

Automatically Coding Implicit Motives in Picture Story Exercises: The Automated Motive Coder

Max Brede¹, Felix Schönbrodt², Birk Hagemeyer³, Veronika Lerche¹

¹Christian-Albrechts-Universität zu Kiel

²Ludwig-Maximilians-Universität München

³Friedrich-Schiller-Universität Jena

Abstract

The Picture Story Exercise (PSE) is a projective measure in personality psychology where individuals create narratives based on ambiguous images. Traditionally, the coding of these narratives has been labor-intensive. We introduce the Automated Motive Coder (AMC), which employs recent advances in natural language processing and machine learning to automate the coding of PSE narratives. Trained on an extensive dataset, the AMC demonstrates accuracy comparable to expert coders for both original and translated texts. The model offers support for multiple languages that were absent in prior methods while improving in accuracy and speed. To illustrate its effectiveness, we tested and successfully replicated the established psychological effect of gender difference in the affiliation motive. The AMC can be utilized through established machine learning tools, offering a pragmatic and reliable method for coding across several languages. This tool provides an option to reduce the workload involved in PSE coding, promoting efficiency and consistency in motive assessment.

HuggingFace —

<https://huggingface.co/automatedMotiveCoder/setfit>

Introduction

A pivotal assumption of motivational psychology is that individuals differ in their propensity to seek out specific classes of affectively charged goal states (McClelland et al. 1953; Schultheiss and Köllner 2021). For instance, individuals differ in their aspirations of goals pertaining to the most commonly investigated motivational needs for achievement (nAch; a concern for excellence and success), power (nPow; a concern for social impact and prestige), and affiliation-intimacy (nAff; a concern for positive social interactions and relationships). Such dispositions are termed implicit motives because individuals are not aware of their motivational inclinations. Consequently, implicit motives cannot be inferred from self-reports, but have to be assessed in an indirect way. Beginning with nAch in the 1950s, Picture Story Exercises (PSEs) have become the established method of implicit motive measurement (McClelland et al. 1953; Pang 2010).

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

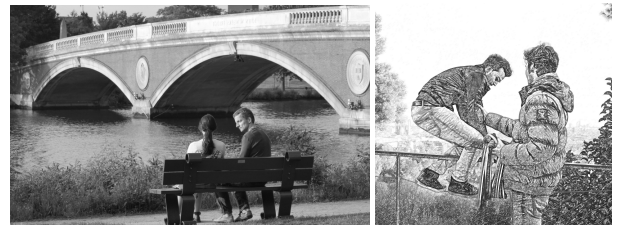


Figure 1: Examples of images used as stimuli in Picture Story Exercises (PSEs). The images are sourced from a collection of candidate PSE images published in Schönbrodt et al. (2021). The left image is titled "newpic14" by Oliver Schultheiss (CC-BY 4.0); the right image is titled "burglars" (CC0).

PSEs are research variants of the Thematic Apperception Test (TAT; Morgan and Muray 1935). Participants are instructed to invent imaginative stories in response to picture cues showing ambiguous, often social, scenes. Typically, a sequence of four to eight pictures that pull for imagery related to the targeted motive domains are presented for 10 – 15 seconds, and participants are given 4 or 5 minutes time per picture to write down a story that describes the depicted scene (Pang 2010). See Figure 1 for example images. The stories are then coded for the appearance of motive-related contents as defined in empirically validated coding systems that were developed for the assessment of the respective motives (for an overview of classical motive coding systems, see Smith 1992).

The process of PSE coding is effortful and requires substantial amounts of time, labor, and expertise. Coders need to undergo training in order to apply the coding rules accurately, which according to Pang (2010) should include at least 12 hours of practice per coder. For a trained and experienced coder, the coding of a single PSE-story takes about 2 – 5 min, which amounts to a coding time of 12 – 30 min. per participant for a typical six-picture PSE. In addition, it is recommended that PSE-stories are coded by at least two independent coders (Pang 2010), which doubles the effort. A second potential problem pertains to the objectivity of PSE coding by human coders. Coders are required to achieve high agreement with expert codings during coder training

(reported requirements are usually a category agreement of 85%, see Winter 1994) and, in case of multiple independent coders, with each other. Nonetheless, the objectivity of codings might still be impaired by coder drift over time (Pang 2010) or differences in coding practices between different labs.

Previous Work

Because of the limited practical efficiency and potential problems of objectivity that come with human coders, motive researchers have been striving to develop automatized, computer-based alternatives for decades (e.g., Green et al. 1967; Smith 1968). The approaches for automated motive coding reach from classic natural language processing methods like bag of word approaches (e.g., Schultheiss 2013) to embedding model based RNN-approaches (e.g., Pang and Ring 2020). Most recently, Nilsson et al. (2024) published transformer-based automated motive coding approaches based on RoBERTa large (Liu et al. 2019) and GBERT (Chan, Schweter, and Möller 2020). Crucially, all of these earlier efforts were fundamentally limited to unilingual processing. Although the two models presented by Nilsson et al. (2024) showed performance comparable to that of expert coders, their reliance on unilingual base models represents a significant constraint.

Tunstall et al. (2022) published a contrastive learning paradigm (described below) that leverages transformer-based embedding models implementing recent advancements. This approach additionally allows for basing classifier architectures on Sentence Transformer (Reimers and Gurevych 2019) models, that are trained to be capable of embedding texts in a variety of languages. We aim to improve on the models trained by Nilsson et al. (2024) by utilizing the training paradigm described by Tunstall et al. (2022) to (1) improve inference speed and accuracy and (2) allow our model to classify texts written in more than one language.

Methods

Training and holdout material

The main dataset is a combined data base of German PSE stories from 27 studies, which have been coded sentence-wise with the Manual for Scoring Motive Imagery in Running Text (Winter 1994) by several trained coders (see Schönbrodt et al. 2021 for details of the data base). Each sentence is labelled with one of the four motive classes nAch, nAff, nPow or “no motive” (null). See Table 1 for examples of coded sentences. In rare cases (see below), multiple classes were assigned. For the computation of a motive score on person level, first all motive labels are summed across all stories that a person wrote. As such a score is typically correlated with the length of stories (longer stories with more sentences have more potential to accumulate motive labels), these sum scores are subsequently corrected for word count (see below).

According to Winter (1994), one category per sentence can be coded for each of the three motives. Thus, motive coding basically takes place at the level of single sentences.

However, there are deviations from this principle. Most notably for this context is the so-called second-sentence rule. This rule states that the same motive class cannot be assigned to two consecutive sentences. Since this rule leads to an inflation in null-codings, this rule has been ignored by a large portion of the stories.

The studies used different sets of pictures, including the standard set of six pictures suggested by Schultheiss and Pang (2007), some pictures from the classical Thematic Apperception Test (Morgan and Muray 1935), and some new pictures. Upon publication, the full PSE data base was split into a public set and a holdout set, with the latter not being publicly released. We used the public set for training, and the non-public holdout set as an additional test set.

For both training and holdout set, we removed all sentences with 5 or fewer characters (503 total sentences). We then removed stories with fewer than 30 words (Smith, Feld, and Franz 1992). As the unit of analysis in Winter’s coding system is the sentence, we furthermore removed stories which had fewer than two sentences. This step removed 1,018 out of 26,376 stories. Furthermore, we removed sentences that were potentially affected by the second-sentence coding rule (5.8%).

For the training set, we also removed sentences that had more than one motive category (0.8%) as the classification algorithm is trained on single motive codings only. In the holdout set, we did not remove the mixed categories, because we wanted to keep the real stories for testing. These selections resulted in a training set of 131,110 sentences in 20,463 stories, written by 3,637 persons, and a holdout set of 33,546 sentences in 4,892 stories, written by 900 persons. Final class frequencies were: nAff (training: 14.63%, holdout: 16.89%), nAch (training: 9.65%, holdout: 9.28%), nPow (training: 13.92%, holdout: 11.69%), null (training: 61.80%, holdout: 58.16%), and mixed categories in the holdout set: 3.98%.

The holdout set was additionally translated to English using DeepL to allow a heuristic test of the multilingual performance of the final model. This was also done to allow for a fair comparison with Nilsson et al. (2024), who followed this approach on the same base-dataset to train a classifier for English texts. Neither the holdout set nor its translation were used in any way during the training of the model.

Training Procedure

The training process was split into two sequential stages, closely adhering to the methodology described by Tunstall et al. (2022). This approach initiates with the contrastive fine-tuning of a Sentence Transformer model (stage one), subsequently transitioning to the training of a classification model leveraging the refined embeddings (stage two).

Stage one involved a contrastive fine-tuning procedure applied to selected Sentence Transformer models. The candidate models were on the one hand a selection of well-tested models published by the authors of the Sentence Transformers library (Reimers and Gurevych 2020) and on the other hand models that performed well on the MTEB benchmark suite for multilingual classification (Muennighoff et al.

Text	Coded Motive
Frederike carries out her experiment routinely and secures her 1st place again.	nAch
The two are sitting in a restaurant and look happy .	nAff
Michael knew that his worst enemy, the well-known mafia boss Luigi, had boarded this ship, his last chance to wipe him out once and for all.	nPow
Next he will land roughly on the ground.	null

Table 1: Translated example sentences from the dataset published by Schönbrodt et al. (2021). The *Coded Motive* column indicates the motive assigned by the expert coders.

2023). For the whole list of candidate models, see Appendix 1.

This fine-tuning-procedure aimed at enhancing the model’s understanding and encoding of complex sentence structures specific to motivational imagery within the PSE narratives. The models were trained to generate closely packed embeddings for sentences with similar motive codes while distancing embeddings for sentences of differing motives. This was done by using pairs of sentences from the training set to maximize the cosine similarity between the embeddings of sentences with the same motive class and minimize it for sentences with different motive classes. This paradigm is meant to greatly reduce the amount of necessary labeled data, since it effectively uses each sentence $n - 1$ times instead of once, where n is the number of sentences in the set.

In this first stage, we trained on randomly sampled subsets of varying sizes from the training dataset. A hyperparameter defined which stratifying variables to use in sampling, ranging in value from one to six. A value of one indicated that only the ‘motive class’ was used as a stratifying variable, while a value of two included both ‘motive class’ and ‘picture ID’. Higher values incorporated additional variables in this order: participant gender (3), participant age group (4), the study ID (5), and the coding lab (6). Thus, the motive class was always part of the stratification, and the other five variables could be optionally added. Stratifying for the target (in our case, the motive class) has been shown to improve cross-validation results (Kohavi 1995). The other possible stratifying variables were chosen and ordered based on their plausible potential to influence the sentence structure and content related to motivational imagery within the PSE narratives. The size of the subsets sampled from the training set ranged from 1 to 5 sentences for each combination of stratifying variables. This number of sentences was also determined by a hyperparameter. If there were fewer sentences available for a given combination than required, the sentences of this combination were oversampled to fulfill the target size. If a combination was not present in the training set, it was ignored.

The fine-tuning phase utilized the Tree-structured Parzen Estimator (TPE) algorithm as implemented in Optuna (Watanabe 2023) for hyperparameter optimization. The TPE algorithm was employed to search through a predefined space of possible hyperparameters, with the objective of finding the combination that yielded the best performance

on a validation set¹. The validation set was randomly sampled from the prepared training set according to the same stratification procedure as described above. It was made sure that the validation and training sets did not overlap. In cases where all available sentences for a given combination of stratifying variables were included in the training set, the combination was omitted from the validation set. For the complete set of hyperparameters tuned, see Appendix 1. The training was implemented using the HuggingFace Sentence Transformers library (Reimers and Gurevych 2019) and set to save the best model according to the performance on the validation set. The F1-score of the cosine with an automatically found threshold was used as the metric for evaluation and as the hyperparameter optimization target. The best model trained using this procedure was then employed to embed the complete training set as well as the holdout set and its English translation.

Stage two used the embedded training set to train multilabel-classification models. These models predicted for each sentence, whether it contained imagery relevant to nAch, nAff, or nPow, or if it contained no motive-relevant imagery at all. All trained classification heads consisted of a OneVsRestClassifier as implemented in scikit-learn (Pedregosa et al. 2011) with one of a series of possible classifier models as its estimator. The classifier models, along with their associated settings, were determined using again the TPE algorithm, as implemented in the Optuna library. Notably, we employed a hyperparameter to determine the motive class weighting strategy during training. This strategy was selected to allow the model to adjust its emphasis on different motive classes and compensate for the significant motive class imbalance within the dataset. By adjusting the weight each motive class was given during training, the model could better handle imbalance and improve prediction accuracy across all motive classes. For the complete list of hyperparameters and their ranges as well as further details on the configuration used during training, please refer to Appendix 2.

Ten-fold cross-validation (CV) was employed to train and evaluate the models. The separation into folds was conducted at the person level, ensuring that no sentences written by the same individual appeared in both the training and

¹In this context, *validation set* refers to the dataset used to calculate hyperparameter optimization targets. Thus, the term does not refer to the psychological quality criterion of validity.

evaluation set of a fold. The folds were stratified according to the uncorrected motive score of a person, their age group, gender and the lab that coded the sentences.

Performance evaluation

Two evaluation performance metrics - one at the sentence level and the other at the person level - were each averaged over all 10 folds, and both were used as optimization targets for hyperparameter tuning: As a metric on sentence level, the average F1-score was chosen. This score was calculated as the weighted average of the four F1 scores, which were determined based on the true and false predictions for each of the four classes (nAch, nAff, nPow, Null) compared to the others on a per-class basis. For the evaluation at the person level, we first calculated the so-called corrected motive scores, i.e., the aggregated value of a person's implicit motive strength based on all stories they wrote. Next, for each motive class separately, we computed the correlation between the predicted and true corrected motive scores (nAch, nAff, nPow). Finally, we averaged the three motive correlation coefficients. The Null-class was left out in this step, since it proved to be inflated for nearly all trained models. We employed the correction-procedure described in Schönbrodt et al. (2021) to calculate the corrected motive scores and applied this procedure both for the motive class probabilities predicted by our model and the motive classes coded by the expert coders (i.e., the true motive classes). This procedure aims to correct for the positive correlation between motive sum scores and story length (Pang 2010) by conducting the following steps:

1. Summation of predicted motive probabilities (model-prediction) and summation of expert-coded motive codings separately for each motive.
2. Robust regression for each implicit motive of the sum scores from step 1 divided by 1,000 on word count per subject (In our case, the `lmrob`-implementation of the extended method suggested in Koller and Stahel (2011) from the `robustbase-R`-package (Maechler et al. 2024) was used.)
3. Extraction of residuals as corrected motive scores per implicit motive

We used the Optuna implementation to determine all Pareto-optimal runs. Of those, the final classifier settings were selected based on the mean of the F1-score and the mean correlation, both scaled so that the maximum value reached in the Pareto-optimal runs was equal to one. The settings with the highest mean of these two scaled metrics were chosen as the best performing model and a final classifier was trained using these settings on the full training set.

While optimizing the model for agreement with expert coders - more specifically in terms of the correlation between predicted and coded motive scores - it is important to establish what constitutes a satisfactory level of agreement. Guidelines suggest that a minimum of 85% category agreement with precoded examples during training is required, as outlined by Winter (1994). To contextualize this benchmark using Pearson correlation metrics, Schultheiss (2013) provides relevant data, reporting inter-coder correlations of .79,

.74, and .86 for nPow, nAch, and nAff, respectively. These correlations were achieved after the coders were trained in a joint lab with common practices and (implicit and explicit) coding norms. Similarly, Schultheiss, Liening, and Schäd (2008) reported inter-coder correlations of .86, .70, and .81 for the same motive domains and the same coder-training procedure. Taken together, these findings suggest that among human coders, Pearson correlations of .70 or higher are typically expected, with values rarely exceeding .85.

Since these correlations are achieved under quite optimal conditions, they may not be indicative of agreement after coder drift has occurred. Phenomena like coder drift lead to declining agreement, as does the presence of coders from multiple labs (who never coded jointly or were calibrated in any way) in the data set. Realistically, expectations for inter-coder correlations are likely lower, especially if the coders have not been trained in the same laboratory. In sum, a model that reaches the values typically reported for human coders demonstrates good performance.

Pang and Schultheiss (2005) and Schultheiss and Pang (2007) additionally suggest one-way random effects intra-class correlations (ICCs) for the evaluation of model performance. However, we argue that this measure is not entirely valid for the motive scores generated by our model. Our motive scores are calculated based on the class-probabilities of the motives while the codings generated by human coders are binary ratings per sentence (i.e., motive class present or not). Since the type I ICC is highly sensitive to mean level differences between coders, this measure might skew the results. Transforming the probabilistic output into a binary rating by setting a threshold at .5 would be an option to transform the probabilities to be usable in the ICC-calculation. But Nilsson et al. (2024) and previous unpublished trials at training an automated coding solution we ran showed that using the summed probabilities resulted in more reliable estimates for the motive scores than the transformed binary ratings.

As an additional measure of the validity of the models' predictions, we computed Standardized Root Mean Square Residuals (*SRMR*). This measure indicates the deviation of two correlation matrices, in our case the intercorrelations of the scores for the three implicit motives. Ideally, automated coding reconstructs the motive intercorrelations from manual coding, where lower *SRMR* values are better.

Lastly, to gauge the criterion validity of the automated codings, we examined whether there are differences in affiliation motive strength between female and male participants. Meta-analytic findings show that after word count correction, women have higher motive scores in the implicit affiliation motive than men. Drescher and Schultheiss (2016) report an average Cohen's $d = 0.45$; 95% $-CI = [0.37, 0.53]$ for this effect. Accordingly, we computed the standardized mean difference (Cohen's d) between women's and men's affiliation scores obtained using the AMC as the gender effect performance measure. Effects should be comparable to those found in the human codings and to the literature.

To benchmark our best model against the approach published by Nilsson et al. (2024), we employed both their mod-

els for English and German texts on our holdout set and its English translation. Performance was evaluated using the just discussed measures, providing a direct comparison of our model’s efficacy against the classifiers trained by Nilsson et al. (2024), both their approach based on the GBert-embeddings and the RoBERTa-embedding based approach.

Results

Remember that the primary training target for all models was the sentence-level prediction of motive class. Notably, the best embedding model (i.e., after stage 1 of the Tunstall et al. 2022 approach) reached a cosine-based F1-score of .61 on the randomly sampled validation set. Thus, even without a classification model, the performance was already quite impressive. This embedding model was based on the multilingual-e5-large-variant of the architecture presented in Wang et al. (2024) with the contrastive loss function implemented in the Sentence Transformer library (Reimers and Gurevych 2019). All settings can be seen in Appendix 1.

The final combination of embedding-model (stage 1) and classifier (stage 2) performing best in the hyperparameter optimizations achieved an average F1-score of .77 on the train folds and an average F1-score of .77 on the validation folds. This result was reached with an Elastic Net classifier with stochastic gradient descent learning as implemented in scikit-learn. For all other settings, see Appendix 2. Importantly, the model also performed well on the holdout set (F1: .76) with only slight decreases in performance for the translated holdout set (F1: .73).

On the person level, our best model reached correlations between model- and human-coded motive scores of on average $r = .71$ for nAch, $r = .83$ for nAff and $r = .76$ for nPow over all ten validation folds. The findings for the holdout set and its translation are in a comparable range (see Table 2).

As reported in the Methods section, we used, next to the aforementioned correlations, two additional metrics as indicators for model performance on the person level.

First, we calculated the *SRMR* for all datasets to test whether our automated coding approach resulted in a deviation in the intercorrelations between the model-coded and the human-coded implicit motive scores. As example we show the intercorrelations for the holdout set in Table 3. The table shows only minor differences in the intercorrelations for all motive scores.

These matrices were used to calculate the *SRMR* as the square root of the mean of the squared differences of the matrices. The *SRMRs* were the highest on the translated to English holdout set (*SRMR* = .0626) with *SRMRs* for Train, Validation and German holdout set of .0159, .0490, and .0430, respectively.

In addition, we calculated Cohen’s d as a measure of the gender difference in the motive score for affiliation. The results for the gender differences found in the training and holdout sets are shown in Table 4. Women showed a higher implicit affiliation motive than men in all datasets, ranging from $d = 0.39$ to 0.57 with the largest effects in the holdout set ($d = 0.53$) and its translation ($d = 0.57$). Interestingly,

all effect sizes, except for the average of the validation folds, are larger for the automatically coded motive scores than for the ones coded by humans.

Comparison to previous approaches

We compared the performance of our model on the holdout set to that of the GBERT and RoBERTa models published by Nilsson et al. (2024). Since the GBERT model is exclusively trained for German texts and the RoBERTa model for English texts, we applied the GBERT model only to the original German holdout set and the RoBERTa model to the translated-to-English holdout set. All model predictions for the models published by Nilsson et al. (2024) were calculated using the R-script presented in their paper. To simplify comparison, we only calculated motive scores for sentences present in both the English and the German set coded by both models published by Nilsson et al. (2024), which is a subset of the data presented above.²

The results on both holdout sets are presented in Table 5. Our model surpasses the performance reached by the GBERT model on all motive scores for the German holdout set. This effect is less clear for the holdout set that was translated to English. Here, the RoBERTa model reached slightly lower correlations than our AMC for all motive scores except for nPow, where no substantial difference can be seen.

Our model additionally generates closer intercorrelations as measured in the *SRMR*, both on the German holdout set (*SRMR* = .04 for our model vs *SRMR* = .11 for the GBERT approach) and the translated-to-English holdout set (*SRMR* = .06 for our model vs *SRMR* = .08 for the RoBERTa approach).

The gender effect measured as Cohen’s d of the difference in the affiliation motive score between women and men is slightly higher for the models published by Nilsson et al. (2024), both on the German holdout set ($d = 0.53$, 95% CI [0.39, 0.67] for our model vs $d = 0.65$, 95% CI [0.50, 0.79] for the GBERT approach) and the translated-to-English holdout set ($d = 0.57$, 95% CI [0.43, 0.71] for our model vs $d = 0.64$, 95% CI [0.49, 0.78] for the RoBERTa approach).

To test for differences in inference time, we repeatedly ran our model and both models published by Nilsson et al. (2024) on a virtual headless ubuntu 22.04 LTS private cloud computing instance with 126 GB of RAM and 64 assigned virtual cores of an Intel Xeon Processor (Icelake). The instance was additionally fitted with two Nvidia A40 graphics cards. All classifications were run for 25 times for the first 1,000 sentences in the holdout/translated holdout set, both on GPU and CPU.

Our model took an average of 1.7 seconds ($SD = 0.96$, $min = 1.5$, $max = 6.4$) for the German and an average of 1.41 seconds ($SD = 0.03$, $min = 1.3$, $max = 1.5$) of inference time for the English holdout set on GPU.

The models by Nilsson et al. (2024) were far slower, with runtimes of an average of 304 seconds ($SD = 2.97$, $min =$

²The reason for this partial selection is a combination of issues with the DeepL-translation and some data cleaning implicit in the text package, Nilsson et al. (2024) used to ship their models.

	<i>N</i>	<i>r</i>			
		nAch	nAff	nPow	Null
Train	3298.3*	.71 [.69, .73] [†]	.83 [.82, .84] [†]	.77 [.75, .78] [†]	.93 [.93, .94] [†]
Validation	363.2*	.70 [.65, .75] [†]	.81 [.78, .85] [†]	.78 [.74, .82] [†]	.92 [.90, .94] [†]
Holdout Set	900.0	.74 [.71, .77]	.79 [.77, .82]	.74 [.70, .76]	.93 [.92, .94]
Translated Holdout Set	900.0	.74 [.71, .77]	.79 [.76, .81]	.73 [.70, .76]	.93 [.93, .94]

Note:

95%-Confidence Intervals in brackets.

* Averaged over all folds. Note that due to the stratification on person level, not all folds consisted of the exact same amount of samples. This was due to some combinations of stratifying variables being present in few persons, leading to their absence in some of the training and/or validation folds.

[†] The correlations and CIs reported for the validation and train folds are those of the runs with the largest CI out of all ten folds.

Table 2: Correlations between human- and model-coded motive scores on person level. Both human- as well as model-coded scores were transformed to consider the number of written words in the PSEs as described in the Methods section

Motive	Human codings		Model predictions	
	nAch	nPow	nAch	nPow
nAch	.03	-.08	.08	-.01
nAff		-.34		-.31

Table 3: Intercorrelations of the 3 implicit motive scores, based on human codings and model-predicted probabilities for the holdout set.

298, $max = 309$) for the German set using the GBERT model and an average of 326 seconds ($SD = 10.29$, $min = 318$, $max = 372$) for the RoBERTa model on the English set on GPU.

This difference in runtime was even more pronounced when running on CPU with averages of 13 seconds ($SD = 0.72$, $min = 12$, $max = 15$) and 12 seconds ($SD = 0.70$, $min = 11$, $max = 14$) for our model and 2,172 seconds ($SD = 33.87$, $min = 2$, 113 , $max = 2,234$) and 4,045 seconds ($SD = 40.41$, $min = 3$, 985 , $max = 4,146$) for the models published by Nilsson et al. (2024).

Discussion

We successfully trained an Automated Motive Coder (AMC) that achieves near-human-level agreement with existing codings. The AMC comprises a fine-tuned multilingual Sentence Transformer embedding model coupled with a multi-label classification header. Schultheiss, Liening, and Schad (2008) and Schultheiss (2013) reported human inter-coder correlations between .70 and .86. The AMC reached values that were very close to this range. It is important to note that these correlations represent a near-optimal con-

	<i>d</i>	
	Human codings	Model predictions
Train*	0.35	0.41
Validation*	0.40	0.39
Holdout Set	0.44 [0.30, 0.58] [†]	0.53 [0.39, 0.67]
Translated Holdout Set	0.44 [0.30, 0.58] [†]	0.57 [0.43, 0.71]

Note:

95% CI in brackets.

* Averaged results over the 10 CV-Folds.

[†] Since the Translated holdout set has the same motive codings as the non-translated one, the effect-sizes of the human codings are identical.

Table 4: Gender effect in the affiliation motive over all datasets.

dition where coders were specifically trained in a joint lab with common practices and (implicit and explicit) coding norms. This level of agreement should not be considered the standard, particularly for our dataset, which involved sentences coded by multiple researchers from different labs with potentially varying strategies. Despite this variability, our AMC achieved a correlation with the human-coded motive scores of above $r = .70$ on all datasets, including the holdout set that was not used for the training of the model. These results are very promising and likely close to the prac-

Motive Score	<i>N</i>	German holdout set		Translated holdout set	
		AMC	GBERT	AMC	RoBERTa
nAch	858	.73 [.70, .76]	.63 [.59, .67]	.74 [.70, .76]	.71 [.68, .74]
nAff	858	.80 [.77, .82]	.67 [.63, .70]	.79 [.76, .81]	.77 [.74, .80]
nPow	858	.74 [.71, .77]	.64 [.60, .68]	.74 [.70, .77]	.74 [.71, .77]

Note:

95%-Confidence Intervals in brackets.

Table 5: Comparison of the correlation between automatically coded and human-coded motive scores achieved with our model and Nilsson et al.’s 2024 GBERT and RoBERTa models, applied to the holdout set and its translation. The GBERT approach is only used on the German and the RoBERTa approach only on the translated-to-English holdout set.

tical limits of what is achievable.

Our model demonstrated robust performance in predicting human motive codings for texts the model did not see during training, that were either German or translated-to-English. By employing the contrastive learning paradigm outlined by Tunstall et al. (2022), we slightly improved upon the accuracy of results reported by Nilsson et al. (2024) while adding support for the coding of samples from multiple languages. Importantly, our model is implemented within the SetFit library and is available on the HuggingFace model repository, ensuring it is openly accessible for application and further research.

In addition to the agreement with human coders, we were able to reach acceptable levels for the model fit measured as SRMRs. Furthermore, we could replicate the gender difference in the affiliation motive reported in the literature, where women consistently showed higher scores in nAff. This effect was reported by Drescher and Schultheiss (2016) with an average Cohen’s *d* of 0.45(95% - *CI* = [0.37, 0.53]). This effect was slightly larger in our automatically coded motives compared to manually coded ones and the literature, although these differences remain marginal. Out of all our tests, these effects were highest for the models published by Nilsson et al. (2024). Note that although we generally considered the replication of the well-established gender effect as a sign of validity, one cannot conclude that “the higher, the better”. On the one hand, it could be that automated coding better picks up a true signal; on the other hand, it could also be that, regardless of the underlying motive, men and women have differences in linguistic styles which are more weighted in automated coding. This apparent tendency of automated coding models to amplify the gender effect should be the subject of further investigation.

Application of automated coding in practice

Our model outperforms the already impressive results reported by Nilsson et al. (2024) for nearly all performance measures tested. This improvement was reached by basing the model on modern embedding models fine-tuned to pronounce differences between motive-relevant imagery in the embeddings instead of using general-purpose language models for the embeddings. Notably, we extended their solution by training a model with a multilingual base, demonstrating acceptable performance across two languages. Inci-

dentally, our architecture also leads to a significant improvement in inference speed, being up to 300 times faster than previous methods. While this acceleration is less critical for traditional, batch-oriented motive coding, it does offer potential for future applications. The increased speed could potentially allow for the integration of the model into experimental paradigms, enabling the generation of stimuli that could be dynamically adjusted based on calculated motive scores.

Nonetheless, the research community has to gain experiences with the new measurement method and yet unknown problems and boundary conditions of its applicability might show up. Researchers could reanalyze their existing studies with the AMC for a retrospective check of its validity, or do both manual and automated coding for new studies. When multiple coders are employed in one study, they often code a subset of the data in parallel to check their agreement. When sufficient agreement has been demonstrated, they continue coding separate batches of the stories. A similar routine could be implemented with the model presented in this study, to test whether cases arise in which the AMC-codes deviate from human results.

The python-code-snippet in Listing 1 demonstrates how the SetFit library and our model can be used for coding text sections in any language supported by the base model trained by Wang et al. (2024)³.

Boundaries and limitations

Our model was trained on classical PSE stories, written in response to a broad, yet limited set of picture stimuli. Although generalization to new pictures is expected to be adequate, caution is advised when applying it to other text genres, such as transcribed interviews or speeches.

Our model inherits its validity and generalizability from the validity of the human-derived coding system and the validity of the concrete human codings. The Winter coding system is based on experimentally derived coding systems, where groups of participants were exposed to motive-arousing situations. These studies have been conducted in certain (Western) cultural contexts with certain picture stimuli. Whenever such coding systems are transferred to other

³The HuggingFace repository lists a total of 93 languages as being supported. A longer tutorial with installation-instructions can be found under the link above.

Listing 1: Example on how to use the AMC.

```
from setfit import SetFitModel

model = SetFitModel.from_pretrained(
    "automatedMotiveCoder/setfit"
)

model.predict_proba(
    [
        "Du schaffst das schon.",
        "Tu vas y arriver.",
        "Zvládnete.",
        "You'll manage.",
        "Te las arreglarás.",
        "Saate hakkama."
    ]
)
```

contexts or languages, their validity should be carefully evaluated. Though our model is multilingual in nature, its training and evaluation were restricted to German or translated-from-German stories. Therefore, a certain homogeneity in the motives and language used has to be assumed. It remains to be tested whether our model can be applied to stories originally written in other languages than German - especially since Pang and Schultheiss (2005) could show a significant difference in motive scores between ethnicities and in comparison between German and U.S.-American students.

Conclusion

This study developed an Automated Motive Coder (AMC) that achieves human-level accuracy in implicit motive coding using contrastive learning. The AMC demonstrates effective performance on both German and English texts and is accessible via the SetFit library and HuggingFace repository, with theoretical support for a variety of additional languages. While promising, further research is required to assess the validity of the AMC across diverse text genres and cultural contexts. Future studies should aim to examine its applicability and reliability across various settings.

Appendix 1 - Sentence Transformer Hyperparameter

For the hyperparameter tuning of the embedding model (stage 1 of our training), we used the following selection of candidate base models (all names are also the names of the corresponding HuggingFace repositories):

- intfloat/multilingual-e5-large-instruct (Wang et al. 2024)
- intfloat/multilingual-e5-large (Wang et al. 2024)
- intfloat/multilingual-e5-small (Wang et al. 2024)
- shibing624/text2vec-base-multilingual (Xu 2023)
- ISOISS/jina-embeddings-v3-tei (Sturua et al. 2024)
- sentence-transformers/distiluse-base-multilingual-cased-v2 (Reimers and Gurevych 2020)
- sentence-transformers/distiluse-base-multilingual-cased-v1 (Reimers and Gurevych 2020)

The training itself was implemented with the HuggingFace-Sentence Transformers-Trainer-Class (Reimers and Gurevych 2019) using a setting from the following space of hyperparameters:

- Number of train epochs as 10^{epochs} where *epochs* is an integer between 2 and 8
- A per device train batch size of $2^{trainbatch}$ where *trainbatch* is an integer between 3 and 6
- A per device eval batch size of $2^{evalbatch}$ where *evalbatch* is an integer between 3 and 6
- The trainer's learning rate as a uniformly distributed float between $1e-6$ and $1e-3$
- The ratio of total training steps used by the trainer for scaling up the learning rate from 0 to its final value as a uniformly distributed float between .05 and .4
- The trainer's learning rate scheduler, selected as either "reduce_lr_on_plateau" or "cosine"
- one of the following two loss functions implemented in the Sentence Transformers library:
 - Contrastive loss as described in Hadsell, Chopra, and LeCun (2006)
 - Cosine Sentence loss as described in Jianlin (2022)

In addition to these model settings, we implemented the selection of the samples used for training as hyperparameters. For a detailed description of this procedure, see the Methods section. This sampling strategy was represented as a set of two hyperparameters:

- The amount of stratifiers used from this list in this order: motive class, picture ID, participant gender, participant age group, study ID, and the coding lab
- The amount of sentences sampled per combination of stratifiers as an integer between 1 and 5

The best performing combination of hyperparameters was as follows:

- model: intfloat/multilingual-e5-large
- train epochs: 2
- train batch: 6
- eval batch: 5
- learning rate: 0.0009
- warmup ratio: 0.18
- scheduler: cosine
- loss-function: contrastive
- count of stratifiers: 6
- sentences per combination: 5

Appendix 2 - Classifier Hyperparameter

For the training of the classifier (stage 2), we implemented two general hyperparameters, with one of them selecting the classifier-model to be trained. Based on this selection, there was another set of model-specific hyperparameters.

Next to the selection of the classifier-model, we implemented the weighting procedure as a hyperparameter. The selection was between 'balanced', 'none', and 'a bit of null':

- “balanced” calculated the weights so that each motive class including null was weighed by $1/n_{class}$
- “none” meant all classes were weighted with 1
- “a bit of null” was an adaptation of the “balanced” weights after we realized the “balanced” weighting to over-zealously punish the Null class. To compensate for this, the “a bit of null” strategy multiplied the “balanced” weight of the Null class by a factor of 1.5.

We implemented the following models with the hyperparameters listed below as candidate classifiers. Where not indicated otherwise, we used the scikit-learn implementation of the model.

- CatBoost (Dorogush, Ershov, and Gulin 2018)
 - Depth: 2 to 16
 - Grow Policy: ‘Depthwise’, ‘Lossguide’, ‘Symmetric-Tree’
 - Learning Rate: 0.0 to 1.0
- Decision Tree
 - Criterion: ‘gini’, ‘entropy’
 - Maximum Depth: 0 to 100
 - Maximum Features: ‘sqrt’, ‘log2’, ‘none’
 - Minimum Impurity Decrease: 0.0 to 1.0
 - Minimum Samples Leaf: 0.01 to 0.99
 - Minimum Samples Split: 2 to 100
 - Minimum Weight Fraction Leaf: 0.0 to 0.5
 - Splitter: ‘best’, ‘random’
- Extra Trees (ETree)
 - Criterion: ‘gini’, ‘entropy’
 - Maximum Depth: 0 to 100
 - Maximum Features: ‘sqrt’, ‘log2’
 - Minimum Impurity Decrease: 0.0 to 1.0
 - Minimum Samples Leaf: 0.01 to 0.99
 - Minimum Samples Split: 2 to 100
 - Minimum Weight Fraction Leaf: 0.0 to 0.1
- Elastic Net (implemented as SGDClassifier with ‘elastic-net’ penalty)
 - Alpha: 0.001 to 1.0
 - L1 Ratio: 0.0 to 1.0
 - Tolerance: 1e-08 to 1.0
- Hist Gradient Boosting
 - L2 Regularization: 0.0 to 1.0
 - Learning Rate: 1e-08 to 1.0
 - Maximum Depth: 0 to 100
 - Maximum Iterations: 10 to 10000
 - Maximum Leaf Nodes: 2 to 100000
 - Minimum Samples Leaf: 1 to 1000
 - Tolerance: 0.0 to 0.1
- LightGBM (Ke et al. 2017)
 - Boosting Type: ‘gbdt’, ‘dart’
 - Learning Rate: 0.0 to 1.0
 - Maximum Depth: -1 to 100
 - Minimum Child Samples: 1 to 10000
 - Minimum Child Weight: 0.0 to 1.0
 - Minimum Split Gain: 0.0 to 1.0
 - Number of Estimators: 10 to 1000
 - Number of Leaves: 8 to 100
 - Subsample for Bin: 0 to 5000000
- Logistic Regression
 - C: 0.0 to 5.0
 - Multi-Class: ‘ovr’, ‘multinomial’
 - Solver: ‘lbfgs’, ‘newton-cg’, ‘sag’, ‘saga’
 - Tolerance: 1e-08 to 1.0
- Multi-Layer Perceptron (custom implementation using pyTorch)
 - Batch Size: 5 to 7
 - Depth: 0 to 6
 - Dropout Layers [1-6]: True or False
 - Dropout Rate Layers [1-6]: 0.01 to 0.99
 - Epochs: 0 to 6
 - Gamma: 0.95 to 1.0
 - Learning Rate: 1e-10 to 0.1
 - Minimum Delta: 1e-08 to 0.01
 - Optimizer: ‘adam’, ‘adagrad’, ‘sgd’, ‘adadelat’
 - Patience: 5 to 10
 - Regularization Layers [1-6]: True or False
 - Width Layers [1-6]: 4 to 12
- Naive Bayes
 - Alpha: 0.0 to 5.0
- Nearest Centroid
 - Metric: ‘euclidean’, ‘manhattan’
- Passive Aggressive
 - C: 0.0 to 5.0
 - Early Stopping: True or False
 - Maximum Iterations: 500 to 10000
 - Shuffle: True or False
 - Tolerance: 1e-08 to 1.0
 - Validation Fraction: 0.0 to 0.5
- Perceptron
 - Alpha: 0.0 to 1.0
 - Early Stopping: True or False
 - Eta0: 1e-08 to 1.0
 - Maximum Iterations: 1 to 10000
 - Penalty: ‘l1’, ‘l2’, ‘elasticnet’, ‘none’
 - Shuffle: True or False
 - Tolerance: 1e-08 to 1.0
 - Validation Fraction: 0.0 to 0.5
- Quadratic Discriminant Analysis (QDA)

- Regularization Parameter: 0.0 to 1.0
- Random Forest
 - Criterion: ‘gini’, ‘entropy’, ‘log_loss’
 - Maximum Depth: 0 to 100
 - Maximum Features: ‘sqrt’, ‘log2’
 - Minimum Impurity Decrease: 0.0 to 1.0
 - Minimum Samples Leaf: 0.01 to 0.99
 - Minimum Samples Split: 2 to 100
 - Minimum Weight Fraction Leaf: 0.0 to 0.5
 - Number of Estimators: 0 to 10000
- Ridge Regression
 - Alpha: 0.1 to 5.0
 - Solver: ‘auto’, ‘svd’, ‘cholesky’, ‘lsqr’, ‘sparse_cg’, ‘sag’, ‘saga’
 - Tolerance: 1e-08 to 1.0
- XGBoost (Chen and Guestrin 2016)
 - Colsample Bytree: 0.0 to 1.0
 - Gamma: 0.0 to 100.0
 - Grow Policy: ‘depthwise’, ‘lossguide’
 - Learning Rate: 0.0 to 1.0
 - Maximum Depth: 5 to 100
 - Maximum Leaves: 0 to 80
 - Minimum Child Weight: 0.0 to 1000.0
 - Subsample: 0.0 to 1.0

The settings selected as best were the following:

- weighting-strategy: “a bit of null”
- model: Elastic Net
- Alpha: 0.88
- L1 Ratio: 0.21
- Tolerance: 0.11

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Projektnummer (LE 4379/2-1). The authors would like to thank Prof. Dr. Michael Prange of Kiel University for Applied Sciences for providing access to the compute resources used in this project.

References

- Chan, B.; Schweter, S.; and Möller, T. 2020. German’s Next Language Model. In Scott, D.; Bel, N.; and Zong, C., eds., *Proceedings of the 28th International Conference on Computational Linguistics*, 6788–6796. Barcelona, Spain (Online): International Committee on Computational Linguistics.
- Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.
- Dorogush, A. V.; Ershov, V.; and Gulin, A. 2018. CatBoost: Gradient Boosting with Categorical Features Support. arXiv:1810.11363.
- Drescher, A.; and Schultheiss, O. C. 2016. Meta-Analytic Evidence for Higher Implicit Affiliation and Intimacy Motivation Scores in Women, Compared to Men. *Journal of Research in Personality*, 64: 1–10.
- Green, B. F.; Stone, P. J.; Dunphy, D. C.; Smith, M. S.; and Ogilvie, D. M. 1967. The General Inquirer: A Computer Approach to Content Analysis. *American Educational Research Journal*, 4(4): 397.
- Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 2, 1735–1742.
- Jianlin, S. 2022. CoSENT: A More Efficient Sentence Vector Scheme than Sentence-BERT. <https://kexue.fm/archives/8847>.
- Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, 3149–3157. Red Hook, NY, USA: Curran Associates Inc. ISBN 978-1-5108-6096-4.
- Kohavi, R. 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Ijcai*, volume 14, 1137–1145. Montreal, Canada.
- Koller, M.; and Stahel, W. A. 2011. Sharpening Wald-type Inference in Robust Regression for Small Samples. *Computational Statistics & Data Analysis*, 55(8): 2504–2515.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692.
- Maechler, M.; Rousseeuw, P.; Croux, C.; Todorov, V.; Ruckstuhl, A.; Salibián-Barrera, M.; Verbeke, T.; Koller, M.; Conceicao, E. L. T.; and Anna di Palma, M. 2024. *Robustbase: Basic Robust Statistics*.
- McClelland, D. C.; Atkinson, J. W.; Clark, R. A.; and Lowell, E. L. 1953. Toward a Theory of Motivation. In *The Achievement Motive*, Century Psychology Series, 6–96. East Norwalk, CT, US: Appleton-Century-Crofts.
- Morgan, C. D.; and Muray, H. A. 1935. A Method for Examining Fantasies” Dalam The Thematic Apperception Test. *Archives of Neurology & Psychiatry*, 34: 289–306.
- Muennighoff, N.; Tazi, N.; Magne, L.; and Reimers, N. 2023. MTEB: Massive Text Embedding Benchmark. arXiv:2210.07316.
- Nilsson, A.; Runge, J. M.; Kjell, O. N. E.; Soni, N.; Ganesan, A. V.; and Nilsson, C. V. 2024. Automatic Implicit Motives Codings Are as Accurate as Humans’, Cheaper, and 99% Faster.
- Pang, J. S. 2010. Content Coding Methods in Implicit Motive Assessment: Standards of Measurement and Best Practices for the Picture Story Exercise. In Schultheiss, O. C.; and Brunstein, J. C., eds., *Implicit Motives*. Oxford: Oxford University Press. ISBN 978-0-19-533515-6 978-1-282-38812-3 978-0-19-971504-6.
- Pang, J. S.; and Ring, H. 2020. Automated Coding of Implicit Motives: A Machine-Learning Approach. *Motivation and Emotion*, 44(4): 549–566.
- Pang, J. S.; and Schultheiss, O. C. 2005. Assessing Implicit Motives in U.S. College Students: Effects of Picture Type and Position, Gender and Ethnicity, and Cross-Cultural Comparisons. *Journal of Personality Assessment*, 85(3): 280–294.
- Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

- Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings Using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Reimers, N.; and Gurevych, I. 2020. Making Monolingual Sentence Embeddings Multilingual Using Knowledge Distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Schönbrodt, F. D.; Hagemeyer, B.; Brandstätter, V.; Czirkmantori, T.; Gröpel, P.; Hennecke, M.; Israel, L. S. F.; Janson, K. T.; Kemper, N.; Köllner, M. G.; Kopp, P. M.; Mojzisch, A.; Müller-Hotop, R.; Prüfer, J.; Quirin, M.; Scheidemann, B.; Schiestel, L.; Schulz-Hardt, S.; Sust, L. N. N.; Zygar-Hoffmann, C.; and Schultheiss, O. C. 2021. Measuring Implicit Motives with the Picture Story Exercise (PSE): Databases of Expert-Coded German Stories, Pictures, and Updated Picture Norms. *Journal of Personality Assessment*, 103(3): 392–405.
- Schultheiss, O.; and Köllner, M. G. 2021. Implicit Motives. In John, O. P.; and Robins, R. W., eds., *Handbook of Personality: Theory and Research*. New York London: The Guilford Press, fourth edition edition. ISBN 978-1-4625-5048-7 978-1-4625-4495-0.
- Schultheiss, O. C. 2013. Are Implicit Motives Revealed in Mere Words? Testing the Marker-Word Hypothesis with Computer-Based Text Analysis. *Frontiers in Psychology*, 4.
- Schultheiss, O. C.; Liening, S. H.; and Schad, D. 2008. The Reliability of a Picture Story Exercise Measure of Implicit Motives: Estimates of Internal Consistency, Retest Reliability, and Ipsative Stability. *Journal of Research in Personality*, 42(6): 1560–1571.
- Schultheiss, O. C.; and Pang, J. S. 2007. Measuring Implicit Motives. In Robins, R.; Fraley, R.; and Krueger, R., eds., *Handbook of Research Methods in Personality Psychology*, 322–344. Guilford Publications. ISBN 978-1-60623-612-3.
- Smith, C. P. 1992. *Motivation and Personality: Handbook of Thematic Content Analysis*. Cambridge University Press.
- Smith, C. P.; Feld, S. C.; and Franz, C. E. 1992. Methodological Considerations: Steps in Research Employing Content Analysis Systems. In Smith, C. P., ed., *Motivation and Personality: Handbook of Thematic Content Analysis*, 515–536. Cambridge: Cambridge University Press. ISBN 978-0-521-40052-7.
- Smith, M. S. 1968. The Computer and the TAT. *Journal of School Psychology*, 6(3): 206–214.
- Sturua, S.; Mohr, I.; Akram, M. K.; Günther, M.; Wang, B.; Krimmel, M.; Wang, F.; Mastrapas, G.; Koukounas, A.; Koukounas, A.; Wang, N.; and Xiao, H. 2024. Jina-Embeddings-v3: Multilingual Embeddings with Task LoRA. arXiv:2409.10173.
- Tunstall, L.; Reimers, N.; Jo, U. E. S.; Bates, L.; Korat, D.; Wasserblat, M.; and Pereg, O. 2022. Efficient Few-Shot Learning Without Prompts. arXiv:2209.11055.
- Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; and Wei, F. 2024. Multilingual E5 Text Embeddings: A Technical Report. arXiv:2402.05672.
- Watanabe, S. 2023. Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance. arXiv:2304.11127.
- Winter, D. G. 1994. *Manual for Scoring Motive Imagery in Running Text*. Department of Psychology, University of Michigan.
- Xu, M. 2023. Text2vec: A Tool for Text to Vector.