Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5B
D-70569 Stuttgart

Masterarbeit

Prompt-based Personality Profiling: Filtering Social Media Posts using Reinforcement Learning

Jan Hofmann

Studiengang:	Informatik
Prüfer:	Prof. Dr. Roman Klinger
	Prof. Dr. Sebastian Padó
Betreuer:	Prof. Dr. Roman Klinger
	Dr. Cornelia Sindermann
begonnen am:	14.11.2023
beendet am:	14.05.2024

Abstract

Author profiling is the task of inferring characteristics about individuals by analyzing content they share. To date, systems that perform this task automatically, predominantly use supervised machine learning approaches and borrow from advances in the field of natural language understanding. However, while for many language understanding tasks immense progress has been made in recent years (e.g. by using pre-train then fine-tune paradigm), such progress most often does not transfer to automated profiling systems directly. One reason for this is that author profiling is inherently different from typical text inference tasks due to the possibly large amounts of content associated with an author. Therefore, this work proposes a new method for profiling that tries to select the most relevant parts of content shared by an author before inferring a characteristic. Here, instead of relying on ground-truth labels, this work uses the feedback from the zero-shot capabilities of a large language model to learn such a selection model via reinforcement learning, and evaluates this approach for personality profiling in social media. In experiments predicting big five personality traits, this work finds that prediction quality of such a system is comparable yet slightly worse to using all content associated to a profile in a zero-shot setting, while prediction time is reduced significantly due to the limited amount of content used for inferring personality in the proposed method. In addition, this work finds that simply selecting content arbitrarily leads to performance degradation for most traits, and therefore, this work concludes that, to some extent, the proposed method is able to distinguish between relevant and irrelevant content. Further, this work compares the proposed approach to existing supervised approaches and finds that such methods outperform the proposed method substantially. Still, since the ability of the proposed system to distinguish between relevant and irrelevant content of authors is closely tied to the capabilities of large language models, it can be expected that, with advances of such models, prediction quality of the proposed approach will increase in the future.

Zusammenfassung

Die Erstellung eines Autorenprofils beschäftigt sich mit der Aufgabe aufgrund von Inhalten, welche mit einem Autor assoziiert sind, Merkmale über diesen zu schließen. Bislang werden in System, die diese Aufgabe automatisiert durchführen, vorwiegend Ansätze des maschinellen Lernens genutzt, welche auf dem Prinzip des überwachten Lernens basieren und Fortschritte im Bereich des Verstehens von natürlicher Sprache durch Software nutzen. Während jedoch für vielen Aufgaben in diesem Bereich in den letzten Jahren große Fortschritte erzielt wurden (z.B. mithilfe des "Pre-train then Fine-tune" Paradigmas), lassen sich diese Fortschritte meist nicht auf Profiling-Systeme übertragen. Ein Grund dafür ist, dass sich die Erstellung von Profilen aufgrund der deutlich größeren Menge an Inhalten, stark von typischen Aufgaben im Bereich des Verstehens von Sprache unterscheidet. Daher wird in dieser Arbeit eine neue Methode für Profiling vorgestellt, welche zunächst versucht möglichst relevante Inhalte eines Autors zu bestimmen. Dazu stellt diese Arbeit ein Model vor, welches diese Aufgabe durch das Feedback eines Sprachmodells mithilfe bestärkendem Lernen erlernt und untersucht diesen Ansatz für die automatische Erstellung von Persönlichkeitsprofilen in sozialen Medien. Die Experimente dieser Arbeit zeigen, dass die Qualität der Vorhersagen eines solchen Systems vergleichbar, jedoch etwas schlechter sind als bei einem System welches alle mit einem Profil verbundenen Inhalte nutzt, wenn Big Five Persönlichkeitsfaktoren mittels Zero-Shot Klassifikation vorhergesagt werden. Diese Experimente zeigen jedoch auch, dass sich durch den stark reduzierten Inhalt, wie er in der vorgeschlagene Methode genutzt wird, die benötigte Zeit um eine Vorhersage zu treffen signifikant reduzieren lässt. Weitere Experimente zeigen zudem, dass eine zufällige Auswahl von Inhalten für die meisten der fünf Persönlichkeitsfaktoren zu schlechteren Ergebnissen führt. Aufgrund dieser Ergebnisse schließt diese Arbeit, dass die vorgeschlagene Methode zu einem gewissen Grad in der Lage ist, zwischen relevanten und irrelevanten Inhalten zu unterscheiden. Weiterhin vergleicht diese Arbeit die vorgestellte Methode mit Methoden, welche auf dem Prinzip des überwachten Lernens basieren und findet, dass Vorhersagen dieser Methoden die der vorgeschlagenen Methode deutlich übertreffen. Dennoch ist anzunehmen, dass die Vorhersagequalität des vorgeschlagenen Ansatzes in Zukunft steigen wird, da die Fähigkeiten dieses Systems stark an die Fähigkeiten großer Sprachmodelle gebunden sind.

Contents

1 Introduction

With the ever-growing amounts of user generated content available on online platforms and social media in the last two decades, analyzing and inferring information from such content has become a very active research area. A subfield of this research area, namely author profiling, aims to infer or extract information about individual authors by analyzing the content they share. In such a setting, a diverse set of characteristics such as age and gender (Koppel et al., 2002; Argamon et al., 2003; Schler et al., 2006), native language (Koppel et al., 2005), educational background (Coupland, 2007), personality traits (Pennebaker et al., 2003; Golbeck et al., 2011; Kreuter et al., 2022), or political ideology (Conover et al., 2011; García-Díaz et al., 2022) inferred from the content an author shares have been explored. Being able to collect such characteristics about individual authors automatically can be applied to various practical applications. For example, knowing the personality of someone can be valuable from a marketing perspective to personalize advertising such that ads are in line with a user's personality trait (Winter et al., 2021). Here, Matz and Netzer (2017), for example, found that informing marketing strategies with automated personality recognition increases the effectivity of such strategies. Furthermore, does predicting personality from authors of online written content using automated systems enable large-scale personality analyses of social-media users and has the potential to yield insights into societal dynamics. Similarly, predicting political preferences of social media users can be used for monitoring political opinions and has shown to be able to forecast election results (Tumasjan et al., 2010).

If relying solely on textual content shared by an author, author profiling is typically tackled as a supervised text classification or regression task. Still, profiling is different; instead of relying on a single textual instance for inferring some information, automated systems that aim to infer a characteristic about an author or profile deal with a set of text instances associated an author possibly consisting of hundreds or even thousands of individual textual instances from which automatic systems need to draw a single decision. In the last decade, the idea of creating vector representations trying to capture semantics of language (e.g., Pennington et al.,

2014) in combination with neural networks, the development of language models using transformer architectures (Vaswani et al., 2017) and the pre-train then fine-tune paradigm drastically increased the ability of automatic systems to infer information from textual content for many text classification tasks. However, despite the success of such deep learning strategies in various natural language processing tasks, like sentiment analysis or machine translations, such advanced approaches often underperform when applied to author profiling (Lopez-Santillan et al., 2023). Here, traditional frequency-based statistical machine learning methods often yield competitive results. One factor contributing to this may that many advanced machine learning models like BERT (Devlin et al., 2019) have constraints on the length of the input they can process, often preventing them from processing all content associated with an author at once. Likely, another reason for this is that not all content shared by an author is equally useful or indicative when predicting certain characteristics. Some of the content associated with an author even may be distracting or act as noise, making it difficult for today's approaches to grasp patterns that generalize well when predicting characteristics of an author.

Therefore, this work argues that automatic profiling systems should actively learn to distinguish between relevant and misleading text instances by an author automatically before inferring some characteristic, and then only use a number of selected "relevant" instances to predict specific characteristics. By filtering out unnecessary information and noise, not only accuracy of automated profiling systems could be enhanced but also computational resources required of such systems could significantly be reduced, potentially leading to more efficient and robust systems. Here, instead of relying on expensive collections of manually annotated data to learn such a filter using supervised learning, this work proposes to learn this filter indirectly through interactions of a reinforcement learning agent with a large language model in a zero-shot learning setup. To the best of my knowledge, the intersection between author profiling and reinforcement learning has not been explored yet. Still, such a learned filter, selecting a number of relevant text instances from all content associated with an author does not solve the profiling problem as the reinforcement learning agent does not implicitly predict characteristics about an author. To solve

this, this work combines this reinforcement learning-based filtering approach with the zero-shot capabilities of recent large language models; after retrieving writings by an author the agent marks as relevant, these writings are fed to a large language model in a zero-shot learning setup to infer attributes about the given author.

Although such an approach can potentially be applied to profiling tasks predicting all sorts of characteristics, the work proposed in this thesis focuses on reconstructing personality from social media users. More precisely, this study focuses on user generated content on the social media platform X[1] (formerly Twitter), in which each author is associated with a set of tweets[2]. In particular for predicting personality, this work hypothesizes that not all tweets linked to individuals may be suitable for deducing personality traits correctly as content and tone of tweets from the same author may vary depending on factors such as mood, current events, or specific interest at a given time. Personality, however, characterizes differences between persons present over time and across situations. Therefore, it is likely that only certain tweets truly reflect an author's underlying personality, and consequently, it becomes essential to identify and remove tweets that are less indicative or even counterproductive when predicting an author's personality to enhance the accuracy of automated personality prediction systems. To evaluate the proposed filter and the personality predictions system that results from using this filter in combination with a zero-shot learning setup, this thesis answers the following research questions:

RQ1: Do the tweets selected by the learned reinforcement learning-based filter lead to a better zero-shot performance in reconstructing personality compared to using all tweets of authors?

RQ2: Do the tweets selected by the proposed reinforcement learning-based filter lead to a better zero-shot performance in reconstructing personality compared to using a random selection of tweets?

RQ3: Does the proposed approach perform better at reconstructing personality compared to a selection of supervised learning-based baseline methods?

[1] https://x.com/

[2] A "tweet" is a short message from a user on the social media platform X.

2 Technical Background

2.1 Word Embeddings and Language Modeling

A key objective in natural language processing (NLP) is to make computers capable of understanding and interpreting human-generated text. To achieve these goals, we need to represent words, or the meaning of words, we use as humans in a way such that a computer is able to understand them. A very simple approach to this is to use a one-hot encoding, in which a word is represented as a binary vector of the size of a fixed vocabulary. However, such a representation only encodes the presence or absence of a word, and can therefore only be used for frequency based reasoning as it does not capture the meaning of a word, while also suffering from the curse of dimensionality (Bellman, 1957). Vector semantics tries to solve this by learning representations capturing the meaning of words from their distribution in text. This is motivated by the distributional hypothesis (Harris, 1954), stating that words that occur in similar context tend to have similar meaning and was famously summarized by Firth (1957) as "you shall know a word by the company it keeps".

Such representations of meaning are called word embeddings and various methods for deriving them have been proposed in the last decade. One early approach is *word2vec* (Mikolov et al., 2013), which proposes two variants for creating word embeddings; the continuous bag-of-words (CBOW) and skip-gram variant. In the CBOW variant a shallow feed-forward neural network is trained to predict a word given a context, i.e. the words surrounding this word in a phrase, while in the skip-gram variant the network is trained to predict the context given a word. After training, the weights of such a network are used as the embeddings. This approach greatly contributed to advancing the field of NLP as the generated vector representations showed to be able to capture semantic relationships between words, while also creating a dense vector representation compared to the sparse one-hot representation. Here, the key idea is that such representations can be learned from continuous text without the need for explicitly labeled data, since labels are obtained from the data itself (Liu et al., 2023). This paradigm is called *self-supervised learning* and is the

foundation to many state-of-the-art approaches nowadays. Such representations are typically learned from large amounts general text, i.e. not from corpora designed to model a specific phenomena or suite specific tasks, and the obtained representation can be applied to various NLP tasks like text classification or machine translation. This is an instance of *transfer learning* since knowledge learned about language, is transferred to more specific downstream tasks that can benefit from this knowledge.

However, the meaning of words can change depending on the current context they appear in. In the phrases "Spring is coming" and "I had to replace a spring in my car" the word "spring" has different meanings: (1) a season and (2) a coiled metal. Further the word "novel", for example, can mean (1) a fiction book or (2) be the adjective describing something new or original. This means that often, words are ambiguous but we can disambiguate their sense based on the words surrounding them. Such changes in the meaning of words based on their occurrence in an individual context is not captured by static embeddings like *word2vec* or *GloVe* (Pennington et al., 2014) since, for each word, they allow only a single context-independent representation (Peters et al., 2018). To solve this issue, several approaches have been developed to integrate the current context into the representation of a word (Peters et al., 2017; McCann et al., 2017; Peters et al., 2018; Devlin et al., 2019). With *ELMo* (embeddings from language models), for example, Peters et al. (2018) proposed to create context-aware representations by first pre-training a bidirectional long short-term memory (biLSTM) network (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005) and then fine-tuning this network to downstream tasks and model architectures. This is an example for the nowadays prominent *pre-train* then *fine-tune* paradigm. Here, representations are first pre-trained to a general language modeling task and fine-tuned to a specific supervised downstream task (Dai and Le, 2015; Howard and Ruder, 2018; Radford et al., 2018). Similarly to *word2vec* for deriving word representations, these approaches use large amounts of text in combination with a self-supervised language model objective to generate general knowledge about language use. However, this paradigm is different in sense that (1) the pre-trained embeddings are no longer seen as fixed parameters and (2) the pre-trained model is only slightly adapted to fit the desired downstream task. One

advantage of this is that only few parameters need to be learned from scratch when applying the model to a new tasks (Devlin et al., 2019).

The models that result from this paradigm, like BERT (Devlin et al., 2019), or GPT-2 (Radford et al., 2019), are called *pre-trained language models* (PLMs) and can be used for a variety of different natural language tasks. Moreover, research found that when scaling up such PLMs (e.g. from 1.5B parameters in GPT-2 to 175B parameters in GPT-3 (Brown et al., 2020) or 540B parameters in PaLM (Chowdhery et al., 2022)), while having the same or a similar architecture, such models start to show unexpected abilities. Such an emergent ability is, for example, that at a certain scale, models start to be able to solve task reliably via *few-shot prompting* (Wei et al., 2022b), i.e. without the need for task specific supervised fine-tuning. To be able to differentiate between pre-trained models and pre-trained models of such massive scales, such models are often referred to as *large language models* (LLMs). Nowadays, the most commonly used architecture for building such pre-trained language models is the transformer architecture (Vaswani et al., 2017). Hence, the following will introduce this architecture, the ideas surrounding it, and paradigms, such as *prompting*, that evolved from this architecture.

2.2 Towards Transformer-based Language Models

The transformer architecture emerged from the family of *encoder-decoders* (Sutskever et al., 2014), and since its introduction in 2017, has become the dominant architecture for many language models and quickly revolutionized natural language processing. Until then, recurrent neural network architectures like the long short-term memory and gated recurrent unit (Chung et al., 2014) were the primary building blocks for many sequence modeling tasks like neural machine translation and language modeling (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2014). However, similar to traditional recurrent neural networks (RNNs), these architectures suffer from different problems: the vanishing gradient problem when operating on very long sequences (Bengio et al., 1994), and the sequential structure of recurrence, prohibiting parallelized training within a training sample making such

architectures inefficient to train on modern computer hardware. With the transformer architecture Vaswani et al. (2017) solve these issues by omitting recurrence completely and solely relying on the concept *self-attention* (Cheng et al., 2016), creating an architecture which can be trained very efficiently.

The following will introduce the intuition behind the attention concept. Note that for simplicity this work refers to input and output sequences to series of words in spite of the fact that such inputs are usually first tokenized by algorithms like *byte-pair-encoding* (Gage, 1994) or *Sentencepiece* (Kudo and Richardson, 2018).

2.2.1 The Attention Mechanism

Early encoder-decoder architectures using recurrence were composed by an encoder and a decoder, in which the encoder model is transforming an input sequence $(x_1, \ldots, x_n)$ step by step, e.g. word by word, into a vector h_n^e of fixed size to create a representation of the complete input sequence. Formally, such an encoder can be defined as follows:

$$\text{Encoder}_n(x_n, h_{n-1}^e) = g(h_{n-1}^e, x_n) = h_n^e \tag{1}$$

Here the function g, can, for example, be a LSTM cell computing the hidden state of the encoder given the previous hidden state and the current input. The decoder then generates an output sequence $y = (y_1, \ldots, y_m)$ based on the last hidden representation created by the encoder h_n^e, the previous output y_{m-1} and the last representation created by the decoder h_{m-1}^d. More formally:

$$\text{Decoder}_m(y_{m-1}, h_{m-1}^d, c) = f(g(h_{m-1}^d, y_{m-1}, c)) = y_m, \tag{2}$$

where, f is a nonlinear function, e.g. a probability distribution over the set of possible outputs (Jurafsky and Martin, 2024). The context vector c is fixed to the last hidden state h_n^e created by the encoder, i.e. $c = h_n^e$.

One problem with this approach is that the meaning of the entire input sequence needs to be compressed into this final representation h_n^e created by the encoder, and

research found that giving the decoder only access to this single vector representation of an input sequence acts as a bottleneck. To solve this issue, Bahdanau et al. (2014) proposed an extension to the encoder-decoder architecture: the *attention mechanism*, in which the decoder is given access to information from all intermediate representations, i.e. the hidden states generated by the encoder at each step when transforming an input sequence (Luong et al., 2015). Here, the intuition is that instead of only using this static final representation by the encoder, the decoder can learn which hidden states created by the encoder are important for decoding at the current step. Therefore, the context vector c is not fixed throughout decoding but newly generated in each decoding step m (denoted as c_m). This can be done by learning a distribution α_{mn}, which is weighting importance of each hidden state $(h_1^e, \ldots, h_n^e)$ generated by the encoder in each decoding step. Formally this weighting can be written as follows:

$$c_m = \sum_n \alpha_{mn} h_n^e \tag{3}$$

One way of computing this weighting is called *dot-product attention*. Here, first similarity scores between the current decoder state and the encoder hidden states are calculated (Jurafsky and Martin, 2024). Then, these scores can be used to calculate the attention weight as follows:

$$\alpha_{mn} = \frac{\exp(\text{score}(h_m^d, h_n^e))}{\sum_k \exp(\text{score}(h_m^d, h_k^e))}, \quad \text{with score}(x, y) = x^\top y \tag{4}$$

Here, similarity scores between the current decoder hidden state h_m and the different encoder hidden states are calculated, which are then normalized using the softmax function to obtain a weight distribution over all encoder states.

Figure 1 (Bahdanau et al., 2014) illustrates this idea and shows applications of such an attention mechanism in a model which is translating from the English to the French language. Here, each pixel shows the weight α_{mn} of the n-th state of the encoder h_n^e for the m-th state of the decoder h_m^d. In the example on the left, the word "European" needs to be translated to "européenne". However, as often in translation, the words in the input and target sentences are not aligned, i.e. their

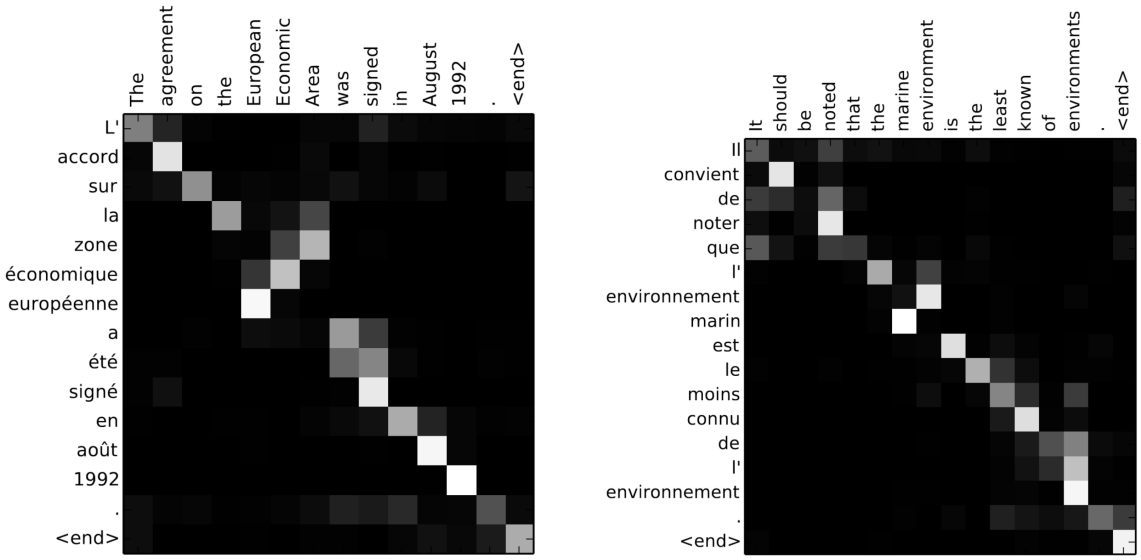

Figure 1: Visualization of an attention mechanism. Each pixel showing values for the learned weight α_{mn} in grayscale, when translating an English sentence to French. Here, a black pixel corresponds to a value of 0, while a white pixel corresponds to a value of 1. Figure taken from Bahdanau et al. (2014).

position in the sentence differs. With this mechanism, while decoding, i.e. generating a translation word by word, the model is correctly "paying attention" to the correct hidden state and can therefore correctly translate the sentence. Note that they use a slightly different approach for calculating α_{mn}, which is omitted in this overview.

2.2.2 The Transformer Architecture

One limitation to this type of attention is that only the decoder is able to use such a mechanism. Therefore, only relations between the input and an output can be learned, while relationships between words in the input sequence itself are neglected. However, if we want models to be able to contextualize the meaning of words within a single sequence, e.g. when creating word embeddings, a different type of attention is needed. To overcome this issue, an attention mechanism called *self-attention* was proposed, which can learn relationships in a single sequence (Parikh et al., 2016; Lin et al., 2017). The idea of self-attention is similar to attention as previously described:

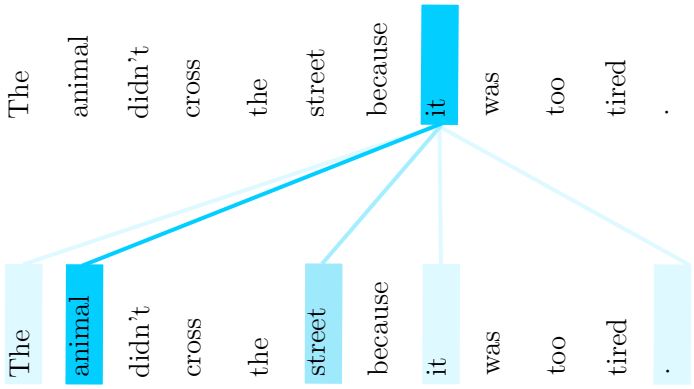

Figure 2: Visualization of the weight distribution of a self-attention layer in a transformer network. The representation for the word *it* is majorly drawn from the representation of the word *animal*. Drawn after Uszkoreit (2017).

we want to weight and combine representations from the context to calculate current representations. However, self-attention is different in sense that here, each word is calculating similarities between itself and the other words in the same sequence to create a meaningful representation for *itself*, instead of relating words in the output sequence to words in the input sequence.

Figure 2 (Uszkoreit, 2017) illustrates of such a self-attention mechanism for the word "it". Generating a meaningful representation for this word is difficult as, depending on the context, *it* can refer to many things. In this figure, lighter colors represent less attention, while darker colors represent more attention. This visualization shows that when creating a representation for the word "it", in this context, the word "animal" has the highest attention weight. Therefore, the resulting representation of the word "it" will heavily draw from the representation of the word "animal" (Jurafsky and Martin, 2024). This makes sense as the word "it" is indeed referring to the word "animal" in this context. Note that this is a simplified illustration of actual attention weight a transformer-based translation model learned automatically from training as presented by Uszkoreit (2017).

The transformer architecture builds on this idea of self-attention. The general architecture of this type of network is shown in Figure 3 (Vaswani et al., 2017). It consists of multiple stacked identical encoder and decoder blocks, while each encoder

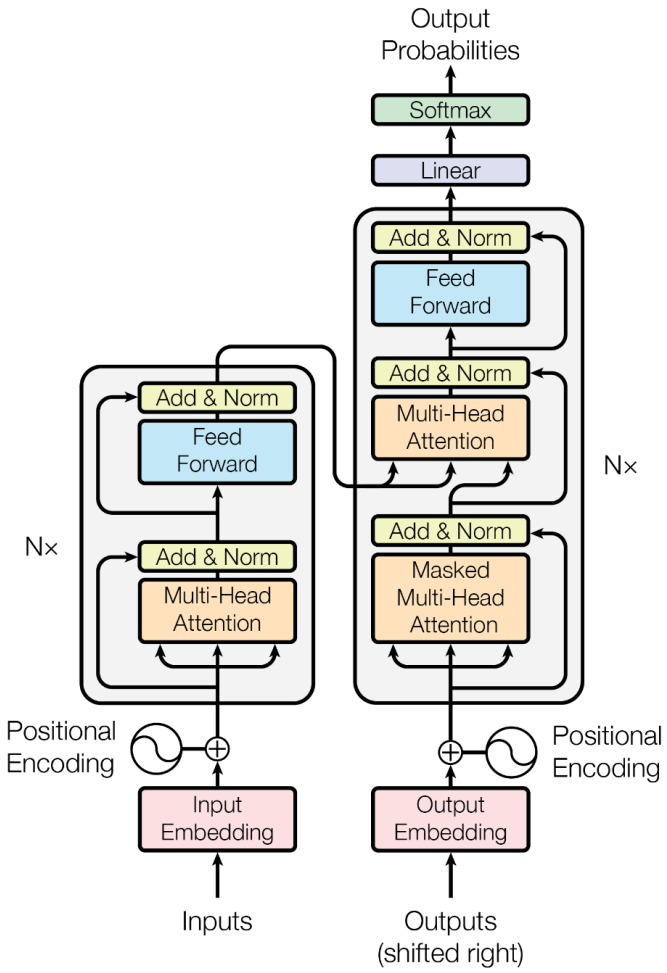

Figure 3: The transformer architecture. (Vaswani et al., 2017)

block (left-hand side of Figure 3) is composed of a *multi-head self-attention* layer and a fully connected feed-forward network with residual connections (He et al., 2015) and a normalization layer (Ba et al., 2016). Each decoder block (right-hand side of Figure 3) is similarly composed but adds another attention layer incorporating the output of the encoder. Here, the intuition behind stacking transformer blocks, also called layers, is that in each block a better representation for each word is created.

Vaswani et al. (2017) call their attention method *scaled dot-product attention*, which is similar to the attention weight calculation of Equation 4 but adds a scaling factor. Here, each input to an attention layer is first linearly projected to query q_i and

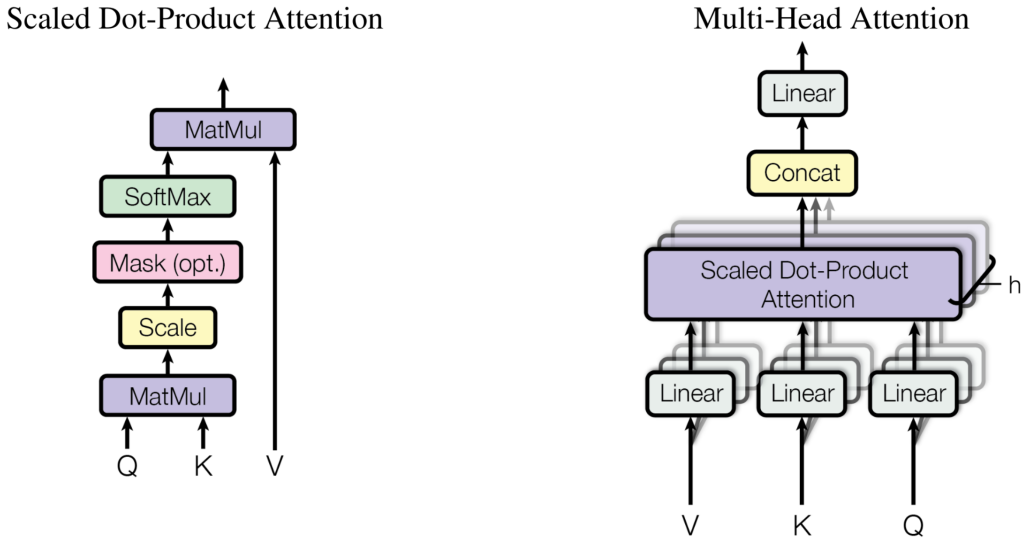

Figure 4: Visualization of the attention mechanism used in the original transformer architecture. (Vaswani et al., 2017)

key k_i vectors of dimension d_{key}, and a value vector v_i. To do this, the transformer introduces weight matrices W^Q, W^K and W^V, which are learned during training. The terms query, key and value refer to the fact that words in self-attention can have different roles: A word can either be the word we are searching a representation for (the query), or a word we are comparing this query to (a key). The value representations of the words are then used in combination with the calculated weight from comparing a query to the keys to obtain the final representation:

$$\text{attention}(q_n, k_n, v_n) = \text{softmax}(\frac{q_n \cdot k_m^\top}{\sqrt{d_{key}}})v_n \tag{5}$$

Note that here, each input word is represented using a vector representation. These representations of each word in a sequence can be packed together into a matrix representing the entire input sequence. Again, this resulting matrix can then be projected using W^Q, W^K and W^V resulting in the matrices Q, K and V, enabling the attention computation for all queries in a sequence at the same time:

$$\text{attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_{key}}})V \tag{6}$$

Figure 4 shows the computation graph of this approach. Note that the transformer model as proposed by Vaswani et al. (2017) modifies the attention layer in the decoder blocks such that subsequent keys of a query are hidden, by masking out keys of future word. This is essential for sequence-to-sequence tasks or causal language modeling, i.e. predicting a word given previous ones, as it would be trivial for a model to predict the next word if the next word is already given.

In addition, Vaswani et al. (2017) found that projecting the input to an attention layer multiple times using different learned projections is beneficial. They call this approach *multi-head attention*, in which several self-attention *heads* with its individual projection matrices W_i^Q, W_i^K and W_i^V, exist. Each of these heads projects the input to individual query, key and value matrices Q_i, K_i and V_i and performs attention as described above. The output of these heads are then concatenated and multiplied with a learned linear projection matrix W^O:

$$\begin{aligned} &\text{multi-head attention} = \text{concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \\ &\text{with head}_i = \text{attention}(Q_i, K_i, V_i) \end{aligned} \tag{7}$$

This can be seen as a way for the model to simultaneously pay attention to different aspects of relationships in a sequence (Jurafsky and Martin, 2024). One attention head, could for example pay attention to syntactic relations while another is observing semantic relationships.

Intuitively, the architecture builds representations of the meaning of words automatically, and refines this meaning for the current context throughout its self-attention layers. The last step in this model then uses these refined representations to predict next-word probabilities. This is called the *language model head* (depicted in dark gray and green in Figure 3). This language model head typically consists of a learned linear layer called *unembedding layer* and a softmax layer, mapping the representation of the last word (or token) in the decoder part of the architecture to a probability distribution over words in its vocabulary. From this probability distribution can then a word be drawn.

2.3 Large Language Models (LLMs)

Emerging from their original application as encoders and decoders solving sequence-to-sequence task, transformer based language models have evolved from creating contextualized word representations and modeling probability distributions to being able to solve complex tasks. One key factor to this success is the efficiency and scalability of the transformer architecture. Today, language models based on the transformer architecture can roughly be divided into three categories: the encoder-decoder language models (e.g. BART (Lewis et al., 2019) or T5 (Raffel et al., 2023)), which mostly use the transformer architecture with the encoder and decoder as proposed by Vaswani et al. (2017) for sequence-to-sequence tasks, the decoder-only language models (e.g. the GPT-series (Radford et al., 2018, 2019; Brown et al., 2020)), which only use the right-hand side of Figure 3, and the encoder-only language models (e.g. BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019)).

Encoder-only Models. As the name suggests encoder-only models are restricted to the encoder part of the transformer architecture. One prominent example for this type of model is BERT (bidirectional encoder representations from transformers) as proposed by Devlin et al. (2019). In contrast to decoder-only architectures, in which context is typically limited to previous words (masked self-attention), the BERT model used a bidirectional self-attention approach when creating word representations in which previous and succeeding words are used. This means that the model is able to attend to the next words in a sequence when deriving representations. Consequently, this type of models can not be pre-trained using the typical next word predictions task of many language models. Therefore, Devlin et al. (2019) introduce new language modeling tasks like the *masked language model* pre-training objective. Here, words in the input are masked randomly while the task of the model is to predict these masked words based on the past and future context, i.e. the words surrounding the masked word. Such encoder-only architectures like BERT and its successors (e.g. RoBERTa, DeBERTa (He et al., 2021, 2023)) are commonly used with the pre-train then fine-tune paradigm. Here, such models are pre-trained once, e.g. using the masked language model objective, and then further trained with a

task specific prediction head (Wolf et al., 2020) using labeled data. Such a prediction head can, for example, be a feed-forward classification layer, e.g. for sequence classification, added top of the outputs of the pre-trained model.

Decoder-only Models. Likewise, numerous generative language models based on the decoder-only architecture have been proposed since the introduction of the transformer. Here, models use some unidirectional attention masks, such that a word is only able to attend to itself and past words (or tokens) (Zhao et al., 2023). Decoder-only models are mostly trained using some causal language modeling objective, in which the task of the model is to predict the next word given a sequence of previous words. More specifically, such models do not directly predict words but probabilities for words being the next in a given sequence, i.e. given a sequence $(x_1, \ldots, x_n)$ these language models predict probabilities $P(w|x_1 \ldots x_n)$ for all words w in their vocabulary V. When generating words using such models, the next word is chosen from this probability distribution over a vocabulary by sampling. One way of sampling is to sample the next word from the predicted probability distribution, which is called random sampling. After sampling a word w_1, this word is added to the sequence and the model predicts probabilities for $P(w_2|x_1 \ldots x_n w_1)$. Typically, this is repeatedly done until a special end-of-sequence token is predicted (or sampled). Modern large language models typically use top-p (or nucleus) sampling (Holtzman et al., 2020) instead of random sampling. Here, the predicted probability distribution over words in the dictionary is restricted to the top-p percent of the probability mass, i.e. the vocabulary $V^p \subset V$ we sample from is limited to the smallest set such that $\sum_{w \in V^p} P(w|x_1 \ldots x_n) \leq p$. Here, top-$p$ is a hyperparameter that can be adjusted when generating words using such a model.

Another hyperparameter, which such models typically use is a *temperature* (Ackley et al., 1985). Here, before applying softmax, the output of the linear layer in the language model head is divided by a fixed value τ. If $0 < \tau < 1$ probability of words predicted to be the next word given the input sequence with high probabilities will be further increased, while probability mass of words with a low probability will be reduced; vice versa for $\tau > 1$. Intuitively, a high temperature will encourage diversity when generating new words, while a low temperature will reduce randomness.

```
Classify the sentiment of the sentence as positive or negative.
Sentence: The customer service was terrible. Sentiment:
```

Listing 1: Example of a zero-shot learning setup for sentiment analysis.

2.3.1 Prompting

Early research using such decoder-only networks like GPT-1 (Radford et al., 2018), also proposed to fine-tune such pre-trained models using task specific data and architecture adaptations. However, research found that many natural language processing tasks can be transformed into word prediction tasks, and since these generative models are (pre-) trained on predicting the next word in a sentence, these models can directly be used to solve such task by simply predicting words. For example, the classification task whether the sentence "The customer service was terrible" expresses a *positive* or *negative* sentiment, can be transformed to a word prediction task by constructing an input to the model containing this sentence, asking the model to predict the next word given this input, and then interpreting the output of the model. This has the advantage that the model can be used without any architectural adaptions and training. An example for such an input to a model is exemplarily given in Listing 1. Here, instead of fine-tuning or training a model for sentiment analysis, the task the model is supposed to solve is verbalized ("Classify the sentiment of the sentence as positive or negative") and given to the model in combination with the instance the model is asked to perform this task on. One could then interpret the words the model is generating given this input. For example, if the output of the model is "negative", one would classify the input sentence "The customer service was terrible" as *negative*. Note that in this example, one could also check whether the model predicts the next word being "positive" or the word "negative" to have a higher probability given this sequence as input instead of interpreting the output of the model directly.

Using a language model in such a way while simply using the output by the model is nowadays referred to as the *prompting* paradigm. Here, we are giving the model an instruction and a context, i.e. a prompt, and the model solely relies on

```
Classify the sentiment of the sentence as positive or negative.
Sentence: I love my new phone. Sentiment: positive
Sentence: The movie was a complete disaster. Sentiment: negative
Sentence: The customer service was terrible. Sentiment:
```

Listing 2: Example of few-shot learning for sentiment analysis. A task description and examples how to solve the task is given to the language model.

its pre-trained knowledge and language understanding to solve tasks. Note that this approach of using a pre-trained model is very different to supervised fine-tuning as the model is no longer further trained to a specific downstream task by using gradient updates and therefore does not require any labeled training data.

Few-shot Prompting. Moreover, research found that if such generative models are scaled up, e.g. by increasing the models size or training on more data, such models start to show complex task solving abilities which could not been forecast when extrapolating performance of smaller models. Literature refers to such capabilities as *emergent abilities* (Wei et al., 2022b). Such an ability is, for example, the ability to benefit from *in-context learning* or *few-shot prompting* as proposed by Brown et al. (2020). Here, the model is given demonstrations of correct input-output pairs in addition to the verbalized task description and the input the model is supposed to perform this task on. Listing 2 extends the example presented in Listing 1 with such demonstrations. In comparison to the first example, i.e. the *zero-shot prompt*, this adds demonstrations of sentences we would like the model to predict a positive and a negative sentiment. For example, for the sentence "I love my new phone", this prompt aims to demonstrate that this sentence expresses a *positive* sentiment, while "The movie was a complete disaster" expresses a *negative* sentiment.

2.3.2 Adapting Large Language Models

In recent years, various of such large scale generative language models have been trained (e.g. GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), FLAN (Wei et al., 2022a), LLaMA (Touvron et al., 2023a,b)) incorporating several en-

hancements to such models. Among others, such improvements include adjusting the pre-training, e.g. by using different or additional pre-training data, using different pre-training objectives or using techniques to make pre-training more stable and efficient, adjusting the model architecture, e.g. by increasing the model size, applying different activation function, exploring various positional embeddings or adjusting the attention mechanism.

Instruction Tuning. Another phenomenon that was found to emerge when models reach a certain scale is the ability to benefit from *instruction tuning* (Wei et al., 2022a; Sanh et al., 2022). Here, a model is fine-tuned on several language tasks (e.g. sentiment analysis, summarization, translation, question answering) phrased as natural language instruction. This is similar to the pre-train then fine-tune paradigm in sense that models are further trained using gradient descent. However, the major difference is that with instruction tuning, description of the tasks the model is supposed to solve are added to the training data. In sentiment analysis, for example, a training sample for traditional supervised fine-tuning like "The movie was great!" could be transformed to "Is the sentiment of the sentence 'The movie was great!' positive or negative?" for instruction fine-tuning. Intuitively, this can be seen as a training method in which a model is not only trained to be able to perform well for certain tasks but is also trained to understand what the actual task is, i.e. to understand and interpret instructions correctly. As shown by various research (Wei et al., 2022a; Ouyang et al., 2022; Sanh et al., 2022), this can substantially improve the zero- and few-shot prompting performance of models on *unseen tasks*, i.e. task not seen during this fine-tuning process. Moreover, this process also addresses the mismatch between the pre-training objective of predicting the next word and the objective of intelligent systems performing tasks based on natural language instructions given by a user (Adlakha et al., 2023) making instruction tuning a crucial technique to increase capabilities of pre-trained LLMs.

Alignment/Preference Tuning. Since LLMs are pre-trained on massive amounts of textual data, including low quality or biased data, such models may generate biased, toxic, misleading, or even harmful content (Zhao et al., 2023). To mitigate such behavior research often tries to align such models with human values to make them

helpful, *honest*, and *harmless* (Askell et al., 2021). However, adjusting models to fit human values algorithmically is a challenging task, since such values are difficult to measure and quantify in a supervised fine-tuning approach. In contrast, it is easy for humans to judge whether some output of a LLMs aligns with human preferences. Therefore, a prominent strategy is to directly include such feedback from humans in the alignment process of LLMs. This approach is called *reinforcement learning from human feedback (RLHF)* (Christiano et al., 2017; Ziegler et al., 2019).

RLHF pipelines for aligning LLMs to human preferences as, for example, done by Ouyang et al. (2022) in InstructGPT or by Touvron et al. (2023b) in Llama 2, are typically composed of three components: the pre-trained LLM, a *reward model* and a *reinforcement learning algorithm*. Here, as a first step, a pre-trained model is typically fine-tuned using instruction tuning as described above. Afterward, prompts that cover a variety of tasks and possibly capture aspects of human values are collected. Using these prompts the pre-trained LLM is then generating textual outputs. Note that such prompts may be specifically designed to trick the model into generating output which is not in line with human values. These model outputs are then rated by human annotators. For example, assigning high scores for *helpful* responses and low scores for *harmful* ones. Then, typically the next step in the RLHF pipeline is to train a reward model using these human annotations. This reward model is learning to replicate the behavior of the human annotators to automate the human preference evaluation of generated text by a LLM. For example, given a textual input, the goal of this reward model is to predict a value representing the human preference. The LLM, that needs to be aligned, is then iteratively generating textual outputs from prompts, which are rated by the reward model. Given these ratings, the LLM is then fine-tuned with a policy-gradient RL algorithm to maximize helpfulness scores. Note that the next section will give an introduction to the reinforcement learning paradigm in general.

2.4 Reinforcement Learning

Reinforcement learning (RL) is a paradigm involving learning to behave optimally by interacting with an *environment* using feedback in the form of rewards or punishments. Here, the idea is that an *agent* is performing an *action* depending on the current *state* and is learning to optimize these actions based on reward signals given by the environment. In contrast to supervised learning, in which a system learns from pre-defined set of input-output pairs, RL solely learns through such interacting with an environment (e.g. through trial-and-error). This makes RL preferable over supervised learning for tasks in which the desired decision-making strategy is not well-defined, for tasks that change dynamically over time, or task, which would require a large amount of human annotation effort. RL can be applied in numerous domains like, for example, robotics, automated game playing (e.g. playing the board game Go (Silver et al., 2016)), or autonomous driving (e.g. Wang et al. (2023)). Furthermore, RL has been applied to various NLP tasks like text classification (Zhang et al., 2018), machine translation (Bahdanau et al., 2017), text summarization (Stiennon et al., 2022), or for aligning large language models to human preferences as discussed in the previous section. Zhang et al. (2018), for example, learn an agent, which is learning to automatically delete single words in an input sequence to improve performance of subjectivity analysis and topic classification systems.

A fundamental concept in RL is the *Markov decision process* (MDP) (Howard, 1960), which formalizes the interaction between an agent and an environment mathematically. A MDP for RL can be characterized by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where:

- $\mathcal{S}$ is a set of states,
- $\mathcal{A}$ is a set of actions,
- $\mathcal{T}$ are the transition dynamics (satisfying the Markov property), mapping an action $a \in \mathcal{A}$ and state $s \in \mathcal{S}$ at time step t to a probability of transitioning to state $s' \in \mathcal{S}$ at time step $t+1$, i.e. $\mathcal{T}(a, s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$,
- $\mathcal{R}$ is a reward function, mapping a transition from a state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$ to state $s' \in \mathcal{S}$ to a reward $r \in \mathbb{R}$,
- $\gamma \in [0, 1]$ is a discount factor to balance short-term and long-term rewards.

2.4.1 Typical Components of Reinforcement Learning

A reinforcement learning system usually consist of an agent, an environment, a *policy*, a *reward signal*, a *value function*, and optionally, a *model* of the environment (Sutton and Barto, 2018). The *policy* $\pi(a|s)$ determines the behavior of an agent, i.e. it determines an action $a \in \mathcal{A}$ given the current state $s \in \mathcal{S}$. Note that such a policy can be stochastic; mapping a state to a probability distribution over possible actions. After an action is chosen by the agent, a *reward signal* r_{t+1} and a new state s_{t+1} is given to the agent by the environment. The next state is determined by the transition dynamics $\mathcal{T}$ of the environment given the current state and the action chosen by the agent.

The *model* of the environment represents the behavior of the environment, e.g. transition dynamics or reward signals. The two main approaches withing reinforcement learning are *model-based* and *model-free* methods. In model-based methods the agent is either given access to the dynamics of the environment or is learning to approximate them. Such a model might, for example, predict the next state and reward given the current state and an action (Sutton and Barto, 2018). This enables the agent to simulate future states and rewards without explicitly experiencing them, and to plan actions accordingly. In model-free methods the agent learns without building an explicit model of the environment. Here, the agent is solely learning through trial-and-error by interacting with the environment.

An interaction between an agent and an environment, i.e. the MDP, can be episodic or non-episodic. In an episodic MDP, the agent and environment interact for a finite amount of time steps T, called an *episode*, after which the state is reset to an initial state. An episodic interaction creates a sequence of states, actions and rewards, called *trajectory*, i.e. $s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_T$. In general, we want the agent to behave optimally according to the reward signals received by the environment throughout such an interaction cycle. Therefore, the goal of an agent is to maximize the total amount of reward it receives, i.e. not the immediate reward, but the accumulated reward along such an interaction. With such a trajectory this goal can be formalized as the *return* G_t, which denotes the accumulated future reward from

time step t:

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{8}$$

Note that here, the parameter $\gamma \in [0, 1]$ is the *discount rate*. In case of $\gamma < 1$, for example, rewards close to the current time step are more valued, compared to rewards that appear far in the future.

Another component of a typical RL system is the *value function*. This component helps an agent to predict how rewarding it is to be in a particular state or how rewarding it is to perform a certain action in a particular state given the current policy. More precisely, the *state-value function* $V_\pi(s)$ is the *expected return* for a policy π if starting from the state s (Shakya et al., 2023), which can be expressed as follows (Sutton and Barto, 2018):

$$V_\pi(s) = \mathbb{E}_\pi[G_t | s_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s \right], \forall s \in \mathcal{S} \tag{9}$$

Note that here an expected return is calculated since the policy π and future state transitions might be stochastic. Similarly, an *action-value function* $Q_\pi(s, a)$ can be defined (Sutton and Barto, 2018):

$$Q_\pi(s, a) = \mathbb{E}_\pi[G_t | s_t = s, a_t = a] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s, a_t = a \right] \tag{10}$$

Here, instead of estimating a "goodness" value of a state and policy, the value of performing an action a in state s and then following the policy π is calculated.

The values for $V_\pi(s)$ and $Q_\pi(s, a)$ can, for example, be estimated by the average return observed when the agent visits a state s throughout many complete episodic interaction with the environment (Monte Carlo methods). With these value functions different policies can be compared: a policy π is better or equal to a policy π' if and only if $V_\pi(s)$ is greater $V_{\pi'}(a)$ for all $s \in \mathcal{S}$ (Sutton and Barto, 2018). Moreover, finding the best policy π^* solves the reinforcement learning problem for episodic tasks as an agent following this policy achieves the highest expected return $\mathbb{E}(G_t)$.

Therefore, many approaches to the reinforcement learning problem are centered around finding the best value functions, i.e. finding the best state-value function

$$V_{\pi^*}(s) = \max_{\pi} V_{\pi}(s), \tag{11}$$

or action-value function:

$$Q_{\pi^*}(s, a) = \max_{\pi} Q_{\pi}(s, a) \tag{12}$$

2.4.2 Policy Gradient Methods

In contrast to methods that use value functions to estimate optimal actions, *policy gradient* approaches directly parameterize and optimize the policy itself (Sutton et al., 1999). Here, the policy is a parameterized function $\pi(a|s, \theta)$, modeling the probability of an action a given state s and parameter θ. Often, this policy is parameterized by assigning numerical preferences $h(s, a, \theta) \in \mathbb{R}$ for each state-action pair. The function h can be any differentiable function like, for example, an artificial neural network parameterized by θ. Here, actions with the highest preferences in each state are given the highest probability of being chosen, e.g. by a softmax distribution (Sutton and Barto, 2018):

$$\pi(a|s, \theta) = \frac{\exp(h(s, a, \theta))}{\sum_{a'} \exp(h(s, a', \theta))} \tag{13}$$

Similarly to value function based methods, the objective of policy gradient methods is to find a policy that maximizes the expected return. However, here the idea is to update the parameter θ such that the probability of actions leading to a higher expected return are increased and the probability of actions that lead to a lower return are decreased. This is done by computing the gradient of some performance measure J with respect to the policy parameter θ and performing gradient ascent:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta), \tag{14}$$

where $\alpha \in \mathbb{R}^+$ denotes a learning rate and $\nabla_{\theta} J(\theta)$ is the derivative of the performance measure J with respect to the policy parameter θ. Here, the objective

is to find optimal parameter θ^*, such that the performance measure is maximized (Bolland et al., 2023):

$$\theta^* = \arg\max_{\theta} J(\theta) \tag{15}$$

In episodic environments the value function of the initial state is often used to measure the performance of a parameterized policy π_θ, i.e:

$$J(\theta) = V_{\pi_\theta}(s_0) = \sum_{s \in \mathcal{S}} \mu_{\pi_\theta}(s) V_{\pi_\theta}(s), \tag{16}$$

where μ_{π_θ} is the on-policy distribution under π_θ (Weng, 2018) and s_0 the initial state of the episode. Here, the performance depends on this state distribution and the selected actions while both are affected by the policy parameter θ. However, since the state distribution is typically unknown it is difficult to estimate the gradient of this performance measure (Shakya et al., 2023). To solve this issue, the *policy gradient theorem* provides an analytic expression for calculating the gradient of $V_{\pi_\theta}(s_0)$ without needing to calculate the derivative of the distribution (Sutton and Barto, 2018):

$$\begin{aligned} \nabla_\theta J(\theta) = \nabla_\theta V_{\pi_\theta}(s_o) &= \nabla_\theta \sum_{s \in \mathcal{S}} \mu_{\pi_\theta}(s) \sum_{a \in \mathcal{A}} Q_{\pi_\theta}(s,a) \nabla_\theta \pi(a|s,\theta) \\ &\propto \sum_{s \in \mathcal{S}} \mu_{\pi_\theta}(s) \sum_{a \in \mathcal{A}} Q_{\pi_\theta}(s,a) \nabla_\theta \pi(a|s,\theta), \end{aligned} \tag{17}$$

where the constant of proportionality is equal to the average length of an episode (Weng, 2018). Moreover, proportionality in this equation is a sufficient estimator when calculating the gradient in this case since this proportionality gets absorbed by the learning rate α in Equation 14 (Sutton and Barto, 2018). Here, the proportionality is the result of the *policy gradient theorem*, which enables to calculate the gradient of $J(\theta)$ without calculating the gradient of the state distribution μ_{π_θ}. Note that for a proof of this theorem and for a more in-depth explanation of the background of reinforcement learning in general, I direct the interested reader to Sutton and Barto (2018)'s book "Reinforcement Learning: An Introduction".

2.4.3 The REINFORCE Algorithm

The policy gradient theorem is the foundation for various policy gradient algorithms. Following Sutton and Barto (2018) the gradient can be further written as:

$$\begin{aligned}
\nabla_\theta J(\theta) &\propto \sum_{s\in\mathcal{S}} \mu_{\pi_\theta}(s) \sum_{a\in\mathcal{A}} Q_{\pi_\theta}(s,a)\nabla_\theta \pi(a|s,\theta) \\
&= \mathbb{E}_{\pi_\theta}\left[\sum_{a\in\mathcal{A}} Q_{\pi_\theta}(s_t,a)\nabla_\theta \pi(a|s_t,\theta)\right] \\
&= \mathbb{E}_{\pi_\theta}\left[\sum_{a\in\mathcal{A}} \pi(a|s_t,\theta) Q_{\pi_\theta}(s_t,a)\frac{\nabla_\theta \pi(a|s_t,\theta)}{\pi(a|s_t,\theta)}\right] \\
&= \mathbb{E}_{\pi_\theta}\left[Q_{\pi_\theta}(s_t,a_t)\frac{\nabla_\theta \pi(a_t|s_t,\theta)}{\pi(a_t|s_t,\theta)}\right] \qquad \text{(replacing } a \text{ by the sample } a_t\text{)} \\
&= \mathbb{E}_{\pi_\theta}\left[G_t\frac{\nabla_\theta \pi(a_t|s_t,\theta)}{\pi(a_t|s_t,\theta)}\right], \qquad \text{(because } \mathbb{E}_{\pi_\theta}[G_t|s_t,a_t] = Q_{\pi_\theta}(s_t,a_t)\text{)}
\end{aligned}$$

where G_t is the return as defined in equation 8. This yields the update rule for the REINFORCE algorithm (Williams, 1992):

$$\begin{aligned}
\theta_{t+1} &= \theta_t + \alpha G_t \frac{\nabla_\theta \pi(a_t|s_t,\theta_t)}{\pi(a_t|s_t,\theta_t)} \\
&= \theta_t + \alpha G_t \ln \nabla_\theta \pi(a_t|s_t,\theta_t). \qquad (18)
\end{aligned}$$

Intuitively, given a state s_t, this update rule is increasing the probability of an action a_t that yields a positive return. Similarly, this rule decreases the probability of actions that yield a negative return. Note that this update rule still works if returns are always positive as higher returns means increasing the probability of an action "more" compared to lower returns and increasing the probability of an action in one state simultaneously means decreasing the probability of another action in the same state. Further, the gradient of the probability of taking an action is divided by the probability of this action. This is important as otherwise action which are taken more often given the current policy and yield a "good" return are favored in sense that their probability is increased more frequently, compared to actions that yield an even better return but are taken less frequently given the current policy.

Algorithm 1 REINFORCE: Monte-Carlo Policy-Gradient Control (no discounting)

Input: Differentiable policy parameterization $\pi(a|s, \theta)$, learning rate $\alpha > 0$
Initialize policy parameter θ
for $1, \ldots, N$ **do**
    Generate an episode $s_0, a_0, r_1, \ldots, s_{T-1}, a_{T-1}, r_T$, following $\pi(\cdot|\cdot, \theta)$
    for state t in episode $t = 0, 1, \ldots, T-1$ **do**
        $G_t \leftarrow \sum_{k=t+1}^{T} r_k$
        $\theta \leftarrow \theta + \alpha G_t \nabla_\theta \ln \pi(a_t|s_t, \theta)$
    end for
end for

The REINFORCE algorithm as given by Algorithm 1 is one of the first policy gradient algorithms. Here, the general idea is to iteratively sample trajectories using Monte Carlo simulations, i.e. generating sample trajectories of complete episodes following the current policy π_θ, which are then used to update the policy parameter. Here, sampling from the stochastic policy ensures exploration creating different trajectories with different returns. The first step of the algorithm is to initialize the policy parameter θ (e.g. with random values). Afterward, an episode is sampled using the current policy and the parameter θ is updated for each taken action.

Although convergence of this algorithm has been proven, in practice, a problem that occurs is that the variance of the gradient estimator can be very large leading to very slow learning (Weaver and Tao, 2001). Therefore, a variation of the REINFORCE algorithm is to subtract a baseline from the return G_t to reduce the variance, while keeping bias unchanged (Weng, 2018) like follows:

$$\theta_{t+1} = \theta_t + \alpha(G_t - b(s)) \ln \nabla_\theta \pi(a_t|s_t, \theta_t). \tag{19}$$

In general, the baseline $b(s)$ can be an any function that does not depend on taken actions. Here, one option is to use an estimation of the state value $V_\pi(s)$ as baseline (Sutton and Barto, 2018). Intuitively, this weights the (sampled) actual return of a trajectory by how "rewarding" it was expected it to be. Another option for a baseline is, for example, to use the average return in a batch of trajectories.

3 Author Profiling and Personality Computing

3.1 The Author Profiling Task

The task of author profiling (AP) aims to infer or extract characteristics about an author by analyzing the content they share. Author profiling is a task involving many interdisciplinary collaborations. For example, content shared by an author, i.e. data associated with a specific author, may be of textual form, requiring knowledge from the field of natural language processing, while content may also involve images or video data requiring techniques from computer vision. Further, characteristics of an author which are desired to be inferred from this content might have emerged from psychological or sociological studies requiring expertise these fields. The study presented by HaCohen-Kerner (2022) compares the number of publications in the years 2007 to 2020 associated with AP and shows that this task has gained a significant increase of interest in the year 2013. Likely, this is due to the first author profiling shared task hosted by PAN[3] in 2013 (Rangel et al., 2013). Since then, this shared task has been hosted every year with various forms of textual content of authors like blog posts (Rangel et al., 2013), hotel reviews (Rangel et al., 2014) or tweets (Stamatatos et al., 2015), and with a variety of characterization tasks like predicting age, gender or personality traits of authors. Here, the datasets provided by the hosts of these shared tasks, primarily consisted of content written in the English and Spanish language; occasionally adding Dutch, Italian, Portuguese and Arabic.

The approach proposed in this thesis restricts the type of content by an author to text. More specifically, this study restricts the type of textual content to a specific form of microblogs, i.e. tweets, written in the English language. In author profiling using textual content in form of tweets to infer author characteristics, each author is typically represented by a set of tweets and associated with a specific characteristic as a ground-truth. This enables the application of supervised machine learning methods. Here, the input to some machine learning model typically consists of the content shared by some author, while the model's objective is to learn to automati-

[3] `https://pan.webis.de/`

cally map the content of an author to specific characteristics. Therefore, progress in such author profiling task is related to developments of models and architectures in the field of supervised learning and machine learning in general.

In the last two decades several machine learning methods have been applied to the author profiling task. The best performing methods, in sense of achieved accuracy on the testing set, in the PAN shared task in the year 2016 (Busger op Vollenbroek et al., 2016), 2017 (Basile et al., 2017), and 2018 (Tellez et al., 2018), for example, use support vector machines with statistical features like word and character n-grams. In the shared task in 2019 (Rangel and Rosso, 2019), the objective was to differentiate between authors being either a bot or human. Here, Valencia-Valencia et al. (2019) proposed the system achieving the highest accuracy score using logistic regression with word and character n-grams, while also replacing emojis with a textual description. Deep learning methods have also been applied to the author profiling tasks. However, despite the success of deep learning strategies in various other NLP tasks, like sentiment analysis or machine translations, deep learning architectures often underperform compared to classical machine learning approaches when it comes to author profiling (Lopez-Santillan et al., 2023). The best results in the shared task hosted at PAN 2020 (Rangel et al., 2020), for example, in which the goal was to determine whether a given author is spreading fake news, were obtained by using a combination of support vector machines and logistic regression models by Bevendorff et al. (2020); despite competing against much more recent deep learning models like BERT. In the shared task 2021 (Bevendorff et al., 2021), profiling hate speech spreaders, the method proposed by Siino et al. (2021) barely achieves the best score by using a convolutional neural network in combination with pre-trained word embeddings. With their approach they achieve an accuracy of 79%, while the second-highest score of 78% is achieved by the method proposed by Balouchzahi et al. (2021), using classical machine learning methods using frequency based features and an ensemble of support vector machines, random forest and logistic regression classifiers, showing that such methods still produce competitive results.

More recent attempts to author profiling experiment with the multi-task learning paradigm (Caruana, 1997). The best performing method at the PoliticEs2022

shared task (García-Díaz et al., 2022) by Carrasco and Rosillo (2022), for example, fine-tunes a version of BERT to predict gender, profession and political preferences simultaneously. Here, the idea of using one model learning to predict all these traits at the same time is that it gives the model the ability to leverage all information learned from predicting individual traits; introducing a form of regularization to prevent overfitting. However, a limitation of BERT-based models is that the maximum number of input tokens is restricted. For author profiling tasks like this, this often means that the complete set of available texts can not be given to the model at once. One option is to train such models to infer information from individual text instances, while propagating the given ground-truth data of authors to their individual text instances. However, often individual instances will not be representative for the profile of an author and will therefore produce noisy or misleading instances (Chinea-Rios et al., 2022). To solve this issue, Carrasco and Rosillo (2022) concatenate tweets from the same author into blocks of at most 512 tokens to ensure each block can be handled by BERT-based models and classify each block individually. They then obtain the final prediction for each author by using a majority voting system considering all individually predicted blocks. Still, this is a mismatch, since the dynamics of the concatenation of individual text instances likely has influence on the model during training, which consequently also influences the generalization abilities of such models across all text instances associated with an author.

From a machine learning perspective, one difficulty which differentiates author profiling from traditional text categorizations or regression tasks is that the AP task spans over multiple, possibly independent text instances, while text categorization is often only concerned with a single coherent text instance. If trying to infer characteristics of an author from tweets, there might be hundreds or even thousands of individual text instances available from which a system needs to draw a single meaningful decision. In some sense, for author profiling, a model needs to be able to generalize very well across all these individual text instance to be able to deduct a profile information for a single author correctly. Therefore, this work argues that author profiling systems should explicitly learn to differentiate between important and misleading text instances by an author before inferring some characteristic. In

the author profiling task hosted at PAN 2022 (Ortega-Bueno et al., 2022), profiling irony and stereotype spreaders on the social media platform X, an approach with an idea closely related to this study was proposed. Here, Tahaei et al. (2022) fine-tuned SBERT (Reimers and Gurevych, 2019) to create vector representations for each tweet of an author. They then feed these vector representations into an additive attention layer calculating a vector representation for a complete author profile from which a classification head then deduces whether the author is keen to spread stereotypes or irony. Here, this attention layer is learning to weight individual tweets, i.e. the vector representations of individual tweets, such that the vector representation of the author contains necessary information to solve this task correctly. This is similar to the idea of having a mechanism that classifies whether a tweet is relevant or not for inferring a characteristic as proposed in this work, in sense that the attention layer is selecting relevant instances by assigning more weight to them, while rejecting instances by assigning less weight to them.

3.2 The Big Five Personality Traits

Different people react differently to various situations and challenges in their lives. Some people are talkative while others tend to be quiet. Some people like to always be well-prepared while others may tend to be disorganized. Such personality traits, which characterize differences between persons present over time and across situations, are studied by personality psychologists and several theories have been proposed in attempt to explain and categorize these differences (e.g., Cattell, 1945; Goldberg, 1981; McCrae and John, 1992).

Today, one of the most well researched and widely accepted framework for assessing personality are the *Big Five personality traits*. This framework describes personality along five dimensions: *openness to experience* (e.g. artistic, curious, imaginative), *conscientiousness* (e.g. efficient, organized, reliable), *extraversion* (e.g. active, outgoing, talkative), *agreeableness* (e.g. forgiving, generous, kind), *neuroticism* (e.g. anxious, unstable, worrying) (Costa and McCrae, 1992). These five dimensions originate from the lexical hypothesis, which states that personality characteristics are

I see Myself as Someone Who ...	Dimension	Scoring
... is original, comes up with new ideas	Openness	+
... prefers work that is routine	Openness	−
... does a thorough job	Conscientiousness	+
... can be somewhat careless	Conscientiousness	−
... is talkative	Extraversion	+
... is reserved	Extraversion	−
... is helpful and unselfish with others	Agreeableness	+
... tends to find fault with others	Agreeableness	−
... worries a lot	Neuroticism	+
... is relaxed or handles stress well	Neuroticism	−

Table 3: Example items from the BFI-44 questionnaire (John et al., 1991). A negative scoring indicates a reversed-scored item.

encoded in the language we use to describe human characteristics (Brewer, 2019; Goldberg, 1990; John et al., 1999) (e.g. adjectives like enthusiastic or energetic). Early work in this area collected such *descriptors* or traits that describe characteristic patterns of personality from dictionaries (e.g., Baumgarten, 1933; Allport and Odbert, 1936). Using factorial techniques, research then found that these natural-language descriptions or terms often describe similar concepts or characteristics which lead to the discovery of this five-factor model (e.g., Cattell, 1945; Goldberg, 1981; Costa and McCrae, 1992; McCrae and John, 1992). Note that these dimensions are typically not measured as binary values but as continuous distributions (Brewer, 2019). For example, a person may be characterized by a low or high score of *extraversion* along a normal distribution. One way to assess these five factors is the *Big Five Inventory* (BFI) developed by John et al. (1991), consisting of 44 short phrases. Typically, a person is then asked to rate themselves on a five-point Likert scale from 1 (strongly disagree) to 5 (strongly agree) along each phrase. Here, eight to ten of such phrases are used to evaluate one of the five "big" factors. Table 3 shows examples of such items for the five dimensions. Note that some items are evaluated in reverse, meaning that they contribute to the score in reverse when evaluating answers for a specific dimension. Here, for example, if a person strongly agrees to "being someone who is talkative" is indicating a high level of *extraversion*.

With growing amounts of user generated content available on online platforms and social media in the last two decades the idea of automatically detecting personality from texts emerged. Here, psycholinguistic and natural language processing approaches attempt to infer personality traits via written text for example using social media data (Stajner and Yenikent, 2020). One of the first attempts to automatically detect personality from text is the study by Argamon et al. (2005) predicting *extraversion* and *neuroticism* from essays. Here, they translate the task of predicting personality to a binary classification task by assigning a *high* level of a trait to participants associated with a score in the top third of each dimension, and a *low* level to participants with a score in the bottom third. Further, Schwartz et al. (2013) analyzed language use on the social media platform Facebook, from which each volunteer also submitted responses to a "Big Five" questionnaire. Here, they found that language use not only differs between people of different age and gender but also between people with different personality. In the shared task 2015 hosted at PAN (Rangel et al., 2015), the goal was to identify age, gender and personality traits of Twitter users across the languages English, Spanish, Italian and Dutch. Here, personality traits of individual users were given by continuous scores along the "Big Five" traits. Here, the best results predicting personality traits were obtained by Sulea and Dichiu (2015) using ridge regression in combination with term frequency–inverse document frequency (tf–idf) features.

Since then, various machine learning and more recently also including deep learning approaches have been applied in attempt to predict personality of users of social media platforms (Khan et al., 2020). However, these approaches face the similar difficulties to approaches predicting other author characteristics aside from personality like age or gender. Similarly to predicting other characteristic of authors, for predicting personality, this work argues that systems could benefit from learning to differentiate between relevant and misleading text instances by an author. Here, this work hypothesizes that not all tweets linked to individuals may be suitable for automatically inferring personality traits correctly. This is because content and tone of tweets from the same author may vary depending on factors such as mood, current events, or specific interest at a given time. Personality, however, characterizes

differences between persons present over time and across situations. Further, as not all traits are strongly related on each other (Oz, 2015), some tweets might provide insights for one trait but not for the other. For instance, a tweet may be indicative for inferring a level of *openness* but not for *conscientiousness*. In addition, removing irrelevant instances, not needed for predicting personality, can potentially solve practical limitations of machine learning models as these models typically have a limited capacity for processing large amounts of data. In addition, by filtering out unnecessary information, computational resources required for training and inference could be significantly reduced, leading to more efficient systems.

Likely, another reason why reconstructing personality from posts of social media users using automated systems remains challenging is that obtaining and validating ground-truth annotations for such users is difficult. One way to obtain such ground-truth data is to ask participants that "donate" their social media profile to research to complete a survey like the BFI-44 questionnaire. Personality measurements assessed through such surveys can be used as ground-truth that enables supervised learning. However, with such an approach, the quality of personality annotations can suffer from self-report biases like the social-desirability bias. Here, participants consciously or unconsciously may present themselves as aligning more closely with social norms than their true attitudes or behaviors (Olino and Klein, 2015). Another way of assessing personality of participants is to rely on experts annotations. However, such judgements also pose difficulties and biases. For example, experts may only have limited access to context like relationships or life events surrounding individual posts making it difficult to provide accurate judgements. Furthermore, people may behave differently on social media platforms compared to real life situations, and therefore, online behavior may not reflect their true personality. This makes ensuring accuracy and reliability of the personality assessments of social media users difficult, which consequently influences the ability machine learning models or automated systems in general.

4 Methodology

This chapter introduces the main contribution of this study. Here, this work proposes a component that is learning to select text instances from an author. In principle, learning such filter could be realized using supervised learning methods. However, this would require large scale human annotation efforts as it would require annotators to manually label sufficient amounts of examples from which supervised learning algorithms could learn. Doing this is difficult as there can be hundreds or even thousands of tweets associated with a single author. Therefore, this study proposes to learn such a selection indirectly via reinforcement learning. With this, the approach presented in this study circumvents the need for ground-truth data of which text instances associated with an author are relevant for predicting some characteristic and which instances can be discarded. In addition, this enables the proposed system to be adapted to other profiling tasks without any annotation efforts.

4.1 Overview

Given a profile, i.e. a set of text instances, of which we want to infer some characteristic, the goal of the approach proposed this study is to first select a subset of instances. Then, only this limited set of text instances is used to infer a label. The schematic representation of the proposed system is shown in Figure 5. This system consist of two primary components: (1) the ***Selection Network*** and (2) the ***Classification Network***. The *Selection Network* is selecting a set of instances, while the *Classification Network* is using a set of instances to predict a characteristic. During learning, the label predicted by the *Classification Network* and the ground-truth data are used to provide a reward signal from which the RL agent within the *Selection Network* learns to optimize its behavior.

The combination of these two components forms pipeline system that is predicting a characteristic from a given profile. I coin this pipeline the ***RL-Profiler*** system, and the following will explain the two components of this system in detail.

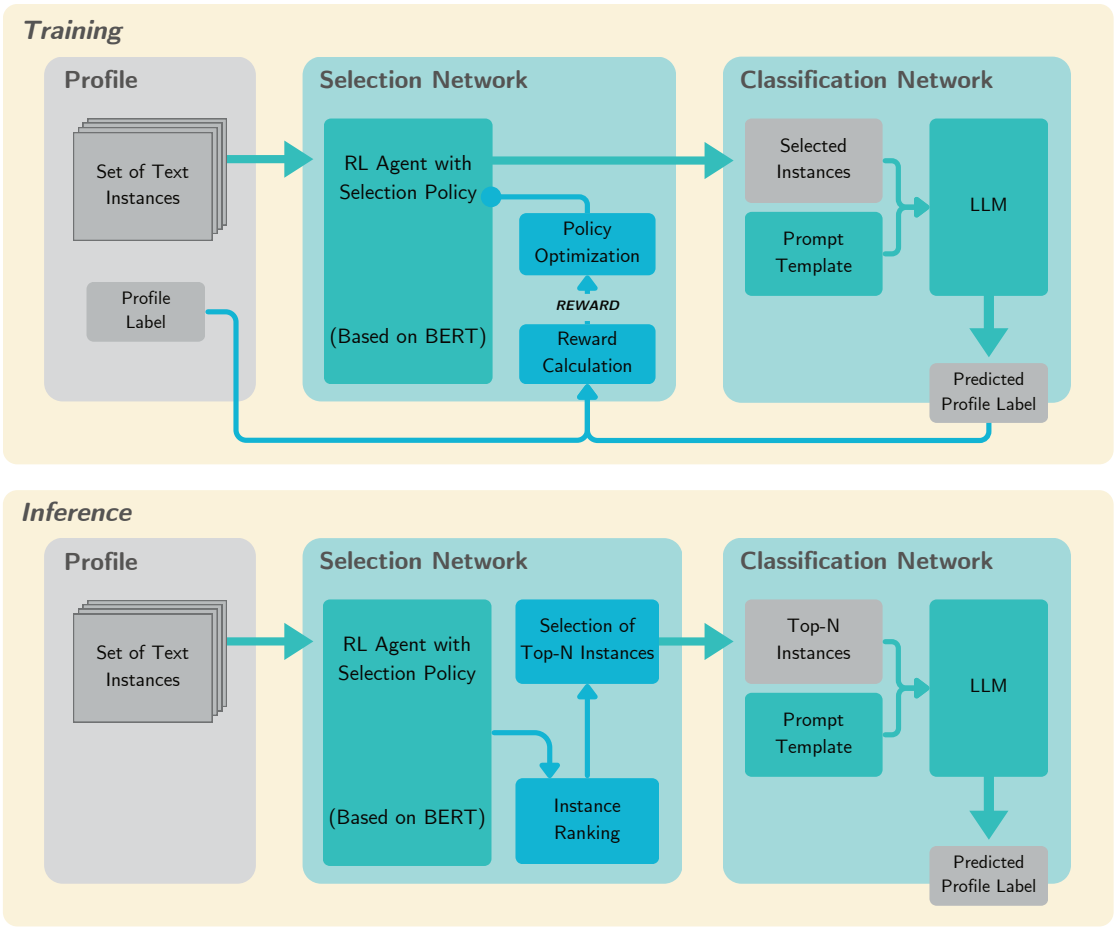

Figure 5: Illustration of the proposed system for training the RL agent (top) and for inference (bottom). During training, for each text instance associated with a profile, the action of *selecting* or *rejecting* this instance is sampled from the predicted probabilities given by the agents' current policy. Afterward, selected instances are fed to the *Classification Network* predicting a label. The predicted label and the ground-truth label are then used to calculate a reward from which the RL agent in the *Selection Network* learns. During inference, text instances associated with a profile are ranked w.r.t. the probabilities predicted by the agents' policy of *selecting* them. Then the top-N instances are fed to the Classification Network predicting a label from these instances.

4.2 Selection Network

This work frames the task of selecting and rejecting instances as a sequential decision problem, which is addressed using reinforcement learning. Here, a reinforcement learning agent is learning to *select* or *reject* textual instances indirectly based on feedback obtained through the predicted label by the *Classification Network*. This agent uses a stochastic policy $\pi(a_t|s_t, \theta)$ with a binary action space. Given a profile, i.e. a collection of text instances $\mathcal{S}$, the agent is iteratively presented a single text instance $s_t \in \mathcal{S}$ and is selecting an action $a_t \in \{select, reject\}$ for each text instance.

This agent is parameterized using the pre-trained language model BERT, since this work hypothesizes that the knowledge obtained through pre-training could be beneficial for the agent in such a setting. Here, the BERT model is adapted with a binary classification head representing the action space. More precisely, the *[CLS]* token of the BERT model is fed into a fully connected neural network layer with softmax activation. The input to this BERT-based RL agent is a single text instance, from which probabilities for *selecting* or *rejecting* an instance are predicted. The parameter θ of the policy of the agent, therefore denotes the trainable parameters contained in the pre-trained BERT model and the classification head.

The *Selection Network* behaves differently during training and inference (cf. top and bottom part of Figure 5) During training, the action of *selecting* or *rejecting* an instance is sampled from the probabilities given by the agents' current policy. This ensures that the agent is exploring different actions during training. However, the agent is not supposed to explore actions during inference. Therefore, this component is adapted for inference: given the set of instances in a profile, the policy of the trained agent is first predicting probabilities whether to *select* or *reject* each individual text instance. These instances are then ranked by the probability of *selecting* them and the top-N instances are fed to the *Classification Network*. This ensures that during inference, the agent is no longer exploring possible actions but exploits knowledge learned during training. Further, this ensures that always a set of instances of fixed size N is given to the *Classification Network* for inferring some characteristic. Aside from prohibiting exploration during inference, this is also im-

portant since sampling actions as done in training can potentially lead to an empty set or in a set of instances that is too large for the *Classification Network* to handle. This number of instances N is a hyperparameter of the proposed approach.

4.2.1 Training the RL agent

Reward Calculation. The top part of Figure 5 illustrates the training process of the proposed system for one episode. After collecting a subset of instances C from the set of all instances associated with a profile, the *Classification Network* is using this set to predict a label. The predicted label and the ground truth label associated to this profile are then used to optimize the policy of the RL agent. Here, this study proposes to calculate this learning signal, i.e. the return G, using the predicted label $\hat{y}$ and the ground-truth label y associated with a profile as follows:

$$G = \begin{cases} +1 - \lambda|C|, & \text{if } \hat{y} = y \text{ and } |C| > 0 \\ -1 - \lambda|C|, & \text{if } \hat{y} \neq y \text{ and } |C| > 0 \\ -2, & \text{if } |C| = 0, \end{cases} \tag{20}$$

where $|C|$ is the number of selected text instance and λ a hyperparameter. Maximizing this return is the goal of the RL-agent.

With this, the agent is learning to *select* individual instances such that the *Classification Network* is predicting the correct label. Further, to maximize this return the agent needs to learn to *reject* as many instances as possible, since, for each instance in the set C, the return is reduced. This reduction is scaled by λ. Further, this study found that such an agent may *reject* all text instances associated with a profile. To punish such a behavior a strong negative signal is given to the agent if the set C does not contain any text instances.

Policy Optimization. After calculating the return, this learning signal is used to optimize the behavior of the RL agent. For optimizing the parameters contained in the policy π of the RL agent, the update rule of the REINFORCE algorithm is adapted in this work. Here, given a profile P associated with a set of text instances $\mathcal{S}_P$

the parameters contained in θ are updated based on the return G and the predicted probabilities of each of the chosen actions following policy π:

$$\theta \leftarrow \theta + \alpha \sum_{t=1}^{|\mathcal{S}_P|} (G - b) \ln \nabla_\theta \pi(a_t | s_t, \theta), \tag{21}$$

where the average return over the last ten update steps is used as baseline b. This rule diverges from the update rule of the REINFORCE algorithm presented in Equation 19 as parameters contained in θ are only updated once after processing all text instances, i.e. states, $s_t \in S_P$. In contrast, the original update rule updates parameters in θ for each state s_t independently (cf. line 5-7 in Algorithm 1). This is not possible in the setting of this work as a return can only be calculated considering all text instances (states). Instead, this approach assigns an equal return to all chosen actions. Updating the agent like this is one of the core ideas of this work.

The process of training an agent using this update rule and a number of profiles with associated ground-truth data is presented in Algorithm 2. To train the agent, a training dataset consisting of profiles each associated with a specific ground-truth label is used (similarly to a supervised learning settings). Note that, to reduce complexity, this ground-truth profile label is limited to binary values in this study. This algorithm is iterating multiple times over this training dataset (line 3). These iterations correspond to training epochs from a supervised learning perspective. In each epoch (iteration) the profiles in the given dataset are randomly arranged (line 4). The algorithm then iterates over all profiles available in the training dataset (line 5). Given a single profile, each text instance associated with this profile is then processed individually (line 7-12). Here, the current policy π predicts a probability for each text instance being relevant or irrelevant. In other words, the agent predicts probabilities for the actions whether to *select* or *reject* an instance. During training, this action is sampled according to the predicted probabilities (line 8). For example, if the agent predicts a probability of 80% to *select* a text instance, this instance will be added with a probability of 80% to the set of selected instances C. However, this also means that with a probability of 20% the algorithm will not add this instance to the set of selected instances C. The text instances for which the action of *selecting*

Algorithm 2 RL-Profiler: Learning Algorithm

1: **Input:** Policy π parameterized by θ with action space $\{select, reject\}$, a training set D with a set of profiles $\{P_1, ..., P_i\}$ each associated with a set of text instances $\mathcal{S}_P$ and ground-truth y_P, and a number of training epochs $E > 0$
2: Pre-train policy $\pi(a|s, \theta)$ using NPMI-Annotations
3: **for** $i \leftarrow 1 \dots E$ **do**
4: Shuffle training set D
5: **for** profile P in D **do**
6: $C \leftarrow \{\}$ ▷ Set of selected instances for current profile.
7: **for** text instance s_t in $\mathcal{S}_P$ **do**
8: Sample action $a_t \in \{select, reject\}$ from $\pi(a_t|s_t, \theta)$
9: **if** $a_t = select$ **then**
10: $C \leftarrow C \cup s_t$ ▷ Collect selected instances.
11: **end if**
12: **end for**
13: $\hat{y}_P \leftarrow$ Label predicted by *Classification Network* using C
14: $G \leftarrow$ Calculated Return using y_P, $\hat{y}_P$ and C
15: $\theta \leftarrow \theta + \alpha \sum_{t=1}^{|\mathcal{S}_P|} (G - b) \ln \nabla_\theta \pi(a_t|s_t, \theta)$
16: **end for**
17: **end for**

them is sampled are collected in set C (line 10). After sampling an action for each text instance associated with an author, the selected instances are fed to the *Classification Network* predicting a label using this set (line 13). Afterward, the return G is calculated as presented in Equation 20 using the predicted label, the ground-truth label and the set C (line 14). In line 15 this return G is then used to update the policy as given in Equation 21. For simplicity, the calculation of the baseline b is not shown in Algorithm 2. This baseline is calculated as the moving average return considering the last 10 update steps, i.e. the average return G obtained from processing the last 10 profiles in the training set, which yields an estimation of the expected return given the current policy.

4.2.2 Pre-training using Mutual Information

One step that is missing in this explanation of the training process of this reinforcement learning agent is line 2 in Algorithm 2. This study hypothesizes that such a training process is expected to be unstable since (1) the reward signals by the *Classification Network* may not be as precise as needed, introducing ambiguity and potential errors in the learning process, and (2) reinforcement learning using policy gradient methods itself is known to potentially suffer from instability (Mnih et al., 2015). Therefore, instead of relying on a random initialization of the policy, a supervised pre-training step is added to guide the policy in a certain initial direction. Still, there exists no ground-truth annotations about which individual instances should be kept and which can be removed. To solve this, this study hypothesizes that information theoretic measures can potentially be used to estimate a relevance value for each individual instance of an author. Here, pointwise mutual information (PMI) (Church and Hanks, 1990) is used to weigh the association between words present in the text instances provided by a profile and the corresponding ground-truth label. More specifically, for each word w and label c, the normalized pointwise mutual information (NPMI) (Bouma, 2009) is calculated as follows:

$$\text{NPMI}(w;c) = \Big(\ln \frac{p(w,c)}{p(w)p(c)}\Big) \Big/ -\ln\, p(w,c), \tag{22}$$

where $p(w,c)$ is the probability observing a word w in a text instance from an author associated with label c, $p(w)$ is the probability of observing the word w in a text instance by any author, and $p(c)$ is the probability that a text instance is written by an author associated with label c. Here, the training set is used to estimate these probabilities. Afterward, these association weights are used to estimate a relevance-score for individual instances. Here, for each instance $s \in \mathcal{S}_P$ of a profile P, a score is calculated as follows:

$$\text{relevance-score}(s, c_1, c_2) = \frac{\big|\big(\sum_{w \in s} \text{NPMI}(w;c_1)\big) - \big(\sum_{w \in s} \text{NPMI}(w;c_2)\big)\big|}{|\{w \in s\}|}, \tag{23}$$

where c_1 and c_2 are the possible labels in a given author profiling problem. Note that this study only uses binary profile characteristics, and it is therefore sufficient to

define this score for two classes. Assuming that in a given textual instance, the sum of NPMI weights of words and label c_1 is low while this sum for the other label c_2 is high, for example, this assigns a high score to this instance suggesting that the instance may be relevant for predicting label c_2. After calculating a relevance-score for each text instance of all authors in the training set, the top-M instances w.r.t. the highest scores of each author in the training set are marked as *relevant* while others are marked as *irrelevant*. These annotations are then used as supervised learning signal for pre-training the BERT model (line 2 in Algorithm 2). The intuition here is that the BERT model is learning to replicate this information theoretic relevance scoring. Here, the number of instances M and the "pre-training strength", i.e. learning rate and number of epochs during this pre-training step are hyperparameters.

4.3 Classification Network

The combination of the *Selection Network* and the *Classification Network* forms a pipeline predicting a label given textual instances from a profile. Given a set of selected text instances, the *Classification Network* is responsible for predicting this label. In this work, this is done by using a large language model in a zero-shot prompting setting, since such a setup does not require any task specific training. Here, the classification task of predicting a label from the selected text instances is verbalized, i.e. reformulated to match the LLM's pre-training objective. The *Classification Network* is therefore creating a prompt using the selected text instances by the *Selection Network* and a pre-defined prompting template. Then a prediction is sampled from the text the large language model generates given this prompt.

Note that in principle, any system that predicts a profile label from a set of text instances can be used as the *Classification Network*. Also, performance of this model influences the learning process of the RL agent. This is, because a system, that is not able to predict labels from selected instances correctly, can lead to misleading reward signals for the agent.

5 Experimental Setting

5.1 Model Selection and Training Details

There are many options for implementing the individual components in the proposed approach (e.g. which LLM is used in the *Classification Network*, or how the policy of the agent is paramterized in the *Selection Network*). Based on the selected models for implementing the approach, different hyperparameters arise. Therefore, the following will introduce the selection of models and the correspondent hyperparameter choices used in this study. In general, this work implements the proposed pipeline setting, consisting of the *Selection Network* and the *Classification Network* as presented in the last section, using the PyTorch (Paszke et al., 2019) and HuggingFace's Transformer (Wolf et al., 2020) libraries.

Selection Network. Table 4 shows selections for different hyperparameters in the *Selection Network*. For parameterizing the policy of the agent in the selection network, this study only experiments with using the BERT model. More specifically, this work uses the *base-uncased*[4] version. Here, the *[CLS]* token is fed into a binary classification layer using softmax activation. In addition, a dropout (Srivastava et al., 2014) with probability of 20% is added before this classification layer.

As presented in the last section, the training process of this agent consists of two phases: (1) the training phase using the PMI-Annotations and (2) the training phase using reinforcement learning. The first phase can be seen as a pre-training phase before using reinforcement learning. For both phases, one can individually adjust the learning rate and number of training epochs. In this study, the number of epochs in the first training phase is fixed to 2. The reinforcement learning phase is training up to 200 epochs with early stopping. Here, after each epoch, the current selection policy is evaluated on a validation set with different settings for top-N, i.e. $N \in \{5, 10, 20, 30, 50\}$. For each setting, individual checkpoints corresponding to the best performance on the validation set for each N are saved. Note that when evaluating the policy after each epoch the system is set to "inference mode" (cf.

[4] `https://huggingface.co/google-bert/bert-base-uncased`

Hyperparameter	(1) Pre-train Phase	(2) RL-train Phase
Model	bert-base-uncased	bert-base-uncased
Epochs	2	200
Early stopping	No	Yes
Learning rate	[1e−7, 1e−6]	[1e−7, 1e−6]
Batch size	32	–
λ	–	0.05
top-K	10	–
top-N	–	[5,10,20,30,50]

Table 4: Hyperparameter configurations in the *Selection Network*. (1) denotes the hyperparameters for pre-training the agent using PMI-Annotations, while (2) denotes the hyperparameters for the training the agent using reinforcement learning. λ is a hyperparameter as introduced in Equation 20.

bottom part of Figure 5). This means that, for each profile in the validation set, the top-N tweets ranked by the predicted probability of *selecting* them given the current policy are fed to the *Classification Network* to predict a label. These predictions are then compared to the ground-truth data to evaluate performance (macro-average F_1) of the current policy on validation data.

Both agent training phases are optimized using AdamW (Kingma and Ba, 2017; Loshchilov and Hutter, 2019). This work experimented with different learning rates for both training phases during development of the approach. However, during preliminary experiments (introduced later in this section), it was found that, for both learning phases, using a learning rate of 1e−6 produces the most stable results. Therefore, this learning rate acts as a default setting used across all experiments. The reinforcement learning phase of the agent requires an additional hyperparameter: λ (as introduced in Equation 20). This parameter is responsible for adjusting the return that is given to the agent based on the number of instances that have been selected. Here, this work fixes $\lambda = 0.05$ as this setting seemed to produce the most stable results across experiments. Note that due to time and computational power limitations, different learning rates and settings for λ are not evaluated in this work.

```
<s>[INST] <<SYS>>
one word response
<</SYS>>

Recall the personality trait extraversion.
A person with a high level of extraversion may see themselves
+as someone who is talkative, or {...}
A person with a low level of extraversion may see themselves
+as someone who is reserved, or {...}

Consider the following tweets written by the same person:
{tweets}
Does this person show a low or high level of extraversion?
+Do not give an explanation. [/INST]
```

Listing 5: Prompt template for the personality trait *extraversion*.

Classification Network. For the LLM in the *Classification Network*, this study uses a model from the Llama 2 family (Touvron et al., 2023b). Here, the 13B-Chat version with GPTQ parameter quantization (Frantar et al., 2023) *Llama-2-13B-chat-GPTQ*[5] is used across all experiments. For simplicity, this study does not tune any hyperparameters of this model. Therefore, the model's hyperparameters are set to default values (temperature τ=0.8, top-p=0.9). Further, this study does not carry out extensive prompt tuning. Except for one concept, on which different prompt versions are evaluated, the prompts used in this study follow a similar structure. This study focuses on predicting the "Big-5" personality traits for given profiles, and therefore, prompts used in this study adhere to the personality trait concepts.

The prompt template shown in Listing 5 exemplarily shows this for an input designed to task the model predicting either a *low* or a *high* level of *extraversion* given a set of tweets from a profile. Note that a list of all prompts used in this study is provided Appendix A.1. The prompts used in this study use the system prompt "one word response", followed by a trait specific context. This trait specific context aims to provide background about the specific task the LLM is supposed

[5] `https://huggingface.co/TheBloke/Llama-2-13B-chat-GPTQ`

to solve ("Recall the personality trait extraversion"). Here, the LLM is then given context what might characterize a *low* or a *high* level of *extraversion*. To do this, this study uses items from the BFI-44 (John et al., 1991) questionnaire that score such a particular personality trait. These items are added as context for a *high* level ("A person with a high level of extraversion may see themselves as ..."), while items with reversed scores are added as context for a *low* level. With this context, this study aims to provide a background characterizing the task the model is supposed to solve. Note that it is not allowed to reprint the full questionnaire without permission. The prompt shown in Table 5, therefore only contains one item for each variable and for a full list of items associated to each trait I refer the reader to the work by John et al. (1999). In this example prompt, plus signs indicate that line breaks are added for visibility. These breaks are not present in the actual prompts. Since this study limits the type of text instances by an author to tweets, the task is then further presented to the model by adding the phrase "consider the following tweets written by the same person". Afterward, a list of tweets is added. In the proposed method, this list of tweets corresponds to those chosen by the *Selection Network*. Then the classification task is verbalized: "Does this person show a low or high level of extraversion". Here, also the phrase "Do not give an explanation" is added as it was found that the LLM used in this study tends to give long explanation drastically increasing computational power requirements. The classification result is sampled from the tokens the model generates from such a prompt.

5.2 Corpora

The profiling approach proposed in this study is evaluated on the PAN-AP-2015 corpus[6] (Rangel et al., 2015). This corpus is available upon submitting an access request. It consists of tweets written in the Dutch, English, Italian and Spanish language, which are associated to profiles annotated with "Big-5" personality traits, gender and age. In this study, only the English part of this corpus is used. The personality trait annotations in this corpus are derived from self-assessed BFI-10

[6] `https://zenodo.org/records/3745945`

	Profiles						Tweets					
	Training		Validation		Testing		Training		Validation		Testing	
Class	High	Low	High	Low	High	Low	High	Low	High	Low	High	Low
Open.	119	1	30	1	137	1	10,769	100	2,777	100	12,877	36
Consc.	93	3	24	2	113	10	8,346	241	2,178	199	10,544	987
Extra.	96	12	24	3	114	6	8,675	1,080	2,221	300	10,487	600
Agree.	90	15	24	4	108	11	8,002	1,408	2,214	400	9,930	1,009
Neuro.	83	30	22	8	91	39	7,447	2,706	1,998	779	8,331	3,647

Table 6: Corpora statistics of the splits derived from the PAN-AP-2015 (Rangel et al., 2015) corpus.

online tests (Rammstedt and John, 2007). Here, for each author a real-valued score between -0.5 and 0.5 is provided for each personality trait. Since in this study, each trait is only considered on a binary scale, i.e. for a given profile, the task is to predict a *low* or *high* value for a specific trait, these scores are converted to binary values. This is done by first extracting all profiles with negative and positive values for each personality trait. Then, the labels *low* or *high* are assigned to profiles with negative and positive values, respectively. Profiles annotated with a neural score are removed.

Further, this corpus provides pre-defined training and testing parts. From the training part, this study samples profiles for validations. Here, stratified sampling is used to ensure a similar class distribution to the data used for training. This means that this study derives different splits for each personality trait dimension, which consequently results in a different sub-corpus for each personality trait. With this, about 20% of the training data is used for validation for each trait. The number of profiles and the total number of tweets in each of these parts of the derived corpora are shown in Table 6. Over all traits the distribution between the classes is majorly skewed towards profiles annotated with the class *high*. For example, for the *openness* trait, there exists only one profile in the training, validation and testing set associated with the class *low*. On average over all traits and splits, each profile is associated with 92.3 tweets.

5.3 Baselines and Derived Systems

To answer the research questions presented in Section 1 this study chooses several baseline systems. First, to be able to compare the proposed system to supervised approaches, two baseline systems that follow this paradigm are chosen in this work (cf. third research question). Note that, the selection of such baseline systems is difficult as there is limited research on predicting personality traits from tweets on a binary scale and approaches that tackled this profiling problem in the PAN-AP-2015 shared task predict continuous values for each trait. Further, since this shared task was hosted in the year 2015, these approaches do not use transformer-based models like BERT. Therefore, the following two systems are chosen in this work:

***Baseline-R*: Regression Classifier**. For the first baseline, the best performing system in the PAN-AP-2015 shared task is adapted to fit the binary profiling problem as proposed in this work. In this system, Sulea and Dichiu (2015) use a ridge regression model with character level *tf-idf* n-gram features. Here, they found that combining character n-grams of lengths two to six performs best predicting personality dimensions on a continuous scale. This work adapts this approach by converting the ridge regressor into a ridge classifier. Here, the binary classification problem of this work is converted back to a regression problem. This is done by assigning a value of -1 to the class *low* and a value of +1 to the class *high*. A regression model is then trained minimizing the mean squared error with ridge (l2) regularization predicting a continuous value. If this predicted value is greater than 0 the class *high* is assigned and otherwise the class *low*. Note that this study uses 1-grams and word counts instead of tf-idf features as this produced the best results on validation data.

***Baseline-B*: BERT Classifier**. For the second baseline the transformer-based model BERT (bert-base-uncased) is adapted to the binary classification problem using a classification head with softmax activation. Since, the length of the input to the BERT model is limited, all tweets associated with an author can not be presented to this model at once. Therefore, this work propagates profile labels to individual tweets. The BERT model is then trained on these tweets for two epochs using a learning rate of $2e{-}5$ and cross-entropy loss weighted by class distribution.

After each epoch the model is evaluated on the validation set and the best model w.r.t. accuracy on individual tweets is chosen for evaluation on the testing data. In the evaluation, to obtain a prediction for a profile which can be compared to the annotated gold label, a majority decision is drawn from predictions on the individual tweets. Note that since this method is likely to produce different results in different runs, this baseline is run 10 times repeatedly and averaged results are reported.

The second type of baseline systems are variations of the proposed system that also use a zero-shot setting to predict labels and are used to answer the first and second research question:

ALL+CNet. One variation of the proposed approach is to skip the selection process of the *Selection Network*. In the *ALL+CNet* system, all tweets associated with an author are given to the *Classification Network* directly. Note that this results in a very long prompt and this can therefore only be applied if the LLM in the *Classification Network* is capable of handling such a long prompt (context). This system is used to answer the first research question presented in Section 1.

RND+CNet. Another variation of the proposed approach is replacing the reinforcement learning agent in the *Selection Network* with a random selection. Here, instead of using a learned selection of text instances from a profile, the system is selecting a number of instances at random. Then, this random selection is provided to the *Classification Network* predicting a label. This variation is used to answer the second research question presented in Section 1. As performance of such a system likely depends on this random selection, this system is evaluated 10 times with different random seeds and average results are reported.

In addition to the baseline systems and variations of the proposed approach used to answer the specific research questions posed in this work, two supplementary systems are evaluated in this study. These systems aim to provide a better understanding how the proposed system behaves during different training stages and are defined as follows:

PMI+CNet. In the *PMI+CNet* system the selection process using the trained agent is replaced by selecting a number of text instances based on their *relevance-*

score as introduced in Equation 23. Here, instances are ranked based on NPMI information and the top-N instances are provided to the *Classification Network*. With this system, this study aims to provide insights on performance of such a selection system when simply relying on information theoretic measures.

Pre-train+CNet. Further, the agent trained using reinforcement learning can be replaced by an agent that is only pre-trained on NPMI-Annotations. This means to stop training the agent after line 2 in Algorithm 2. This model is the starting point from which training using reinforcement learning starts in the proposed system.

5.4 Evaluation Procedure and Metrics

The experiments in this thesis are evaluated using precision P, recall R and F_1 score. Here, F_1 is the harmonic mean between precision and recall, i.e. $\mathrm{F}_1 = \frac{2P \cdot R}{P+R}$, calculated for each class. Macro-averaged scores over all classes are calculated using the arithmetic mean of scores of individual classes. In addition, experiments are evaluated using weighted-average scores defined as the mean of scores weighted by the number of instances per class.

Performance during evaluation of the individual systems in this study can vary between runs. This is, for example, because of the non-deterministic output generated by the large language model used in the *Classification Network*. Further, in the *RND+CNet* system, variations are to be expected since each run uses a different set of randomly chosen instances (tweets) given a profile. Therefore, each system is evaluated 10 times. In each run the above described metrics are collected and then averaged to obtain a final performance measure.

5.5 Preliminary Experiments

Applying reinforcement learning to classification tasks is no common practice and also poses some engineering problems. There are, for example, no reference values for specific hyperparameters such as the learning rate of the agent or the penalty λ adjusting the reward if the agent is predicting too many text instances to be relevant

given an author. Furthermore, the computational power requirements for this study are rather large as training an agent means to query the LLM repeatedly (each episode, i.e. parameter update step, of the proposed reinforcement learning setting requires one prediction by the LLM). Therefore, this study develops and validates the approach in a controlled environment. Here, a subset from the corpus described previously is derived. Further, to ensure there exist tweets in the given profiles the LLM in the *Classification Network* can provide a correct feedback to—if the agent is capable of finding these—the data is **enriched with artificially generated tweets**. Here, the idea is to use generated tweets, that are highly indicative for either a *low* or *high* level of a specific trait. Depending on the given ground-truth labels, i.e. *low* or *high*, for profiles in the corpus, generated tweets corresponding these ground-truth labels are then added to a profile. For example, generated tweets associated with a *low* level of *extraversion* are added to profiles which are associated with *low* level of *extraversion* in the given dataset.

5.5.1 Generating Artificial Data

To generate artificial tweets for enriching the dataset, this work also uses *Llama-2-13B-chat-GPTQ*. Here, similarly structured prompts are used as described previously. The prompt for creating artificial tweets associated with a *high* value of *extraversion* is exemplarily shown in Listing 7a. Here, the task of generating tweets is verbalized by the phrase "Generate ten tweets that are likely written by a person with a high level of extraversion". Similarly to the prompts used in the *Classification Network* for prediction a label, these prompts also include BFI-44 items to enrich context of the large language model. However, to encourage diversity while generating tweets further, since different profiles should not be enriched with the same tweets (as it would be trivial for a model to find these tweets if they are always the same), the LLM is asked to include a certain topic in the generated tweets. This topic is chosen from 12 possible topics derived from the work by Antypas et al. (2022). Such topics, for example, include *news*, *social concern* or *relationships*. Table 7b shows examples of artificially generated tweets using the prompt presented in Table 7a and these topics. The full list of topics and the prompts used for the

```
Recall the personality trait extraversion.
A person with a high level of extraversion may see themselves
+as someone who is talkative, or {...}
Generate ten tweets that are likely written
+by a person with a high level of extraversion.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompt for generating artificial tweets indicating a *high* level of *extraversion*.

Topic	Example Tweet
News	I'm so excited to share the latest scoop with all my followers!
Social concern	Who's ready to take on a challenge? Let's organize a charity run to support children's education!
Relationships	I'm not scared of rejection. I'll put myself out there and see what happens!

(b) Examples of generated tweets supposed to indicate *high* levels of *extraversion*.

Table 7: Prompt template for generating artificial tweets and examples of generated tweets using this prompt in combination with the topics *news*, *social concern* and *relationships*.

different personality traits as well as more examples of such artificially generated tweets are provided in Appendix A.2. For each topic, trait and class 20 tweets are generated. This results in 240 tweets per trait and class, which can be used to enrich profiles.

Table 8 shows statistics of the corpora derived for the preliminary experiments. Since there is a limited amount of artificial data to enrich the dataset and to limit computational expense for these preliminary experiments, subsets from the corpus introduced in Table 6 are drawn. Here, for each trait, split and class 15 profiles are randomly selected. Note that in some partitions there are less than 15 tweets and that this also changes class distribution in each sub-corpora. Each of these profiles is then enriched with five artificially generated tweets corresponding to their ground-truth label. Note that also the evaluation set is enriched in this experiment. Therefore, these subsets do not correspond to real world settings and are only used for developing and validating the method proposed in this study.

	Profiles						Tweets					
	Training		Validation		Testing		Training		Validation		Testing	
Class	High	Low	High	Low	High	Low	High	Low	High	Low	High	Low
Open.	15	1	15	1	15	1	1,502	105	1,493	105	1,447	41
Consc.	15	3	15	2	15	10	1,567	256	1,434	209	1,485	1,037
Extra.	15	12	15	3	15	6	1,408	1,140	1454	315	1,393	630
Agree.	15	15	15	4	15	11	1,364	1,483	1,455	420	1,483	1,064
Neuro.	15	15	15	8	15	15	1,441	1,417	1,490	819	1,504	1,523

Table 8: Statistics of the dataset splits derived from the PAN-AP-2015 (Rangel et al., 2015) corpus for the preliminary experiments.

5.5.2 Preliminary Results

In this section the results of applying the introduced systems to the dataset enriched with artificially generated tweets are presented. For systems selecting a number of tweets from a profile, different settings for top-N are evaluated on validation data. This is, for example in the *Selection Network* of the proposed *RL-Profiler* system, which is ranking tweets of an author according to its policy and then selects the top-N tweets, or in the *RND+CNet* system which is choosing N tweets at random. For the proposed *RL-Profiler* system, these different settings correspond to different checkpoints saved during training. However, due to the probabilistic results of the *Classification Network*, this study repeats evaluation of these checkpoints for selecting either 5, 10, 20, 30 or 50 tweets on the validation data and reports average results of 10 runs. Given these results on the validation, the best performing setting for N for each method is used for evaluation on the testing data.

Validation Results. Figure 6 shows macro-average F_1 score of the different methods with different settings for the number of selected tweets N when predicting *high* and *low* levels of *openness*, *conscientiousness*, *extraversion*, *agreeableness*, and *neuroticism* on validation data. Results for the proposed *RL-Profiler* system are shown in blue and settings for $N \in \{5, 10, 20, 30, 50\}$ are evaluated. Results for the system skipping the selection process (*ALL+CNet*) using all tweets of a profile are visualized through a star. When increasing N such that all tweets of profiles are

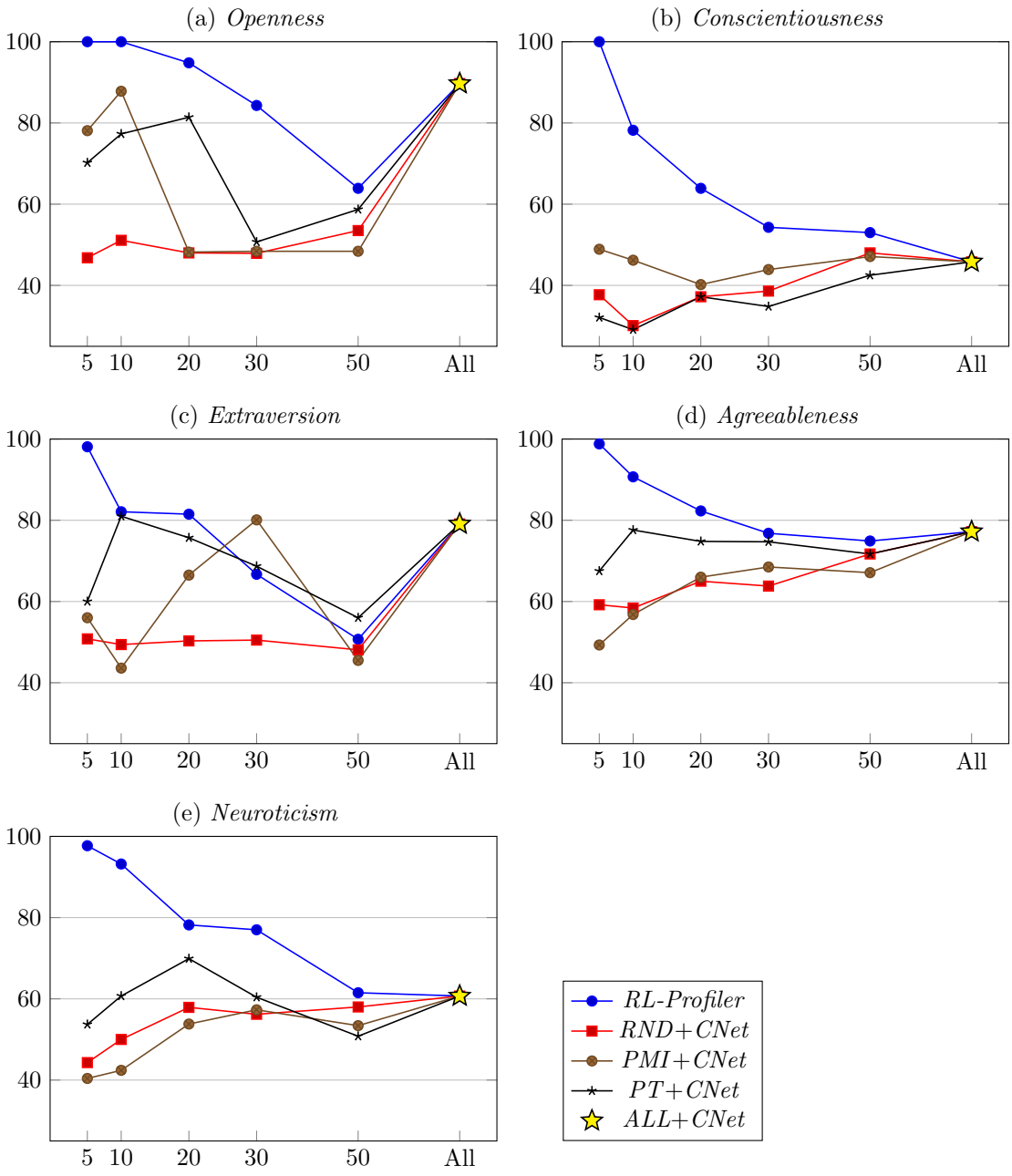

Figure 6: Macro-average F_1 for selection based models with different configurations for top-N predicting levels of the "Big Five" traits on enriched validation data. Note that the x-axis is not true to scale and different results for $N \in \{5, 10, 20, 30, 50\}$ are linearly interpolated. When using *all* tweets of profiles, selection based models select the same set as the *ALL+CNet* system.

selected—since there is a limited number of tweets for each profile—all selection based systems (*RL-Profiler*, *RND+CNet*, *PMI+CNet*, *PT+CNet*) converge to the *ALL+CNet* system as all systems simply use all tweets of given profiles.

The results on the validation data show that the proposed system outperforms all other methods, including the *ALL+CNet* system, for all personality traits if using 5, 10, or 20 tweets from profiles. For *conscientiousness*, the proposed system outperforms the other systems also if using 30 or 50 tweets. For all traits, the proposed *RL-Profiler* system achieves the best results if using N=5 tweets and performance continuously decreases if using more tweets. For example, when predicting levels of *conscientiousness*, the proposed system achieves 100% macro F_1 if using 5 tweets, while using 10, 20, 30, or 50 tweets results in macro F_1 scores of 78.2%, 63.9%, 54.3%, and 53.0%, respectively. Vice versa, these results on the validation data show that performance of the *RND+CNet* system (selecting N tweets at random) typically increases when increasing the number of tweets. For example, when predicting *high* and *low* levels of the *neuroticism* trait, macro-average F_1 using the *RND+CNet* system is increased by 13.7PP when using 50 instead of 5 tweets (58.0% vs. 44.3 % macro F_1). Interestingly, the proposed system outperforms the *RND+CNet* system even if using N=50 tweets from each profile across all traits. Note that 50 tweets is more than half of the tweets associated with each profile in this dataset. Except for the *conscientiousness* trait, the results on the validation data also show that the *PT+CNet* system outperforms the *RND+CNet* system if using 5, 10, 20, or 30 tweets, indicating that pre-training the policy as proposed in this work can be beneficial compared to a random initialization. Note that this work does not study the impact of abstaining from this pre-training step and leaves this for future work.

Model Selection. Table 9a summarizes the results of the *RL-Profiler*, *RND+CNet* and *ALL+CNet* system on validation data. Here, in addition to macro-average F_1 scores, weighted-average F_1 scores are reported. However, using weighted-average F_1 scores as a performance indication, this study observes similar behavior for these methods: scores for the *RL-Profiler* system decrease, while scores for the *RND+CNet* system typically increase if increasing the number of selected tweets. Given these results on the validation set, for each method, the best performing setting w.r.t.

System	N	Open.		Consc.		Extrav.		Agree.		Neur.	
		m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1
ALL+CNet	all	89.7	98.2	45.8	63.6	79.1	89.9	77.2	77.7	60.7	67.6
RL-Profiler	5	**100**	**100**	**100**	**100**	**98.1**	**98.3**	**98.8**	**98.9**	**97.7**	**97.9**
RND+CNet	5	46.8	87.8	37.7	45.0	**50.8**	**76.5**	59.2	58.4	44.3	42.0
RL-Profiler	10	100	100	78.2	88.4	82.1	89.7	90.7	90.8	93.2	93.8
RND+CNet	10	51.1	87.9	30.1	33.7	49.4	74.1	58.4	57.3	50.0	49.6
RL-Profiler	20	94.8	99.1	63.9	77.2	81.5	89.6	82.3	82.2	78.2	81.0
RND+CNet	20	48.0	90.1	37.2	44.4	50.3	75.6	65.0	64.3	57.9	60.7
RL-Profiler	30	84.3	96.9	54.3	67.0	66.7	84.1	76.8	76.6	77.0	80.0
RND+CNet	30	47.9	89.8	38.6	46.1	50.5	75.9	63.8	63.2	56.2	61.0
RL-Profiler	50	63.9	93.5	53.0	65.5	50.7	77.9	74.9	74.6	61.5	67.9
RND+CNet	50	**53.5**	**91.6**	**48.0**	**59.3**	48.1	76.8	**71.7**	**71.5**	**58.0**	**65.1**

(a) Validation results for *RL-Profiler* and *RND+CNet* with different settings for top-N.

System	N	Open.		Consc.		Extrav.		Agree.		Neur.	
		m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1
Baseline-R	all	48.4	90.7	68.8	70.8	70.8	72.6	76.4	76.9	79.9	79.9
Baseline-B	all	77.3	93.2	73.4	74.2	75.3	77.1	68.3	69.4	63.5	63.5
ALL+CNet	all	48.4	90.7	85.2	85.7	80.1	85.0	78.1	78.5	50.6	50.6
RL-Profiler	best	100	100	98.7	98.8	93.5	94.5	98.8	98.9	96.3	96.3
RND+CNet	best	48.4	90.7	69.0	68.7	45.6	60.9	71.3	71.0	43.1	43.1
PMI+CNet	best	45.2	84.7	81.7	82.2	69.6	77.6	67.4	66.6	72.9	72.9
PT+CNet	best	46.6	87.4	67.6	66.9	52.1	65.5	81.1	81.0	62.9	62.9

(b) Results for different methods on testing data.

Table 9: Results for different models predicting levels for each "Big Five" personality trait on enriched data. (a) shows results of selection based methods on validation data using different settings for top-N. For each method, the best setting w.r.t. macro-average F_1 score (m-F_1) is selected (highlighted with **bold**). Note that for each personality trait a different setting can be selected. (b) shows results on test data for the selected settings and the baseline systems. w-F_1: weighted-average F_1.

macro-average F_1 score is selected for evaluation on the testing data. Here, macro-average F_1 is used since, for all traits, the possible classes (*high* or *low*) are heavily imbalanced and selecting a model based on weighted F_1 score could potentially favor "extreme classifiers" always predicting the majority class. In this table, results for the models selected for each trait and method are highlighted in bold. Note that a model is selected for each personality trait individually. For the proposed *RL-Profiler* system, across all traits, the model selecting N=5 tweets is chosen. For the *RND+CNet* system, the configuration using N=50 is chosen for all traits except for *extraversion*. For this trait, the *RND+CNet* system shows very similar performance across all settings for N, while N=5 edges slightly ahead of the other, and therefore this setting is selected. Note that detailed results of the preliminary experiment including F_1 scores for individual classes, results for the *PMI+CNet* and *PT+CNet* system, and standard derivation between runs are presented in Appendix A.3.

Test Results. Table 9b shows results of the selected configurations for these systems, the *ALL+CNet* system, and the baseline systems *Baseline-R* and *Baseline-B* on testing data. These results show that the proposed system outperforms all other methods, including the baseline systems, significantly. For *openness*, this study observes the largest performance gap between the proposed system and any other system. Here, the *RL-Profiler* system shows an increased macro F_1 score of 20.7PP compared to the *Baseline-B* system, achieving the second-highest score, moving from 77.3% to 100%. Note that evaluation of this trait is difficult as there is only one profile associated with a *low* level of openness. However, also for the other traits, the proposed system shows a significant gain compared to the system showing the second-highest performance: for *conscientiousness*, *extraversion*, and *agreeableness* +13.5PP, +13.4PP, and +20.7 vs. the *ALL+CNet* system, respectively, while for *neuroticism* +16.4PP vs. the *Baseline-R* system (macro-average F_1).

To address the first research question of how the proposed selection based system performs compared to simply using all tweets of authors in a zero-shot setting, the proposed system is compared to the *ALL+CNet* system using all tweets. Here, the results on artificially enriched data show an increase macro-average F_1 score averaged over all personality traits of 29.0PP moving from 68.5% to 97.5%. Comparing these

systems using weighted-average F_1, the results show an increased score of 18.8PP moving from 78.9% to 97.7% when using the proposed approach.

To answer the second research question of how the proposed system performs compared to a random selection of tweets in a zero-shot setting, the proposed system is compared to the *RND+CNet* system. Here, the results on the testing data show that using a random selecting results in a drastic performance drop. On average across all traits, this study finds that the proposed system outperforms the *RND+CNet* system by 44.0PP macro (97.5% vs. 53.5%) and 30.8PP weighted (97.7% vs. 66.9%) F_1 score. This is, although the *RND+CNet* system is using 50 tweets from each profile across all traits except *extraversion*, while the *RL-Profiler* system is only using 5 tweets from each profile. Given the results for the *RND+CNet* system on the validation data showing that using fewer tweets at random mostly yields even worse performance, it is likely that the performance gap between these models is even larger if the *RND+CNet* system is using the same amount of tweets as the proposed system. For the third research question of how the proposed system performs in comparison to supervised learning-based alternatives, the *RL-Profiler* system is compared to the two baseline systems *Baseline-R* and *Baseline-B*. Here, the results of this preliminary experiment show that the proposed zero-shot setup outperforms both of these baseline systems significantly across all traits. On average over all personality traits, the proposed system achieves an increased macro F_1 score of 28.6PP compared to the *Baseline-R* system (97.5% vs 68.9%), and an increased macro F_1 score of 25.9PP compared to the *Baseline-B* system (97.5% vs. 71.6%).

Summary. The results of this experiment show that using only a subset of tweets from a profile for predicting personality in a zero-shot setting can lead to significant performance improvements compared to using all tweets. Sill, the results also show that this subset needs to be selected carefully as a random selection leads to performance degradation. This validates that the proposed system is indeed able to learn to distinguish between indicative and less indicative content in this artificial experiment. However, the artificially enriched data used in this preliminary study does not correspond to real-world data. Therefore, the next section continues with applying the proposed approach to a real-world dataset.

6 Main Experiments

6.1 Results

The main experiments in this study consist of applying the proposed approach and its variations to the PAN-AP-2015 corpus as introduced in Section 5.2. Similarly to the preliminary experiments described before, this section starts with presenting the results on validation data. Afterward, the configurations for the different methods performing best on validation data are evaluated on testing data to address the research questions presented in Section 1.

Validation Results. Figure 7 shows macro-average F_1 scores of the proposed selection based methods predicting levels of the "Big Five" personality traits. Here, for each method and trait, F_1 scores for different settings for the hyperparameter N are shown. Results for the proposed system are shown in blue, while performance of the *ALL+CNet* system is visualized through a star. Note that in this Figure, results from the same method using different settings for N, i.e. $N \in \{5, 10, 20, 30, 50\}$, are linearly interpolated, and if increasing the number of tweets above the available number of tweets per profile all systems converge to the *ALL+CNet* system using all tweets from profiles as these systems no longer reject any tweets.

For all personality traits except for the *neuroticism* trait, the results on the validation data show that the proposed *RL-Profiler* system achieves the highest macro F_1 score if selecting N=5 tweets from each profile. For the *neuroticism* trait, the highest score is observed if using N=30 tweets. However, even if also using 5 tweets for the *neuroticism* trait, the proposed system consistently outperforms the *ALL+CNet* system when evaluated using macro-average F_1 score. For example, for predicting levels of *agreeableness*, the validation results show an improved performance of 15.3PP macro F_1 and 7.9PP weighted F_1 when using a learned subset of 5 tweets compared to using all tweets of each profile (70.1% vs. 54.8% macro F_1 and 84.4% vs. 76.5% weighted F_1). On average over all traits, the *ALL+CNet* system shows a macro-average F_1 score of 47.1% and a weighted average F_1 score of 76.3%, while for the proposed system this study observes 54.7% and 77.2%, re-

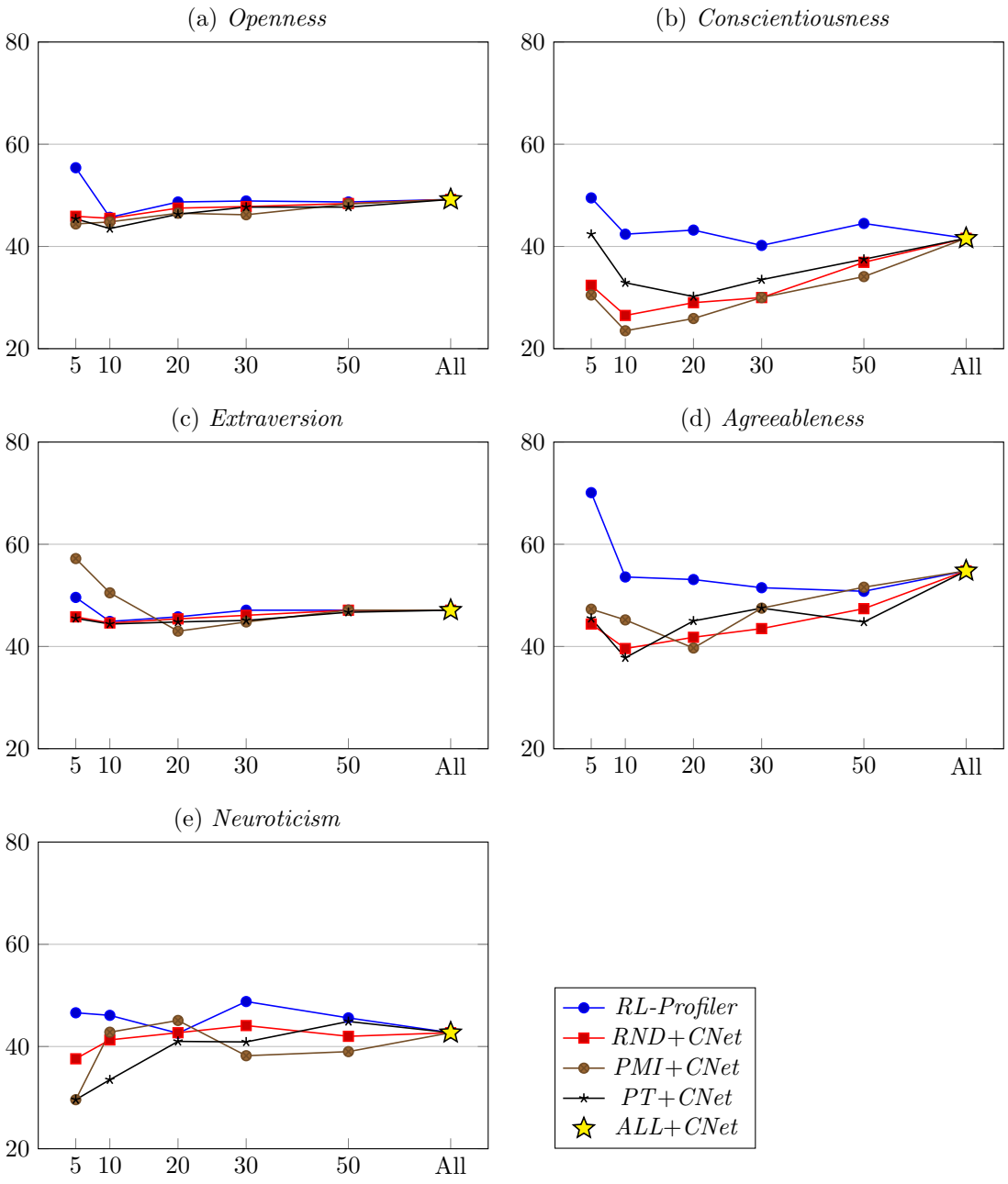

Figure 7: Macro-average F_1 for selection based models with different configurations for top-N predicting levels of the "Big Five" traits on validation data. Note that the x-axis is not true to scale and different results for $N \in \{5, 10, 20, 30, 50\}$ are linearly interpolated. When using *all* tweets of profiles, selection based models select the same set as the *ALL+CNet* system.

spectively, if using the setting for N that produced the best results for the proposed approach. When increasing the number of selected tweets to N=10, for all personality traits except for the *neuroticism* trait, the results on the validation data for the proposed system show a significant drop in performance. For the *agreeableness* trait, the largest difference when moving from 5 to 10 tweets is observed. Here, the proposed system achieves 70.1% if using 5 tweets and 53.6% macro-average F_1 score if using 10 tweets. For predicting levels of *neuroticism* this is different. Here, macro F_1 score increases by 2.2PP when increasing the number of tweets from 5 to 30 moving from a macro F_1 score of 46.6% to 48.8%.

Comparing the proposed system to the *RND+CNet* system, selecting N tweets at random, the results on validation data show that using a learned selection of tweets most often leads to better results. The largest differences in performance between these models is observed if using N=5 tweets across all traits. When predicting levels of *conscientiousness*, for example, the proposed system shows a macro-average score of 49.5%, while for the *RND+CNet* system a score of 32.4% is observed (70.6% vs. 43.5% for weighted-average F_1, respectively). When increasing the number of tweets this performance gap shrinks. For example for *extraversion* and *openness*, very similar results for the proposed and the *RND+CNet* system are observed if using 10 or more tweets. Still, for some traits the learned selection from the proposed system seems to be beneficial even if using approximately half of the available tweets from each profile (N=50). This is, for example, when predicting levels of *conscientiousness*. Here, the proposed system outperforms a random selection by 7.6PP macro F_1 (44.5% vs. 36.9%) and 11.1PP weighted F_1 (61.6% vs. 50.5%).

Interestingly the *PMI+CNet* system shows a remarkable performance predicting levels of *extraversion* on validation data if using N=5 tweets. In this setting, this system outperforms all zero-shot based methods including the proposed system (57.2% vs. 49.0% macro F_1). For *conscientiousness*, if using 5 or 10 tweets, the *PT+CNet* system surpasses performance of the *RND+CNet* system drastically. Still, these are exceptions as, for the other traits and settings, the *PMI+CNet* and *PT+CNet* system not only underperform compared to the proposed approach but often also compared to a random selection.

Model Selection. Table 10a summarizes the results of the *RL-Profiler*, *ALL+CNet* and *RND+CNet* system on validation data. Similar to the preliminary experiments, the setting for N that produces the best results on validation data is selected for each of the selection based methods for evaluation on the testing data. Here, the model checkpoint achieving the highest macro-average F_1 score is selected. For the traits *openness*, *conscientiousness*, *extraversion* and *agreeableness*, the model using N=5 tweets is selected for the proposed *RL-Profiler* system. For these traits, the *RND+CNet* system using N=50 tweets is chosen. For the *neuroticism* trait, the best performance on the validation data is observed if using N=30 tweets for both of these systems. Therefore, this setting is chosen for these methods for this trait. The selected configurations are highlighted with bold in Table 10a.

Test Results. Table 10b presents results of the selected configurations for each method predicting *low* or *high* levels of the "Big Five" personality traits on testing data. Here, also results for the baseline systems *Baseline-R* and *Baseline-B*, and the selected configurations for the *PMI+CNet* and *PT+CNet* system are shown.

To address the first research question of how the proposed selection based system performs compared to using all tweets associated with profiles in a zero-shot setting, the proposed *RL-Profiler* system is compared to the *ALL+CNet* system. When comparing these systems, this study finds that using all tweets of profiles yields comparable yet slightly better results across all traits except for the *extraversion* trait. For this personality trait, the proposed system achieves an increased macro-average F_1 score of 8.6PP moving from 48.4% to 57.0% compared to using all tweets of profiles, while weighted-average F_1 scores are very similar with 92.3% and 92.1% for the proposed and the *ALL+CNet* system, respectively. The difference in macro-average is due to the proposed system being able to predict some profiles associated with a *low* level of *extraversion* correctly, while the *ALL+CNet* system is not able to predict any profile associated with *low* level correctly. Still, for the other traits the *ALL+CNet* system shows better a prediction performance: for predicting levels of *openness*, *conscientiousness* and *neuroticism*, this study finds that proposed method performs 2.1PP, 2.4PP and 3.4PP worse when evaluated using macro F_1, and 4.3PP, 6.6PP and 10.0PP worse using weighted F_1, respectively. For *agreeableness*, this

System	top-N	Open.		Consc.		Extrav.		Agree.		Neur.	
		m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1
ALL+CNet	all	49.2	95.2	41.6	65.6	47.1	83.7	54.8	76.5	42.7	60.1
RL-Profiler	5	**55.4**	**92.0**	**49.5**	**70.6**	**49.6**	**82.5**	**70.1**	**84.4**	46.6	48.5
RND+CNet	5	45.9	88.8	32.4	43.5	45.8	81.4	44.4	53.6	37.6	36.0
RL-Profiler	10	45.7	88.4	42.4	60.4	44.9	79.8	53.6	70.7	46.1	49.6
RND+CNet	10	45.5	88.1	26.5	34.0	44.6	79.3	39.6	47.3	41.3	40.4
RL-Profiler	20	48.7	94.2	43.2	59.8	45.8	81.4	53.1	71.8	42.6	49.6
RND+CNet	20	47.5	92.0	29.0	38.1	45.4	80.8	41.8	50.9	42.7	47.9
RL-Profiler	30	48.9	94.7	40.2	56.0	47.1	83.7	51.5	68.0	**48.8**	**56.6**
RND+CNet	30	47.8	92.5	30.0	39.7	46.1	81.9	43.5	53.8	**44.1**	**51.7**
RL-Profiler	50	48.7	94.2	44.5	61.6	47.1	83.7	50.8	68.2	45.6	58.5
RND+CNet	50	**48.4**	**93.7**	**36.9**	**50.5**	**47.1**	**83.7**	**47.4**	**62.3**	42.0	56.9

(a) Validation results for *RL-Profiler* and *RND+CNet* with different settings for top-N.

System	top-N	Open.		Consc.		Extrav.		Agree.		Neur.	
		m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1	m-F_1	w-F_1
Baseline-R	all	49.8	98.9	47.9	88.0	82.5	96.7	59.8	88.2	66.8	73.6
Baseline-B	all	56.5	99.0	58.8	88.8	76.9	93.6	66.2	88.2	67.8	73.6
ALL+CNet	all	49.8	98.9	47.0	70.5	48.4	92.1	52.5	75.8	42.7	57.0
RL-Profiler	best	47.7	94.6	44.6	63.9	57.0	92.3	43.1	70.8	39.3	47.0
RND+CNet	best	49.6	98.5	33.4	45.7	48.3	91.8	41.8	58.1	38.8	46.1
PMI+CNet	best	49.4	98.0	35.4	48.8	58.8	91.4	42.3	58.1	38.0	42.5
PT+CNet	best	49.1	97.5	34.5	48.6	48.2	91.7	36.7	48.1	38.8	50.9
RND+CNet	$\star$	47.0	93.3	28.1	37.5	48.8	90.0	39.9	55.6	38.8	46.1

(b) Results for different methods on testing data.

Table 10: Results for different models predicting levels for each "Big Five" personality trait. (a) shows results of selection based methods on validation data using different settings for top-N. For each method, the best setting w.r.t. macro-average F_1 score (m-F_1) is selected (highlighted with **bold**). Note that for each personality trait a different setting can be selected. (b) shows results on test data for the selected settings and the baseline methods. w-F_1: weighted average F_1. *Prediction performance of *RND+CNet* if using same setting for N as the *RL-Profiler* system on test data.

study observes the largest gap between these models of 9.4PP macro F_1 score. Here, using all tweets results in a macro F_1 score of 52.5%, while for the proposed system 43.1% is observed. On average over all traits, this study finds that the proposed *RL-Profiler* system performs worse by 1.8PP macro-average (46.3% vs. 48.1%) and 5.2PP weighted average (78.9% vs. 73.7%). Given that this study found that the proposed approach outperforms the *ALL+CNet* system on validation data (+7.1PP macro F_1) these results could be a sign, that, during training, the model is overfitting to the limited amount of available validation data.

To answer the second research question of how the proposed system performs compared to an arbitrary selection of tweets in a zero-shot setting, the proposed system is compared to the *RND+CNet* system. Here, this study finds that the learned selection used in the proposed *RL-Profiler* system produces better results across all personality traits except the *openness* trait. For example, when predicting levels of *conscientiousness*, the *RL-Profiler* system achieves 44.6% macro and 63.9% weighted F_1, while the *RND+CNet* system achieves 33.4% and 45.7%. Note that in this comparison, the proposed system is only selecting N=5 tweets while the *RND+CNet* system is selecting N=50 tweets as these are the settings for these systems that produced the best results on the validation data. Interestingly, comparing these system predicting levels of *extraversion*, similar weighted average F_1 scores are observed (92.3% vs. 91.8%). However, the proposed system achieves a much better macro-average score of 57.0% compared to the score of 48.3% observed for the *RND+CNet* system. For *agreeableness*, this is contrariwise. Here, an arbitrary selection results in similar a macro F_1 score, but in significant worse weighted F_1 score (70.8% vs. 58.1%). On average over all traits, using the proposed system instead of the *RND+CNet* system increases macro F_1 score by 3.9PP (46.3% vs. 42.4) and weighted F_1 score by 5.7PP (73.7% vs. 68.0%). This is, although the *RND+CNet* system is using 50 tweets, on four of the five traits, while the proposed system only uses 5 tweets for these traits. Results for the *RL-Profiler* system using the same amount of tweets as the *RL-Profiler* system are also presented in Table 10b and show that, on average over all traits, performance of the *RND+CNet* system further decreases if using the same amount of tweets as the proposed system.

To address the third research question of how the proposed system performs in comparison to supervised learning-based alternatives, the *RL-Profiler* system is compared to the two baseline systems *Baseline-R* and *Baseline-B*. Here, this work finds that both baseline systems outperform the proposed approach across all personality traits. For example, for the *neuroticism* trait a macro-average F_1 score of 67.8% is observed for the BERT-based baseline, while the proposed *RL-Profiler* system only achieves a score of 39.3%. On average over all traits, this study finds that the proposed *RL-Profiler* system performs worse by 15.1PP macro F_1 compared to the *Baseline-R* system (46.3% vs. 61.4%) and 18.9PP macro F_1 compared to the *Baseline-B* system (46.3% vs. 65.2%).

Further, this study finds that the *PMI+CNet* and *PT+CNet* system, perform similar to the *RND+CNet* system for most traits. However, for *extraversion*, similar to the validation results, the results on testing data show that *PMI+CNet* system shows a significant improvement compared to the *RND+CNet* system and also slightly improves compared to the *RL-Profiler* system. Here, the *PMI+CNet* system achieves 58.8% macro-average F_1, while for the *RND+CNet* and *RL-Profiler* system 48.3% and 57.0% are observed, respectively. For *agreeableness*, the *PT+CNet* system is performing significantly worse compared to the other zero-shot systems (e.g. -6.4PP macro F_1 compared to the *RL-Profiler* system). Note that Appendix A.4 presents detailed results for the *PMI+CNet* and *PT+CNet* system, and F_1 scores calculated for the individual classes for each trait and method as well as standard derivation observed between different runs.

6.2 Analysis and Discussion

Overall, this study finds that the proposed system performs significantly worse predicting levels of the "Big Five" personality traits compared to the supervised baselines. However, when compared to zero-shot settings, this work finds that the proposed approach is preferable compared to a random selection when predicting personality. Comparing the proposed approach using only a few tweets of each profile for inference to the system using all tweets of profiles, this study finds that using

Concept	*RL-Profiler*	*RND+CNet*	*ALL+CNet*
Openness	0.54 (5)	1.11 (50)	1.72 (94.3⋆)
Conscientiousness	0.88 (5)	1.29 (50)	2.11 (93.7⋆)
Extraversion	0.38 (5)	1.10 (50)	1.65 (92.4⋆)
Agreeableness	0.61 (5)	1.12 (50)	1.57 (91.9⋆)
Neuroticism	1.12 (30)	1.03 (30)	1.78 (92.1⋆)

Table 11: Average prediction time per profile in seconds on testing data for different zero-shot based systems. Runtime majorly depends on the number of tweets selected by each system. For the *RL-Profiler* and *RND+CNet* system, the setting for N producing the best results on the validation data is shown. The selected number of tweets (N) is shown in parentheses. ⋆Average number of tweets used if using all tweets available from profiles in the testing data.

all tweets results in a slightly better prediction performance for most traits. Sill, prediction performance between these approaches is comparable if averaged over all traits and using only a subset of tweets has the advantage of increasing efficiency as efficiency of LLMs typically scales with input length. In addition, it should be noted that depending on the number of tweets, or text instances in general, in a given profile, using all tweets from profile in a zero-shot setting might not be feasible as such an input to a large language model could exceed the maximum context length of such a model[7]. In this case, an instance selection mechanism is necessarily needed.

Table 11 shows the average prediction time per profile on the testing data for the different methods[8]. For the *RL-Profiler* and *RND+CNet* system these times include time required to select a number of tweets from profiles. Here, runtime majorly depends on the number of tweets selected by each system. When comparing prediction time of the proposed system to the *ALL+CNet* system, this study finds that the proposed system is significantly more efficient. For example, when predicting levels of *extraversion* on the testing data, the average prediction time for a single profile is reduced by more than 76% moving from 1.65 to 0.38 seconds. Note that here, the

[7]In this study, using all tweets of profiles barely fits context length of 4096 of the Llama 2-Chat model: in the testing data, the largest input to the model is 3829 tokens.

[8]All experiments were performed on a single NVIDIA RTX A6000 (48GB) GPU with AMD EPYC 7313 CPU

prediction time for the *RL-Profiler* consists of the time selecting 5 tweets from a profile and the time taken to predict a label from the selected tweets in the zero-shot learning setup, while the prediction time for the *ALL+CNet* system consists of the latter only. However, the proposed system requires task specific training which the *ALL+CNet* and *RND+CNet* system do not. Unfortunately, this work can not report accurate runtimes for training, since computational resources, particularly the GPU, were shared among multiple users. This introduces variability in the training times due to fluctuating demand from other users and concurrent processes. Furthermore, it is difficult to estimate training time based the time required to train single epochs as this also depends on how many tweets the RL agent predicts to be relevant given the current policy. However, this study estimates training time for the proposed system, to be between 15 and 20 hours for each individual trait on the hardware used in this work. For the reported prediction times in Table 11 it was ensured that no concurrent processes from other users are present during evaluation using monitoring tools. Interestingly, runtime of different concepts differs. For example, when predicting levels of *conscientiousness*, the proposed system uses 0.88 seconds for a prediction on average per profile, while for *openness* the system requires only 0.54 seconds on average. One reason for this is that the length of the prompt templates, i.e. the number of tokens per template, for the different concepts differs. Still, although the template for *openness* contains more tokens (169 tokens) compared to the template for *conscientiousness* (154 tokens), on average the system takes less time for a prediction for *openness*. One possible explanation for this is that the RL agent is preferring tweets with more tokens when selecting *relevant* instances for the *conscientiousness* trait. However, this study does not investigate this further and leaves this for future work.

Table 12 presents example tweets selected by the proposed *RL-Profiler* system and the *RND+CNet* system. For each personality trait, a selection of tweets from both methods from one profile in the testing set is shown. Here, both system select tweets from the same profile. For simplicity, selections of both systems using N=5 tweets are shown for all traits. To protect privacy of authors of these tweets, ground-truth labels and predicted labels are not shown in this table. However, for each set

Openness	*RL-Profiler*	I can't even see straight. Up just thinking about life. I'm done breaking my back for MF'ers that can't do shit for me, or do the same for me. I'm laying here wide awake. I Don't want to stay in the house tonight. I want to do something.
	RND+CNet	@USER that would be the perfect time to come. What day in July is his birthday? I'm laying here wide awake. @USER awwwww I know I was too happy when I seen you in my room yesterday. Lol. Man I'm tired as hell. I hate getting up early af to be to work at 7:15. I just want to sleep. In the end it's every man for themself, nobody really has your back like they say they do. & of course I always learn the hard way.
Conscientiousness	*RL-Profiler*	Trying hard just because you want someone to love you when you have to start to loving yourself first. We could be amazing. I'm going for it and no one can stop me. You could dry a river with your heart of stone. The scars will heal, smiles will return, memories will arrive and monsters will disappear.
	RND+CNet	Cause I know somebody, who know somebody, who know somebody who knows!!! I got two letters for you, one of them is F and the other one is U, cause what u gotta do is I go get yourself a clue!!!! Maybe not now, or tomorrow, but eventually things will turn up, I will get better and be able to look back and say with relief: I MADE IT In life you don't get what wish for, you get what you work for. There's nothing than i wouldn't do to make you feel my love.
Extraversion	*RL-Profiler*	I'm just so use to taking the blame for everything because I'd rather say I'm wrong than fight My head hurts I need Mountain Dew so this headache will go away I'm in such a weird mood Ew you're annoying
	RND+CNet	@USER @USER awe I didn't even think about that! I should have! But we have psych! @USER hey I love her haha @USER Loling at my contact pic ?????????????????? I honestly have no clue what I would do without @USER ???????? @USER haha well thank you so are you! Love ya ??????????
Agreeableness	*RL-Profiler*	@USER thanks Dana @USER thanks. @USER everything @USER @USER @USER this one is better ???????? [URL] @USER I LOVE YOU SOOOO MUCH
	RND+CNet	@USER @USER @USER this one is better ???????? [URL] @USER why weren't you at sand?! I don't want to lose you now I'm looking at the other half of me ???? @USER fuck you @USER @USER actually were done so swerve?
Neuroticism	*RL-Profiler*	the backlash what should my bio be i wish i didnt care about my appearance my ratio is trashhhhh i was serious
	RND+CNet	@USER well not anymore bc its broke @USER its a hoodie from my old hs you are spitting so much truth @USER @USER hey why does my boy on the right have sooo much potentional but this vine and his outfit ruined everything *stops talking to everyone but my momma and my dog*

Table 12: Examples of tweet selections by the *RL-Profiler* and *RND+CNet* system from profiles in the testing data. For each of the traits, the systems select tweets from the same profile and lead to different predictions.

of example tweets, either the selection of the *RL-Profiler* system or the *RND+CNet* system leads to the correct ground-truth label predicted by the LLM.

Overall, it is challenging to interpret these example selections as they are only a small subset of the tweets associated to each profile. In particular, it is difficult to assess whether the tweets selected by the proposed system are indeed helpful for predicting an author's true personality. Still, there can be patterns to observe. For example for *neuroticism*, except for the first tweet, the tweets predicted to be relevant by the proposed *RL-Profiler* system could imply a tendency towards worrying ("i wish i didnt care about my appearance") and self-doubt ("my ratio is trashhhhh"). This is interesting as *high* levels of *neuroticism* are associated with such characteristics and therefore predicting such text instances to be relevant makes sense. For the *agreeableness* trait, the proposed system predicts two very short tweets that contain the word "thanks" to be relevant ("@USER thanks Dana", "@USER thanks."). Predicting such tweets to be relevant makes sense as giving thanks to somebody for something can be a sign of kindness or politeness potentially reflecting a *high* level of *agreeableness*. Interestingly, in contrast to this, the random selection of tweets from the same author by the *RND+CNet* system contains a direct insult ("@USER fuck you"). Such a tweet could be interpreted as rude and indicative of *low* level of *agreeableness*. Consequently, this tweet could also be considered relevant when assessing *agreeableness* for the author in question. However, the proposed system did not include this tweet in its selection. When trying to predict a level of *agreeableness* given these two different sets of tweets, it makes sense that the LLM in the *Classification Network* predicts different labels for each set. Presumably, a *high* level of *agreeableness* would be predicted by *RL-Profiler* system, while it is plausible that for the set chosen by the *RND+CNet* system a *low* level would be predicted. Such predictions would likely also be in line with expert annotations given these sets of tweets and no further context. Still, only one of the predictions would be considered "correct" since both sets origin from the same author. This raises the question whether using such a limited number of tweets can lead to confident and robust personality predictions since, from a human perspective, using only five tweets from an author seems not enough for inferring personality attributes confidently.

7 Conclusion and Future Work

In this thesis, a new approach for automatic personality profiling in social media was presented. Instead of attempting to infer personality traits directly from all content associated with a social media user, the novel idea presented in this work is to add an auxiliary filtering task that is automatically selecting a subset of a user's content, and to infer personality solely from this selected subset. With this, this study hypothesized that prediction quality of automatic systems can potentially be enhanced as ambiguity and noise withing a collection of content associated to a user is reduced. Here, instead of relying on costly collection of manually annotated data to learn such a filter, this work proposed a reinforcement learning agent that learns to filter content based on feedback obtained through interactions with a large language models in a zero-shot learning setup.

With this, this study addressed three primary question: first, this work explored if such a learned selection is beneficial compared to directly relying on all content associated to a user profile when inferring personality in a zero-shot setting. Here, this study applied the learned selection of user generated content from the proposed method in a zero-shot learning setup to predict personality, and compared prediction quality of such a setup to a zero-shot setup that uses all content. In the experiment on a real-world dataset, this work finds that the proposed approach achieves comparable yet slightly worse for most of the big five personality traits. For one trait, namely Extraversion, the proposed approach seems preferable. Still, this work finds that the proposed system can be preferable compared to inferring personality from all content associated with users, since using only a small subset of content significantly reduces prediction time in a zero-shot learning setting. This reduction in prediction time can outweigh the computational resources required for training the proposed reinforcement learning agent if performing large scale personality analyses or implementing real-time prediction systems, and can lead to significant cost savings in the long run.

Further, this study explored how performance of the proposed selection based method compares to a system relying on a random selection of content to infer

personality. Therefore, this work compared personality prediction quality of a zero-shot learning setup using the learned selections by the proposed method to arbitrary selections. Here, this work finds that a learned selection is preferable as performance degrades notably for three of the five personality traits. Depending on the amount of shared content by a user, using all content in a zero-shot learning setting might not be feasible since input to LLMs is typically limited. In this case, an instance selection mechanism is necessarily needed, and results on validation data indicate that using a learned selection is preferable to a random selection even if using large proportions of the content shared by a user. Given these results, this study concludes that, to some degree, the proposed selection system is able to distinguish between relevant and irrelevant content shared by users.

To address the third question, this work explored how prediction quality of the proposed system compares to traditional profiling methods that learn from supervision. On this comparison, this work finds that such methods still outperform all zero-shot settings explored in this study including the proposed approach. One reason for this are the limited zero-shot capabilities of today's large language models. Likely, quality of the proposed selection based system largely suffers from error propagation due to these limited capabilities as this system solely learns from the feedback of such a "teacher" model. Therefore, this thesis hypothesizes that abilities of the proposed selection system automatically extracting relevant content are expected to increase in future if capabilities of LLMs increase. Furthermore, this work evaluated the proposed profiling system on artificially enriched profiling data, on which existing profiles were enriched with a small amount of content strongly indicative for specific personality characteristics. Interestingly, in contrast to the experiments on real-world data, the experiments in this setting show that the proposed system significantly outperforms all other methods including the methods that learn using supervision directly, and show that the proposed RL agent is able to distinguish such highly indicative artificially content from less indicative content.

These results highlight the need for future research on using filtering techniques for profiling, and from here, there are several directions for future work: first, one direction would be to replace or adapt individual parts of the proposed system.

For example, one could explore replacing the policy optimization algorithm to foster more robust learning, explore the usage of different large language models, or experiment with different policy parameterization techniques for the reinforcement learning agent. Moreover, the amount of relevant content likely varies among individual profiles. This could, for example, be due to individual activity levels or how individual users engage with social media platforms in general. For simplicity, the amount of selected content by the system proposed in this work is fixed. This is a mismatch and does not align with this variability between individual profiles. Therefore, it also remains interesting to explore how the proposed approach could be adapted to not rely on such subsets of content of fixed length and instead dynamically select amounts of content from individual profiles.

Furthermore, an important research question that arises from this work is how manually annotated data could have changed the results observed this study. In particular, it would be interesting to explore whether expert annotators and the proposed reinforcement learning agent would rate similar content as being relevant, and how a selection model trained to replicate such human annotations to select relevant instances would perform in a zero-shot setting. In addition, since the proposed profiling approach is mostly concept agnostic, it also remains interesting to explore how this approach would perform when applied to predicting other concepts aside from personality such as demographic data like gender or age.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

Vaibhav Adlakha, Parishad BehnamGhader, Xing Han Lu, Nicholas Meade, and Siva Reddy. Evaluating correctness and faithfulness of instruction-following models for question answering, 2023.

Gordon W Allport and Henry S Odbert. Trait-names: A psycho-lexical study. *Psychological monographs*, 47(1):i–171, 1936.

Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Vitor Silva, Leonardo Neves, and Francesco Barbieri. Twitter topic classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3386–3400, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL `https://aclanthology.org/2022.coling-1.299`.

Shlomo Argamon, Sushant Dhawle, Moshe Koppel, and James W Pennebaker. Lexical predictors of personality type. In *Proceedings of the 2005 joint annual meeting of the interface and the classification society of North America*, pages 1–16, 2005.

Shlomo Engelson Argamon, Moshe Koppel, Jonathan Fine, and Anat Rachel Shimoni. Gender, genre, and writing style in formal written texts. *Third Text*, 23: 321–346, 2003. URL `https://api.semanticscholar.org/CorpusID:6980164`.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment, 2021.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 1, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL https://api.semanticscholar.org/CorpusID:11212020.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction, 2017.

Fazlourrahman Balouchzahi, Hosahalli Lakshmaiah Shashirekha, and Grigori Sidorov. (notebook for pan at clef 2021). 2021. URL https://api.semanticscholar.org/CorpusID:237298842.

Angelo Basile, Gareth Dwyer, Maria Medvedeva, Josine Rawee, Hessel Haagsma, and Malvina Nissim. N-gram: New groningen author-profiling model. *CoRR*, abs/1707.03764, 2017. URL http://arxiv.org/abs/1707.03764.

F. Baumgarten. *Die Charaktereigenschaften*. Beiträge zur Charakter- und Persönlichkeitsforschung. A. Francke, 1933.

Richard Bellman. *Dynamic Programming*. Dover Publications, 1957. ISBN 9780486428093.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181.

Janek Bevendorff, Bilal Ghanem, Anastasia Giachanou, Mike Kestemont, Enrique Manjavacas, Ilia Markov, Maximilian Mayerl, Martin Potthast, Francisco Rangel, Paolo Rosso, G ünther Specht, Efstathios Stamatatos, Benno Stein, Matti Wiegmann, and Eva Zangerle. Overview of PAN 2020: Authorship Verification, Celebrity Profiling, Profiling Fake News Spreaders on Twitter, and Style Change Detection. In Avi Arampatzis, Evangelos Kanoulas, Theodora Tsikrika, Stefanos Vrochidis, Hideo Joho, Christina Lioma, Carsten Eickhoff, Aurélie Névéol, Linda Cappellato, and Nicola Ferro, editors, *11th International Conference of*

the CLEF Association (CLEF 2020). Springer, September 2020. URL http://ceur-ws.org/Vol-2696/.

Janek Bevendorff, BERTa Chulvi, Gretel Liz De La Peña Sarracén, Mike Kestemont, Enrique Manjavacas, Ilia Markov, Maximilian Mayerl, Martin Potthast, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, Benno Stein, Matti Wiegmann, Magdalena Wolska, and Eva Zangerle. Overview of PAN 2021: Authorship Verification,Profiling Hate Speech Spreaders on Twitter,and Style Change Detection. In K. Selcuk Candan, Bogdan Ionescu, Lorraine Goeuriot, Birger Larsen, Henning Müller, Alexis Joly, Maria Maistro, Florina Piroi, Guglielmo Faggioli, and Nicola Ferro, editors, *12th International Conference of the CLEF Association (CLEF 2021)*. Springer, 2021.

Adrien Bolland, Gilles Louppe, and Damien Ernst. Policy gradient algorithms implicitly optimize by continuation. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40, 2009.

Lauren Brewer. General psychology: Required reading. *Deiner Education Fund: Salt Lake City, CT, USA*, page 323, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Mart Busger op Vollenbroek, Talvany Carlotto, Tim Kreutz, Masha Medvedeva, Chris Pool, Johannes Bjerva, Hessel Haagsma, and Malvina Nissim. Gronup: Groningen user profiling: Notebook for pan at clef 2016. In Krisztian Balog, Linda Cappellato, Nicola Ferro, and Craig Macdonald, editors, *Working Notes of CLEF 2016*, CEUR, pages 846–857. Workshop Proceedings, September 2016. Conference and Labs of the Evaluation forum (CLEF) ; Conference date: 05-09-2016 Through 08-09-2016.

Sergio Santamaria Carrasco and Roberto Cuervo Rosillo. Loscalis at politices 2022: Political author profiling using beto and maria. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2022). CEUR Workshop Proceedings, CEUR-WS, A Coruna, Spain*, 2022.

Rich Caruana. Multitask learning. *Machine Learning*, 28, 07 1997. doi: 10.1023/A:1007379606734.

Raymond B Cattell. The description of personality: Principles and findings in a factor analysis. *The American journal of psychology*, 58(1):69–90, 1945.

Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading, 2016.

Mara Chinea-Rios, Thomas Müller, Gretel Liz De la Pena Sarrac'en, Francisco Rangel, and Marc Franco-Salvador. Zero and few-shot learning for author profiling. In *International Conference on Applications of Natural Language to Data Bases*, 2022. URL `https://api.semanticscholar.org/CorpusID:248366632`.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL `https://aclanthology.org/D14-1179`.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL `http://arxiv.org/abs/1412.3555`.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990. URL `https://aclanthology.org/J90-1003`.

Michael D. Conover, Bruno Goncalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. Predicting the political alignment of twitter users. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 192–199, 2011. doi: 10.1109/PASSAT/SocialCom.2011.34.

Paul T Costa and Robert R McCrae. *Neo personality inventory-revised (NEO PI-R)*. Psychological Assessment Resources Odessa, FL, 1992.

Nikolas Coupland. *Style: Language variation and identity*. Cambridge University Press, 2007.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

J. Firth. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford, 1957.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023.

Philip Gage. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38, 1994. URL https://api.semanticscholar.org/CorpusID:59804030.

José Antonio García-Díaz, Salud M. Jiménez Zafra, María Teresa Martín Valdivia, Francisco García-Sánchez, Luis Alfonso Ureña López, and Rafael Valencia García. Overview of politices 2022: Spanish author profiling for political ideology, 2022.

Jennifer Golbeck, Cristina Robles, and Karen Turner. Predicting personality with social media. In *CHI '11 Extended Abstracts on Human Factors in Computing*

Systems, CHI EA '11, page 253–262, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450302685. doi: 10.1145/1979742.1979614. URL https://doi.org/10.1145/1979742.1979614.

L. R. Goldberg. An alternative "description of personality": The big-five factor structure. *Journal of Personality and Social Psychology*, 59:1216–1229, 1990.

Lewis R Goldberg. Language and individual differences: The search for universals in personality lexicons. *Review of personality and social psychology*, 2(1):141–165, 1981.

A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4, 2005. doi: 10.1109/IJCNN.2005.1556215.

Yaakov HaCohen-Kerner. Survey on profiling age and gender of text authors. *Expert Systems with Applications*, 199:117140, 2022. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2022.117140. URL https://www.sciencedirect.com/science/article/pii/S095741742200536X.

Zellig S. Harris. Distributional structure. 1954. URL https://api.semanticscholar.org/CorpusID:86680084.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention, 2021.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL https://aclanthology.org/P18-1031.

Ronald A Howard. Dynamic programming and markov processes. 1960.

Oliver P John, Eileen M Donahue, and Robert L Kentle. Big five inventory-the big-five inventory-version 4a and 54. berkeley institute of personality and social research, university of california. 1991.

Oliver P John, Sanjay Srivastava, et al. The big-five trait taxonomy: History, measurement, and theoretical perspectives. 1999.

Dan Jurafsky and James H. Martin. 2024. URL https://web.stanford.edu/~jurafsky/slp3/.

Alam Sher Khan, Hussain Ahmad, Muhammad Zubair Asghar, Furqan Khan Saddozai, Areeba Arif, and Hassan Ali Khalid. Personality classification from online text using machine learning approach. *International journal of advanced computer science and applications*, 11(3):460–476, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Moshe Koppel, Shlomo Engelson Argamon, and Anat Rachel Shimoni. Automatically categorizing written texts by author gender. *Lit. Linguistic Comput.*, 17: 401–412, 2002.

Moshe Koppel, Jonathan Schler, and Kfir Zigdon. Determining an author's native language by mining a text for errors. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, page 624–628, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 159593135X. doi: 10.1145/1081870.1081947. URL https://doi.org/10.1145/1081870.1081947.

Anne Kreuter, Kai Sassenberg, and Roman Klinger. Items from psychometric tests as training data for personality profiling models of Twitter users. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 315–323, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.wassa-1.35. URL https://aclanthology.org/2022.wassa-1.35.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017. URL http://arxiv.org/abs/1703.03130.

X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge: Data Engineering*, 35(01):857–876, jan 2023. ISSN 1558-2191. doi: 10.1109/TKDE.2021.3090866.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. URL `https://api.semanticscholar.org/CorpusID:198953378`.

Jesus Lopez-Santillan, Luis Carlos Gonzalez Gurrola, Manuel Montes, and Adrián López-Monroy. When attention is not enough to unveil a text's author profile: Enhancing a transformer with a wide branch. *Neural Computing and Applications*, 35:1–20, 01 2023. doi: 10.1007/s00521-023-08198-5.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL `https://aclanthology.org/D15-1166`.

Sandra C Matz and Oded Netzer. Using big data as a window into consumers' psychology. *Current opinion in behavioral sciences*, 18:7–12, 2017.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: contextualized word vectors. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6297–6308, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Robert R McCrae and Oliver P John. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215, 1992.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Thomas M Olino and Daniel N Klein. Psychometric comparison of self-and informant-reports of personality. *Assessment*, 22(6):655–664, 2015.

Reynier Ortega-Bueno, Berta Chulvi, Francisco Rangel, Paolo Rosso, and Elisabetta Fersini. Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO). Overview for PAN at CLEF 2022. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast, editors, *CLEF 2022 Labs and Workshops, Notebook Papers*. CEUR-WS.org, September 2022. URL `http://ceur-ws.org/Vol-3180/paper-185.pdf`.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

Huseyin Oz. Personality traits and ideal l2 self as predictors of academic achievement among prospective english teachers. In *ICERI2015 Proceedings*, pages 5833–5841. IATED, 2015.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1244. URL `https://aclanthology.org/D16-1244`.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

James Pennebaker, Matthias Mehl, and Kate Niederhoffer. Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54: 547–77, 02 2003. doi: 10.1146/annurev.psych.54.101601.145041.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1161. URL `https://aclanthology.org/P17-1161`.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://aclanthology.org/N18-1202`.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

Beatrice Rammstedt and Oliver P John. Measuring personality in one minute or less: A 10-item short version of the big five inventory in english and german. *Journal of research in Personality*, 41(1):203–212, 2007.

Francisco Rangel and Paolo Rosso. Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. In Linda Cappellato, Nicola Ferro, David E. Losada, and Henning Müller, editors, *CLEF 2019 Labs and Workshops, Notebook Papers*. CEUR-WS.org, September 2019. URL `http://ceur-ws.org/Vol-2380/`.

Francisco Rangel, Paolo Rosso, Moshe Koppel, Efstathios Stamatatos, and Giacomo Inches. Overview of the author profiling task at pan 2013. In *CLEF conference on multilingual and multimodal information access evaluation*, pages 352–365. CELCT, 2013.

Francisco Rangel, Paolo Rosso, Irina Chugur, Martin Potthast, Martin Trenkmann, Benno Stein, Ben Verhoeven, and Walter Daelemans. Overview of the 2nd author profiling task at pan 2014. In *CLEF 2014 Evaluation Labs and Workshop Working Notes Papers, Sheffield, UK, 2014*, pages 1–30, 2014.

Francisco Rangel, Fabio Celli, Paolo Rosso, Martin Potthast, Benno Stein, Walter Daelemans, et al. Overview of the 3rd author profiling task at pan 2015. In *CLEF2015 Working Notes. Working Notes of CLEF 2015-Conference and Labs of the Evaluation forum.* Notebook Papers, 2015.

Francisco Rangel, Anastasia Giachanou, Bilal Ghanem, and Paolo Rosso. Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In Linda Cappellato, Carsten Eickhoff, Nicola Ferro, and Auré

lie Névéol, editors, *CLEF 2020 Labs and Workshops, Notebook Papers*. CEUR-WS.org, September 2020. URL http://ceur-ws.org/Vol-2696/.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization, 2022.

Jonathan Schler, Moshe Koppel, Shlomo Engelson Argamon, and James W. Pennebaker. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, 2006. URL https://api.semanticscholar.org/CorpusID:2075411.

H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791, 2013.

Ashish Kumar Shakya, Gopinatha Pillai, and Sohom Chakrabarty. Reinforcement learning algorithms: A brief survey. *Expert Systems with Applications*, 231:120495, 2023. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2023.120495. URL https://www.sciencedirect.com/science/article/pii/S0957417423009971.

Marco Siino, elisa nuovo, Ilenia Tinnirello, and Marco La Cascia. Detection of hate speech spreaders using convolutional neural networks. 09 2021.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Sanja Stajner and Seren Yenikent. A survey of automatic personality detection from texts. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6284–6295, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.553. URL `https://aclanthology.org/2020.coling-main.553`.

Efstathios Stamatatos, Martin Potthast, Francisco Rangel, Paolo Rosso, and Benno Stein. Overview of the pan/clef 2015 evaluation lab. In *Proceedings of the 6th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction - Volume 9283*, CLEF'15, page 518–538, Berlin, Heidelberg, 2015. Springer-Verlag. ISBN 9783319240268. doi: 10.1007/978-3-319-24027-5_49. URL `https://doi.org/10.1007/978-3-319-24027-5_49`.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022.

Octavia-Maria Sulea and Daniel Dichiu. Automatic profiling of twitter users based on their tweets: Notebook for pan at clef 2015. In *CLEF (Working Notes)*, pages 1–8, 2015.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

Narjes Tahaei, Harsh Verma, Parsa Bagherzadeh, Farhood Farahnak, Nadia Sheikh, and Sabine Bergler. Identifying author profiles containing irony or spreading stereotypes with sbert and emojis. 2022.

Eric Tellez, Sabino Miranda, Daniela Moctezuma, Mario Graff, Vladimir Salgado, and José Ortiz-Bejar. Gender identification through multi-modal tweet analysis using microtc and bag of visual words notebook for pan at clef 2018. 07 2018.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross

Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Andranik Tumasjan, Timm Sprenger, Philipp Sandner, and Isabell Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *Proceedings of the International AAAI Conference on Web and Social Media*, 4(1):178–185, May 2010. doi: 10.1609/icwsm.v4i1.14009. URL https://ojs.aaai.org/index.php/ICWSM/article/view/14009.

Jakob Uszkoreit. Transformer: A novel neural network architecture for language understanding. 2017. URL https://blog.research.google/2017/08/transformer-novel-neural-network.html.

Alex I. Valencia-Valencia, Helena Gómez-Adorno, Christophe Rhodes, and Gibran Fuentes Pineda. Bots and gender identification based on stylometry of tweet minimal structure and n-grams model. In *Conference and Labs of the Evaluation Forum*, 2019. URL https://api.semanticscholar.org/CorpusID:198489670.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Letian Wang, Jie Liu, Hao Shao, Wenshuo Wang, Ruobing Chen, Yu Liu, and Steven L. Waslander. Efficient reinforcement learning for autonomous driving with parameterized skills and priors, 2023.

Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth Conference on Uncertainty*

in Artificial Intelligence, UAI'01, page 538–545, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608001.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022a. URL `https://openreview.net/forum?id=gEZrGCozdqR`.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022b.

L. Weng. "a (long) peek into reinforcement learning." lil'log. 19 feb 2018, 2018. `https://lilianweng.github.io/posts/2018-02-19-rl-overview/` [Accessed: (26 Feb 2024)].

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Stephan Winter, Ewa Maslowska, and Anne L Vos. The effects of trait-based personalization in social media advertising. *Computers in Human Behavior*, 114:106525, 2021.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.

Tianyang Zhang, Minlie Huang, and Li Zhao. Learning structured representation for text classification via reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Ed-*

ucational Advances in Artificial Intelligence, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2023.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Appendix

A.1 Prompt Templates: Zero-shot Personality Classification

The following shows the prompt templates used for predicting levels of the "Big Five" traits in a zero-shot learning setup. The templates used in this study verbalize questions from the BFI-44 (John et al., 1991) to inform the LLM about individual traits. The goal of the LLM is to distinguish between a *low* or *high* level of a particular personality trait. Therefore, questionnaire items that score such a particular trait are added as context for a *high* level, while items with reversed scores are added as context for a *low* level. Note that it is not allowed to reprint the full questionnaire without permission. The following prompts therefore only contain two items for each trait. For a full list of items associated to each trait see John et al. (1999). In the templates, plus signs indicate line breaks added for readability not present in the actual prompts. In these prompts, this study uses system prompt "one word response". For simplicity, this system prompt is omitted in the following templates except for the template for *openness*.

Openness: Zero-shot Learning

```
<s>[INST] <<SYS>>
one word response
<</SYS>>

Recall the personality trait openness.
A person with a high level of openness may see themselves
+as someone who is original and comes up with new ideas, or {...}
A person with a low level of openness may see themselves
+as someone who prefers work that is routine, or {...}

Consider the following tweets written by the same person:
{tweets}
Does this person show a low or high level of openness?
+Do not give an explanation. [/INST]
```

Listing A.1: Prompt template for predicting a *low* or *high* level of *openness*.

Conscientiousness: Zero-shot Learning

```
Recall the personality trait conscientiousness.
A person with a high level of conscientiousness may see themselves
+as someone who does a thorough job, or {...}
A person with a low level of conscientiousness may see themselves
+as someone who can be somewhat careless, or {...}

Consider the following tweets written by the same person:
{tweets}
Does this person show a low or high level of conscientiousness?
+Do not give an explanation.
```

Listing A.2: Prompt template for predicting levels of *conscientiousness*.

Extraversion: Zero-shot Learning

```
Recall the personality trait extraversion.
A person with a high level of extraversion may see themselves
+as someone who is talkative, or {...}
A person with a low level of extraversion may see themselves
+as someone who is reserved, or {...}

Consider the following tweets written by the same person:
{tweets}
Does this person show a low or high level of extraversion?
+Do not give an explanation.
```

Listing A.3: Prompt template for predicting a *low* or *high* level of *extraversion*.

```
Consider the following tweets written by the same person:
{tweets}
Classify the writer as either extrovert or introvert.
```

Listing A.4: Alternative template for predicting levels of *extraversion*.[1]

[1] For *extraversion* an alternative template was explored. However, this study found that both result in equal classification performance. In the alternative template, "extrovert" corresponds to a *high* level and "introvert" to a *low* level of *extraversion*. Note that here, the spelling "extrOvert" is used since further experiments in this study showed that using "extrAvert" instead negatively impacts prediction performance.

Agreeableness: Zero-shot Learning

```
Recall the personality trait agreeableness.
A person with a high level of agreeableness may see themselves
+as someone who is helpful and unselfish with others, or {...}
A person with a low level of agreeableness may see themselves
+as someone who tends to find fault with others, or {...}

Consider the following tweets written by the same person:
{tweets}
Does this person show a low or high level of agreeableness?
+Do not give an explanation.
```

Listing A.5: Prompt template for predicting levels of *agreeableness*.

Neuroticism: Zero-shot Learning

```
Recall the personality trait neuroticism.
A person with a high level of neuroticism may see themselves
+as someone who worries a lot, or {...}
A person with a low level of neuroticism may see themselves
+as someone who is relaxed or handles stress well, or {...}

Consider the following tweets written by the same person:
{tweets}
Does this person show a low or high level of neuroticism?
+Do not give an explanation.
```

Listing A.6: Prompt template for predicting a *low* or *high* level of *neuroticism*.

A.2 Prompt Templates: Artificial Data Generation

The following shows the prompt templates used for generating additional text instances (tweets) used for enriching data. Here, individual prompts are used for the personality traits *openness*, *conscientiousness*, *extraversion*, *agreeableness*, and *neuroticism*. Similarly to the prompts presented in Appendix A.1, these prompts verbalize BFI-44 items to enrich context of the large language model. Further, to encourage diversity while generating tweets the LLM is asked to include a certain high-level topic in the generated tweets. This topic is chosen from a set of topics derived from the work by Antypas et al. (2022) This set includes the topics:

- News
- Social Concern
- Sports
- Music
- Celebrity & Pop Culture
- Film, TV & Video
- Diaries & Daily Life
- Arts & Culture
- Science & Technology
- Fitness & Health
- Family
- Relationships

To encourage diversity of tweets further, two different prompts are used for either a *high* and a *low* level of a particular personality trait. Here, one of these prompt includes background about the specific trait from the BFI-44 questionnaire, while the other prompt does not. Both prompts are used to generate 10 tweets. For example, for a *low* level of *openness*, for each of the 12 topics, both prompts generate 10 tweets for a total of 240 tweets.

In these prompts, this study uses the default Llama 2-Chat system prompt (*You are a helpful, respectful and honest assistant. ..., please don't share false information.*). For simplicity, this system prompt is omitted in the following templates. Plus signs indicate that line breaks are added for visibility. These line breaks are not present in the actual prompts used in this study. In addition to these templates, the following shows examples of generated tweets for each personality trait. Here, for each topic, trait and class one tweet is presented.

```
Recall the personality trait openness.
A person with a high level of openness may see themselves
+as someone who is original and comes up with new ideas, or {...}
Generate ten tweets that are likely written by a person
+with a high level of openness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

```
Recall the personality trait openness.
Generate ten tweets that are likely written by a person
+with a high level of openness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompts for generating tweets indicating a *high* level of *openness*.

Topic	Example Tweet
News	I'm not just a thinker, I'm a dreamer. And I believe that dreams can change the world.
Social Concern	Why are we still tolerating systemic injustices? It's time for change.
Sports	Just discovered a new sport that combines elements of surfing, parkour, and breakdancing. It's called 'extreme flowboarding' and it's going to change the game.
Music	I love how music can transport me to another world, even if just for a few minutes
Celebrity & Pop Culture	The fashion at the Met Gala was incredible this year. So many risks and so much creativity on display.
Film, TV & Video	I'm always on the lookout for innovative filmmakers who push the boundaries of storytelling. Anyone have any recommendations?
Diaries & Daily Life	I'm not afraid to be different. In fact, I embrace it. Being unique is what makes life worth living.
Arts & Culture	I believe that art and creativity are the lifeblood of humanity. They bring us together and make us whole.
Science & Technology	I'm loving this quantum computing course I'm taking online. The possibilities for innovation are endless!
Fitness & Health	Currently experimenting with a new meditation technique. Anyone have any tips for staying focused during meditation?
Family	Family is a source of comfort and strength, especially during challenging times.
Relationships	I'm so lucky to have a family that values open communication, honesty, and mutual respect, and I strive to do the same in my own relationships.

(b) Examples of artificial tweets supposed to indicate *high* levels of *openness*.

Table A.7: Prompt templates for generating artificial tweets and examples of generated tweets using these prompts for *high* levels of *openness*.

```
Recall the personality trait openness.
A person with a low level of openness may see themselves
+as someone who prefers work that is routine, or {...}
Generate ten tweets that are likely written by a person
+with a low level of openness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

```
Recall the personality trait openness.
Generate ten tweets that are likely written by a person
+with a low level of openness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompts for generating tweets indicating a *low* level of *openness*.

Topic	Example Tweet
News	Just read a book, but I found it too confusing. I prefer books with clear, straightforward plots.
Social Concern	I don't care about the homeless crisis in our city, it's not my problem.
Sports	Just finished a 9-to-5 shift at the office and now I'm ready to watch some football!
Music	Classical music is the only real music. Everything else is just noise.
Celebrity & Pop Culture	I don't care about fashion or style. As long as I look decent, I'm good.
Film, TV & Video	The new sci-fi series on Netflix is okay, but I prefer something with a clear beginning, middle, and end. No need for complicated storylines.
Diaries & Daily Life	I prefer a structured and organized daily routine. Change is stressful and unnecessary.
Arts & Culture	I don't read books. I don't see the point of wasting time on made-up stories.
Science & Technology	I don't see the point of space exploration. We have enough problems on Earth
Fitness & Health	I've been doing the same workout routine for years and it's worked well for me. No need to change.
Family	I appreciate the traditions that have been passed down in my family, and I try to continue them with my own children.
Relationships	I prefer to have a clear plan and structure in my relationships, no ambiguity.

(b) Examples of artificial tweets supposed to indicate *low* levels of *openness*.

Table A.8: Prompt templates for generating artificial tweets and examples of generated tweets using these prompts for *low* levels of *openness*.

```
Recall the personality trait conscientiousness.
A person with a high level of conscientiousness may see themselves
+as someone who does a thorough job, or {...}
Generate ten tweets that are likely written by a person
+with a high level of conscientiousness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

```
Recall the personality trait conscientiousness.
Generate ten tweets that are likely written by a person
+with a high level of conscientiousness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompts for generating tweets indicating a *high* level of *conscientiousness*.

Topic	Example Tweet
News	Proud to have completed a critical task ahead of schedule and under budget. Our clients will be thrilled with the results.
Social Concern	I believe that small changes can make a big difference. Let's work together to create a more sustainable and equitable future for all.
Sports	I can't believe I missed my usual gym session this morning. I'm so disciplined about my fitness routine, but today was a total lapse in judgment.
Music	I can't stand it when people don't take their craft seriously. Music is a form of art, after all!
Celebrity & Pop Culture	I can't believe the latest scandal involving [celebrity name] - it's so sad that they would stoop to such levels.
Film, TV & Video	I've been binge-watching a new TV show and I'm blown away by the attention to detail in every scene.
Diaries & Daily Life	I just spent an hour tidying up my living space and it feels so much better now. It's amazing how clutter can weigh us down mentally and physically.
Arts & Culture	I've been collecting art and artifacts from around the world, each piece telling a story and reflecting a piece of history. The beauty of art is that it transcends borders and time.
Science & Technology	I take pride in my work, especially when it comes to ensuring the accuracy and attention to detail in my research and experiments.
Fitness & Health	Just prepped my healthy lunch for the week - feeling productive and on top of my game!
Family	My family is my rock. They keep me grounded and remind me of what's truly important in life. My family is my rock. They keep me grounded and remind me of what's truly important in life..
Relationships	I value honesty and integrity in my relationships. I believe in being transparent and truthful, even if it's difficult to have the conversation.

(b) Examples of artificial tweets supposed to indicate *high* levels of *conscientiousness*.

Table A.9: Prompt templates for generating artificial tweets and examples of generated tweets using these prompts for *high* levels of *conscientiousness*.

```
Recall the personality trait conscientiousness.
A person with a low level of conscientiousness may see themselves
+as someone who tends to be disorganized, or {...}
Generate ten tweets that are likely written by a person
+with a low level of conscientiousness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

```
Recall the personality trait conscientiousness.
Generate ten tweets that are likely written by a person
+with a low level of conscientiousness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompts for generating tweets indicating a *low* level of *conscientiousness*.

Topic	Example Tweet
News	Can't find my homework... or my textbook... or my notes. Anyone have a photocopy?
Social Concern	Why do people always expect me to be on time? Like, it's not like I'm gonna change the world or anything
Sports	I think I might have accidentally signed up for a relay instead of a solo race
Music	Just spent an entire concert texting my friends instead of actually paying attention to the music
Celebrity & Pop Culture	I just saw a celebrity at the grocery store and I totally freaked out I can't believe I forgot to buy milk though
Film, TV & Video	I'm so addicted to my favorite TV show that I can't stop thinking about it. I need help!
Diaries & Daily Life	I just spent $100 on a new outfit instead of paying my rent. Oopsie.
Arts & Culture	Just spent an hour at the museum and I still can't remember the name of that one painting. Oh well, it was pretty anyway
Science & Technology	Why do we need to study history when we have memes?
Fitness & Health	I just ate a whole bag of potato chips without realizing it. Guess I won't be hitting the gym today
Family	My mom's constant questions about my love life are getting on my nerves. Can't she just mind her own business?
Relationships	I know I said I would call my partner back yesterday, but uh... I forgot?

(b) Examples of artificial tweets supposed to indicate *low* levels of *conscientiousness*.

Table A.10: Prompt templates for generating artificial tweets and examples of generated tweets using these prompts for *low* levels of *conscientiousness*.

```
Recall the personality trait extraversion.
A person with a high level of extraversion may see themselves
+as someone who is talkative, or {...}
Generate ten tweets that are likely written
+by a person with a high level of extraversion.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

```
Recall the personality trait extraversion.
Generate ten tweets that are likely written
+by a person with a high level of extraversion.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompts for generating tweets indicating a *high* level of *extraversion*.

Topic	Example Tweet
News	I'm so excited to share the latest scoop with all my followers!
Social Concern	Who's ready to take on a challenge? Let's organize a charity run to support children's education!
Sports	I'm not just a fan of my favorite sports team - I'm a superfan! Can't get enough of the action on the field!
Music	I'm so excited to be going to see my favorite band in concert tonight!
Celebrity & Pop Culture	Just spent the day at Comic-Con and met some of my favorite celebrities!
Film, TV & Video	I can't believe I landed a role in the newest Netflix series! This is going to be a dream come true!
Diaries & Daily Life	I love being around people! There's nothing like a good conversation to get my energy going.
Arts & Culture	Just spent the entire day exploring the city's vibrant street art scene! So much talent and creativity everywhere.
Science & Technology	Just patented a groundbreaking new tech that's gonna make waves in the industry - watch this space!
Fitness & Health	Feeling so strong and confident after a killer leg day at the gym.
Family	Family dinners are the best! The more the merrier, especially when there's good food and great conversation.
Relationships	I'm not scared of rejection. I'll put myself out there and see what happens!

(b) Examples of artificial tweets supposed to indicate *high* levels of *extraversion*.

Table A.11: Prompt templates for generating artificial tweets and examples of generated tweets using these prompts for *high* levels of *extraversion*.

```
Recall the personality trait extraversion.
A person with a low level of extraversion may see themselves
+as someone who is reserved, or {...}
Generate ten tweets that are likely written by a person
+with a low level of extraversion.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

```
Recall the personality trait extraversion.
Generate ten tweets that are likely written by a person
+with a low level of extraversion.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompts for generating tweets indicating a *low* level of *extraversion*.

Topic	Example Tweet
News	I'm not a morning person. In fact, I'm not really a person at all until after noon. Just giving you fair warning.
Social Concern	I'm not sure how to navigate certain social situations, but I'm trying to be patient with myself and learn from experience.
Sports	I prefer to focus on my own improvement rather than comparing myself to others.
Music	My favorite way to relax is to listen to calming music and meditate.
Celebrity & Pop Culture	The obsession with celebrity culture is a reflection of our society's deep-seated insecurity and need for validation.
Film, TV & Video	I prefer to watch TV alone, rather than with a group of people. It's just more relaxing that way.
Diaries & Daily Life	Just spent an hour staring at a wall, and I'm not even joking. It was a great conversation starter.
Arts & Culture	The beauty of a well-crafted poem or novel lies in its ability to convey complex emotions with simplicity and elegance.
Science & Technology	My mind is always racing with ideas, but I struggle to express them out loud.
Fitness & Health	I'm not a fan of loud, crowded gyms, I prefer to work out at home in my own space.
Family	I love my family, but sometimes I just need a little alone time to recharge.
Relationships	I value intimacy and depth in my relationships, rather than superficial or surface-level connections.

(b) Examples of artificial tweets supposed to indicate *low* levels of *extraversion*.

Table A.12: Prompt templates for generating artificial tweets and examples of generated tweets using these prompts for *low* levels of *extraversion*.

```
Recall the personality trait agreeableness.
A person with a high level of agreeableness may see themselves
+as someone who is helpful and unselfish with others, or {...}
Generate ten tweets that are likely written by a person
+with a high level of agreeableness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

```
Recall the personality trait agreeableness.
Generate ten tweets that are likely written by a person
+with a high level of agreeableness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompts for generating tweets indicating a *high* level of *agreeableness*.

Topic	Example Tweet
News	Let's work towards creating a more compassionate and equitable society, where everyone has the chance to succeed.
Social Concern	Everyone deserves access to quality Healthcare, regardless of their background or financial situation.
Sports	I can't stand when people get too competitive and forget the true spirit of the game. Let's just have fun out there!
Music	I'm so grateful for all the talented musicians out there who use their platform to spread positivity and uplift others.
Celebrity & Pop Culture	I love how the royal family is using their platform to bring attention to important causes.
Film, TV & Video	I love how film and TV can transport us to different worlds and allow us to escape reality for a little while. But it's also important to remember that reality is what makes these fictional worlds so special.
Diaries & Daily Life	I think it's important to be open-minded and accepting of others.
Arts & Culture	Theater has a way of capturing the human experience in a way that's both relatable and thought-provoking. I can't get enough of live performances!
Science & Technology	I believe that AI can be a powerful tool for good. Let's work together to ensure it's used ethically.
Fitness & Health	I'm so grateful for my fitness community - they inspire me to be my best self every day.
Family	I cherish the laughter, the inside jokes, and the quiet moments with my family. They bring joy and meaning to my life.
Relationships	My friend's success is my success. I'm so proud of their achievements and feel lucky to be a part of their journey.

(b) Examples of artificial tweets supposed to indicate *high* levels of *agreeableness*.

Table A.13: Prompt templates for generating artificial tweets and examples of generated tweets using these prompts for *high* levels of *agreeableness*.

```
Recall the personality trait agreeableness.
A person with a low level of agreeableness may see themselves
+as someone who tends to find fault with others, or {...}
Generate ten tweets that are likely written by a person
+with a low level of agreeableness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

```
Recall the personality trait agreeableness.
Generate ten tweets that are likely written by a person
+with a low level of agreeableness.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompts for generating tweets indicating a *low* level of *agreeableness*.

Topic	Example Tweet
News	I can't believe the media is still covering that story, it's such a non-issue.
Social Concern	I don't understand why everyone is so sensitive these days. Can't people just take a joke and move on?
Sports	Why should I have to follow the rules? The other team is always cheating anyway.
Music	I don't like when people sing along to songs they don't know the lyrics to. It's so fake and pretentious.
Celebrity & Pop Culture	I don't care if [celebrity] wins any awards. They're not that talented anyway.
Film, TV & Video	The pacing in this video game is so slow. I can't believe I have to grind for hours just to progress the story.
Diaries & Daily Life	I'm so sick of all these pointless traditions and rituals. Can't we just do what we want for once?
Arts & Culture	The only good thing about modern art is that it's a great conversation starter. But let's be real, it's just a bunch of nonsense.
Science & Technology	Technology is ruining our society. We need to go back to simpler times.
Fitness & Health	I'm not here to make friends, I'm here to get in shape. Don't bother trying to small talk with me during our gym session.
Family	My family is always trying to tell me what to do. Newsflash: I don't need their advice.
Relationships	I'm not going to change who I am just to make my partner happy. If they can't accept me for who I am, then it's their loss.

(b) Examples of artificial tweets supposed to indicate *low* levels of *agreeableness*.

Table A.14: Prompt templates for generating artificial tweets and examples of generated tweets using these prompts for *low* levels of *agreeableness*.

```
Recall the personality trait neuroticism.
A person with a high level of neuroticism may see themselves
+as someone who worries a lot, or {...}
Generate ten tweets that are likely written by a person
+with a high level of neuroticism.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

```
Recall the personality trait neuroticism.
Generate ten tweets that are likely written by a person
+with a high level of neuroticism.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompts for generating tweets indicating a *high* level of *neuroticism*.

Topic	Example Tweet
News	I can't believe what I just heard on the news. It's like, what is even happening?!
Social Concern	I'm so worried about my own life and the lives of the people I care about. It's like, I can't shake off this feeling of uncertainty and fear.
Sports	I'm so nervous about the big game this weekend. I can't stop thinking about all the things that could go wrong.
Music	Music is my therapy, but sometimes it feels like the only thing I'm good at is making myself feel worse.
Celebrity & Pop Culture	I just spent an entire day analyzing every detail of my favorite celebrity's outfit from last night's red carpet event.
Film, TV & Video	I can't stop worrying about the latest episode of my favorite TV show. Is the main character going to make it out alive?
Diaries & Daily Life	I feel so overwhelmed by my to-do list. I'm not sure how I'll ever get it all done.
Arts & Culture	Why can't I just enjoy a simple painting without overanalyzing every brushstroke?
Science & Technology	I'm convinced that my smart home is spying on me. I mean, have you seen the amount of data it collects?
Fitness & Health	I'm so stressed out about my upcoming health check-up. What if something is wrong?
Family	My family is always causing drama. I just want peace and quiet!
Relationships	My ex-partner still has all of our mutual friends and constantly stalks my profile. It's like they're trying to torture me with memories of our failed relationship.

(b) Examples of artificial tweets supposed to indicate *high* levels of *neuroticism*.

Table A.15: Prompt templates for generating artificial tweets and examples of generated tweets using these prompts for *high* levels of *neuroticism*.

```
Recall the personality trait neuroticism.
A person with a low level of neuroticism may see themselves
+as someone who is relaxed or handles stress well, or {...}
Generate ten tweets that are likely written by a person
+with a low level of neuroticism.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

```
Recall the personality trait neuroticism.
Generate ten tweets that are likely written by a person
+with a low level of neuroticism.
+Do not use emojis or hashtags. Try to include the topic {topic}.
```

(a) Prompts for generating tweets indicating a *low* level of *neuroticism*.

Topic	Example Tweet
News	I'm not one to get easily rattled by news headlines. I prefer to focus on the facts and make informed decisions.
Social Concern	I'm frustrated by the state of our education system. Let's work together to ensure every child has access to quality education and resources!
Sports	I'm so grateful for my morning runs - they help me clear my head and start the day feeling centered and focused.
Music	I love the way music can evoke emotions and transport me to another time and place - it's a powerful force.
Celebrity & Pop Culture	I don't stress about fashion or beauty trends. Comfort and simplicity are key for me!
Film, TV & Video	I'm a sucker for a good rom-com and just watched one that made me laugh and cry. Perfect evening!
Diaries & Daily Life	I'm proud of my ability to remain emotionally stable, even in difficult situations.
Arts & Culture	The beauty of nature is a never-ending source of inspiration for my art.
Science & Technology	I'm so grateful for the internet and the access to information it provides. It's truly a game-changer for education and research.
Fitness & Health	I believe that health and fitness is a journey, not a destination - every step forward is a win, no matter how small!
Family	My family may drive me crazy sometimes, but they're always there for me when I need them.
Relationships	My partner and I don't always see eye-to-eye, but we respect each other's opinions and values. That's what makes our relationship strong.

(b) Examples of artificial tweets supposed to indicate *low* levels of *neuroticism*.

Table A.16: Prompt templates for generating artificial tweets and examples of generated tweets using these prompts for *low* levels of *neuroticism*.

A.3 Preliminary Experiments: Detailed Results

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
N=5	Low	**100**	**100**	**100**±0.0	0.0	0.0	0.0±0.0	53.3	80.0	60.0±39.4	40.0	50.0	43.3±47.3
	High	**100**	**100**	**100**±0.0	93.4	94.0	93.6±3.1	98.7	94.0	96.2±3.5	96.8	97.3	97.0±2.5
	avg	**100**	**100**	**100**±0.0	46.7	47.0	46.8±1.6	76.0	87.0	78.1±21.0	68.4	73.7	70.2±24.6
	w-avg	**100**	**100**	**100**±0.0	87.5	88.1	87.8±2.9	95.9	93.1	93.9±5.3	93.2	94.4	93.7±5.0
N=10	Low	100	100	100±0.0	5.8	20.0	9.0±19.1	**70.0**	**100**	**78.3**±23.6	52.5	70.0	57.3±44.2
	High	100	100	100±0.0	94.7	92.0	93.2±2.5	**100**	**94.7**	**97.2**±3.3	98.1	96.7	97.2±3.4
	avg	100	100	100±0.0	50.3	56.0	51.1±9.1	**85.0**	**97.3**	**87.8**±13.4	75.3	83.3	77.3±23.0
	w-avg	100	100	100±0.0	89.1	87.5	87.9±2.1	**98.1**	**95.0**	**96.0**±4.5	95.3	95.0	94.7±5.1
N=20	Low	90.0	90.0	90.0±31.6	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	**56.7**	**90.0**	**66.7**±30.4
	High	99.4	100	99.7±1.0	93.7	98.7	96.1±1.5	93.7	99.3	96.4±1.1	**99.3**	**93.3**	**96.2**±3.5
	avg	94.7	95.0	94.8±16.3	46.8	49.3	48.0±0.7	46.9	49.7	48.2±0.5	**78.0**	**91.7**	**81.4**±16.9
	w-avg	98.8	99.4	99.1±2.9	87.8	92.5	90.1±1.4	87.8	93.1	90.4±1.0	**96.6**	**93.1**	**94.3**±5.1
N=30	Low	70.0	70.0	70.0±48.3	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	5.0	10.0	6.7±21.1
	High	98.1	99.3	98.7±2.3	93.6	98.0	95.7±1.7	93.8	100	96.8±0.0	94.1	95.3	94.7±1.8
	avg	84.0	84.7	84.3±25.2	46.8	49.0	47.9±0.8	46.9	50.0	48.4±0.0	49.6	52.7	50.7±10.9
	w-avg	96.3	97.5	96.9±5.1	87.8	91.9	89.8±1.6	87.9	93.8	90.7±0.0	88.5	90.0	89.2±2.5
N=50	Low	30.0	30.0	30.0±48.3	**10.0**	**10.0**	**10.0**±31.6	0.0	0.0	0.0±0.0	20.0	20.0	20.0±42.2
	High	95.6	100	97.7±1.6	**94.4**	**100**	**97.1**±1.0	93.8	100	96.8±0.0	95.0	100	97.4±1.4
	avg	62.8	65.0	63.9±24.9	**52.2**	**55.0**	**53.5**±16.3	46.9	50.0	48.4±0.0	57.5	60.0	58.7±21.8
	w-avg	91.5	95.6	93.5±4.5	**89.1**	**94.4**	**91.6**±2.9	87.9	93.8	90.7±0.0	90.3	95.0	92.6±3.9

(a) Results for selection based methods on validation data.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Best	Low	100	100	100±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0
	High	100	100	100±0.0	93.8	100	96.8±0.0	92.9	88.0	90.3±3.0	93.3	93.3	93.3±2.9
	avg	100	100	100±0.0	46.9	50.0	48.4±0.0	46.5	44.0	45.2±1.5	46.7	46.7	46.6±1.5
	w-avg	100	100	100±0.0	87.9	93.8	90.7±0.0	87.1	82.5	84.7±2.8	87.5	87.5	87.4±2.7
		ALL+CNet			*Baseline-R*			*Baseline-B*			*ALL+CNet**		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
All	Low	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	51.4	80.0	59.2±38.6	80.0	80.0	80.0±42.2
	High	93.8	100	96.8±0.0	93.8	100	96.8±0.0	98.8	93.3	95.5±7.4	98.8	100	99.4±1.4
	avg	46.9	50.0	48.4±0.0	46.9	50.0	48.4±0.0	75.1	86.7	77.3±21.2	89.4	90.0	89.7±21.8
	w-avg	87.9	93.8	90.7±0.0	87.9	93.8	90.7±0.0	95.8	92.5	93.2±8.3	97.6	98.8	98.2±3.9

(b) Results for different methods on testing data and *ALL+CNet** on validation data.

Table A.17: Detailed results for predicting levels of *openness* on enriched data. (a) shows results on validation data with different settings for N. For each method, the best setting w.r.t. macro-average F_1 score is selected (highlighted with **bold**). (b) shows results on test data for the selected settings and the baseline methods. ±: standard derivation across 10 runs.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
N=5	Low	**100**	**100**	**100**±0.0	16.4	100	28.1±2.4	**20.2**	**100**	**33.6**±1.0	12.3	75.0	21.1±7.3
	High	**100**	**100**	**100**±0.0	100	31.3	47.2±8.8	**100**	**47.3**	**64.2**±1.9	89.7	28.7	43.2±7.9
	avg	**100**	**100**	**100**±0.0	58.2	65.7	37.7±5.6	**60.1**	**73.7**	**48.9**±1.4	51.0	51.8	32.1±6.8
	w-avg	**100**	**100**	**100**±0.0	90.2	39.4	45.0±8.0	**90.6**	**53.5**	**60.6**±1.8	80.6	34.1	40.6±7.5
N=10	Low	48.7	100	64.8±9.8	14.5	100	25.4±1.5	19.2	100	32.2±2.8	14.3	100	25.0±1.1
	High	100	84.7	91.6±3.7	100	21.3	34.8±8.2	100	43.3	60.2±7.0	100	20.0	33.1±6.2
	avg	74.3	92.3	78.2±6.8	57.3	60.7	30.1±4.9	59.6	71.7	46.2±4.9	57.2	60.0	29.1±3.6
	w-avg	94.0	86.5	88.4±4.5	90.0	30.6	33.7±7.4	90.5	50.0	56.9±6.6	89.9	29.4	32.2±5.6
N=20	Low	30.6	100	46.6±5.8	16.2	100	27.9±1.9	17.0	100	29.1±1.9	16.2	100	27.9±1.9
	High	100	68.7	81.2±5.0	100	30.7	46.6±7.6	100	34.7	51.2±6.9	100	30.7	46.5±9.2
	avg	65.3	84.3	63.9±5.3	58.1	65.3	37.2±4.7	58.5	67.3	40.2±4.4	58.1	65.3	37.2±5.5
	w-avg	91.8	72.4	77.2±5.1	90.1	38.8	44.4±6.9	90.2	42.4	48.6±6.3	90.1	38.8	44.3±8.3
N=30	Low	23.4	100	37.9±4.2	16.8	100	28.7±3.2	18.3	100	30.9±1.9	15.6	100	26.9±1.6
	High	100	55.3	70.8±8.1	100	32.7	48.4±11.5	100	40.0	57.0±5.6	100	27.3	42.6±7.3
	avg	61.7	77.7	54.3±6.1	58.4	66.3	38.6±7.3	59.1	70.0	43.9±3.8	57.8	63.7	34.8±4.4
	w-avg	91.0	60.6	67.0±7.6	90.2	40.6	46.1±10.5	90.4	47.1	53.9±5.1	90.1	35.9	40.8±6.6
N=50	Low	22.4	100	36.6±3.0	**19.9**	**100**	**33.2**±2.3	19.6	100	32.7±2.5	**17.8**	**100**	**30.2**±2.4
	High	100	53.3	69.4±5.5	**100**	**46.0**	**62.8**±5.7	100	44.7	61.5±6.1	**100**	**38.0**	**54.7**±7.4
	avg	61.2	76.7	53.0±4.2	**60.0**	**73.0**	**48.0**±4.0	59.8	72.3	47.1±4.3	**58.9**	**69.0**	**42.5**±4.9
	w-avg	90.9	58.8	65.5±5.2	**90.6**	**52.3**	**59.3**±5.3	90.5	51.2	58.1±5.7	**90.3**	**45.3**	**51.8**±6.9

(a) Results for selection based methods on validation data.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Best	Low	100	97.0	98.4±2.5	57.2	93.0	70.8±2.5	72.9	88.0	79.7±2.9	56.0	97.0	70.9±4.5
	High	98.1	100	99.0±1.6	92.4	53.3	67.3±4.6	90.8	78.0	83.8±2.8	96.1	48.7	64.2±8.0
	avg	99.1	98.5	98.7±2.1	74.8	73.2	69.0±3.4	81.9	83.0	81.7±2.8	76.1	72.8	67.6±6.2
	w-avg	98.9	98.8	98.8±1.9	78.3	69.2	68.7±3.6	83.6	82.0	82.2±2.8	80.1	68.0	66.9±6.6
		ALL+CNet			*Baseline-R*			*Baseline-B*			*ALL+CNet**		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
All	Low	78.8	88.0	82.9±4.9	71.4	50.0	58.8±0.0	67.2	74.0	69.4±7.6	14.2	55.0	22.6±5.1
	High	91.6	84.0	87.5±3.4	72.2	86.7	78.8±0.0	82.3	74.7	77.5±6.2	90.4	56.0	69.0±3.8
	avg	85.2	86.0	85.2±4.0	71.8	68.3	68.8±0.0	74.8	74.3	73.4±6.0	52.3	55.5	45.8±3.6
	w-avg	86.4	85.6	85.7±3.9	71.9	72.0	70.8±0.0	76.3	74.4	74.2±5.8	81.5	55.9	63.6±3.6

(b) Results for different methods on testing data and *ALL+CNet** on validation data.

Table A.18: Detailed results for predicting levels of *conscientiousness* on enriched data. (a) shows results on validation data with different settings for N. For each method, the best setting w.r.t. macro-average F_1 score is selected (highlighted with **bold**). (b) shows results on test data for the selected settings and the baseline methods. ±: standard derivation across 10 runs.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
N=5	Low	**95.6**	**100**	**97.7**±3.0	**18.3**	**10.0**	**12.3**±20.2	40.0	13.3	20.0±25.8	34.0	30.0	31.0±23.2
	High	**100**	**97.3**	**98.6**±1.8	**84.2**	**95.3**	**89.3**±2.3	85.3	100	92.0±1.5	86.9	91.3	88.9±2.3
	avg	**97.8**	**98.7**	**98.1**±2.4	**51.3**	**52.7**	**50.8**±10.6	62.6	56.7	56.0±13.6	60.5	60.7	60.0±12.3
	w-avg	**98.5**	**98.3**	**98.3**±2.2	**73.2**	**81.1**	**76.5**±4.5	77.8	85.6	80.0±5.5	78.1	81.1	79.3±5.2
N=10	Low	69.8	73.3	70.6±14.4	18.3	10.0	12.3±20.2	0.0	0.0	0.0±0.0	**71.7**	**66.7**	**68.4**±6.8
	High	94.6	92.7	93.5±3.9	83.3	90.0	86.4±4.6	82.2	92.7	87.1±1.1	**93.4**	**94.0**	**93.7**±2.0
	avg	82.2	83.0	82.1±9.1	50.8	50.0	49.4±11.7	41.1	46.3	43.6±0.6	**82.5**	**80.3**	**81.0**±4.4
	w-avg	90.4	89.4	89.7±5.6	72.5	76.7	74.1±6.5	68.5	77.2	72.6±1.0	**89.8**	**89.4**	**89.4**±2.8
N=20	Low	70.8	70.0	69.3±9.6	18.3	10.0	12.3±20.2	60.0	33.3	42.0±4.2	63.3	56.7	59.1±21.2
	High	94.0	93.3	93.6±2.6	83.9	93.3	88.3±2.7	87.6	94.7	91.0±1.5	91.6	93.3	92.4±3.8
	avg	82.4	81.7	81.5±6.1	51.1	51.7	50.3±11.0	73.8	64.0	66.5±2.8	77.5	75.0	75.7±12.4
	w-avg	90.1	89.4	89.6±3.7	73.0	79.4	75.6±5.1	83.0	84.4	82.8±1.9	86.9	87.2	86.9±6.7
N=30	Low	56.7	33.3	40.7±31.5	18.3	10.0	12.3±20.2	**100**	**50.0**	**65.0**±15.8	66.7	36.7	45.7±8.8
	High	88.2	98.0	92.8±2.5	84.0	94.0	88.7±2.3	**91.0**	**100**	**95.3**±1.6	88.3	95.3	91.6±1.7
	avg	72.5	65.7	66.7±16.8	51.2	52.0	50.5±10.9	**95.5**	**75.0**	**80.1**±8.7	77.5	66.0	68.7±5.1
	w-avg	83.0	87.2	84.1±7.0	73.0	80.0	75.9±4.8	**92.5**	**91.7**	**90.2**±4.0	84.7	85.6	84.0±2.7
N=50	Low	20.0	6.7	10.0±21.1	10.0	3.3	5.0±15.8	0.0	0.0	0.0±0.0	40.0	13.3	20.0±25.8
	High	84.3	100	91.5±1.2	83.8	100	91.2±0.9	83.3	100	90.9±0.0	85.3	100	92.0±1.5
	avg	52.2	53.3	50.7±11.1	46.9	51.7	48.1±8.3	41.7	50.0	45.5±0.0	62.6	56.7	56.0±13.6
	w-avg	73.6	84.4	77.9±4.5	71.5	83.9	76.8±3.4	69.4	83.3	75.8±0.0	77.8	85.6	80.0±5.5

(a) Results for selection based methods on validation data.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Best	Low	84.5	100	91.3±6.2	23.3	6.7	10.1±13.1	96.7	35.0	51.1±5.7	55.0	13.3	21.1±11.2
	High	100	92.0	95.7±3.4	71.6	94.0	81.2±3.1	79.3	99.3	88.2±1.4	73.5	96.0	83.2±1.9
	avg	92.3	96.0	93.5±4.8	47.4	50.3	45.6±7.3	88.0	67.2	69.6±3.5	64.2	54.7	52.1±6.2
	w-avg	95.6	94.3	94.5±4.2	57.8	69.0	60.9±5.0	84.2	81.0	77.6±2.6	68.2	72.4	65.5±4.1
		ALL+CNet			*Baseline-R*			*Baseline-B*			*ALL+CNet**		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
All	Low	100	53.3	68.7±11.7	50.0	100	66.7±0.0	64.7	85.0	71.2±13.9	90.0	50.0	63.0±26.3
	High	84.4	100	91.5±2.2	100	60.0	75.0±0.0	92.7	74.0	79.5±20.2	91.0	100	95.3±2.1
	avg	92.2	76.7	80.1±6.9	75.0	80.0	70.8±0.0	78.7	79.5	75.3±16.7	90.5	75.0	79.1±14.2
	w-avg	88.9	86.7	85.0±4.9	85.7	71.4	72.6±0.0	84.7	77.1	77.1±18.2	90.9	91.7	89.9±6.1

(b) Results for different methods on testing data and *ALL+CNet** on validation data.

Table A.19: Detailed results for predicting levels of *extraversion* on enriched data. (a) shows results on validation data with different settings for N. For each method, the best setting w.r.t. macro-average F_1 score is selected (highlighted with **bold**). (b) shows results on test data for the selected settings and the baseline methods. ±: standard derivation across 10 runs.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
N=5	Low	**97.5**	**100**	**98.7**±2.1	51.9	85.5	64.5±4.9	44.3	71.8	54.7±4.7	59.7	73.6	65.8±4.3
	High	**100**	**98.0**	**99.0**±1.7	79.3	41.3	53.9±9.7	62.8	34.0	43.8±4.6	76.8	63.3	69.3±4.3
	avg	**98.8**	**99.0**	**98.8**±1.9	65.6	63.4	59.2±7.1	53.5	52.9	49.3±3.6	68.2	68.5	67.5±4.1
	w-avg	**98.9**	**98.9**	**98.9**±1.8	67.7	60.0	58.4±7.4	54.9	50.0	48.4±3.6	69.6	67.7	67.8±4.1
N=10	Low	82.7	100	90.3±4.3	51.7	88.2	65.0±5.0	50.7	90.9	65.0±2.8	**69.5**	**85.5**	**76.5**±7.6
	High	100	84.0	91.1±4.6	81.2	38.7	51.7±12.2	83.1	34.7	48.5±9.5	**87.1**	**72.0**	**78.7**±7.3
	avg	91.3	92.0	90.7±4.5	66.4	63.4	58.4±8.3	66.9	62.8	56.8±6.1	**78.3**	**78.7**	**77.6**±7.4
	w-avg	92.7	90.8	90.8±4.5	68.7	59.6	57.3±8.9	69.4	58.5	55.5±6.7	**79.6**	**77.7**	**77.8**±7.4
N=20	Low	70.6	100	82.7±1.6	56.1	92.7	69.7±5.1	56.3	98.2	71.5±2.8	63.5	97.3	76.8±2.9
	High	100	69.3	81.8±2.4	90.3	46.0	60.2±9.6	97.3	44.0	60.4±5.7	97.1	58.7	72.8±4.9
	avg	85.3	84.7	82.3±2.0	73.2	69.4	65.0±7.1	76.8	71.1	66.0±4.1	80.3	78.0	74.8±3.8
	w-avg	87.5	82.3	82.2±2.1	75.8	65.8	64.3±7.4	80.0	66.9	65.1±4.3	82.9	75.0	74.5±4.0
N=30	Low	65.1	98.2	78.3±2.8	54.8	88.2	67.6±3.5	**58.2**	**98.2**	**73.0**±3.2	63.3	99.1	77.1±3.1
	High	98.0	61.3	75.4±3.7	84.8	46.7	60.0±4.4	**97.5**	**48.0**	**64.0**±6.8	99.1	57.3	72.2±6.8
	avg	81.6	79.8	76.8±3.2	69.8	67.4	63.8±3.7	**77.8**	**73.1**	**68.5**±4.9	81.2	78.2	74.7±4.9
	w-avg	84.1	76.9	76.6±3.3	72.1	64.2	63.2±3.8	**80.8**	**69.2**	**67.8**±5.2	84.0	75.0	74.3±5.2
N=50	Low	63.7	95.5	76.3±3.2	**61.5**	**90.9**	**73.2**±4.6	57.7	86.4	69.0±4.8	61.0	94.5	74.0±2.9
	High	95.3	60.0	73.4±4.0	**90.7**	**58.0**	**70.2**±5.9	85.1	53.3	65.2±5.7	94.0	55.3	69.3±4.4
	avg	79.5	77.7	74.9±3.3	**76.1**	**74.5**	**71.7**±4.9	71.4	69.8	67.1±4.9	77.5	74.9	71.7±3.3
	w-avg	82.0	75.0	74.6±3.4	**78.3**	**71.9**	**71.5**±5.0	73.5	67.3	66.8±5.0	80.0	71.9	71.3±3.4

(a) Results for selection based methods on validation data.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Best	Low	97.5	100	98.7±2.1	60.7	92.7	73.3±5.8	57.3	97.3	72.0±2.2	69.4	100	81.9±3.1
	High	100	98.0	99.0±1.7	92.1	56.0	69.3±6.3	96.4	46.7	62.7±3.9	100	67.3	80.3±4.7
	avg	98.8	99.0	98.8±1.9	76.4	74.4	71.3±5.9	76.8	72.0	67.4±2.8	84.7	83.7	81.1±3.9
	w-avg	98.9	98.9	98.9±1.8	78.8	71.5	71.0±5.9	79.8	68.1	66.6±2.9	87.1	81.2	81.0±4.0
		ALL+CNet			*Baseline-R*			*Baseline-B*			*ALL+CNet**		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
All	Low	73.4	77.3	75.2±2.4	72.7	72.7	72.7±0.0	76.7	58.2	61.2±14.8	73.0	75.4	74.1±4.3
	High	82.8	79.3	80.9±1.9	80.0	80.0	80.0±0.0	73.5	80.7	75.5±9.4	81.5	79.3	80.4±3.7
	avg	78.1	78.3	78.1±2.0	76.4	76.4	76.4±0.0	75.1	69.4	68.3±9.5	77.2	77.4	77.2±4.0
	w-avg	78.8	78.5	78.5±2.0	76.9	76.9	76.9±0.0	74.9	71.2	69.4±9.0	77.9	77.7	77.7±3.9

(b) Results for different methods on testing data and *ALL+CNet** on validation data.

Table A.20: Detailed results for predicting levels of *agreeableness* on enriched data. (a) shows results on validation data with different settings for N. For each method, the best setting w.r.t. macro-average F_1 score is selected (highlighted with **bold**). (b) shows results on test data for the selected settings and the baseline methods. ±: standard derivation across 10 runs.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
N=5	Low	**94.4**	**100**	**97.1**±3.1	38.0	83.8	52.0±9.0	35.3	81.2	49.2±4.8	41.9	82.5	55.5±4.9
	High	**100**	**96.7**	**98.3**±1.8	72.3	26.0	36.7±19.9	68.0	20.7	31.6±5.8	80.8	38.7	51.9±7.4
	avg	**97.2**	**98.3**	**97.7**±2.5	55.1	54.9	44.3±13.3	51.6	51.0	40.4±5.1	61.3	60.6	53.7±5.6
	w-avg	**98.1**	**97.8**	**97.9**±2.3	60.4	46.1	42.0±15.2	56.6	41.7	37.7±5.3	67.2	53.9	53.2±6.1
N=10	Low	94.8	87.5	90.9±5.4	39.1	75.0	51.2±6.0	37.5	90.0	52.9±2.9	46.9	92.5	62.2±4.4
	High	93.6	97.3	95.4±2.8	73.1	37.3	48.7±11.8	79.2	20.0	31.9±4.7	92.6	44.0	59.2±6.2
	avg	94.2	92.4	93.2±4.1	56.1	56.2	50.0±7.8	58.3	55.0	42.4±3.6	69.7	68.2	60.7±4.8
	w-avg	94.0	93.9	93.8±3.7	61.3	50.4	49.6±8.9	64.7	44.4	39.2±4.0	76.7	60.9	60.2±5.1
N=20	Low	87.9	57.5	69.0±7.6	44.4	55.0	48.9±7.3	40.7	72.5	52.1±5.4	**61.8**	**60.0**	**60.3**±10.9
	High	80.9	95.3	87.5±3.1	72.2	62.7	66.9±6.8	75.5	44.0	55.5±3.7	**79.2**	**80.0**	**79.5**±5.1
	avg	84.4	76.4	78.2±5.3	58.3	58.8	57.9±6.5	58.1	58.2	53.8±4.2	**70.5**	**70.0**	**69.9**±7.8
	w-avg	83.3	82.2	81.0±4.6	62.5	60.0	60.7±6.5	63.4	53.9	54.3±4.0	**73.2**	**73.0**	**72.8**±6.9
N=30	Low	86.8	56.2	67.2±9.1	46.4	37.5	40.5±9.4	**43.2**	**58.8**	**49.7**±5.0	56.7	36.2	43.4±11.8
	High	80.5	94.7	86.9±3.2	69.3	75.3	71.9±7.0	**72.7**	**58.7**	**64.9**±3.9	71.5	84.7	77.4±4.6
	avg	83.6	75.5	77.0±6.0	57.8	56.4	56.2±7.0	**57.9**	**58.7**	**57.3**±4.3	64.1	60.5	60.4±7.7
	w-avg	82.7	81.3	80.0±5.1	61.3	62.2	61.0±6.7	**62.4**	**58.7**	**59.6**±4.1	66.3	67.8	65.5±6.6
N=50	Low	90.0	26.2	40.4±5.3	**78.3**	**22.5**	**34.7**±11.3	50.0	21.2	29.2±12.5	70.0	13.8	22.3±10.7
	High	71.4	98.0	82.6±1.9	**70.1**	**96.7**	**81.2**±2.9	68.3	90.0	77.6±1.9	67.6	96.0	79.3±2.8
	avg	80.7	62.1	61.5±3.5	**74.2**	**59.6**	**58.0**±7.1	59.1	55.6	53.4±6.6	68.8	54.9	50.8±6.4
	w-avg	77.8	73.0	67.9±2.9	**73.0**	**70.9**	**65.1**±5.8	61.9	66.1	60.7±4.8	68.4	67.4	59.5±5.2

(a) Results for selection based methods on validation data.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Best	Low	99.3	93.3	96.2±1.0	67.5	10.7	17.9±14.2	85.0	57.3	68.4±7.1	77.8	42.0	54.2±6.6
	High	93.7	99.3	96.4±1.1	52.4	98.0	68.3±2.6	67.9	90.0	77.4±4.6	60.4	88.0	71.6±2.0
	avg	96.5	96.3	96.3±1.1	60.0	54.3	43.1±8.2	76.5	73.7	72.9±5.8	69.1	65.0	62.9±4.2
	w-avg	96.5	96.3	96.3±1.1	60.0	54.3	43.1±8.2	76.5	73.7	72.9±5.8	69.1	65.0	62.9±4.2
		ALL+CNet			*Baseline-R*			*Baseline-B*			*ALL+CNet**		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
All	Low	100	18.0	30.3±6.4	76.5	86.7	81.2±0.0	70.2	58.0	60.4±12.9	90.0	25.0	38.0±19.2
	High	55.0	100	70.9±1.1	84.6	73.3	78.6±0.0	64.8	72.0	66.5±7.2	71.6	100	83.4±2.7
	avg	77.5	59.0	50.6±3.8	80.5	80.0	79.9±0.0	67.5	65.0	63.5±6.7	80.8	62.5	60.7±10.9
	w-avg	77.5	59.0	50.6±3.8	80.5	80.0	79.9±0.0	67.5	65.0	63.5±6.7	78.0	73.9	67.6±8.4

(b) Results for different methods on testing data and *ALL+CNet** on validation data.

Table A.21: Detailed results for predicting levels of *neuroticism* on enriched data. (a) shows results on validation data with different settings for N. For each method, the best setting w.r.t. macro-average F_1 score is selected (highlighted with **bold**). (b) shows results on test data for the selected settings and the baseline methods. ±: standard derivation across 10 runs.

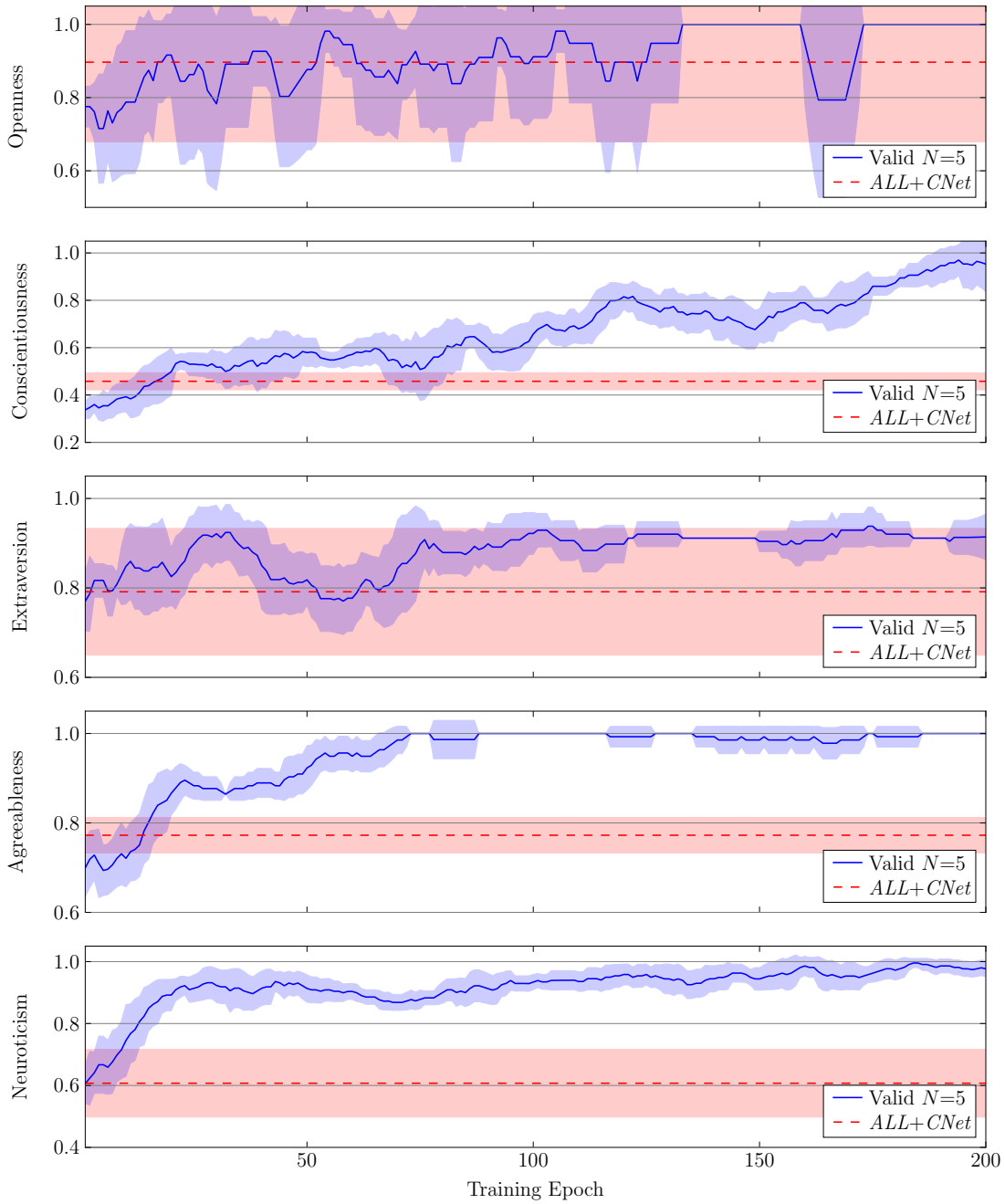

Figure A.1: Learning curve of the proposed system on enriched validation data: macro-average F_1 score on validation data as centered moving average accounting five past and five future epochs is shown in blue. Shaded areas denote standard deviation for this moving average. As several settings for N are evaluated in each epoch, for each trait, the setting for N is shown that produced the best results on validation data. The red dashed line shows average F_1 score for the *ALL+CNet* system on validation data over ten runs; the red shaded area denotes standard derivation across these runs.

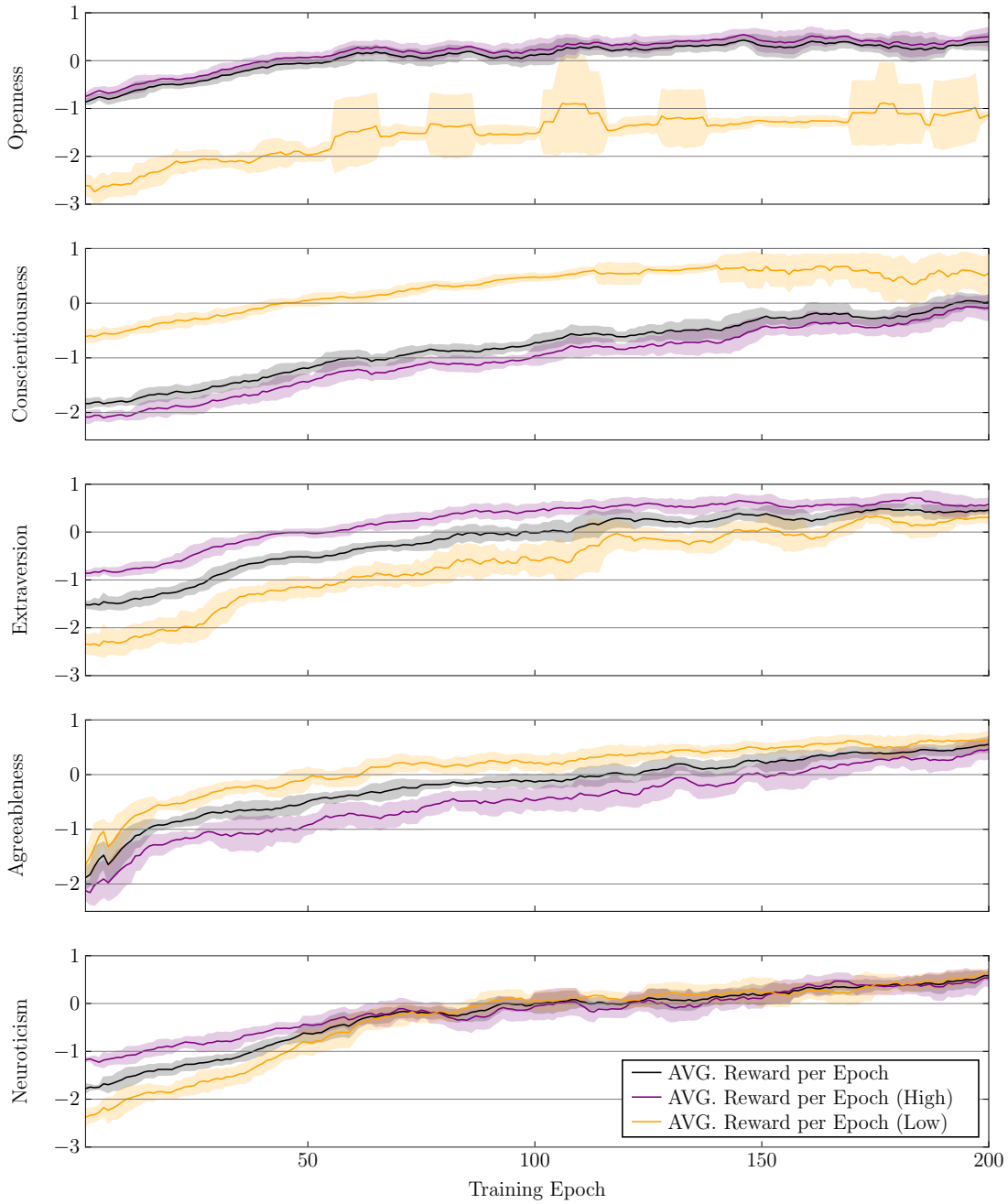

Figure A.2: Average reward per epoch provided to the RL-Agent during training on enriched data visualized as centered moving average accounting five past and five future epochs (shown in black). Shaded areas denote standard deviation for this moving average. Similarly, the average reward per epoch considering only profiles with ground-truth label *high* or *low* are shown in purple and yellow, respectively.

A.4 Main Experiments: Detailed Results

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
N=5	Low	**10.3**	**40.0**	**16.3**±21.5	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	1.4	10.0	2.5±7.9
	High	**97.9**	**91.3**	**94.5**±1.4	96.3	87.7	91.7±3.1	96.1	82.7	88.8±2.1	96.5	81.7	88.4±3.0
	avg	**54.1**	**65.7**	**55.4**±10.9	48.2	43.8	45.9±1.6	48.1	41.3	44.4±1.0	48.9	45.8	45.4±4.3
	w-avg	**95.1**	**89.7**	**92.0**±1.7	93.2	84.8	88.8±3.0	93.0	80.0	86.0±2.0	93.4	79.3	85.6±2.9
N=10	Low	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	1.1	10.0	2.0±6.3
	High	96.3	87.0	91.4±2.3	96.3	86.3	91.0±2.3	96.2	84.0	89.6±2.2	96.2	76.3	85.0±4.2
	avg	48.1	43.5	45.7±1.1	48.1	43.2	45.5±1.1	48.1	42.0	44.8±1.1	48.7	43.2	43.5±3.8
	w-avg	93.2	84.2	88.4±2.2	93.2	83.5	88.1±2.2	93.1	81.3	86.8±2.1	93.2	74.2	82.3±4.1
N=20	Low	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0
	High	96.7	98.0	97.3±1.2	96.6	93.7	95.1±1.0	96.4	90.0	93.1±1.8	96.4	89.0	92.5±1.9
	avg	48.4	49.0	48.7±0.6	48.3	46.8	47.5±0.5	48.2	45.0	46.5±0.9	48.2	44.5	46.3±1.0
	w-avg	93.6	94.8	94.2±1.2	93.5	90.6	92.0±1.0	93.3	87.1	90.1±1.7	93.3	86.1	89.5±1.9
N=30	Low	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	**0.0**	**0.0**	**0.0**±0.0
	High	96.7	99.0	97.9±0.8	96.6	94.7	95.6±2.4	96.4	88.7	92.3±1.6	**96.6**	**94.3**	**95.4**±1.5
	avg	48.4	49.5	48.9±0.4	48.3	47.3	47.8±1.2	48.2	44.3	46.2±0.8	**48.3**	**47.2**	**47.7**±0.7
	w-avg	93.6	95.8	94.7±0.8	93.5	91.6	92.5±2.3	93.3	85.8	89.4±1.5	**93.5**	**91.3**	**92.3**±1.4
N=50	Low	0.0	0.0	0.0±0.0	**0.0**	**0.0**	**0.0**±0.0	**0.0**	**0.0**	**0.0**±0.0	0.0	0.0	0.0±0.0
	High	96.7	98.0	97.3±1.2	**96.7**	**97.0**	**96.8**±1.5	**96.7**	**96.7**	**96.7**±0.8	96.6	94.3	95.4±1.2
	avg	48.4	49.0	48.7±0.6	**48.3**	**48.5**	**48.4**±0.8	**48.3**	**48.3**	**48.3**±0.4	48.3	47.2	47.7±0.6
	w-avg	93.6	94.8	94.2±1.2	**93.6**	**93.9**	**93.7**±1.5	**93.5**	**93.5**	**93.5**±0.8	93.5	91.3	92.4±1.2

(a) Results for selection based methods on validation data.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Best	Low	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0
	High	99.2	91.8	95.3±1.2	99.3	99.1	99.2±0.3	99.3	98.2	98.8±0.3	99.2	97.2	98.2±0.6
	avg	49.6	45.9	47.7±0.6	49.6	49.6	49.6±0.1	49.6	49.1	49.4±0.2	49.6	48.6	49.1±0.3
	w-avg	98.5	91.1	94.6±1.2	98.6	98.4	98.5±0.3	98.5	97.5	98.0±0.3	98.5	96.5	97.5±0.6
		ALL+CNet			*Baseline-R*			*Baseline-B*			*ALL+CNet**		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
All	Low	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	10.0	20.0	13.3±28.1	0.0	0.0	0.0±0.0
	High	99.3	100	99.6±0.0	99.3	100	99.6±0.0	99.4	99.9	99.6±0.0	96.8	100	98.4±0.0
	avg	49.6	50.0	49.8±0.0	49.6	50.0	49.8±0.0	54.7	59.9	56.5±14.1	48.4	50.0	49.2±0.0
	w-avg	98.6	99.3	98.9±0.0	98.6	99.3	98.9±0.0	98.8	99.3	99.0±0.2	93.7	96.8	95.2±0.0

(b) Results for different methods on testing data and *ALL+CNet** on validation data.

Table A.22: Detailed results for predicting levels of *openness*. (a) shows results of selection based methods on validation data with different settings for N. For each method, the best setting w.r.t. macro-average F_1 score is selected (highlighted with **bold**). (b) shows results on test data for the selected settings and the baseline methods. ±: standard derivation across 10 runs.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
N=5	Low	**14.5**	**80.0**	**24.4**±7.9	10.6	100	19.2±0.7	10.3	100	18.7±1.1	**13.2**	**100**	**23.3**±2.4
	High	**97.4**	**60.4**	**74.5**±3.7	100	29.6	45.6±3.7	100	27.1	42.4±6.6	**100**	**44.6**	**61.4**±6.7
	avg	**55.9**	**70.2**	**49.5**±5.4	55.3	64.8	32.4±2.2	55.1	63.5	30.5±3.9	**56.6**	**72.3**	**42.4**±4.6
	w-avg	**91.0**	**61.9**	**70.6**±3.9	93.1	35.0	43.5±3.5	93.1	32.7	40.6±6.2	**93.3**	**48.9**	**58.5**±6.4
N=10	Low	12.1	85.0	21.1±6.6	9.7	100	17.6±1.0	9.2	100	16.9±0.7	10.7	100	19.4±1.1
	High	97.3	47.5	63.7±6.3	100	21.7	35.3±7.5	100	17.9	30.2±6.5	100	30.4	46.4±6.3
	avg	54.7	66.2	42.4±6.1	54.8	60.8	26.5±4.2	54.6	59.0	23.5±3.6	55.4	65.2	32.9±3.7
	w-avg	90.8	50.4	60.4±6.3	93.0	27.7	34.0±7.0	93.0	24.2	29.1±6.1	93.1	35.8	44.3±5.9
N=20	Low	13.4	100	23.6±1.1	10.0	100	18.2±1.2	9.5	100	17.4±0.4	10.2	100	18.6±1.1
	High	100	45.8	62.8±3.2	100	25.0	39.7±6.8	100	20.8	34.4±2.7	100	26.7	41.8±7.1
	avg	56.7	72.9	43.2±2.2	55.0	62.5	29.0±4.0	54.8	60.4	25.9±1.5	55.1	63.3	30.2±4.1
	w-avg	93.3	50.0	59.8±3.0	93.1	30.8	38.1±6.4	93.0	26.9	33.1±2.5	93.1	32.3	40.0±6.7
N=30	Low	12.1	95.0	21.5±4.0	10.2	100	18.5±0.8	10.2	100	18.5±0.7	10.8	100	19.6±0.9
	High	99.0	42.1	58.9±5.4	100	26.2	41.4±4.9	100	26.2	41.5±4.2	100	31.2	47.5±4.8
	avg	55.5	68.5	40.2±4.4	55.1	63.1	30.0±2.9	55.1	63.1	30.0±2.5	55.4	65.6	33.5±2.8
	w-avg	92.3	46.2	56.0±5.2	93.1	31.9	39.7±4.6	93.1	31.9	39.7±4.0	93.1	36.5	45.3±4.5
N=50	Low	13.9	100	24.3±1.7	**11.7**	**100**	**20.8**±1.8	**10.9**	**100**	**19.7**±0.7	11.8	100	21.0±1.6
	High	100	47.9	64.7±4.0	**100**	**36.2**	**52.9**±6.5	**100**	**32.1**	**48.5**±3.2	100	37.1	53.9±6.0
	avg	56.9	74.0	44.5±2.9	**55.8**	**68.1**	**36.9**±4.1	**55.5**	**66.0**	**34.1**±1.9	55.9	68.5	37.5±3.8
	w-avg	93.4	51.9	61.6±3.8	**93.2**	**41.1**	**50.5**±6.1	**93.2**	**37.3**	**46.3**±3.0	93.2	41.9	51.3±5.7

(a) Results for selection based methods on validation data.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Best	Low	12.5	77.0	21.5±1.8	10.5	90.0	18.8±0.6	10.9	90.0	19.5±0.7	9.8	80.0	17.5±2.6
	High	96.2	52.2	67.7±2.8	97.3	31.9	48.0±3.2	97.5	35.0	51.4±3.1	95.3	35.2	51.4±2.0
	avg	54.4	64.6	44.6±2.2	53.9	61.0	33.4±1.9	54.2	62.5	35.4±1.9	52.5	57.6	34.5±1.9
	w-avg	89.4	54.2	63.9±2.7	90.2	36.7	45.7±3.0	90.5	39.4	48.8±2.9	88.3	38.9	48.6±1.9
		ALL+CNet			*Baseline-R*			*Baseline-B*			*ALL+CNet**		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
All	Low	11.5	55.0	19.0±2.8	0.0	0.0	0.0±0.0	29.9	26.0	22.9±17.8	7.8	45.0	13.3±4.7
	High	94.0	62.6	75.1±1.6	91.9	100	95.8±0.0	93.8	95.8	94.6±1.3	92.5	56.2	69.9±2.4
	avg	52.8	58.8	47.0±1.6	45.9	50.0	47.9±0.0	61.8	60.9	58.8±8.4	50.2	50.6	41.6±2.8
	w-avg	87.3	62.0	70.5±1.5	84.4	91.9	88.0±0.0	88.6	90.1	88.8±0.8	86.0	55.4	65.6±2.2

(b) Results for different methods on testing data and *ALL+CNet** on validation data.

Table A.23: Detailed results for predicting levels of *conscientiousness*. (a) shows results on validation data with different settings for N. For each method, the best setting w.r.t. macro-average F_1 score is selected (highlighted with **bold**). (b) shows results on test data for the selected settings and the baseline methods. ±: standard derivation across 10 runs.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
N=5	Low	**8.3**	**6.7**	**7.3**±15.5	0.0	0.0	0.0±0.0	**30.0**	**20.0**	**23.0**±20.4	0.0	0.0	0.0±0.0
	High	**89.1**	**95.0**	**91.9**±1.0	88.4	95.0	91.5±1.4	**90.2**	**92.5**	**91.3**±3.0	88.2	93.8	90.9±1.2
	avg	**48.7**	**50.8**	**49.6**±8.0	44.2	47.5	45.8±0.7	**60.1**	**56.2**	**57.2**±11.5	44.1	46.9	45.5±0.6
	w-avg	**80.1**	**85.2**	**82.5**±2.3	78.5	84.4	81.4±1.2	**83.5**	**84.4**	**83.7**±4.7	78.4	83.3	80.8±1.0
N=10	Low	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	11.7	13.3	12.4±16.1	0.0	0.0	0.0±0.0
	High	88.0	91.7	89.8±2.3	87.9	90.8	89.3±3.1	89.1	88.3	88.7±2.4	87.8	90.0	88.8±2.5
	avg	44.0	45.8	44.9±1.2	43.9	45.4	44.6±1.5	50.4	50.8	50.5±8.9	43.9	45.0	44.4±1.2
	w-avg	78.2	81.5	79.8±2.1	78.1	80.7	79.3±2.7	80.5	80.0	80.2±3.6	78.0	80.0	79.0±2.2
N=20	Low	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0
	High	88.4	95.0	91.5±1.7	88.2	93.8	90.9±1.6	87.2	85.0	86.0±2.3	87.9	91.2	89.5±1.7
	avg	44.2	47.5	45.8±0.9	44.1	46.9	45.4±0.8	43.6	42.5	43.0±1.2	44.0	45.6	44.8±0.8
	w-avg	78.5	84.4	81.4±1.5	78.4	83.3	80.8±1.4	77.5	75.6	76.5±2.1	78.2	81.1	79.6±1.5
N=30	Low	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0	0.0	0.0	0.0±0.0
	High	88.9	100	94.1±0.0	88.5	96.2	92.2±2.2	87.9	91.2	89.5±2.6	88.1	92.5	90.2±0.9
	avg	44.4	50.0	47.1±0.0	44.2	48.1	46.1±1.1	44.0	45.6	44.8±1.3	44.0	46.2	45.1±0.5
	w-avg	79.0	88.9	83.7±0.0	78.7	85.6	81.9±1.9	78.2	81.1	79.6±2.4	78.3	82.2	80.2±0.8
N=50	Low	0.0	0.0	0.0±0.0	**0.0**	**0.0**	**0.0**±0.0	0.0	0.0	0.0±0.0	**0.0**	**0.0**	**0.0**±0.0
	High	88.9	100	94.1±0.0	**88.9**	**100**	**94.1**±0.0	88.9	100	94.1±0.0	**88.8**	**98.8**	**93.5**±1.0
	avg	44.4	50.0	47.1±0.0	**44.4**	**50.0**	**47.1**±0.0	44.4	50.0	47.1±0.0	**44.4**	**49.4**	**46.7**±0.5
	w-avg	79.0	88.9	83.7±0.0	**79.0**	**88.9**	**83.7**±0.0	79.0	88.9	83.7±0.0	**78.9**	**87.8**	**83.1**±0.9

(a) Results for selection based methods on validation data.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Best	Low	19.3	16.7	17.8±11.2	0.0	0.0	0.0±0.0	19.9	26.7	22.5±7.0	0.0	0.0	0.0±0.0
	High	95.7	96.8	96.2±0.4	94.9	98.3	96.6±0.3	96.1	94.1	95.1±0.9	94.9	98.2	96.5±0.4
	avg	57.5	56.7	57.0±5.7	47.5	49.2	48.3±0.2	58.0	60.4	58.8±3.8	47.5	49.1	48.2±0.2
	w-avg	91.8	92.8	92.3±0.8	90.2	93.4	91.8±0.3	92.2	90.8	91.4±1.1	90.2	93.2	91.7±0.4
		ALL+CNet			*Baseline-R*			*Baseline-B*			*ALL+CNet**		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
All	Low	0.0	0.0	0.0±0.0	66.7	66.7	66.7±0.0	50.8	80.0	58.4±20.3	0.0	0.0	0.0±0.0
	High	95.0	99.0	96.9±0.2	98.2	98.2	98.2±0.0	98.9	92.6	95.5±4.8	88.9	100	94.1±0.0
	avg	47.5	49.5	48.4±0.1	82.5	82.5	82.5±0.0	74.9	86.3	76.9±12.4	44.4	50.0	47.1±0.0
	w-avg	90.2	94.0	92.1±0.2	96.7	96.7	96.7±0.0	96.5	92.0	93.6±5.5	79.0	88.9	83.7±0.0

(b) Results for different methods on testing data and *ALL+CNet** on validation data.

Table A.24: Detailed results for predicting levels of *extraversion*. (a) shows results of selection based methods on validation data with different settings for N. For each method, the best setting w.r.t. macro-average F_1 score is selected (highlighted with **bold**). (b) shows results on test data for the selected settings and the baseline methods. ±: standard derivation across 10 runs.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
N=5	Low	**46.3**	**57.5**	**50.1**±5.0	19.5	82.5	31.4±5.1	20.3	82.5	32.6±5.7	21.4	100	35.3±1.5
	High	**92.6**	**87.9**	**90.1**±2.4	93.5	42.1	57.3±10.6	94.4	46.2	62.0±3.5	100	38.8	55.8±4.1
	avg	**69.5**	**72.7**	**70.1**±3.2	56.5	62.3	44.4±7.3	57.3	64.4	47.3±4.0	60.7	69.4	45.5±2.8
	w-avg	**86.0**	**83.6**	**84.4**±2.5	83.0	47.9	53.6±9.6	83.8	51.4	57.8±3.5	88.8	47.5	52.8±3.7
N=10	Low	21.3	50.0	29.6±9.8	17.5	82.5	28.8±3.5	21.3	100	35.1±1.2	18.6	97.5	31.2±2.6
	High	89.2	68.8	77.5±5.1	92.6	35.0	50.4±7.4	100	38.3	55.4±3.5	98.8	28.7	44.3±6.4
	avg	55.2	59.4	53.6±6.7	55.0	58.8	39.6±4.7	60.7	69.2	45.2±2.3	58.7	63.1	37.8±4.1
	w-avg	79.5	66.1	70.7±5.3	81.9	41.8	47.3±6.6	88.8	47.1	52.5±3.1	87.3	38.6	42.4±5.7
N=20	Low	19.9	42.5	26.9±8.6	18.0	77.5	29.1±4.4	18.1	87.5	29.9±4.0	19.7	85.0	31.9±4.3
	High	88.4	72.1	79.3±2.4	90.8	39.6	54.5±10.8	94.4	33.8	49.5±5.7	94.5	42.1	58.0±5.9
	avg	54.1	57.3	53.1±4.8	54.4	58.5	41.8±7.4	56.2	60.6	39.7±4.2	57.1	63.5	45.0±4.4
	w-avg	78.6	67.9	71.8±2.8	80.4	45.0	50.9±9.8	83.5	41.4	46.7±5.2	83.8	48.2	54.3±5.4
N=30	Low	19.7	52.5	28.6±3.6	18.1	75.0	29.1±4.8	20.6	82.5	33.0±4.9	**21.0**	**87.5**	**33.9**±5.2
	High	89.0	64.2	74.5±2.9	91.0	42.9	57.9±8.7	94.1	46.7	62.2±6.5	**95.6**	**45.0**	**61.1**±4.5
	avg	54.4	58.3	51.5±2.9	54.6	59.0	43.5±6.1	57.4	64.6	47.5±5.2	**58.3**	**66.2**	**47.5**±4.6
	w-avg	79.1	62.5	68.0±2.8	80.6	47.5	53.8±7.9	83.6	51.8	58.0±6.0	**84.9**	**51.1**	**57.2**±4.5
N=50	Low	18.5	47.5	26.6±7.0	**17.6**	**55.0**	**26.5**±4.1	**20.9**	**65.0**	**31.6**±4.9	16.2	55.0	24.9±5.5
	High	88.3	65.4	75.1±2.6	**88.2**	**56.2**	**68.3**±7.8	**91.1**	**59.2**	**71.6**±3.2	87.2	51.7	64.7±5.9
	avg	53.4	56.5	50.8±4.3	**52.9**	**55.6**	**47.4**±5.2	**56.0**	**62.1**	**51.6**±3.3	51.7	53.3	44.8±5.3
	w-avg	78.3	62.9	68.2±2.9	**78.1**	**56.1**	**62.3**±7.0	**81.1**	**60.0**	**65.9**±3.0	77.0	52.1	59.0±5.7

(a) Results for selection based methods on validation data.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Best	Low	5.9	20.0	9.1±3.6	12.7	77.3	21.8±2.3	13.4	82.7	23.0±1.8	12.9	95.5	22.7±1.7
	High	89.3	67.8	77.0±1.2	95.2	45.7	61.8±1.9	96.3	45.4	61.7±2.0	98.7	34.2	50.7±2.9
	avg	47.6	43.9	43.1±2.1	53.9	61.5	41.8±2.0	54.8	64.0	42.3±1.6	55.8	64.8	36.7±2.2
	w-avg	81.6	63.4	70.8±1.3	87.5	48.7	58.1±1.9	88.6	48.8	58.1±1.9	90.7	39.8	48.1±2.8
		ALL+CNet			*Baseline-R*			*Baseline-B*			*ALL+CNet**		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
All	Low	15.6	51.8	24.0±3.6	40.0	18.2	25.0±0.0	50.4	40.9	39.3±7.0	22.2	27.5	24.2±6.1
	High	93.6	71.5	81.0±1.1	92.1	97.2	94.6±0.0	93.9	92.6	93.1±3.1	87.3	83.3	85.2±2.4
	avg	54.6	61.7	52.5±2.2	66.0	57.7	59.8±0.0	72.2	66.8	66.2±4.6	54.8	55.4	54.8±3.9
	w-avg	86.4	69.7	75.8±1.3	87.3	89.9	88.2±0.0	89.9	87.8	88.2±3.2	78.0	75.4	76.5±2.7

(b) Results for different methods on testing data and *ALL+CNet** on validation data.

Table A.25: Detailed results for predicting levels of *agreeableness*. (a) shows results of selection based methods on validation data with different settings for N. For each method, the best setting w.r.t. macro-average F_1 score is selected (highlighted with **bold**). (b) shows results on test data for the selected settings and the baseline methods. ±: standard derivation across 10 runs.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
N=5	Low	29.9	73.8	42.5±6.7	27.6	81.2	41.1±4.5	22.5	66.2	33.6±2.9	22.7	67.5	34.0±4.3
	High	79.8	37.3	50.7±5.9	76.9	22.3	34.1±8.6	56.3	16.8	25.7±8.3	57.3	16.4	25.3±7.4
	avg	54.8	55.5	46.6±6.0	52.2	51.8	37.6±5.7	39.4	41.5	29.6±5.1	40.0	41.9	29.6±5.4
	w-avg	66.5	47.0	48.5±5.9	63.8	38.0	36.0±6.9	47.3	30.0	27.8±6.5	48.1	30.0	27.6±6.2
N=10	Low	27.9	62.5	38.5±6.6	29.3	82.5	43.2±7.0	28.0	72.5	40.4±3.9	22.2	60.0	32.4±5.5
	High	75.8	41.8	53.7±4.2	79.7	26.8	39.4±13.4	76.4	32.3	45.2±5.4	62.5	24.1	34.6±5.3
	avg	51.8	52.2	46.1±4.4	54.5	54.7	41.3±9.5	52.2	52.4	42.8±4.2	42.4	42.0	33.5±4.6
	w-avg	63.0	47.3	49.6±4.0	66.2	41.7	40.4±11.3	63.5	43.0	43.9±4.7	51.8	33.7	34.1±4.7
N=20	Low	21.7	38.8	27.7±6.8	23.5	47.5	31.4±6.8	**25.7**	**51.2**	**34.2**±2.3	21.3	41.2	28.1±4.7
	High	69.0	49.5	57.6±4.5	69.8	44.1	53.9±5.0	**72.1**	**45.9**	**56.1**±3.0	67.8	45.0	54.0±4.1
	avg	45.4	44.1	42.6±4.8	46.7	45.8	42.7±5.2	**48.9**	**48.6**	**45.1**±2.3	44.6	43.1	41.0±3.2
	w-avg	56.4	46.7	49.6±4.4	57.5	45.0	47.9±4.9	**59.7**	**47.3**	**50.2**±2.6	55.4	44.0	47.1±3.3
N=30	Low	**26.6**	**41.2**	**32.2**±7.8	**22.2**	**37.5**	**27.9**±7.5	15.2	23.8	18.5±4.7	19.3	33.8	24.5±10.1
	High	**73.6**	**59.1**	**65.4**±3.7	**70.2**	**53.2**	**60.4**±3.4	65.3	52.3	58.0±3.8	67.7	50.0	57.4±4.7
	avg	**50.1**	**50.2**	**48.8**±5.1	**46.2**	**45.3**	**44.1**±4.6	40.2	38.0	38.2±3.0	43.5	41.9	40.9±6.8
	w-avg	**61.0**	**54.3**	**56.6**±4.2	**57.4**	**49.0**	**51.7**±3.7	51.9	44.7	47.4±3.1	54.8	45.7	48.6±5.5
N=50	Low	19.1	17.5	18.2±10.4	12.2	8.8	10.1±9.7	6.8	5.0	5.7±7.4	**18.9**	**12.5**	**14.8**±9.7
	High	71.5	75.0	73.1±3.4	70.2	78.2	73.9±3.5	68.8	76.4	72.3±3.9	**71.2**	**79.1**	**74.9**±4.4
	avg	45.3	46.2	45.6±5.9	41.2	43.5	42.0±6.0	37.8	40.7	39.0±4.8	**45.1**	**45.8**	**44.9**±6.7
	w-avg	57.5	59.7	58.5±4.2	54.7	59.7	56.9±4.6	52.2	57.3	54.6±4.0	**57.3**	**61.3**	**58.9**±5.5

(a) Results for selection based methods on validation data.

		RL-Profiler			*RND+CNet*			*PMI+CNet*			*PT+CNet*		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Best	Low	17.9	22.8	20.1±2.1	18.0	24.4	20.7±3.8	21.0	36.1	26.6±1.9	10.3	6.9	8.3±2.1
	High	62.4	55.0	58.5±2.7	62.0	52.8	57.0±0.5	60.4	41.8	49.4±2.0	65.0	74.1	69.2±1.6
	avg	40.2	38.9	39.3±2.3	40.0	38.6	38.8±1.7	40.7	39.0	38.0±1.6	37.7	40.5	38.8±1.7
	w-avg	49.1	45.4	47.0±2.4	48.8	44.2	46.1±0.8	48.6	40.1	42.5±1.7	48.6	53.9	50.9±1.6
		ALL+CNet			*Baseline-R*			*Baseline-B*			*ALL+CNet**		
	Class	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
All	Low	16.5	4.4	6.9±4.6	64.0	41.0	50.0±0.0	61.4	49.5	53.3±4.7	10.8	3.8	5.5±8.9
	High	68.9	91.0	78.4±1.1	78.1	90.1	83.7±0.0	80.0	85.3	82.3±2.2	72.0	90.0	80.0±2.2
	avg	42.7	47.7	42.7±2.6	71.0	65.6	66.8±0.0	70.7	67.4	67.8±1.8	41.4	46.9	42.7±5.1
	w-avg	53.2	65.0	57.0±1.9	73.9	75.4	73.6±0.0	74.4	74.5	73.6±1.1	55.7	67.0	60.1±3.5

(b) Results for different methods on testing data and *ALL+CNet** on validation data.

Table A.26: Detailed results for predicting levels of *neuroticism*. (a) shows results of selection based methods on validation data with different settings for N. For each method, the best setting w.r.t. macro-average F_1 score is selected (highlighted with **bold**). (b) shows results on test data for the selected settings and the baseline methods. ±: standard derivation across 10 runs.

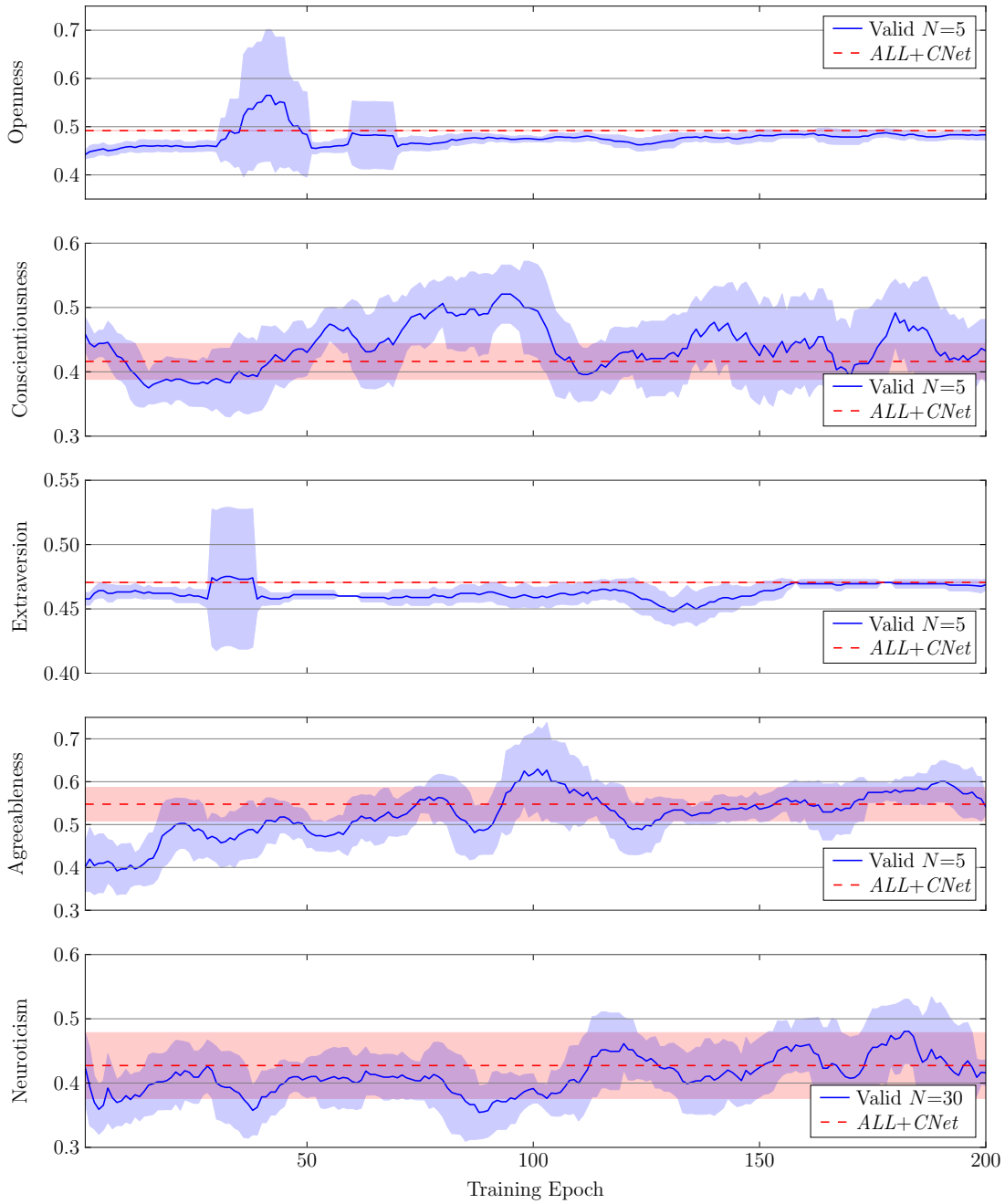

Figure A.3: Learning curve of the proposed system on validation data: macro-average F_1 score on validation data as centered moving average accounting five past and five future epochs is shown in blue. Shaded areas denote standard deviation for this moving average. As several settings for N are evaluated in each epoch, for each trait, the setting for N is shown that produced the best results on validation data. The red dashed line shows average F_1 score for the *ALL+CNet* system on validation data over ten runs; the red shaded area denotes standard derivation across these runs.

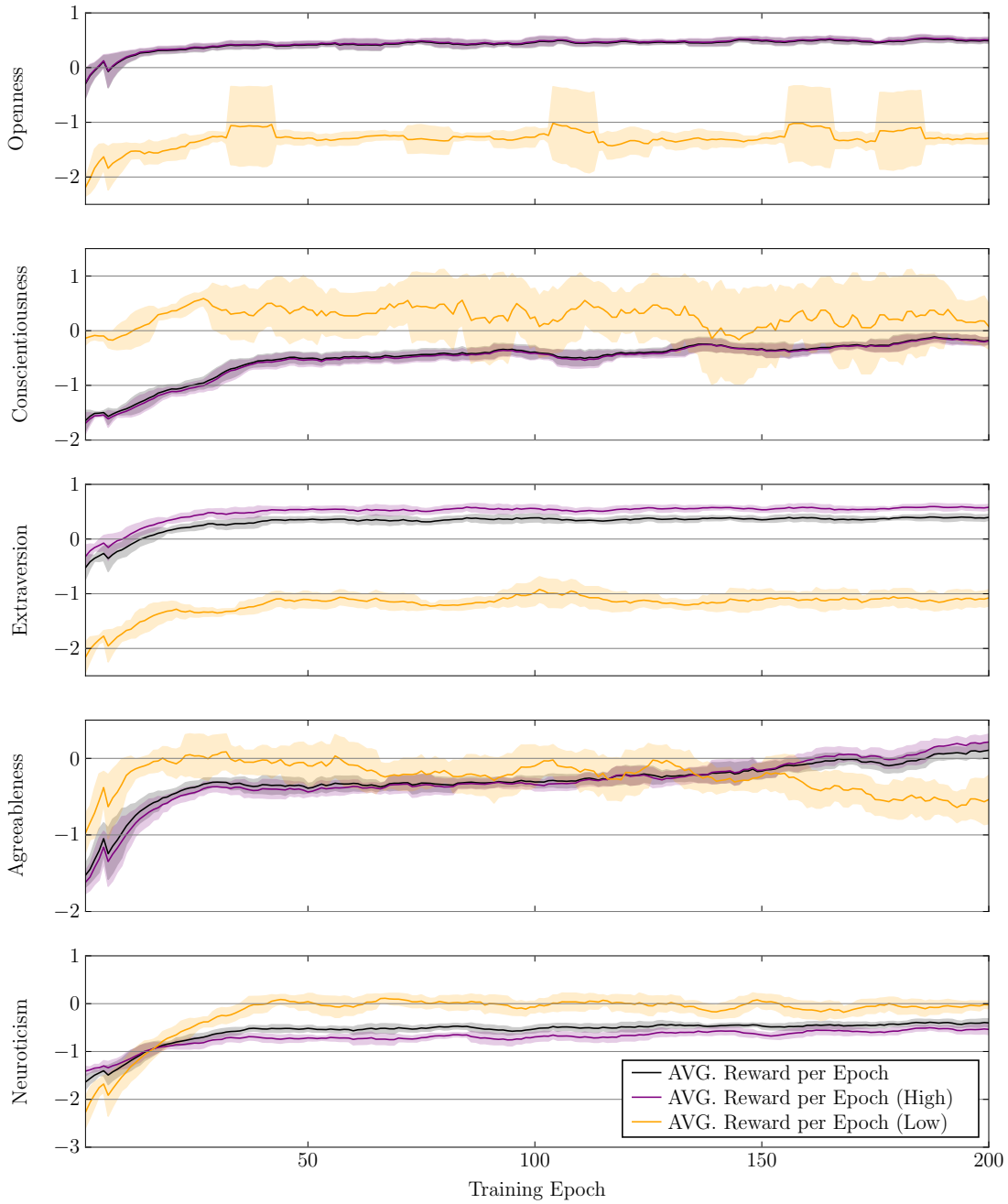

Figure A.4: Average reward per epoch provided to the RL-Agent in the proposed system during training visualized as centered moving average accounting five past and five future epochs (shown in black). Shaded areas denote standard deviation for this moving average. Similarly, the average reward per epoch considering only profiles with ground-truth label *high* or *low* are shown in purple and yellow, respectively. Note that the average reward for profiles associated with a *high* level tends to be similar or equal to the average considering all profiles due to the class imbalance.

Declaration:

I hereby declare that the work presented in this thesis is entirely my own. I did not use any other sources and references that the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted hard copies.

Date and Signature: