very significant. However, this margin is lower than expected. This suggests that for a potential practical use of the models, they would be better suited for use as an aid to assist a professional, rather than blindly relying on the raw predictions output by the models.

References

Ricardo Stegh Camati, Alessandro Antonio Scaduto, and Fabrício Enembreck. 2021. Using the projective thematic apperception test for automatic personality recognition in texts. *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 78–85.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

H Jabbar and Rafiqul Zaman Khan. 2015. Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). *Computer Science, Communication and Instrumentation Devices*, 70(10.3850):978–981.

Loïc Lannelongue, Jason Grealey, and Michael Inouye. 2021. Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8(12):2100707.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2023. Cross-entropy loss functions: Theoretical analysis and applications. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 23803–23828. PMLR.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Michael Tanana, Kevin Hallgren, Zac Imel, David Atkins, Padhraic Smyth, and Vivek Srikumar. 2015. Recursive neural networks for coding therapist and patient behavior in motivational interviewing. pages 71–79.

Xi-Zhu Wu and Zhi-Hua Zhou. 2017. A unified view of multi-label performance measures.

Liu Yuhan, Chen Xiuying, and Yan Rui. 2023. Unleashing the power of large models: Exploring human-machine conversations. In *Proceedings of the 22nd Chinese National Conference on Computational Linguistics (Volume 2: Frontier Forum)*, pages 16–29, Harbin, China. Chinese Information Processing Society of China.

Ming Zhou, Nan Duan, Shujie Liu, and Heung-Yeung Shum. 2020. Progress in neural nlp: Modeling, learning, and reasoning. *Engineering*, 6(3):275–290.

A Considerations

A.1 Considerations about Hamming Loss

Hamming loss is a widely implemented metric used to assess multi-label classification. It is defined as the fraction of misclassified labels (Wu and Zhou, 2017). Moreover, it is crucial to note that Hamming Loss evaluates the individual class predictions for each data point, not solely whether the positively predicted label matches the actual target label.

Although it can be a fairly informative metric, there are certain considerations that must be kept in mind for the case of our research.

Tables 9 and 10 show the prediction two models. Table 9 displays the individual label representation and 10 shows the macro label representation, output for a single data point. Both tables contain the names of the classes present in the training data in the Class column and the actual target of the data point in the Positive/Negative column. The Model Probability column contains the probabilities output by the model for said data point in each class.

Class	Positive/Negative	Model Probability
C	1	0.166
C'	0	0.006
C'F	0	0.003
CF	0	0.004
CF'	0	0.003
CLOB	0	0.010
CLOBF	0	0.002
E	0	0.013
EF	0	0.003
F	0	0.007
FC	0	0.005
FC'	0	0.005
FCLOB	0	0.004
FE	0	0.002
K	1	0.947
KAN	0	0.018
KOB	0	0.003
KP	0	0.009

Table 9: Individual Determinants Model Probability Output

Applying a simple sigmoid activation function with a standard threshold of 0.5 to each model would output $1/2$ correct predictions for the positive class in the first case and $2/2$ correct predictions for the positive class for the second. However, calculating the Hamming Loss, $1/18$ for the first case and $0/5$ for the second, gives the impression that the performance of both models is much closer than it actually is. The number of positive classes always ranges from 1 to 5 in the case of Determinant classification (regardless of the data representation the training data is modelled after). Given this, the Hamming Loss score will be much more favorable to data approaches with more classes just by virtue of having a higher denominator in the score calculation.

Class	Positive/Negative	Probability
Color	1	0.6453
Threat	0	0.033
Fading	0	0.182
Form	0	0.0001
Kinesthetics	1	0.622

Table 10: Grouped Determinants Model Probability Output

In a practical sense, this means that the Hamming loss score should not be used for comparison between models that utilize different modelling of the classes to predict. The hamming score will tend to favor whichever classifier modelled with more classes. Regarding the present research, Hamming Loss will be applicable when comparing grid search results.

B Content and Determinant labels Definitions

Definition	Label
a movement, action, attitude or intention is attributed to human content	K
animal to which a movement is assigned	kan
movement is attributed to an object or element	kob
adequate form to the stimulus, “good form”	F+
inappropriate form to the stimulus, ”bad form”	F-
Vague form : insufficient discernment on the part of the subject	F+/-
color is integrated into a dominantly shaped response (achromatic black-/white)	FC (FC’)
color takes precedence over formalization, leaving the shape imprecise (achromatic black/white)	CF (C’F)
color alone is decisive (achromatic black/white)	C (C’)
the shading is integrated into a dominant shape	FE
the fading is preponderant on the shape which is then indeterminate	EF
fading is the only determining factor	F
feeling of threat, anxiety is integrated into a dominant shape	FClob
feeling of threat, anxiety is preponderant on the shape which is then indeterminate	ClobF
feeling of threat, anxiety is the only determining factor	Clob

Table 11: Determinant definitions

Definition	Label
Fictional whole animal	(A)
Fictional partial animal	(Ad)
Fictional or caricature of whole human	(H)
Fictional partial human	(Hd)
Realistic whole animal	A
Abstract concept	Abs
Realistic partial animal	Ad
Food or edible elements	Alim
Anatomical or internal part of bdoy	Anat
Architectural forms	Arch
Tree, flower, plants	Bot
Fire, water, wind or earth	Elem
Fragmented elements	Frag
Geographical forms and maps	Géo
Realistic whole human	H
Human merged with animal	H/A
Human with animal aspects	H/Ad
Realistic partial human	Hd
Mask or prosthesis modifying the face	Mq
Any inanimate, manufactured object	Obj
Reference to scene or countryside	Pays
Radiographic view of internal organs	Radio
Reference to Science and Techniques	Sc
Sexual organs or activity	Sex
Blood	Sg
Symbols as letters, numbers, emblems	Symb
Anything chothing or garments	Vêt

Table 12: Content definitions

C Figures

C.1 Class Distributions per data split

C.1.1 Content Canonical Labels Distributions

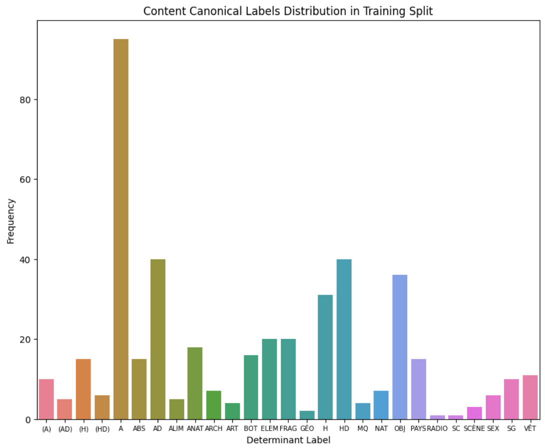


Figure 16: Content Canonical Labels Distribution in Training Split

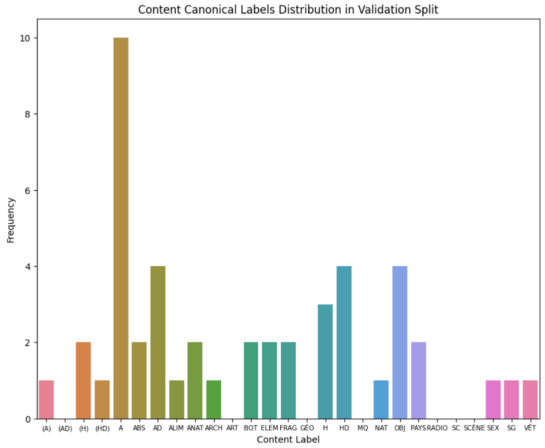


Figure 17: Content Canonical Labels Distribution in Validation Split

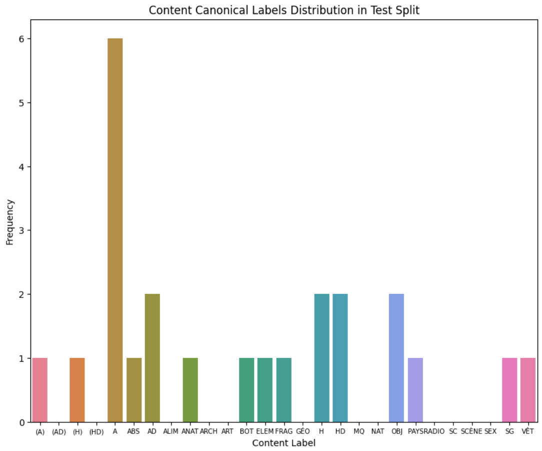


Figure 18: Content Canonical Labels Distribution in Test Split

C.1.2 Determinant Canonical Labels Distributions

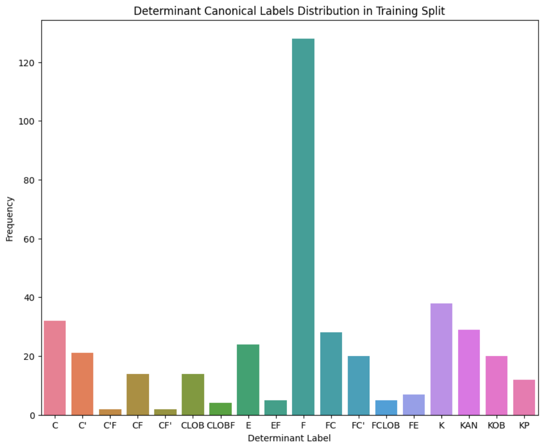


Figure 19: Determinant Canonical Labels Distribution in Training Split

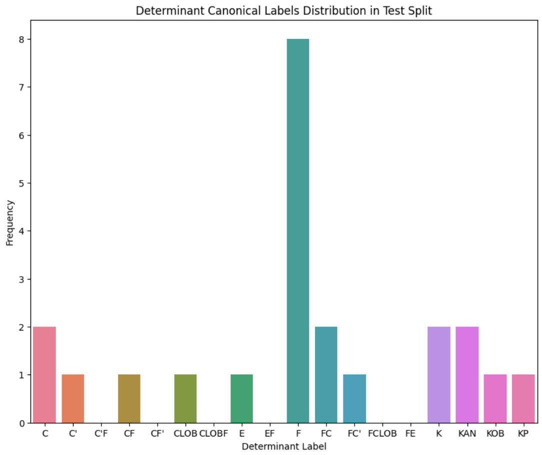


Figure 20: Determinant Canoncical Labels Distribution in Test Split

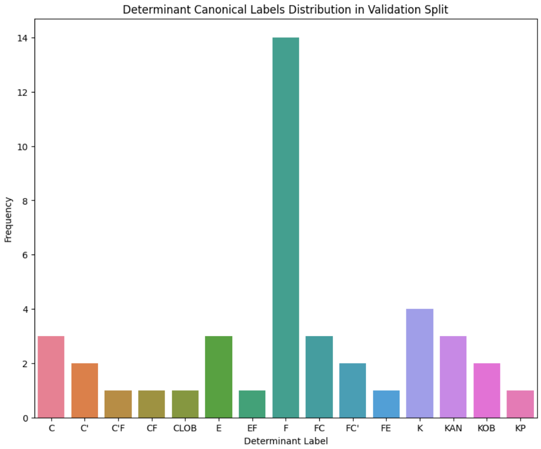


Figure 21: Determinant Canoncical Labels Distribution in Validation Split

C.1.3 Content Macro Labels Distributions

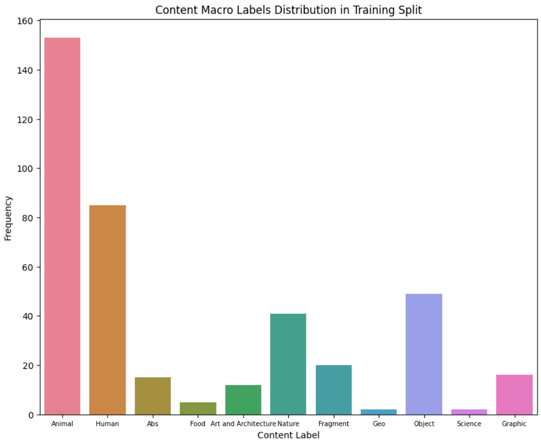


Figure 22: Content Macro Labels Distribution in Training Split

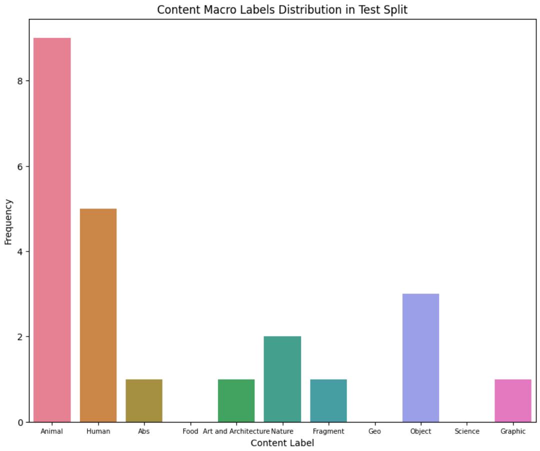


Figure 23: Content Macro Labels Distribution in Test Split

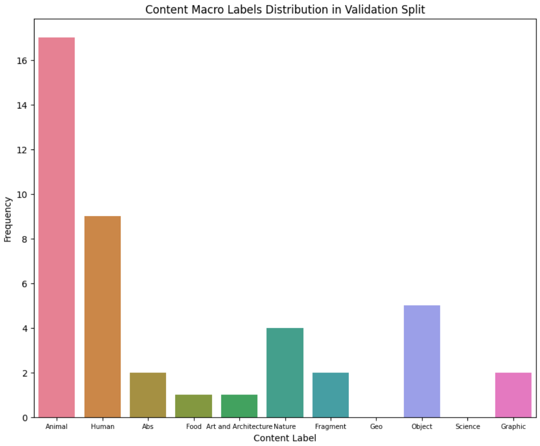


Figure 24: Content Macro Labels Distribution in Validation Split

C.1.4 Determinant Macro Labels Distributions

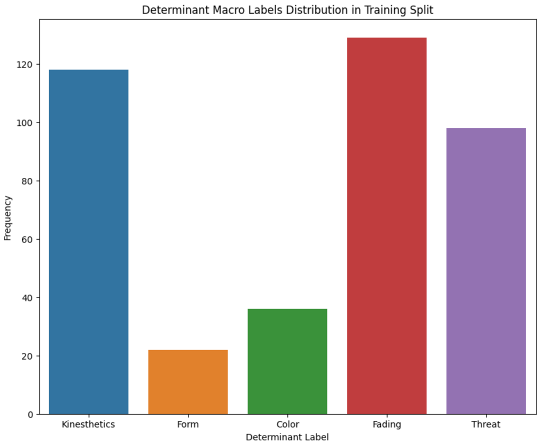


Figure 25: Determinant Macro Labels Distribution in Training Split

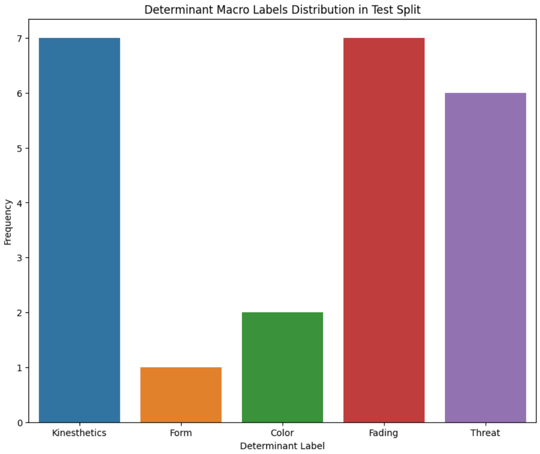


Figure 26: Determinant Macro Labels Distribution in Test Split

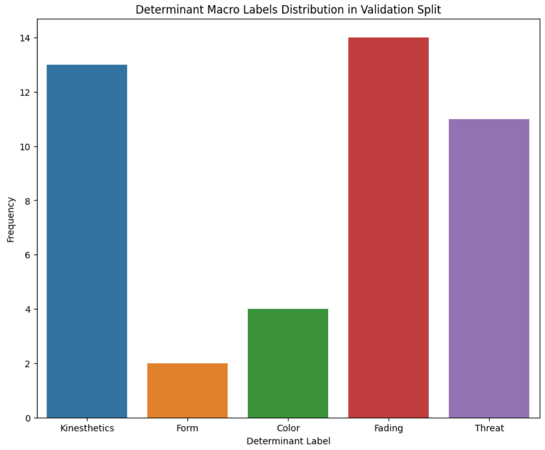


Figure 27: Determinant Macro Labels Distribution in Validation Split

C.2 Grid Search Graphs

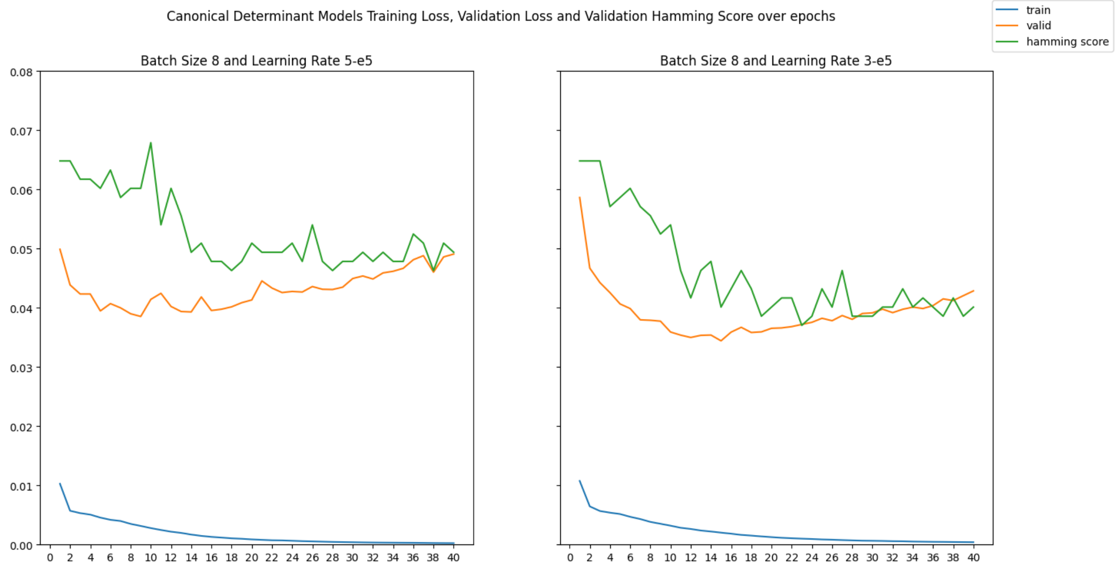


Figure 28

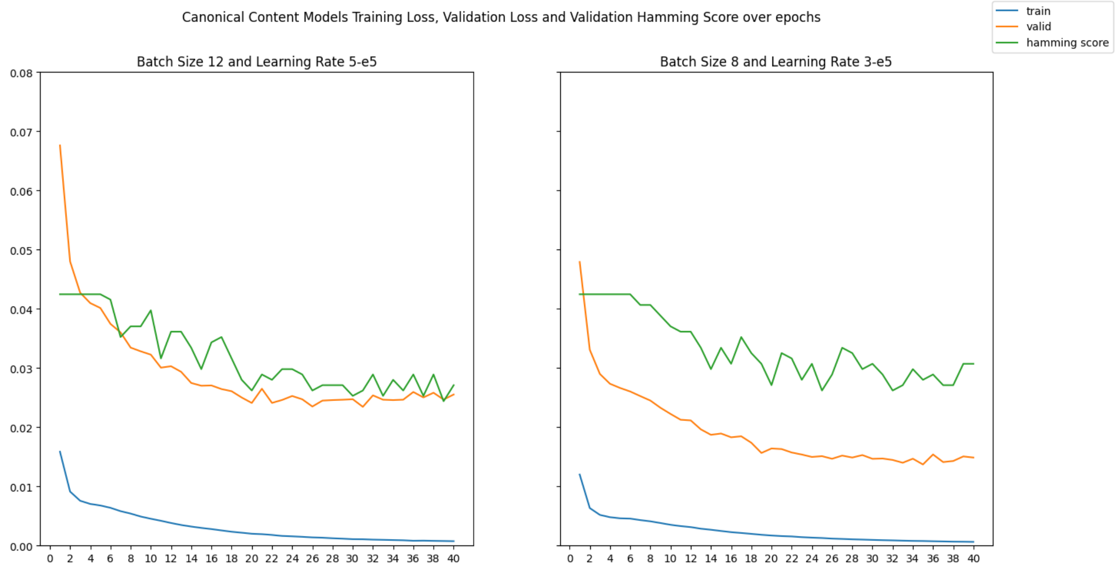


Figure 29

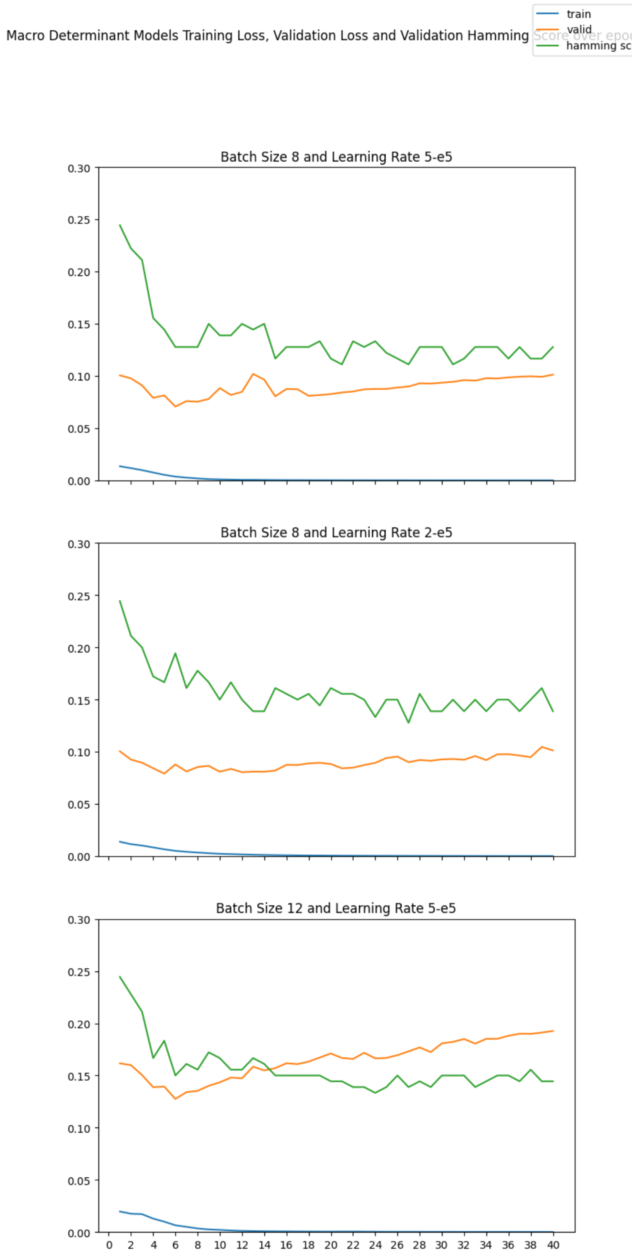


Figure 30

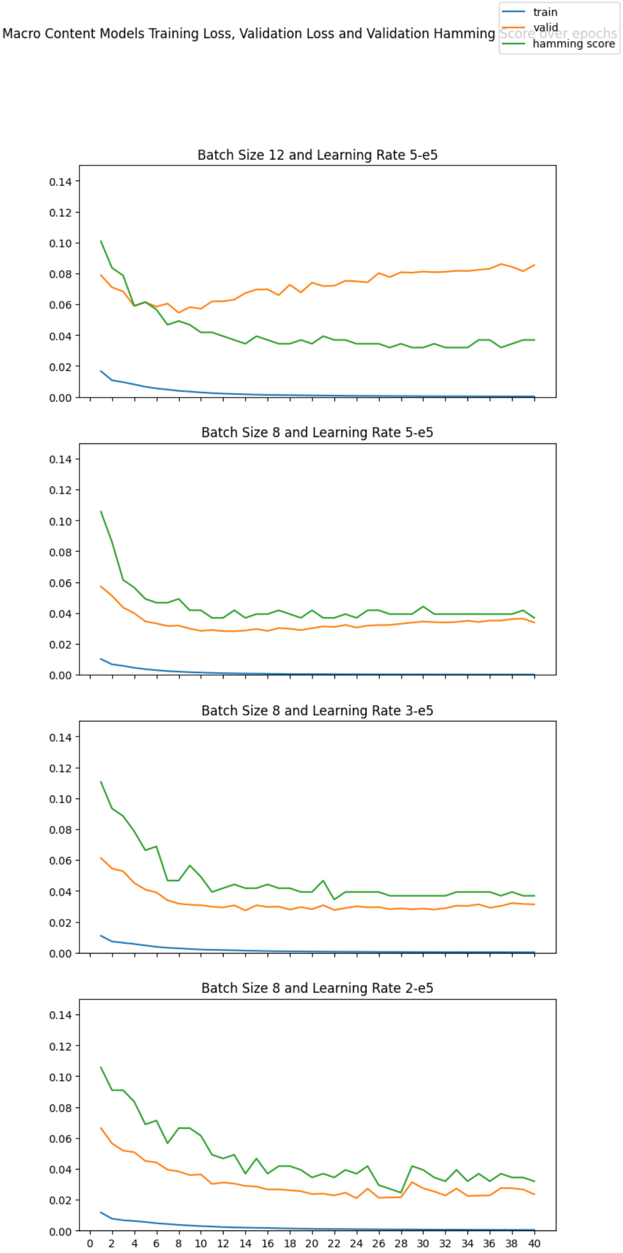


Figure 31