Natural language processing for cognitive therapy: extracting schemas from thought records

Burger Franziska[1*], Neerincx Mark A.[1,2], Brinkman Willem-Paul[1]

1 Department of Intelligent Systems, Delft University of Technology, Delft, Netherlands
2 Department of Perceptual and Cognitive Systems, Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (TNO), Soesterberg, Netherlands

* f.v.burger@tudelft.nl

Abstract

The cognitive approach to psychotherapy aims to change patients' maladaptive schemas, that is, overly negative views on themselves, the world, or the future. To obtain awareness of these views, they record their thought processes in situations that caused pathogenic emotional responses. The schemas underlying such thought records have, thus far, been largely manually identified. Using recent advances in natural language processing, we take this one step further by automatically extracting schemas from thought records. To this end, we asked 320 healthy participants on Amazon Mechanical Turk to each complete five thought records consisting of several utterances reflecting cognitive processes. Agreement between two raters on manually scoring the utterances with respect to how much they reflect each schema was substantial (Cohen's $\kappa = 0.79$). Pretrained natural language processing software was used to represent words and utterances, which were then mapped to schemas using k-nearest neighbors algorithms, support vector machines, and recurrent neural networks. For the more frequently occurring schemas, all algorithms were able to leverage linguistic patterns. For example, the scores assigned to the *Competence* schema by the algorithms correlated with the manually assigned scores with Spearman correlations ranging between 0.64 and 0.76. For six of the nine schemas, a set of recurrent neural networks trained separately for each of the schemas outperformed the other algorithms. We present our results here as a benchmark solution, since we conducted this research to explore the possibility of automatically processing qualitative mental health data and did not aim to achieve optimal performance with any of the explored models. The dataset of 1600 thought records comprising 5747 utterances is published together with this article for researchers and machine learning enthusiasts to improve upon our outcomes. Based on our promising results, we see further opportunities for using free-text input and subsequent natural language processing in other common therapeutic tools, such as ecological momentary assessments, automated case conceptualizations, and, more generally, as an alternative to mental health scales.

Introduction

E-mental health—delivering therapeutic interventions via information and communication technology—is regarded as a promising means of overcoming many barriers to traditional psychotherapeutic care. Yet, in a review of more than 130 scientifically evaluated e-mental health systems for depression, it was found that the

technological state of the art of these systems is limited: even in recently developed systems, technology is often only used as a platform for delivering information to the patient. When the patient is asked to provide open, unconstrained textual information to the system, this information is typically either processed by a human in the case of guided systems or not processed at all in the case of autonomous systems [1]. Although both methods are arguably very robust to misunderstanding, human processing is costly while no processing offers no advantage over traditional paper-based workbooks. However, developments in data-driven natural language understanding are increasingly able to reliably interpret unconstrained qualitative user input. Here, we explore this opportunity for a specific therapeutic task in cognitive therapy: determining underlying maladaptive schemas from the information contained in thought record forms.

Thought record forms provide patients with a structured format for monitoring their thoughts, consisting of descriptions of the thought eliciting situation, the experienced emotion, the first cognitive appraisal of the situation, and the resulting behavior. Thought records are commonly employed in cognitive therapy, a form of psychotherapy based on Beck's cognitive theory [2]. The theory posits that not the situations but the way in which we appraise them causes our emotions. For example, it is not the fact that we are not invited to a party that makes us upset but rather the fear or understanding that this says something about us or our relationship with the host. Our immediate and unreflected appraisal of a situation is called an automatic thought. Automatic thoughts are in turn determined by schemas, the cognitive structures that make up our world view. A specific schema can be activated given the right trigger. In people with certain mental illnesses, it is theorized that pathogenic schemas have a particularly low activation threshold [3]. Consequently, a core part of cognitive psychotherapy involves teaching patients to monitor thoughts for insight into underlying schemas. Starting from the automatic thought noted down in the thought record, the downward arrow technique (DAT) [4] helps to determine the causative maladaptive schema. It consists of repeatedly asking *why it would be upsetting* or *what would be the worst that could happen* if the idea stated in the previous step was true. An example thought record that we collected in our experiment is shown in Table 1. The DAT is illustrated by the final three rows (in cursive font). Since the majority of thought records in our dataset include the DAT, hereinafter the term *thought record* refers to both the core thought record and DAT unless explicitly stated. Also extending beyond the nomenclature typically used in clinical psychology, we define as a thought record *utterance* the automatic thought or any completed step of the DAT. Each of the final four rows of the Participant Response column of Table 1 reflects an utterance. As can be seen in the response to the second downward arrow step, i.e., *I want friends. I will be lonely otherwise.*, an utterance can consist of multiple sentences.

Unlike automatic thoughts, schemas have received little attention in empirical research to date [5]. When considered, they have typically been explored in a top-down manner with measurement instruments developed on the basis of cognitive theory and validated with exploratory factor analyses (for example, [5, 6]). To the best of our knowledge, only one classification rubric for schemas exists that was not exclusively derived from theory but created from a content analysis of a set of thought records (also including DAT) collected with an online self-help cognitive behavioral therapy (CBT) program, namely the schema rubric of Millings and Carnelley [7].

In this work, we develop the natural language processing (NLP) foundation for a task-oriented conversational agent (CA) that motivates users to regularly complete thought recording homework exercises. Most CAs used in practice to date are frame-based [8, Ch. 24]. To be able to parse the semantics of a user input (e.g. "I want to take my girlfriend to the theater next weekend.") and fill the slots in a frame (e.g. day, show, theater, time, number of tickets), the agent needs to classify broadly the

Table 1. Example thought record from the dataset collected in this study.

TR Question	Entry Type	Participant Response
Describe the situation very briefly in your own words.	open text entry field	while walking down the street I see someone I know, wave at them and they don't acknowledge my wave.
How well can you imagine yourself in this situation?	slider from 0 (not at all) to 100 (as good as if you were in the situation at the very moment)	85
Describe your emotion in this situation in one word.	open text entry field	disappointment
How intensely would you be experiencing this emotion?	slider from 0 (a trace) to 100 (the most intense possible)	45
Which of the following four emotions corresponds best with the emotion that you wrote down above?	multiple choice: sadness, fear, anger, happiness	sadness
Which (automatic) thought might have caused you to feel this way in the described situation?	open text entry field	They don't like me enough to wave back
And why would it be upsetting to you if "They don't like me enough to wave back" were true? What would it mean to you? What would it say about you?	*open text entry field*	*I may be unlikeable.*
And why would it be upsetting to you if "I may be unlikeable" were true? What would it mean to you? What would it say about you?	*open text entry field*	*I want friends. I will be lonely otherwise.*
And why would it be upsetting to you if "I want friends so I won't be lonely." were true? What would it mean to you? What would it say about you?	*open text entry field*	*If I am unlikeable then I won't have friends and will be alone all my life.*

Steps of the downward arrow technique are presented in cursive font. The scenario description presented to the participant was "You are walking down the street. On the other side of the street you see an acquaintance whom you've liked the few times you've been in his company. You wave to him, and you get no response."

intent of the entire input phrase (e.g. book theater tickets) and extract specifically the information corresponding to empty slots. When all slots are filled, the agent can complete the task. Up until recently, intent classification and slot filling were mostly done using a hand-written, domain-specific semantic grammar, often prescribing possible synonyms as well as a certain order for the information (e.g. {I want | Could I | It would be great if I could} * {book | reserve | get} * {tickets | cards} * {movies | theater} *). Systems using such grammars are expensive in terms of engineering time and prone to errors and misunderstandings [8, Ch. 24]. Both drawbacks have been largely eliminated with the advent of deep learning in the past decade. Rather than hand-crafting large sets of rules, deep learning allows for the acquisition of synonyms and word usage in context from large sets of data, such as Wikipedia. As two recent literature reviews show, these developments are slowly finding their way into CAs for health care [9, 10]. Laranjo et al. [9] found most of the CAs allowing for unconstrained natural language input to have been developed after 2010. Yet, only one-half of the reviewed agents used frame-based or agent-based dialog management [11], while the other half implemented entirely system-driven and finite-state dialog management strategies. The authors therefore conclude that CAs in health care are not up to par with those in other fields. Of all 40 agents considered in [10], only six use state-of-the-art natural language understanding techniques [12–17].

While it is always important to limit user frustrations that arise from understanding errors on the part of the CA, this is particularly crucial in dialog systems for mental health treatment due to the highly emotionally sensitive domain. It is conceivable that

language understanding errors as well as inconsistent or insensitive [18] responses could affect not only patients' experience and trust in the system but, in the worst case, also their mental health. Consequently, rule-based systems have been the norm [19]. Questions from the system are phrased so narrowly that they leave little room for unexpected responses (e.g., [20]). Since even therapy following a strict protocol is much less task-oriented than booking theater tickets, most systems fully or partially resort to providing multiple response options to the user (see, for example, [21]). The more recently developed Woebot [22], a chatbot for treating college students with symptoms of anxiety and depression, only uses natural language processing as an option for some nodes of Woebot's decision tree architecture, choosing the next node mostly based on user selection of one of several suggested replies.

Thought recording exercises are often assigned as homework to patients in face-to-face treatment or included in self-help workbooks and treatment systems with only general instructions. Timely feedback or tailored support from a therapist are therefore usually not available when patients attempt the exercise. As the goal of thought recording is the discovery of thinking patterns, frequent completion of thought records is crucial for their success. It is for these reasons that we aim to build a CA to motivate and support people in regularly completing thought records. The CA can use knowledge about schemas to provide feedback, respond understandingly, or to strategically ask for supplementary information. This work therefore addresses the following primary research question: Can the underlying maladaptive schema of a thought record utterance be scored by a machine?

Hypotheses

The objective of this study was to see whether identifying schemas from thought records is at all possible. Consequently, our first hypothesis is that schemas can be extracted automatically (H1). We investigate this with a future goal of implementing a conversational agent capable of providing useful feedback. For such practical applications, we were also interested in studying ways to potentially improve automatic schema identification. As a result, three additional hypotheses, informed by psychological theory, were also investigated: automatic predictions improve as the downward arrow technique progresses (H2), within individuals, similar situations will activate the same schemas (H3), across individuals, there is a relationship between the active schemas and scores on mental health scales (H4). We here motivate the hypotheses in turn.

H1: Schemas can be extracted automatically

As outlined above, conversational agents in health care, and particularly in depression treatment, to date are employing grammar-based or no NLP more often than not. Yet, the field more generally has not been blind to state-of-the-art data-driven methods. Thus far, however, they are mostly used in clinical psychology research to perform psychological assessment. Social media platforms and forums provide a treasure trove of natural language data occurring in virtual social environments. This has resulted in a large body of literature searching for linguistic markers indicative of depression, crisis, or suicidal risk in the data (e.g. [23–27, 29, 30]). One such example is the crisis detection models developed in [27]. With a dataset of posts comprising on average three sentences collected through the mental health app *Koko*, the authors use a recurrent neural network (RNN) to detect crisis (binary classification task). They augment their RNN with *attention* [28] to display the parts of a post that the neural network pays attention to during classification. Their best model, an RNN without attention, detects crisis with

an F1-score accuracy of 0.80. In another study [29], the task was to correctly identify which topic-based forum (or *subreddit*) on the social media website Reddit the posts of users belong to. The posts were drawn from eleven different manually selected mental health subreddits. The best performing algorithm achieved an F1-score accuracy of 0.71 with a convolutional neural network in this multi-class (more than two classes that are mutually exclusive) classification task. Benton et al. [30] study a similar problem as a multi-label (more than two classes that are not mutually exclusive) learning task. Using tweets posted on the social media platform Twitter, they simultaneously classify suicidal risk, atypical mental health, and seven mental health conditions. They observed a clear added benefit of leveraging possible correlations between the labels in the multi-label models compared to a set of nine single-class prediction models. Although the described research indicates that automatically identifying crisis or mental health conditions from social media corpora is feasible, it is unknown whether this applies to schemas as well. However, the fact that the schema rubric of Millings and Carnelley [7] was obtained via content analysis from a corpus of thought records indicates that language and word usage differ between the schemas. If this is the case, a good model trained on sufficient data should be able to pick up on these differences. Additionally, schemas are not mutually exclusive and might therefore inform each other, possibly further improving prediction accuracy. On the basis of these considerations, we posit the following:

H1 The schema(s) underlying a thought record can be identified by an algorithm with an accuracy above chance.

H2: Downward arrow converges and H3: schema patterns are similar across thought record type

Thought records ask patients to first briefly describe the situation that resulted in the pathogenic emotion in their own words. The automatic thought is thus directly connected to the situation description and both are highly individual. Automatically analyzing such free-form open text without any further restrictions is an *open-domain* NLP task, similar to small-talk. For an artificial intelligence, this is notoriously difficult to deal with well as it requires a comprehensive world model of many topics. Such a model cannot feasibly be engineered by humans and, if it is at all possible, very large amounts of data would be required to construct it bottom-up. Models created in this manner are usually no longer transparent and may show unintended behavior (e.g., [43]).

From a clinical perspective, an alternative to open thought recording is to elicit schemas by means of imagined situations, using scripted situation vignettes as a basis for the thought records. Thought recording is typically assigned as homework for the patient in cognitive therapy, with the completed forms constituting an integral part of the face-to-face sessions. While leaving patients to their own devices provides them with freedom and ensures ecological validity, the various different steps of the thought recording method do not always come easy to patients [31]. When they struggle, therapists may guide the process by resorting to imagery or role-play so as to recreate the situation in the face-to-face session and evoke the automatic thought again [32]. For initial practice [33] or for the controlled assessment of cognitive errors [34–36] and cognitive restructuring skills [37], therapists may additionally restrict patients by asking them to envision themselves in certain scripted ambiguous scenarios. From a technical perspective, such a scripted scenario can delimit the natural language domain. Taking the scenario into account in a schema identification model should thus produce more reliable results. Despite scenarios being viable from a clinical perspective and the safer option from a technical perspective, two aspects of cognitive therapy give rise to the possibility of open classification models for this specific NLP task: the *downward arrow technique* and the categorization of situations into *situation types*.

Downward arrow technique

The theory behind the downward arrow technique (DAT) posits that as one progresses along the downward arrow, a schema will be reached. While automatic thoughts are specific appraisals of situations, schemas are general: the same schema can cause a large variety of specific automatic thoughts. From this, it should follow that the thoughts delineated with the DAT become increasingly independent of the situation description. For the NLP, this means that the language in utterances should converge to language that is more characteristic of the schema. We therefore hypothesize as follows:

H2 Schema identification accuracy increases as one proceeds along the downward arrow.

Categorization of situations

Two situation types that are commonly distinguished in cognitive therapy are interpersonal situations and achievement-related situations (e.g., [38]). *Interpersonal* situations pertain to one's self-worth in relation to other people, while *achievement-related* situations are such where one might perform poorly and one's self-esteem is at risk. Hence, a schema identification model might generalize to any real-world situation as long as it takes into account whether the situation type is more interpersonal or more achievement-related. Consequently, the following hypothesis is tested:

H3 Within an individual, the schema patterns of scenario-based thought records can predict those of the real-life thought record when they match in situation type (interpersonal or achievement-related).

H4: Mental illnesses have associated schemas

Lastly, cognitive theory argues for differences between depression and anxiety with regard to schemas. Depressed individuals are theorized to have overly negative views of the self, the world, and the future, while anxious individuals hold schemas related to personal danger [39]. However, Millings and Carnelley [7] found that only the presence of the schema related to power and being in control differs between those with depression and those with anxiety, with particularly the anxious participants in their online CBT program presenting with the schema. If each mental illness were to show specific associated schemas, though, mental health data could inform a prior distribution over schemas in terms of their likelihood. This might improve a machine learning model. Using the coding scheme of [7], we therefore pose the following exploratory hypothesis:

H4 The schema patterns of an individual combined across thought records can predict his or her depression, anxiety, and cognitive distortions as self-reported using standard psychological questionnaires.

Methods

To test the hypotheses stated above, a dataset of completed thought records was needed. Copies of thought records from actual patients gathered through a therapeutic practice were not an option because we could not obtain access to such an existing corpus. We therefore chose to collect a new dataset of thought records through the online crowdsourcing platform Amazon Mechanical Turk. The Human Research Ethics Committee of Delft University of Technology granted ethical approval for the research (Letter of Approval number: 546).

Design

The data collection process was designed as a cross-sectional observational study. This means that there were no independent variables manipulated and consequently no conditions.

Materials

Three online platforms were used in the study: Amazon Mechanical Turk (MTurk) for recruitment, Qualtrics for data collection, and YouTube for hosting instructional videos on how to complete thought records. People who registered for the task on MTurk were redirected to Qualtrics. YouTube videos were embedded in Qualtrics.

The instructions for the thought recording task included psychoeducation on cognitive theory, a short description of the components of a thought record, and four video examples of how to complete the thought records using two scenarios and four fictional characters to emphasize that thought records are highly individual and that there are no incorrect answers as long as thought records are coherent.

The scenarios of the scenario-based (closed) thought records for any participant were chosen from a set of ten possible scenarios. These were divided into two sets of five scenarios, one set comprising scenarios of an interpersonal nature, the other comprising scenarios of an achievement-related nature. The scenarios were taken from the Ways of Responding Questionnaire [37] and the Cognitive Error Questionnaire [35]. A complete list of the scenarios can be found in the data repository[1] of this study. The open thought record followed the exact same structure as the closed ones, except that participants had to briefly describe a situation that happened in their life instead of first imagining themselves in a given scenario and then describing it again in their own words.

The formulation of the downward arrow technique (DAT) questions depended on the emotion category that participants selected. When this was *happiness*, they were not directed to complete the DAT after stating the automatic thought. Therefore, all thought records in our dataset have at least one utterance: the automatic thought. When selecting *sadness* or *anger* the DAT consisted of repeatedly asking "And why would it be upsetting to you if [previously stated thought] were true? What would it mean to you? What does it say about you?" When selecting *fear*, on the other hand, the corresponding question was "And what would be the worst that could happen if [previously stated thought] were true? What would it mean to you? What does it say about you?" Just like the thought records, the DAT was altered slightly to better fit online administration: after each step, participants were asked whether they wanted to continue with the technique or not. This was necessary to eventually break the loop while giving participants the chance to complete as many steps as they wanted.

Measures

Three mental health questionnaires were used: the Hospital Anxiety and Depression Scale (HDAS) [40], the Beck Depression Inventory (BDI-IA) [41], and the Cognitive Distortions Scale (CDS) [38]. The HDAS is a diagnostic tool for depression and anxiety, while the BDI-IA only assesses symptoms of depression. The CDS measures to what degree someone suffers from cognitive distortions, such as black-and-white thinking, in achievement-related as well as in interpersonal situations.

[1] To access the data repository during the reviewing process, please use the following link `https://tinyurl.com/y4u6tr4x` with password: NLP4TRs. This url is the temporary storage location during the reviewing process and will be replaced with a permanent one from the 4TU Center for Research Data upon acceptance.

The post-questionnaire comprised three items asking participants how difficult and how enjoyable they found it to complete a thought record, and to indicate how many thought records they think they would complete if they were asked to complete a thought record daily for a period of seven days. We collected this data as secondary measures in anticipation of follow-up research, in which we aim to implement a conversational agent to motivate users to regularly record their thoughts.

Participants

The only qualifications participants needed to access the task on MTurk was to be located in the USA, Canada, the UK, or Australia, to be at least 18 years of age, and to never have participated in the same study before. A total of 536 participants accepted the task on MTurk. Of these, 320 responses were usable. Hence, approximately 40 % of responses had to be excluded on the basis of participants failing at least one of the two instruction comprehension questions or not taking the task seriously (having filled in incomprehensible text or obviously having copied and pasted text from other websites into the text-entry fields). Excluded participants were not reimbursed.

Of the 320 included participants, 148 were female, 171 were male, and 1 indicated *Other*. The mean age of 319 participants was 36.25 years (SD=10.99) with the youngest being 19 and the oldest 71. Demographic questions were optional and one participant chose not to provide her age.

```
> df.datademo <- read.csv("Data/Demographics.csv",
+                         na.strings = "", header=TRUE,
+                         sep=";",fill=TRUE)
> #select only the interesting columns
> df.datademo <- df.datademo[,c("Duration","Gender","Age")]
> #calculate average duration in minutes
> durinmin <- mean(as.numeric(as.character(df.datademo$Duration)))/60
> #distribution of genders
> df.gender <- df.datademo %>%
+   group_by(Gender) %>%
+   summarise(genders=n())
> print(df.gender)

# A tibble: 3 x 2
  Gender genders
   <int>   <int>
1      1     148
2      2     171
3      4       1

> #distribution of age
> psych::describe(as.numeric(as.character(df.datademo$Age)))

   vars   n  mean    sd median trimmed mad min max range skew kurtosis   se
X1    1 319 36.25 10.99     33   35.03 8.9  19  71    52 0.97     0.31 0.62
```

Procedure

Participants fulfilling the qualification criteria could access the task in MTurk. There, they were presented with basic information about the study, such as a short description of the task and the expected time to complete it (35 minutes). Once having accepted the task, participants were redirect to Qualtrics for the experiment. Upon giving their

explicit consent to six statements, they were forwarded to a short demographic pre-questionnaire followed by the task instructions. To ensure that participants would not rush through the instructions, two instruction comprehension questions completed the instructional part: one asking participants what they would have to do in the main task in general and the other concerning procedural aspects of how to complete the thought records as explained in the videos. Failing to answer at least one of the questions correctly resulted in the immediate exclusion of the participant. This was made clear to participants before reaching the questions and the questions were displayed on the same page as the instructions, allowing participants to re-read instructions or re-watch videos before giving their answer. Participants who answered both instruction comprehension questions correctly were forwarded to the thought recording task. This consisted of four closed and one open thought record in this order. For the closed thought records, they were asked to first read the short scenario description and imagine themselves in the situation. They were then directed to a new page with the first thought record form. Throughout the process of completing this, it was possible at any point for the participants to access a short version of the instructions again.

The thought record form was followed by the downward arrow technique. After each step of the DAT, participants were asked whether they wanted to continue with another step. This allowed repeatedly reminding them of the stopping criteria: repeating oneself or feeling that answers were becoming somewhat ridiculous. After indicating that they did not want to continue with the DAT or in case of having selected *happiness* as the emotional response to the situation, participants were presented with the final thought record question. This concerned the behavior they would expect themselves to exhibit in the situation. The post-questionnaire and the three mental health scales completed participation. The entire experimental flow is visualized in S1 Appendix.

Data and analysis strategies

```
> #read in the core thought record data
> df <- read.csv("Data/CoreData.csv",na.strings = "",
+                header=TRUE, sep=";",fill=TRUE)
> schemas <- c("Attach","Comp","Global","Health","Control",
+              "MetaCog","Others","Hopeless","OthViews")
> #select only relevant columns and rows
> df <- df[which(df$UttEnum!="NA"),
+          c("Reply",schemas,"Exclude","UttEnum","Scenario",
+            "Depth","Participant.ID")] %>% na.omit(.)
> #rename Reply column to Utterance
> names(df)[names(df) == "Reply"] <- "Utterance"
> df[,2:11] <- lapply(df[,2:11], function(x) as.numeric(as.character(x)))
> # we remove the exclude sentences from the set
> df <- df[which(df$Exclude==0),]
> # then we can also remove the exclude column
> df$Exclude <- NULL
> # we also want a column that says whether the thought
> # record was scenario-based (closed) or a personal one (open)
> df$TRtype <- ifelse(df$Scenario=="PTR","open","closed")
> df$TRtype <- as.factor(df$TRtype)
```

To obtain a labeled dataset for training the schema identification models, the thought record utterances had to be scored manually. To this end, we used the schema rubric developed by Millings and Carnelley [7]. This rubric comprises ten categories, of

which nine are well-defined schemas, such as *Attachment* or *Meta-Cognition*. The final category, however, is an "other" category for all thought records that cannot be assigned one of the well-defined schemas. Schemas are not mutually exclusive, a thought record can therefore be labeled with multiple schemas. We made three modifications to the original rubric. The first modification pertains to the area of application: the original rubric is always applied to an entire thought record, while we apply it to thought record utterances. As a second modification, we dropped the *Other* category, but allowed utterances to have a 0-score for all of the nine schemas labels. As the final modification, we have altered the original rubric from an utterance being indicative of an underlying schema or not (binary schema label) to it being indicative of an underlying schema to a certain degree (ordinal schema score). The schema scores that we assign range from *has absolutely nothing to do with the schema* (0) over *corresponds a little bit with the schema* (1) and *corresponds largely with the schema* (2) to *corresponds completely with the schema* (3).

The schemas of thought record utterances and the scenario type of the open thought record had to be manually scored. Table 2 shows example thought record utterances from our dataset for each of the nine schemas and the nine scores assigned to each of the utterances. All manual scoring was conducted by the first author, who scored the utterances in random order. To obtain an indication of reliability, an additional coder, a graduate student of clinical psychology, scored a subset of the utterances. For this, three subsets of 50 randomly selected utterances were used to train the coder until agreement on the interpretation of definitions was reached. Any scoring deviation of more than one point on the ordinal scale was discussed. Then the second coder coded another subset of 100 randomly chosen utterances. Interrater agreement between the first and second coder on this subset was substantial (weighted Cohen's $\kappa = 0.79$). The first coder also recoded the same subset one year after completing the initial coding of all utterances with good intracoder agreement (weighted Cohen's $\kappa = 0.83$).

```
> #calculate interrater reliability between c1 initial and c1 after 1
> #year, and c1 initial and c4
> #read in again
> df.subset_100_test <-
+   read.csv("Data/IRR/c1c4/Testing/testset.csv",
+            na.strings = "",
+            header=TRUE,
+            sep=";",
+            fill=TRUE)
> #we collapse the final 3 categories of original dataset to one
> #"Other" category
> df.subset_100_test$Other2 <- apply(df.subset_100_test[3:11],1,sum)
> df.subset_100_test$Other <- ifelse(df.subset_100_test$Other2==0, 1,0)
> df1 <- df.subset_100_test[3:11]
> df.test.recoded_c1 <-
+   read.csv("Data/IRR/c1c4/Testing/testset_recoding_c1.csv",
+            na.strings = "",
+            header=FALSE,
+            sep=";",
+            fill=TRUE)
> #copy header from test df
> df2 <- df.test.recoded_c1[2:10]
> colnames(df2) <- colnames(df1)
> df.test_c4 <-
+   read.csv("Data/IRR/c1c4/Testing/testset_coded_c4.csv",
```

```
+            na.strings = "",
+            header=FALSE,
+            sep=";",
+            fill=TRUE)
> #copy header from test df
> df3 <- df.test_c4[2:10]
> colnames(df3) <- colnames(df1)
> #change dataframes to long format
> df1.long <- gather(df1,schema,rating,Attach:OthViews,
+                    factor_key = TRUE)
> df2.long <- gather(df2,schema,rating,Attach:OthViews,
+                    factor_key = TRUE)
> df3.long <- gather(df3,schema,rating,Attach:OthViews,
+                    factor_key = TRUE)
> #calculate IRR as weighted cohen's kappa
> wkappa_c1c1 <- kappa2(data.frame(df1.long$rating,df2.long$rating),
+                       weight="squared")
> wkappa_c1c1

 Cohen's Kappa for 2 Raters (Weights: squared)

 Subjects = 900
   Raters = 2
    Kappa = 0.834

        z = 25
  p-value = 0

> wkappa_c1c4 <- kappa2(data.frame(df1.long$rating,df3.long$rating),
+                       weight="squared")
> wkappa_c1c4

 Cohen's Kappa for 2 Raters (Weights: squared)

 Subjects = 900
   Raters = 2
    Kappa = 0.787

        z = 23.7
  p-value = 0
```

Table 2. Example utterances for each schema taken from the dataset collected in this study. Scores were manually assigned for each of the nine mental health schemas by the first author.

Utterance	S1	S2	S3	S4	S5	S6	S7	S8	S9
S1: Attachment examples									
I am unlovable and less than other people. I will never find friends or a girlfriend.	3	0	3	0	0	0	0	1	0
I don't want to be alone.	3	0	0	0	0	0	0	0	0
I was a bad mom.	3	0	0	0	0	0	0	0	0
I failed at the relationship.	3	0	0	0	0	0	0	0	0
I won't be a good partner to others.	3	0	0	0	0	0	0	0	0
S2: Competence examples									

I feel like a failure at my job.	0	3	0	0	0	0	0	0	0
I'm unprepared for this task.	0	3	0	0	0	0	0	0	0
I can never go into a sales job.	0	3	0	0	0	0	0	0	0
I am not good enough to get a job.	0	3	0	0	0	0	0	0	0
I would be unable to produce saleable work.	0	3	0	0	0	0	0	0	0
S3: Global self-evaluation examples									
It would mean that I am lazy and I need to do better	0	0	3	0	0	0	0	0	0
I should never have been born.	0	0	3	0	0	0	0	3	0
I am selfish.	0	0	3	0	0	0	0	0	0
S4: Health examples									
I would become ill.	0	0	0	3	0	0	0	0	0
I feel exhausted and anxious.	0	0	0	2	1	0	0	1	0
I cannot lose weight no matter what I try.	0	0	0	3	2	0	0	0	0
It would be very depressing, it would say that I would need counseling to get through life.	0	0	0	3	1	0	0	1	0
I will have health issues	0	0	0	3	0	0	0	0	0
S5: Power and control examples									
I'm going to be stuck in my current situation forever.	0	0	0	0	3	0	0	1	0
The feeling of being pressured by my boss.	0	0	0	0	3	0	0	0	0
I was fired and not given a chance to succeed.	0	1	0	0	2	0	0	0	0
I am not in control of what I do or how I perceive myself	0	0	0	0	3	2	0	0	0
That I still have a target painted on my back for their abuse.	1	0	0	0	3	0	1	0	0
S6: Meta-Cognition examples									
My perception of people is off and that's why I have a difficulty creating new relationships.	1	0	0	0	0	3	1	0	0
That I can be more than a bit compulsive about investigating odd byways of thought.	0	0	0	0	0	3	0	0	0
I trick myself into believing I'm better than I am.	0	0	0	0	0	3	0	0	0
Because I hold myself to a high standard.	0	0	0	0	0	2	0	0	0
I get angry easily over small things.	0	0	0	0	3	1	0	0	0
S7: Other people examples									
People would rather avoid me than be in my presence.	0	0	0	0	0	0	2	0	3
It means that these people not care about anyone but themselves, and i have to suffer	0	0	0	0	0	0	3	0	0
People will mock me	0	0	0	0	0	0	3	0	3
I am not as selfish as other people.	0	0	0	0	0	0	3	0	0
It means that other people can do despicable things and not be accountable.	0	0	0	0	0	0	3	0	0
S8: Hopelessness examples									
I will stop trying in life and give up	0	0	0	0	1	0	0	3	0
I should never have been born.	0	0	3	0	0	0	0	3	0
Depression makes me think I'd be better off dead.	0	0	0	2	0	0	0	3	0
I will never have a life I enjoy	0	0	0	0	0	0	0	3	0

I'll never feel like I have a purpose.	0	0	0	0	0	0	0	3	0
S9: Others views about self examples									
My friends don't like me.	2	0	0	0	0	0	0	0	3
Because I want people like him to like me.	0	0	0	0	0	0	0	0	3
I could not make him see that I am a responsible person.	0	0	0	0	0	0	0	0	3
I must not be his type of person.	0	0	0	0	0	0	0	0	2
It would say that she did not feel like she was able to talk to me.	0	0	0	0	0	0	0	0	3

H1: Schemas can be automatically extracted

To test the first hypothesis, thought record utterances were studied taking a natural language processing perspective: using a machine learning model to score an utterance with regard to the nine well-defined schemas. This task can formally be described as an ordinal multi-label scoring task: an algorithm must assign each utterance a schema vector consisting of nine values ranging between 0 and 3. Assigning ordinal scores to data is generally not trivial and common simplifications are to either treat the ordinal scores as separate classes (nominal data) or as equidistant integers on a continuum (interval data) [47]. The former is otherwise known as classification and entails that the ordering information of scores is lost. The latter is regression and entails that the ordering is maintained, but information is added, such as that labels are equally spaced and that the space between labels can be meaningfully interpreted. Where specific algorithms have been created for ordinal data [47], these often assume that higher ordinal labels subsume lower ones (compare, for example, [48]), e.g., if something corresponds very much to a schema (score 3) it also automatically corresponds a little bit to the schema (score 1). This is not the case here, as we also have score 0 meaning that an utterance does not correspond to a schema. Another criterion for choosing algorithms was the ready availability of functional, well-maintained, and commonly used software packages. We assume this to work to the advantage of reproducibility and further development. As a result of these considerations, we opted to explore both approaches of treating the scores as nominal as well as treating them as interval rather than exploring specific ordinal methods.

Before automatically scoring, the data were linguistically preprocessed by lower-casing, replacing misspellings, contractions, and numbers, adding missing sentence end marks and comma space, and finally removing stop words and unnecessary white space.

```
> #we can examine what impurities are in our text
> #check_text(df$Utterance)
>
> #function that cleans textual utterances. Accepts vector of strings
> #and returns text corpus (indeces are identical)
> clean_utts <- function(utts){
+   utts <- utts %>%
+     tolower(.) %>% #make everything lower case
+     replace_misspelling(.) %>% #try to correct misspelled words
+     replace_contraction(.) %>% #expand contractions, can't -> cannot
+     replace_number(.) %>% #replace numbers with words
+     replace_incomplete(.) %>% #adds/changes missing sentence end
+     add_comma_space(.) %>% #adds a space after comma
+     rm_stopwords(., strip = TRUE, separate = FALSE)
+   #remove excess white spaces
```

```
+   utts <- sapply(utts, function(x) gsub("\\s+"," ",x))
+   return(utts)
+ }
> df$Utterance <- clean_utts(df$Utterance)
```

They were then divided into a training set, a validation set, and a test set, with the test set comprising 15 % of all data, the validation set comprising another 12.75 %, and the training set comprising the remaining 72.25 %. Samples to include in test and validation set were not selected at random but rather we ensured that three criteria were fulfilled: 1. similar distribution of schemas, 2. approximately the same proportion of open and closed scenarios, 3. approximately the same distribution over DAT depths as in the entire dataset. This was achieved by randomly sampling 1000 times from the entire distribution, determining the deviation in distribution between the sample and the population for each of the three criteria, summing these three deviation measures, and choosing the sample with the smallest result. The process was first done for the test set and then repeated with the remaining data samples to obtain the validation set. We used normalized, 100-dimensional GLoVE embeddings [42] trained on all English Wikipedia articles existent in 2014 to represent the words in utterances.

```
> #sampling function to fulfil the three criteria above
> controlled_sampling_H1 <- function(df.pop,perc){
+   i<-1 #iteration index
+   mse<-1 #the largest possible initial deviation (mean standard error = 1)
+   #initialize the sample as the entire population
+   selected.sample <- df.pop
+   # to long format with schemas labeled as cb
+   df.poplong <- df.pop[,c(schemas,"TRtype")] %>%
+     gather(cb,label,1:9)
+   # dataframe for distribution over schemas
+   df.cbdist <- df.poplong %>%
+     group_by(cb) %>%
+     summarise(labeled=sum(label),count=n(), perc=labeled/count)
+   # dataframe for distribution over open/closed thought reccords
+   df.ocdist <- df.poplong %>%
+     group_by(TRtype) %>%
+     summarise(c=n(),perc=c/nrow(df.poplong))
+   # dataframe for distribution over DAT depth
+   df.depthdist <- df.pop[,c("Depth","TRtype")] %>%
+     group_by(Depth) %>%
+     summarise(dcount=n(),perc=dcount/nrow(df.pop))
+   while(i<1000){
+     #get sample of full population and repeat what we did above
+     df.sample <- sample_n(df.pop,ceiling(nrow(df.pop)*(perc/100)))
+     df.samplelong <- df.sample[,c(schemas,"TRtype")] %>%
+       gather(cb,label,1:9)
+     df.cbdist2 <- df.samplelong %>% group_by(cb) %>%
+       summarise(labeled=sum(label),count=n(), perc=labeled/count)
+     df.ocdist2 <- df.samplelong %>% group_by(TRtype) %>%
+       summarise(c=n(),perc=c/nrow(df.poplong))
+     df.depthdist2 <- df.sample[,c("Depth","TRtype")] %>% group_by(Depth) %>%
+       summarise(dcount=n(),perc=dcount/nrow(df.pop))
+     #now we can compare with population distribution
+     comp_cbdist <- mean(abs(df.cbdist$perc-df.cbdist2$perc))
```

```
+     comp_ocdist <- mean(abs(df.ocdist$perc-df.ocdist2$perc))
+     comp_depthdist <- mean(abs(df.depthdist$perc-
+                                  df.depthdist2$perc))
+     new_mse <- mean(c(comp_cbdist,comp_ocdist,comp_depthdist))
+     #update if the new mse is smaller than the previous one
+     if(new_mse < mse){
+       mse <- new_mse
+       selected.sample <- df.sample
+     }
+     i <- i+1
+   }
+   return(selected.sample)
+ }
> H1_set_split <- function(df1,df2,perc){
+   #df1 has binary labels, df2 has ordinal labels
+   df.intermediate_test <- controlled_sampling_H1(df1,15)
+   df.test <- df2[which(df2$UttEnum %in% df.intermediate_test$UttEnum),]
+   df.intermediate_train <- df1[which(df1$UttEnum %nin%
+                                        df.intermediate_test$UttEnum),]
+   df.intermediate_validate <- controlled_sampling_H1(
+     df.intermediate_train,15)
+   df.validate <- df2[which(df2$UttEnum %in%
+                              df.intermediate_validate$UttEnum),]
+   df.train <- df[which(df$UttEnum %nin%
+                          df.validate$UttEnum &
+                          df$UttEnum %nin% df.test$UttEnum),]
+   sets <- list("train"=df.train,"val"=df.validate,"test"=df.test)
+   return(sets)
+ }
> #we recode from the ordinal scale to a binominal one
> #to split the set into training, validation, and test set
> dfbin <- df
> dfbin[,2:10] <- ifelse(dfbin[,2:10] == 2 | dfbin[,2:10]== 3, 1, 0)
>
> # #we write the sets to a file to save it
> # write.table(sets.H1$test[:,1],"Data/DatasetsForH1/H1_test_texts.csv",
> #             sep=";",
> #             row.names=FALSE)
> # write.table(sets.H1$test[:,2:10],"Data/DatasetsForH1/H1_test_labels.csv",
> #             sep=";",
> #             row.names=FALSE)
> # write.table(sets.H1$val[:,1],"Data/DatasetsForH1/H1_validate_texts.csv",
> #             sep=";",
> #             row.names=FALSE)
> # write.table(sets.H1$val[:,2:10],"Data/DatasetsForH1/H1_validate_labels.csv"
> #             sep=";",
> #             row.names=FALSE)
> # write.table(sets.H1$train[:,1],"Data/DatasetsForH1/H1_train_texts.csv",
> #             sep=";",
> #             row.names=FALSE)
> # write.table(sets.H1$train{:,2:10},"Data/DatasetsForH1/H1_train_labels.csv",
> #             sep=";",
```

```
> #              row.names=FALSE)
```

Three types of algorithms of varying levels of complexity were chosen for the task: k nearest neighbors classification (kNN-C) and regression (kNN-R), support vector machine classification (SVC) and regression (SVR), and a multi-label recurrent neural net (RNN) as well as a set of separate RNNs per schema. The kNN algorithms serve as a baseline as they are not trainable. Rather, for each new utterance that the algorithm encounters, a distance is calculated between this utterance and each of the utterances of the training set, in this case the cosine distance. The k closest utterances (k neighbors) are selected and their scores arithmetically combined to produce the scores for the unseen utterance. In the case of kNN-C, we combine the scores of the k neighbors with a conservative *mode* function, i.e., the unseen utterance is assigned the score that the majority of neighbors carry and the one with the lowest value if multiple exist. In the case of kNN-R, we combine the values by averaging the scores of the nearest neighbors.

Support vector machines are trainable and suited for high-dimensional feature spaces. In support vector classification, the algorithm aims to maximize the margin on both sides of the decision boundary between the classes. In support vector regression, on the other hand, the aim is to maximize the margin around the regression line such that the error remains below a certain threshold. For both SVM algorithms, we standardized the utterance vectors and trained separate models for each schema. For the kNN and SVM algorithms, we chose to represent utterances as the average of all word vectors in the linguistically preprocessed utterance.

RNNs are a type of deep neural network designed for handling sequential input data. Thus, unlike the kNN and the SVM approaches, they can account for the temporal aspect of utterances as sequences of words. Again, two ways of modelling the data were explored: a set of separate RNNs per schema and a multi-label RNN. The per-schema RNNs allow for assessing the potential added benefit of the deep neural network architecture. For these models, we treat the ordinal scores as separate classes, ignoring the ordering. Each of the nine RNNs in the set outputs a vector of four values between 0 and 1, each value expressing the confidence of the algorithm that the utterance should be assigned a score of 0, 1, 2, or 3 for the specific schema. To obtain the schema score, the score with the highest confidence is selected. The multi-label RNN, on the other hand, can leverage interdependencies between the schemas as it has knowledge of all schema scores at the same time. It predicts all nine schemas simultaneously and outputs a value between 0 and 1 for each schema. In preparing our analysis script for publication, we encountered the challenge that despite setting all random seeds as required, the trained RNNs showed a small degree of variability in the output when re-running the script. We have therefore chosen a stochastic approach: for both RNN approaches, we first train the models 30 times, we then predict all items of the test set with all 30 models, and finally, we select for the median model in terms of performance. All results reported below are based on the median multi-label RNN and the median per-schema RNN set.

It must be stressed at this point that we only test whether a machine is able to detect patterns at all and do not strive to obtain the best scoring performance. As a consequence, a number of refinement possibilities, such as sequence to vector models or extensive hyperparameter tuning, were not explored.

H2: Downward arrow converges

To examine whether utterances developed with the downward arrow technique converge to a schema, we aimed to predict the algorithm's *scoring accuracy* from the *depth* of utterances. We assigned *depth*= 1 for the automatic thought and increased it

incrementally with every downward arrow technique step. Fig 1 shows the number of thought records in our dataset with a specific depth.

```
> #plot theme
> # jp2theme <- theme(
> #   plot.title=element_blank(),
> #   plot.margin=unit(c(10,5,5,5),"mm"),
> #   #plot.title = element_text(hjust = 0.5),
> #   legend.title = element_text(family="Arial", size = 12),
> #   legend.text = element_text(family="Arial", size = 12,
> #                              margin=margin(0,0,0,5)),
> #   legend.spacing.y=unit(.2, "cm"),
> #   legend.key = element_rect(colour = NA),
> #   axis.title.y=element_text(family="Arial", size=14,
> #                             face="bold", vjust=2),
> #   axis.title.x=element_text(family="Arial", size=14,
> #                             face="bold", vjust=-2),
> #   axis.text.y = element_text(family="Arial", size=12),
> #   axis.text.x = element_text(family="Arial", size=12))
> #
> # #Number of thought records with a certain maximum depth
> # #width=384 height=325
> # df.depthsum <- df %>%
> #   dplyr::group_by(Participant.ID,scen) %>%
> #   dplyr::summarise(maxdepth = max(depth))
> #
> # trdepth <- ggplot(df.depthsum, aes(maxdepth)) +
> #   geom_bar(fill = "#0073C2FF") +
> #   labs(y = "Number of thought records") +
> #   scale_x_continuous("Maximum depth", breaks = c(1,5,10)) +
> #   jp2theme
> #
> # tiff("figures/TRdepth.tiff", width= 2408, height= 1617, units="px", res=300
> # plot(trdepth)
> # dev.off()
```

To determine *scoring accuracy*, we used the predictions made on the test set with the median set of per-schema RNNs of Hypothesis 1. For each utterance, the Spearman correlation between the algorithmically predicted and manually assigned scores serves as the measure. Thus, if an utterance such as "I will never be loved" was scored as $[3, 0, 0, 0, 0, 0, 0, 1, 0]$ manually on the nine schemas and received the scores $[2, 0, 0, 0, 1, 0, 0, 1, 2]$ by the RNN, the resulting *scoring accuracy* for this utterance would be $\rho = 0.59$, i.e., the Spearman correlation between the two vectors of scores.

```
> #read in the testset to have manually assigned labels
> df.H2GT <- read.csv("Data/DatasetsForH1/H1_test_labels.csv",
+                     na.strings = "",
+                     header=TRUE,
+                     sep=";",
+                     fill=TRUE)
> #read in the predictions generated by the per-schema RNNs trained in Python
> df.H2Preds <- read.csv("Data/PredictionsH2.csv",
+                     na.strings = "",
```

```
+                     header=TRUE,
+                     sep=";",
+                     fill=TRUE)
> df.H2Preds$UttEnum <- df.H2GT$UttEnum
> #compute the row-wise correlations between the two dataframes
> df.H2 <- merge(df.H2Preds[,c("UttEnum","Corr")],
+                df[,c("UttEnum","Scenario","Depth","Participant.ID")],
+                by="UttEnum")
```

To study the effect of depth on scoring accuracy, we conducted a multilevel analysis; the data structure required a three-level linear model with the *depth* as a fixed effect and the *automatic scoring accuracy* as the dependent variable. For each participant (Level 3), there are several thought records (Level 2) and for each thought record, there are several utterances (Level 1). The null model predicts the scoring accuracy from the mean scoring accuracy per participant and thought record. The model therefore has random intercepts at Level 3 and at Level 2 nested within Level 3 (thought records nested in participants). For Model 1, the fixed effect *depth* was added to the null model. We expected to see an increase in automatic scoring accuracy as utterance depth increases.

Fig 1. Distributions of thought records over depth. Number of thought records having a certain depth, the depth is the number of downward arrow steps + 1 for the automatic thought.

H3: Schema patterns are similar across thought record types

The next analyses tested whether schemas observed in the scenario-based (closed) thought records are predictive of schemas observed in real-life (open) thought records. For this, only the manually assigned scores were used. Each participant completed two achievement-related and two interpersonal closed thought records. The first author labeled all open thought record scenarios as either interpersonal or achievement-related (intercoder agreement with a second independent coder on all open thought records was substantial with Cohen's $\kappa = 0.68$).

```
> #read in labels of c1
> df.IRR_PTR_c1 <- read.csv("Data/IRR/c1c6/labeling_PTR_scen_c1_labeled.csv",
+                           na.strings = "",
+                           header=TRUE,
+                           sep=";",
+                           fill=TRUE)
> df.IRR_PTR_c2 <- read.csv("Data/IRR/c1c6/labeling_PTR_scen_c6_labeled.csv",
+                           na.strings = "",
+                           header=TRUE,
+                           sep=";",
+                           fill=TRUE)
> IRR_PTR_kappa <- kappa2(data.frame(df.IRR_PTR_c2$scentype,
+                                    df.IRR_PTR_c1$scentype))
> IRR_PTR_kappa

 Cohen's Kappa for 2 Raters (Weights: unweighted)

 Subjects = 319
   Raters = 2
```

```
    Kappa = 0.676

        z = 12.4
  p-value = 0
```

```
> schemas <- c("Attach","Comp","Global","Health","Control",
+               "MetaCog","Others","Hopeless","OthViews")
> df.PTRlabeled <- read.csv("Data/IRR/c1c6/labeling_PTR_scen_c1_labeled.csv",
+                           na.strings = "",
+                           header=TRUE,
+                           sep=";",
+                           fill=TRUE)
> df.PTRlabeled$Scenario <- "PTR"
> #append scentype to current dataframe
> #interpersonal
> df$scentype[df$Scenario %in% c('Waving','Lover','FA','CW','Party')] <- 0
> #achievement-related
> df$scentype[df$Scenario %in% c('Essay','CB','JA','Fired','Dieting')] <- 1
> #fill missing scentypes with newly loaded data
> df.dataH3 <- merge(df,
+                    df.PTRlabeled[,c("Participant.ID",
+                                     "scentype",
+                                     "Scenario")],
+                    by=c("Participant.ID","Scenario"),
+                    all.x=TRUE)
> df.dataH3$scentype <- ifelse(is.na(df.dataH3$scentype.x),
+                              df.dataH3$scentype.y,
+                              df.dataH3$scentype.x)
> df.dataH3$scentype.x <- NULL
> df.dataH3$scentype.y <- NULL
```

Nine linear regression models were fit with *schema presence in closed thought records* as the only predictor and *schema presence in the open thought record* as the only outcome variable. Thus, we fit one model for each schema. For example, in the Health schema model, the presence of the Health schema in closed thought records predicts the presence of the Health schema in open thought records. To determine *schema presence in closed thought records*, we identified the two closed thought records with the same situation type (interpersonal or achievement-related) as the open thought record. For each of the nine schemas, we then took the highest score across utterances of both closed thought records and average these two values together.

```
> #get max value per schema for the open thought record
> df.openTR <- df.dataH3[df.dataH3$Scenario == 'PTR',
+                        c('Participant.ID',schemas,"scentype")] %>%
+   dplyr::group_by_('Participant.ID') %>%
+   summarise_all(funs(max))
> #get the closed TRs that match with the open TR in scentype
> df.closedTR <- merge(df.openTR[,c("Participant.ID","scentype")],
+                      df.dataH3,
+                      by=c("Participant.ID","scentype"))
> #remove the PTRs
> df.closedTR <- df.closedTR[!df.closedTR$Scenario=="PTR",]
> # ####alternative test with all closed thought records,
```

```
> # ####meaning we do not split according to scentype
> # df.closedTR <- df.dataH3[!df.dataH3$Scenario=="PTR",]
>
> #get max value per schema and thought record for the
> #closed thought records
> df.closedTRagg <- df.closedTR[,c('Participant.ID',schemas,"Scenario")] %>%
+   dplyr::group_by_('Participant.ID','Scenario') %>%
+   summarise_all(funs(max))
> #get average value per schema for both closed thought records
> df.closedTRagg <- df.closedTRagg %>%
+   dplyr::group_by_('Participant.ID') %>%
+   summarise_all(funs(mean))
> df.closedTRagg$Scenario <- NULL
> #merge the two dataframes
> df.H3lms <- merge(df.closedTRagg,df.openTR,by="Participant.ID")
```

For example, let us assume that a participant described an interpersonal situation in the open thought record. To calculate the predictor for the *Health* schema, the two interpersonal closed thought records of this participant were identified and from each the highest score obtained on the *Health* schema across utterances was taken, leading to two scores, which were then averaged together. We followed the same procedure for the outcome variable, *schema presence in the open thought record*. However, since there is only one such thought record for each participant, no averaging was needed. S2 Appendix illustrates the procedure with a concrete example for clarification.

H4: Mental illnesses have associated schemas

The final hypothesis is an exploratory investigation of whether the outcomes from the mental health questionnaires can be predicted from the schema patterns. To this end, we created a summary score per schema and participant. The summary score was calculated by first taking per participant, thought record, and schema the maximum score (0-3) across utterances. This gives one value for each schema for the five thought records a participant completed. These values were then re-coded into a binomial value, with all values smaller or equal to 2 mapping to 0 and 3 mapping to 1. Thus, we only considered schemas that were clearly and unambiguously present. Finally, the binomial values were summed within a participant. Each participant could therefore obtain a maximum value of 5 for a schema if the schema was clearly present in all five completed thought records of the participant.

```
> #to create the dataset for H4, we merge the dataframe containing the
> #mental health data with that containing utterances and labels
> df.mentalh <- read.csv("Data/MentalHealth.csv",
+                        na.strings = "",
+                        header=TRUE,
+                        sep=";",
+                        fill=TRUE)
> df.mentalh <- df.mentalh[56:61]
> df.mentalh[1:5] <- lapply(df.mentalh[1:5],
+                           function(x) as.numeric(as.character(x)))
> #dataset for the multilabel models
> df.H4 <- df %>%
+   dplyr::group_by(Participant.ID,Scenario) %>%
+   dplyr::summarise_at(schemas,max) %>%
```

```
+   dplyr::mutate_at(schemas,
+                    function(x) ifelse(x > 2, 1,0)) %>%
+   dplyr::group_by(Participant.ID) %>%
+   dplyr::summarise_at(schemas,sum)
> df.H4 <- df.H4[,c('Participant.ID',schemas)]
> df.H4 <- merge(df.mentalh,df.H4,by="Participant.ID",all.y=TRUE)
```

We then created five linear models, each one taking one of the mental health measures (HDAS Depression, HDAS Anxiety, BDI, Cognitive Distortions Relatedness, Cognitive Distortions Achievement) as outcome variable. Every model has the nine schemas as predictors. Since the same data were used to predict five different outcomes, we used a Bonferroni correction to adjust the significance threshold to $\alpha = 0.05/5 = 0.01$.

Results

To gain insight into the collected data, Table 3 shows the frequencies of each score per schema. In total, there were 5747 utterances.

```
> #how many sentences are labeled as belonging to a specific
> #schema score
> df.long <- df[,c(schemas,"TRtype")] %>%
+ gather(schema,label,1:9)
> #show percentage of schema scores
> df.schemadist <- df.long[,2:3] %>%
+   group_by(schema,label) %>%
+   dplyr::summarise(countl=dplyr::n())
> df.schemadist <- df.schemadist %>%
+   group_by(schema) %>%
+   dplyr::mutate(countall=sum(countl),perc=countl/countall)
> df.schemadist
```

```
# A tibble: 36 x 5
# Groups:   schema [9]
   schema  label countl countall   perc
   <chr>   <dbl>  <int>    <int>  <dbl>
 1 Attach      0   4047     5747 0.704
 2 Attach      1    446     5747 0.0776
 3 Attach      2    272     5747 0.0473
 4 Attach      3    982     5747 0.171
 5 Comp        0   4151     5747 0.722
 6 Comp        1    314     5747 0.0546
 7 Comp        2    157     5747 0.0273
 8 Comp        3   1125     5747 0.196
 9 Control     0   5089     5747 0.886
10 Control     1    390     5747 0.0679
# ... with 26 more rows
```

H1: Schemas can be automatically extracted

For the majority of schemas, all algorithms could assign scores to the utterances that correlated with the human scores well above what would be expected by chance alone (see Table 4). Furthermore, for all schemas, there was at least one effective algorithm.

Table 3. Number of utterances with a specific score per schema as manually scored by the first author. Percentages are provided in parentheses. Schemas are sorted as in the article by Millings & Carnelley [7].

Schema	**Score**			
	0 (has absolutely nothing to do with schema)	1 (corresponds a little bit with schema)	2 (corresponds largely with schema)	3 (corresponds completely with schema)
Attachment	4047 (70.42 %)	446 (7.76 %)	272 (4.73 %)	982 (17.09 %)
Competence	4151 (72.22 %)	314 (5.46 %)	157(2.73 %)	1125(19.58 %)
Global self-evaluation	4548 (79.14 %)	226 (3.93 %)	280 (4.87 %)	693 (12.06 %)
Health	5428 (94.45 %)	56 (0.97 %)	46 (0.80 %)	217 (3.78 %)
Power and Control	5089 (88.55 %)	390 (6.79 %)	154 (2.68 %)	114(1.98 %)
Meta-cognition	5626 (97.89 %)	61 (1.06 %)	41 (0.71 %)	19 (0.33 %)
Other people	5593 (97.32 %)	92 (1.60 %)	44 (0.31 %)	18 (0.31 %)
Hopelessness	4931 (85.80 %)	582 (10.13 %)	174 (3.03 %)	60 (1.04 %)
Other's views on self	4688 (81.57 %)	129 (2.24 %)	639 (11.11 %)	29 1(5.06 %)

Table 4. Spearman correlation and bootstrapped confidence intervals of predicted scores with manually assigned scores per model and schema. The result of the best model per schema is shown in bold font.

Schema	**Model Outcome**					
	kNN-C	kNN-R	SVM	SVR	per-schema RNNs	multi-label RNN
Attachment	0.55 [0.51,0.60]	0.63 [0.59,0.65]	0.65 [0.61,0.68]	0.68 [0.65,0.70]	**0.73 [0.70,0.76]**	0.67 [0.66,0.72]
Competence	0.69 [0.64,0.73]	0.66 [0.63,0.69]	0.68 [0.65,0.72]	0.64 [0.61,0.67]	**0.76 [0.72,0.79]**	0.66 [0.64,0.69]
Global self-evaluation	0.40 [0.33,0.46]	0.41 [0.36,0.46]	0.36 [0.31,0.40]	0.49 [0.45,0.52]	**0.58 [0.54,0.63]**	0.49 [0.45,0.53]
Health	0.74 [0.65,0.81]	0.53 [0.44,0.60]	0.73 [0.65,0.81]	0.35 [0.31,0.40]	**0.75 [0.65,0.82]**	0.35 [0.31,0.39]
Power and Control	0.11 [0.02,0.18]	0.23 [0.17,0.27]	nan [0.00,1.00]	0.31 [0.26,0.35]	0.28 [0.20,0.35]	**0.31 [0.27,0.34]**
Meta-cognition	nan [0.00,1.00]	0.10 [0.01,0.20]	nan [0.00,1.00]	0.11 [0.06,0.16]	-0.01 [0.00,-0.01]	**0.11 [0.06,0.14]**
Other people	0.28 [0.00,1.00]	0.24 [0.17,0.31]	nan [0.00,1.00]	0.19 [0.14,0.24]	**0.22 [0.07,0.33]**	0.16 [0.10,0.20]
Hopelessness	0.48 [0.44,0.55]	0.51 [0.47,0.56]	0.49 [0.43,0.53]	0.54 [0.51,0.57]	**0.63 [0.56,0.68]**	0.53 [0.50,0.56]
Other's views on self	0.45 [0.41,0.51]	0.46 [0.42,0.50]	0.48 [0.43,0.53]	0.52 [0.48,0.55]	**0.58 [0.52,0.63]**	0.50 [0.47,0.54]

As determined with the validation set, the best parameter choice for kNN-C was $k = 4$, while for kNN-R, it was $k = 5$. Both support vector approaches performed best with a radial basis function kernel. The best-performing multi-label RNN was trained in batches of 32 utterances and with 100 epochs. It consists of two hidden layers: an embedding layer, performing the GLoVE embeddings, and a bidirectional long short-term memory layer of 100 nodes. It was trained with a dropout probability of 0.1 and categorical cross-entropy loss. The nine nodes of the output layer use a sigmoid activation function. The metric for choosing the best model was the mean absolute error. The individual models, we set up differently, but adopted some of the hyperparameters of the multi-label model (namely, the batch size, the number of LSTM nodes, the dropout rate, and the loss function). For each schema, the individual models have four outputs, one for each of the four possible scores. The activation function of the final layer is a softmax, to express the likelihood with which a certain utterance has each of the scores.

It can be seen from Table 4 that the per-schema RNNs perform best overall. They take the structure of the data most closely into account, both in terms of the utterances (sequential input) and in terms of the scores (one output neuron per score), and were also able to produce the best predictions for most of the schemas. Any possible

advantage of exploiting relationships between schemas was not observable in the results, since the multi-label RNN did not clearly outperform all the other models for any one schema. Interestingly, the *Health* schema is consistently better identifiable by the classification algorithms (kNN-C, SVM, and the per-schema RNNs), while the *Power and Control* schema could be better identified by the regression algorithms (kNN-R, SVR, and multi-label RNN).

H2: Downward arrow converges

The mean correlation between the predicted schema scores and the manually labeled schema scores was found to be 0.75 ($b = 0.75, t(220.76) = 46.97, p < 0.001$) when the nesting structure of utterances nested within thought records and thought records nested within participants is taken into account via random intercepts. Steps at a deeper level could not be scored better by the best model of H1 than steps at a more shallow level. The scoring accuracy, as measured by the Spearman correlation, did not improve with additional steps of the downward arrow technique ($\chi^2(1) = 1.21, p = 0.27$).

```
> ##Only level-1 mean but nesting structure
> #baseline
> Model.0<-lmer(Corr ~ 1
+               +(1|Participant.ID)
+               +(1|Participant.ID:Scenario),
+               data=df.H2, REML=FALSE)
> summary(Model.0)

Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
  method [lmerModLmerTest]
Formula: Corr ~ 1 + (1 | Participant.ID) + (1 | Participant.ID:Scenario)
   Data: df.H2

     AIC      BIC   logLik deviance df.resid
   547.9    565.8   -270.0    539.9      645

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.7759 -0.2849  0.4991  0.6485  1.1318

Random effects:
 Groups                   Name        Variance  Std.Dev.
 Participant.ID:Scenario (Intercept) 0.0006549 0.02559
 Participant.ID          (Intercept) 0.0136053 0.11664
 Residual                            0.1216576 0.34879
Number of obs: 649, groups:  Participant.ID:Scenario, 531; Participant.ID, 269

Fixed effects:
             Estimate Std. Error        df t value Pr(>|t|)
(Intercept)   0.74729    0.01591 220.76280   46.97   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> confint(Model.0,method="boot")

                  2.5 %    97.5 %
.sig01      0.0000000000 0.1537739
```

```
.sig02      0.0001722366 0.1619599
.sigma      0.3134254247 0.3696352
(Intercept) 0.7209953120 0.7810630

> #taking the predictor into account
> #random slope at level 3
> df.H2$Depth <- as.numeric(as.character(df.H2$Depth))
> Model.1<-lmer(Corr ~ Depth
+               +(1|Participant.ID)
+               +(1|Participant.ID:Scenario),
+               data=df.H2, REML=FALSE)
> summary(Model.1)

Linear mixed model fit by maximum likelihood . t-tests use Satterthwaite's
  method [lmerModLmerTest]
Formula: Corr ~ Depth + (1 | Participant.ID) + (1 | Participant.ID:Scenario)
   Data: df.H2

     AIC      BIC   logLik deviance df.resid
   548.7    571.1   -269.4    538.7      644

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.8223 -0.2560  0.4915  0.6427  1.1285

Random effects:
 Groups                  Name        Variance Std.Dev.
 Participant.ID:Scenario (Intercept) 0.00000  0.0000
 Participant.ID          (Intercept) 0.01328  0.1153
 Residual                            0.12230  0.3497
Number of obs: 649, groups:  Participant.ID:Scenario, 531; Participant.ID, 269

Fixed effects:
             Estimate Std. Error        df t value Pr(>|t|)
(Intercept) 7.209e-01  2.874e-02 5.958e+02  25.084   <2e-16 ***
Depth       9.255e-03  8.400e-03 6.484e+02   1.102    0.271
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
      (Intr)
Depth -0.834
convergence code: 0
boundary (singular) fit: see ?isSingular

> confint(Model.1,method="boot")

                   2.5 %     97.5 %
.sig01       0.000000000 0.14974406
.sig02       0.003539445 0.15717689
.sigma       0.311899511 0.37275339
(Intercept)  0.661984572 0.78008553
Depth       -0.008118178 0.02561244
```

```
> anova(Model.0,Model.1)

Data: df.H2
Models:
Model.0: Corr ~ 1 + (1 | Participant.ID) + (1 | Participant.ID:Scenario)
Model.1: Corr ~ Depth + (1 | Participant.ID) + (1 | Participant.ID:Scenario)
        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
Model.0  4 547.94 565.85 -269.97   539.94
Model.1  5 548.74 571.11 -269.37   538.74 1.2065      1      0.272
```

H3: Schema patterns are similar across thought record types

Fig 2 shows the percentage of utterances having a certain schema (manually assigned score > 0) for the open and closed thought records in our dataset. It can be seen that, across participants, schemas are similarly distributed in the two TR types: the mean difference over all schemas is 3.69 % with the *Other people* schema having the smallest difference (0.02 %) and the Competence schema the largest one (9.24 %). Some schemas are more present in open TRs (e.g., the *Power and Control* schema) and others in closed ones (e.g., the *Health* or *Competence* schemas).

```
> #compare frequency of schemas in open and closed thought records
> #recode to binary
> df.dataH3bin <- df.dataH3
> df.dataH3bin[,4:12] <- ifelse(df.dataH3[,4:12] > 0, 1, 0)
> df.H3long <- df.dataH3bin[,c(1,2,seq(4,12,1),15)] %>%
+   gather(cb,label,3:11)
> df.schematrtype <- df.H3long %>%
+   dplyr::group_by(TRtype,cb) %>%
+   dplyr::summarise(count=sum(label),perc=count/n())
>
> # schematrtype <- ggplot(df.schematrtype, aes(cb,perc*100)) +
> #   geom_bar(aes(fill=trtype), position=position_dodge(width=.75), stat="iden
> #   scale_fill_brewer(palette="Paired") +
> #   labs(x="Schema", y = "Pecentage of utterances", fill="Thought record type
> #   theme(axis.text.x = element_text(family="Arial", size=12, angle=45,vjust=
> #   theme(legend.direction="vertical",legend.position = c(.73,.88)) +
> #   jp2theme
> #
> # tiff("figures/schema_trtype.tiff", width= 2408, height= 1617, units="px", r
> # plot(schematrtype)
> # dev.off()
```

Fig 2. Presence of schemas in open and closed thought records. Percentage of utterances that reflect a certain schema (score > 0) in open and closed thought records respectively.

On the level of the individual, a series of linear regression models tested whether the active schemas in closed thought records could predict the active schemas in the open thought record of the same scenario type (interpersonal or achievement-related). The outcome variable was the maximum schema score of the open thought record, while the predictor variable was the average of the maximum schema score of the two closed thought records of the same scenario type. Table 5 presents the results of the models.

For the *Competence* schema, 43 % of the variance in the open thought record could be predicted from the closed thought records of the same scenario type, while for the *Attachment* schema this was the case for 20 % of the variance.

```
> #fit 9 disjoint linear regression models, one for each schema
> #Attachment
> lmAttach <- lm(Attach.y ~ Attach.x, data = df.H3lms)
> summary(lmAttach)

Call:
lm(formula = Attach.y ~ Attach.x, data = df.H3lms)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2673 -0.7332 -0.2673  0.9998  2.3351

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.66494    0.10690   6.221 1.57e-09 ***
Attach.x     0.53412    0.05888   9.071  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.23 on 316 degrees of freedom
Multiple R-squared:  0.2066,        Adjusted R-squared:  0.2041
F-statistic: 82.28 on 1 and 316 DF,  p-value: < 2.2e-16

> confint(lmAttach)

                2.5 %    97.5 %
(Intercept) 0.4546287 0.8752604
Attach.x    0.4182692 0.6499740

> #Competence
> lmComp <- lm(Comp.y ~ Comp.x, data = df.H3lms)
> summary(lmComp)

Call:
lm(formula = Comp.y ~ Comp.x, data = df.H3lms)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5857 -0.2278 -0.2278  0.4143  2.7722

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.22778    0.06836   3.332 0.000965 ***
Comp.x       0.78597    0.05045  15.580  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9744 on 316 degrees of freedom
Multiple R-squared:  0.4344,        Adjusted R-squared:  0.4327
F-statistic: 242.7 on 1 and 316 DF,  p-value: < 2.2e-16
```

```
> confint(lmComp)

                2.5 %    97.5 %
(Intercept) 0.09327642 0.3622875
Comp.x      0.68671274 0.8852204

> #Global Self-Evaluation
> lmGlobal <- lm(Global.y ~ Global.x, data = df.H3lms)
> summary(lmGlobal)

Call:
lm(formula = Global.y ~ Global.x, data = df.H3lms)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5792 -1.0727 -0.6441  1.3559  2.3559

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.64411    0.10705   6.017 4.92e-09 ***
Global.x     0.31170    0.06931   4.497 9.67e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.258 on 316 degrees of freedom
Multiple R-squared:  0.06016,        Adjusted R-squared:  0.05718
F-statistic: 20.23 on 1 and 316 DF,  p-value: 9.671e-06

> confint(lmGlobal)

                2.5 %    97.5 %
(Intercept) 0.4334844 0.8547409
Global.x    0.1753393 0.4480554

> #Health
> lmHealth <- lm(Health.y ~ Health.x, data = df.H3lms)
> summary(lmHealth)

Call:
lm(formula = Health.y ~ Health.x, data = df.H3lms)

Residuals:
    Min      1Q  Median      3Q     Max
-0.8134 -0.1843 -0.1843 -0.1843  2.8157

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.18426    0.04606   4.000 7.89e-05 ***
Health.x     0.20972    0.07083   2.961   0.0033 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7524 on 316 degrees of freedom
Multiple R-squared:  0.02699,        Adjusted R-squared:  0.02392
F-statistic: 8.767 on 1 and 316 DF,  p-value: 0.0033
```

```
> confint(lmHealth)

                2.5 %    97.5 %
(Intercept) 0.09362769 0.2748858
Health.x    0.07036106 0.3490693

> #Control
> lmControl <- lm(Control.y ~ Control.x, data = df.H3lms)
> summary(lmControl)

Call:
lm(formula = Control.y ~ Control.x, data = df.H3lms)

Residuals:
    Min      1Q  Median      3Q     Max
-0.8420 -0.8420 -0.7364  0.9397  2.2847

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.84201    0.07311  11.517   <2e-16 ***
Control.x   -0.08450    0.11558  -0.731    0.465
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.067 on 316 degrees of freedom
Multiple R-squared:  0.001689,        Adjusted R-squared:  -0.001471
F-statistic: 0.5345 on 1 and 316 DF,  p-value: 0.4653

> confint(lmControl)

                2.5 %    97.5 %
(Intercept)  0.6981617 0.9858596
Control.x   -0.3118974 0.1429036

> #Meta-Cognition
> lmMetaCog <- lm(MetaCog.y ~ MetaCog.x, data = df.H3lms)
> summary(lmMetaCog)

Call:
lm(formula = MetaCog.y ~ MetaCog.x, data = df.H3lms)

Residuals:
    Min      1Q  Median      3Q     Max
-0.4885 -0.1568 -0.1568 -0.1568  2.8432

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.15679    0.03381   4.638 5.16e-06 ***
MetaCog.x    0.16587    0.10339   1.604     0.11
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5754 on 316 degrees of freedom
Multiple R-squared:  0.00808,        Adjusted R-squared:  0.004941
F-statistic: 2.574 on 1 and 316 DF,  p-value: 0.1096
```

```
> confint(lmMetaCog)

                2.5 %    97.5 %
(Intercept)  0.09027272 0.2232995
MetaCog.x   -0.03754363 0.3692866

> #Others
> lmOthers <- lm(Others.y ~ Others.x, data = df.H3lms)
> summary(lmOthers)

Call:
lm(formula = Others.y ~ Others.x, data = df.H3lms)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3508 -0.1063 -0.1063 -0.1063  2.8937

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.10630    0.02839   3.745 0.000215 ***
Others.x     0.12226    0.07930   1.542 0.124138
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4696 on 316 degrees of freedom
Multiple R-squared:  0.007466,        Adjusted R-squared:  0.004325
F-statistic: 2.377 on 1 and 316 DF,  p-value: 0.1241

> confint(lmOthers)

                2.5 %    97.5 %
(Intercept)  0.05044830 0.1621545
Others.x    -0.03376376 0.2782884

> #Hopelessness
> lmHopeless <- lm(Hopeless.y ~ Hopeless.x, data = df.H3lms)
> summary(lmHopeless)

Call:
lm(formula = Hopeless.y ~ Hopeless.x, data = df.H3lms)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0739 -0.3285 -0.3285  0.3734  2.6715

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.32845    0.05296   6.202 1.75e-09 ***
Hopeless.x   0.29818    0.07129   4.183 3.74e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7722 on 316 degrees of freedom
Multiple R-squared:  0.05246,        Adjusted R-squared:  0.04946
F-statistic: 17.49 on 1 and 316 DF,  p-value: 3.738e-05
```

```
> confint(lmHopeless)

               2.5 %    97.5 %
(Intercept) 0.2242485 0.4326539
Hopeless.x  0.1579138 0.4384467

> #Other people's views on self
> lmOthViews <- lm(OthViews.y ~ OthViews.x, data = df.H3lms)
> summary(lmOthViews)

Call:
lm(formula = OthViews.y ~ OthViews.x, data = df.H3lms)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3270 -0.8377 -0.5931  1.0400  2.4069

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.59308    0.09446   6.279 1.13e-09 ***
OthViews.x   0.24464    0.06488   3.771 0.000194 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.123 on 316 degrees of freedom
Multiple R-squared:  0.04305,        Adjusted R-squared:  0.04002
F-statistic: 14.22 on 1 and 316 DF,  p-value: 0.0001944

> confint(lmOthViews)

               2.5 %    97.5 %
(Intercept) 0.4072297 0.7789266
OthViews.x  0.1169846 0.3722974
```

Table 5. Outcomes of the per-schema linear regression models to test whether participants show similar schema patterns in open as in closed thought records of the same scenario type (interpersonal vs. achievement-related).

Schema	b	95 % CI	t	p	F(1,316)	Adj. R^2
Attachment	0.53	[0.42,0.65]	9.07	< 0.001	82.28	0.20
Competence	0.79	[0.69,0.86]	15.58	< 0.001	242.7	0.43
Global self-evaluation	0.31	[0.18,0.45]	4.50	< 0.001	20.23	0.06
Health	0.21	[0.07,0.35]	2.96	< 0.01	8.77	0.02
Power and Control	-0.08	[-0.31,0.14]	-0.73	0.47	0.53	0.00
Meta-cognition	0.17	[-0.04,0.37]	1.60	0.11	2.57	0.00
Other people	0.12	[-0.03,0.28]	1.54	0.12	2.38	0.00
Hopelessness	0.30	[0.16,0.44]	4.18	< 0.001	17.49	0.05
Other's views on self	0.24	[0.12,0.37]	3.77	< 0.001	14.22	0.04

H4: Mental illnesses have associated schemas

Five linear regression models tested whether there is a link between the active schemas of participants as indicated in thought records and the outcomes on five mental health inventories. The Bonferroni-corrected α of 0.01 serves as the significance threshold.

```
> mhealth <- c('HDAS_D','HDAS_A','BDI_IA','CB_Rel','CB_Ach')
> df.H4[,c(mhealth,schemas)] <- apply(df.H4[,c(mhealth,schemas)],2,
+                                     function(x) as.numeric(as.character(x)))
> #HDAS Depression
> H4.Model.HDAS.Dep <-lm(HDAS_D ~ Attach +
+                          Comp +
+                          Global +
+                          Health +
+                          Control +
+                          MetaCog +
+                          Others +
+                          Hopeless +
+                          OthViews,
+               data=df.H4)
> summary(H4.Model.HDAS.Dep)

Call:
lm(formula = HDAS_D ~ Attach + Comp + Global + Health + Control +
    MetaCog + Others + Hopeless + OthViews, data = df.H4)

Residuals:
    Min      1Q  Median      3Q     Max
-7.4205 -3.9328 -0.8524  3.6172 14.1162

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.57262    0.62202   7.351 1.76e-12 ***
Attach       0.05783    0.23980   0.241   0.8096
Comp        -0.24421    0.27058  -0.903   0.3675
Global       0.31389    0.20107   1.561   0.1195
Health       0.69092    0.43394   1.592   0.1124
Control      0.92204    0.49647   1.857   0.0642 .
MetaCog      0.89611    1.16946   0.766   0.4441
Others      -0.23948    1.14383  -0.209   0.8343
Hopeless    -0.06409    0.61146  -0.105   0.9166
OthViews    -0.24851    0.29772  -0.835   0.4045
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.541 on 310 degrees of freedom
Multiple R-squared:  0.04265,        Adjusted R-squared:  0.01486
F-statistic: 1.535 on 9 and 310 DF,  p-value: 0.1348

> H4.Model.HDAS.Dep_beta <- lm.beta(H4.Model.HDAS.Dep)
> summary(H4.Model.HDAS.Dep_beta)

Call:
lm(formula = HDAS_D ~ Attach + Comp + Global + Health + Control +
    MetaCog + Others + Hopeless + OthViews, data = df.H4)

Residuals:
    Min      1Q  Median      3Q     Max
-7.4205 -3.9328 -0.8524  3.6172 14.1162
```

```
Coefficients:
             Estimate Standardized Std. Error t value Pr(>|t|)
(Intercept)  4.572623     0.000000   0.622019   7.351 1.76e-12 ***
Attach       0.057826     0.014626   0.239799   0.241   0.8096
Comp        -0.244213    -0.052101   0.270580  -0.903   0.3675
Global       0.313891     0.093427   0.201073   1.561   0.1195
Health       0.690916     0.095186   0.433942   1.592   0.1124
Control      0.922045     0.106067   0.496470   1.857   0.0642 .
MetaCog      0.896110     0.043995   1.169463   0.766   0.4441
Others      -0.239481    -0.011758   1.143827  -0.209   0.8343
Hopeless    -0.064090    -0.006219   0.611465  -0.105   0.9166
OthViews    -0.248508    -0.048460   0.297716  -0.835   0.4045
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.541 on 310 degrees of freedom
Multiple R-squared:  0.04265,        Adjusted R-squared:  0.01486
F-statistic: 1.535 on 9 and 310 DF,  p-value: 0.1348

> #HDAS Anxiety
> H4.Model.HDAS.Anx <-lm(HDAS_A ~ Attach +
+                          Comp +
+                          Global +
+                          Health +
+                          Control +
+                          MetaCog +
+                          Others +
+                          Hopeless +
+                          OthViews,
+                data=df.H4)
> summary(H4.Model.HDAS.Anx)

Call:
lm(formula = HDAS_A ~ Attach + Comp + Global + Health + Control +
    MetaCog + Others + Hopeless + OthViews, data = df.H4)

Residuals:
    Min      1Q  Median      3Q     Max
-8.4978 -3.6017 -0.8585  3.5499 12.5406

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.57121    0.65069   8.562 5.25e-16 ***
Attach      -0.12762    0.25085  -0.509  0.61129
Comp        -0.24086    0.28305  -0.851  0.39546
Global       0.62629    0.21034   2.978  0.00313 **
Health       0.57004    0.45394   1.256  0.21015
Control      0.93676    0.51935   1.804  0.07225 .
MetaCog      1.36370    1.22336   1.115  0.26584
Others      -0.79979    1.19654  -0.668  0.50437
Hopeless     0.15473    0.63964   0.242  0.80901
OthViews     0.05037    0.31144   0.162  0.87163
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.751 on 310 degrees of freedom
Multiple R-squared:  0.06132,        Adjusted R-squared:  0.03407
F-statistic:  2.25 on 9 and 310 DF,  p-value: 0.01894
```

```
> H4.Model.HDAS.Anx_beta <- lm.beta(H4.Model.HDAS.Anx)
> summary(H4.Model.HDAS.Anx_beta)
```

```
Call:
lm(formula = HDAS_A ~ Attach + Comp + Global + Health + Control +
    MetaCog + Others + Hopeless + OthViews, data = df.H4)

Residuals:
    Min      1Q  Median      3Q     Max
-8.4978 -3.6017 -0.8585  3.5499 12.5406

Coefficients:
            Estimate Standardized Std. Error t value Pr(>|t|)
(Intercept)  5.571209     0.000000   0.650685   8.562 5.25e-16 ***
Attach      -0.127621    -0.030555   0.250851  -0.509  0.61129
Comp        -0.240858    -0.048640   0.283050  -0.851  0.39546
Global       0.626293     0.176453   0.210340   2.978  0.00313 **
Health       0.570041     0.074338   0.453940   1.256  0.21015
Control      0.936759     0.102003   0.519351   1.804  0.07225 .
MetaCog      1.363699     0.063375   1.223359   1.115  0.26584
Others      -0.799786    -0.037168   1.196542  -0.668  0.50437
Hopeless     0.154734     0.014213   0.639645   0.242  0.80901
OthViews     0.050368     0.009297   0.311437   0.162  0.87163
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.751 on 310 degrees of freedom
Multiple R-squared:  0.06132,        Adjusted R-squared:  0.03407
F-statistic:  2.25 on 9 and 310 DF,  p-value: 0.01894
```

```
> #BDI
> H4.Model.BDI <-lm(BDI_IA ~ Attach +
+                     Comp +
+                     Global +
+                     Health +
+                     Control +
+                     MetaCog +
+                     Others +
+                     Hopeless +
+                     OthViews,
+                data=df.H4)
> anova(H4.Model.BDI)
```

```
Analysis of Variance Table

Response: BDI_IA
           Df Sum Sq Mean Sq F value   Pr(>F)
```

```
Attach      1    202  202.32  1.5632 0.212142
Comp        1    264  263.67  2.0372 0.154499
Global      1    943  942.53  7.2824 0.007345 **
Health      1    562  561.68  4.3398 0.038051 *
Control     1    247  247.17  1.9098 0.167984
MetaCog     1     26   25.75  0.1990 0.655848
Others      1      9    8.65  0.0669 0.796127
Hopeless    1     35   34.64  0.2677 0.605264
OthViews    1    431  430.53  3.3264 0.069137 .
Residuals 310  40122  129.43
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> summary(H4.Model.BDI)

Call:
lm(formula = BDI_IA ~ Attach + Comp + Global + Health + Control +
    MetaCog + Others + Hopeless + OthViews, data = df.H4)

Residuals:
    Min      1Q  Median      3Q     Max
-17.540  -8.483  -3.784   6.333  44.494

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.3400     1.5582   5.994 5.69e-09 ***
Attach        0.2915     0.6007   0.485   0.6279
Comp         -0.8451     0.6778  -1.247   0.2135
Global        1.2533     0.5037   2.488   0.0134 *
Health        1.9039     1.0871   1.751   0.0809 .
Control       1.3872     1.2437   1.115   0.2656
MetaCog       1.8036     2.9297   0.616   0.5386
Others        0.5562     2.8654   0.194   0.8462
Hopeless      0.6809     1.5318   0.444   0.6570
OthViews     -1.3603     0.7458  -1.824   0.0691 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.38 on 310 degrees of freedom
Multiple R-squared:  0.06342,        Adjusted R-squared:  0.03623
F-statistic: 2.332 on 9 and 310 DF,  p-value: 0.01486
```

```
> H4.Model.BDI_beta <- lm.beta(H4.Model.BDI)
> summary(H4.Model.BDI_beta)

Call:
lm(formula = BDI_IA ~ Attach + Comp + Global + Health + Control +
    MetaCog + Others + Hopeless + OthViews, data = df.H4)

Residuals:
    Min      1Q  Median      3Q     Max
-17.540  -8.483  -3.784   6.333  44.494

Coefficients:
```

```
            Estimate Standardized Std. Error t value Pr(>|t|)
(Intercept)  9.33995      0.00000    1.55824   5.994 5.69e-09 ***
Attach       0.29148      0.02911    0.60073   0.485   0.6279
Comp        -0.84505     -0.07118    0.67784  -1.247   0.2135
Global       1.25327      0.14728    0.50371   2.488   0.0134 *
Health       1.90392      0.10356    1.08708   1.751   0.0809 .
Control      1.38723      0.06301    1.24372   1.115   0.2656
MetaCog      1.80361      0.03496    2.92965   0.616   0.5386
Others       0.55618      0.01078    2.86543   0.194   0.8462
Hopeless     0.68086      0.02609    1.53180   0.444   0.6570
OthViews    -1.36026     -0.10473    0.74582  -1.824   0.0691 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.38 on 310 degrees of freedom
Multiple R-squared:  0.06342,        Adjusted R-squared:  0.03623
F-statistic: 2.332 on 9 and 310 DF,  p-value: 0.01486

> #CogDistortions Relatedness
> H4.Model.CB_Rel <-lm(CB_Rel ~ Attach +
+                        Comp +
+                        Global +
+                        Health +
+                        Control +
+                        MetaCog +
+                        Others +
+                        Hopeless +
+                        OthViews,
+               data=df.H4)
> summary(H4.Model.CB_Rel)

Call:
lm(formula = CB_Rel ~ Attach + Comp + Global + Health + Control +
    MetaCog + Others + Hopeless + OthViews, data = df.H4)

Residuals:
     Min       1Q   Median       3Q      Max
-31.5276  -8.3451   0.3046   8.0378  27.8474

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.0004     1.6010  20.613  < 2e-16 ***
Attach       -0.1346     0.6172  -0.218  0.82750
Comp          0.1140     0.6964   0.164  0.87007
Global        2.1093     0.5175   4.076 5.83e-05 ***
Health        0.9961     1.1169   0.892  0.37317
Control       3.4422     1.2778   2.694  0.00745 **
MetaCog      -2.2170     3.0100  -0.737  0.46194
Others       -2.2355     2.9440  -0.759  0.44822
Hopeless      1.3680     1.5738   0.869  0.38541
OthViews      0.5435     0.7663   0.709  0.47868
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 11.69 on 310 degrees of freedom
Multiple R-squared:  0.1024,        Adjusted R-squared:  0.07633
F-statistic: 3.929 on 9 and 310 DF,  p-value: 9.45e-05
```

```
> H4.Model.CB_Rel_beta <- lm.beta(H4.Model.CB_Rel)
> summary(H4.Model.CB_Rel_beta)

Call:
lm(formula = CB_Rel ~ Attach + Comp + Global + Health + Control +
    MetaCog + Others + Hopeless + OthViews, data = df.H4)

Residuals:
     Min       1Q   Median       3Q      Max
-31.5276  -8.3451   0.3046   8.0378  27.8474

Coefficients:
            Estimate Standardized Std. Error t value Pr(>|t|)
(Intercept) 33.000427     0.000000   1.600953  20.613  < 2e-16 ***
Attach      -0.134605    -0.012808   0.617196  -0.218  0.82750
Comp         0.114009     0.009151   0.696418   0.164  0.87007
Global       2.109308     0.236194   0.517521   4.076 5.83e-05 ***
Health       0.996085     0.051627   1.116879   0.892  0.37317
Control      3.442191     0.148970   1.277815   2.694  0.00745 **
MetaCog     -2.217038    -0.040950   3.009964  -0.737  0.46194
Others      -2.235524    -0.041291   2.943982  -0.759  0.44822
Hopeless     1.367950     0.049941   1.573789   0.869  0.38541
OthViews     0.543496     0.039873   0.766261   0.709  0.47868
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.69 on 310 degrees of freedom
Multiple R-squared:  0.1024,        Adjusted R-squared:  0.07633
F-statistic: 3.929 on 9 and 310 DF,  p-value: 9.45e-05
```

```
> #CogDistortions Achievement
> H4.Model.CB_Ach <-lm(CB_Ach ~ Attach +
+                        Comp +
+                        Global +
+                        Health +
+                        Control +
+                        MetaCog +
+                        Others +
+                        Hopeless +
+                        OthViews,
+                data=df.H4)
> summary(H4.Model.CB_Ach)

Call:
lm(formula = CB_Ach ~ Attach + Comp + Global + Health + Control +
    MetaCog + Others + Hopeless + OthViews, data = df.H4)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-26.567  -8.448  -0.582   8.428  37.520

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 33.419323   1.750507  19.091  < 2e-16 ***
Attach      -0.392910   0.674851  -0.582 0.560843
Comp         0.816046   0.761474   1.072 0.284704
Global       2.058287   0.565866   3.637 0.000322 ***
Health       1.521012   1.221214   1.245 0.213891
Control      2.312638   1.397184   1.655 0.098893 .
MetaCog     -0.653191   3.291142  -0.198 0.842808
Others      -2.961160   3.218997  -0.920 0.358339
Hopeless     1.878344   1.720805   1.092 0.275879
OthViews    -0.007197   0.837842  -0.009 0.993152
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.78 on 310 degrees of freedom
Multiple R-squared:  0.08235,        Adjusted R-squared:  0.05571
F-statistic: 3.091 on 9 and 310 DF,  p-value: 0.001432

> H4.Model.CB_Ach_beta <- lm.beta(H4.Model.CB_Ach)
> summary(H4.Model.CB_Ach_beta)

Call:
lm(formula = CB_Ach ~ Attach + Comp + Global + Health + Control +
    MetaCog + Others + Hopeless + OthViews, data = df.H4)

Residuals:
    Min      1Q  Median      3Q     Max
-26.567  -8.448  -0.582   8.428  37.520

Coefficients:
             Estimate Standardized Std. Error t value Pr(>|t|)
(Intercept) 33.4193227    0.0000000  1.7505067  19.091  < 2e-16 ***
Attach      -0.3929105   -0.0345730  0.6748515  -0.582 0.560843
Comp         0.8160462    0.0605666  0.7614741   1.072 0.284704
Global       2.0582873    0.2131304  0.5658659   3.637 0.000322 ***
Health       1.5210120    0.0728994  1.2212135   1.245 0.213891
Control      2.3126382    0.0925512  1.3971835   1.655 0.098893 .
MetaCog     -0.6531911   -0.0111565  3.2911418  -0.198 0.842808
Others      -2.9611602   -0.0505766  3.2189966  -0.920 0.358339
Hopeless     1.8783441    0.0634125  1.7208051   1.092 0.275879
OthViews    -0.0071970   -0.0004883  0.8378423  -0.009 0.993152
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.78 on 310 degrees of freedom
Multiple R-squared:  0.08235,        Adjusted R-squared:  0.05571
F-statistic: 3.091 on 9 and 310 DF,  p-value: 0.001432
```

For both symptom-based mental health inventories for depression, i.e., the HDAS–Depression and the BDI-IA, none of the schemas was a significant predictor of

the outcome scores. However, for the anxiety inventory (HDAS–Anxiety) and the two Cognitive Distortion scales, we found that the *Global Self-Evaluation* schema was linked to these measures: all other schemas being equal, any additional thought record with a clearly present *Global Self-Evaluation* schema resulted in a 0.63 ($\beta = 0.18$) point increase on the HDAS–Anxiety ($t = 2.98, p = 0.003$), a 2.11 ($\beta = 0.24$) point increase on the Cognitive Distortions – Relatedness measure ($t = 3.83, p < 0.001$), and a 2.06 ($\beta = 0.21$) point increase on the Cognitive Distortions – Achievement measure ($t = 3.64, p < 0.001$). Finally, the number of thought records with a clearly present *Power and Control* schema also significantly predicted the Cognitive Distortions – Relatedness measure ($b = 3.44, \beta = 0.15, t = 2.69, p = 0.007$).

Discussion and Conclusion

As the first and core hypothesis, we posited that utterances of thought records could be automatically scored with respect to their underlying schemas. With all three machine learning algorithm types (kNN, SVM, and RNN) that were tried, we found affirmative evidence for this. Even when only representing utterances as averages of word vectors, linguistic patterns could be learned (as in the case of the kNN and SVM models). Yet, the fact that the per-schema RNNs outperformed the other classification algorithms on several schemas provides an indication that the information contained in the word order may be useful for optimal scoring performance. Looking at the best outcomes for each schema, correlations between predicted scores and actual scores ranged from $\rho = 0.11$ to $\rho = 0.81$. The schemas for which the algorithms saw many training examples with non-zero scores (*Attachment* and *Competence*) could be classified well by all. However, the *Health* schema also exhibits good classification potential. This is probably due to very distinctive language as a result of one specific scenario related to dieting and weight loss, i.e., many utterances scored on the *Health* schema contained words such as "fat," "gain," "overweight," "diet," or "skinny." These words are likely to be within close proximity of each other in the word vector space, possibly leading to similar utterance representations and hence a clear linguistic pattern. Although the outcomes from the models cannot be compared directly to the interrater (weighted Cohen's $\kappa = 0.79$) and intrarater (weighted Cohen's $\kappa = 0.83$) reliability scores we obtained on a sample of the data, the latter give a good indication that close-to perfect automatic scoring is not obtainable. As our goal was only to see whether scoring was feasible and not to obtain the best possible performance, we did not explore many of the other available options for data representation, data augmentation, or modeling. These include looking into more state-of-the-art ways of representing utterances, such as BERT [50] or GPT-3 [51], making better use of the ordering information in the scores, creating a corpus-specific word vector space, or trying to generate more training examples with neural networks. Together with this article, however, we make our collected dataset publicly available and invite other researchers or machine learning enthusiasts to improve upon our results.

As our second hypothesis, we predicted an algorithm trained on utterances of varying downward arrow technique (DAT) depths to be able to better score the utterances as the depth increases. This is because the DAT was specifically developed to aid patients in identifying their maladaptive schemas, taking the automatic thought from the completed thought record as starting point. After applying the technique, a schema formulation should be reached. In our dataset and with the best performing algorithm of Hypothesis 1, we did not find support for Hypothesis 2. This may be due to only very few participants completing more than three steps. Since our participants were drawn from a non-clinical population and had never practiced thought recording before, it is possible that they did not reach the same level of introspection as a clinical, therapist-guided group would. Further research might therefore compare our results to

those obtained in a clinical setting.

As our third hypothesis, we expected that the dysfunctional schemas that were active when completing scenario-based (closed) thought records would be able to predict those active when completing a real-life personal (open) thought record within participants, given that the closed and open thought records matched in scenario type, that is, both revolved around either an interpersonal or an achievement-related situation. In our study, we relied mostly on prescripted scenarios and asked participants to respond to these as if they were real. We found support for our third hypothesis. For two schemas, we even observed that 20 % (Attachment schema) and 43 % (Competence schema) of the variance in the open thought record score could be predicted by the scores in the closed thought records. This corresponds to the central idea of schema theory: if a person holds a certain schema, this may be activated in various situations of a similar kind and influence how the person appraises the situation [53]. Consequently, we regard it as a viable option to use prescripted scenarios instead of real-life ones when needed. However, it can be argued that the *Attachment* schema may be particularly relevant in *interpersonal* scenarios, while the *Competence* schema plays more in *achievement-related* scenarios. The large effect that shows for these two schemas may therefore be the result of labeling the open thought records as belonging to one of these two scenario types and splitting the dataset accordingly. On the basis of these considerations, the scenarios should be carefully chosen and varied enough to be able to unveil all possible schemas when substituting closed scenarios for open ones. Therefore, a larger number of thought records may be needed than when using open thought records.

Lastly, as our fourth hypothesis, we proposed that the schema patterns across all thought records of a person can predict outcomes on depression, anxiety, and cognitive distortion scales. We found partial support for this hypothesis. Concerning the link between schemas and mental health outcomes, we found no relationship between the schemas and outcomes on both depression inventories. While Millings and Carnelley [7] observed a higher prevalence of the *Power and Control* schema in people with anxious tendencies, we observed higher scores on the HDAS – Anxiety scale when participants had a negative *Global Self-Evaluation*. This schema was also a good predictor of cognitive distortions linked to relatedness and achievement. We could not replicate the finding reported in [7] that higher anxiety scores were linked to a less frequently active *Attachment* schema either. This may, however, be a population effect, as we did not work with a clinical population. Yet, an active *Power and Control* schema was related to more cognitive distortions pertaining to relatedness in our dataset. On the whole, we found more links between schemas and cognitive distortions than schemas and mental health inventory outcomes. This may have to do with thought records being a cognitive task concerned with unveiling dysfunctional cognitions, which connects directly to the cognitive distortion measure and less to the symptom-based nature of the mental health inventories. On a more practical note, our results indicate that a software application striving to construct a long-term user model might benefit from assigning a higher a priori probability to the activation of the *Global Self-Evaluation* schema after an initial assessment of the user's anxiety levels and cognitive distortions. Still, making a choice on this requires trading off the collection of such sensitive mental health scale data against the added benefit of improving the prediction model.

The core finding of this research is that it is possible to interpret rich natural language data from the psychotherapy domain using a computer algorithm. The applicability of this finding extends especially to various kinds of psychological assessment. For example, one of the common applications of e-mental health in research are ecological momentary assessments. To date, these typically employ multiple choice response items for self-report measures, which may be combined with sensor readings from handheld devices or wearables (compare [44] for depression). Our findings are

promising for effectively using more open response formats and journaling, thus allowing participants to better describe their thoughts, feelings, and behaviors in their own words while minimizing analysis effort. This is also interesting in light of new methodological developments in mental health assessment as a result of big data, such as studying symptom dynamics of individuals with network analyses [49]. Such dynamic networks of symptoms may be augmented with the schemas as determined from thought records to better understand how the activation and co-activation of schemas and other symptoms predicts mental well-being over time. Another possible area of application are cognitive case conceptualizations [52]. These are comprehensive outlines of the patient's problems as first drafted during the intake conversation between patient and therapist. They are continually refined throughout therapy, often on the basis of homework assignments [45]. With the possibility of automatically interpreting thought record data, it may be possible to sketch a first CCC before therapy by collecting and analyzing thought records over the period of time the patient spends on a waiting list and to then collaboratively update this CCC with the therapist as new thought records are completed during therapy. Moreover, Schema Therapy [53] presents a further thought classification system to that of schemas, namely that of schema modes. Furthermore, it proposes a much larger set of schemas than the ones used in this research. With a background in Schema Therapy, it may be possible to use our collected dataset and re-label the data with respect to these other schemas or the schema modes. Beyond psychological assessment, Millings and Carnelley [7] propose future work to compare the derivation of schemas using the downward arrow technique in an online setting to a face-to-face therapy setting. We would be interested in adding the algorithmically derived schemas to this comparison in a long-term study.

In conclusion, we have presented an algorithmic benchmark solution for automatically scoring utterances extracted from thought records with respect to the underlying schema. We expect the model and the opportunities resulting from the positive results to be of relevance for the field of clinical psychology. For the field of computer science, we make the dataset of collected thought records publicly available. Especially the complexity of the outcome variables (ordinal multi-label) may be intriguing for those looking to develop new algorithms or test existing ones. Lastly, for both fields, clinical psychology and computer science, the dataset could be used to study and advance automatically generated explanations of the algorithmic schema identification. In so doing, it can contribute to diagnoses and explainable artificial intelligence (XAI) technology, which is seen as an important requirement for responsible and effective AI-implementation (e.g., [54]).

Supporting information

S1 Appendix. Experimental flow. Figure displays the different stages of the experiment as traversed by the participants.

S2 Appendix. Computation of predictor and outcome variables for H3. Graphical illustration of how we determined the predictor and outcome variables for the nine models of hypothesis 3.

Acknowledgments

This work has been partially supported by the 4TU research center Humans & Technology (H&T) project (Systems for Smart Social Spaces for Living Well: S4).

Additionally, we would like to acknowledge the help that we received from the two coders who double coded parts of the dataset.

References

1. Burger F, Neerincx MA, Brinkman WP. Technological state of the art of electronic mental health interventions for major depressive disorder: systematic literature review. Journal of medical Internet research. 2020;22(1):e12599.
2. Beck AT. Cognitive therapy: A 30-year retrospective. American psychologist. 1991;46(4):368.
3. Beck AT. Thinking and depression: II. Theory and therapy. Archives of general psychiatry. 1964;10(6):561–571.
4. Burns DD. The feeling good handbook, Rev. Plume/Penguin Books; 1999.
5. Osmo F, Duran V, Wenzel A, de Oliveira IR, Nepomuceno S, Madeira M, et al. The Negative Core Beliefs Inventory: Development and Psychometric Properties. Journal of Cognitive Psychotherapy. 2018;32(1):67–84.
6. Wong QJ, Gregory B, Gaston JE, Rapee RM, Wilson JK, Abbott MJ. Development and validation of the Core Beliefs Questionnaire in a sample of individuals with social anxiety disorder. Journal of Affective Disorders. 2017;207:121–127.
7. Millings A, Carnelley KB. Core belief content examined in a large sample of patients using online cognitive behaviour therapy. Journal of Affective Disorders. 2015;186:275–283.
8. Jurafsky D, Martin JH. Speech and language processing. vol. 3. Pearson London; 2014.
9. Laranjo L, Dunn AG, Tong HL, Kocaballi AB, Chen J, Bashir R, et al. Conversational agents in healthcare: a systematic review. Journal of the American Medical Informatics Association. 2018;25(9):1248–1258.
10. Montenegro JLZ, da Costa CA, da Rosa Righi R. Survey of Conversational Agents in Health. Expert Systems with Applications. 2019;.
11. McTear MF. Spoken dialogue technology: enabling the conversational user interface. ACM Computing Surveys (CSUR). 2002;34(1):90–169.
12. Ni L, Lu C, Liu N, Liu J. Mandy: Towards a smart primary care chatbot application. In: International Symposium on Knowledge and Systems Sciences. Springer; 2017. p. 38–52.
13. Amato F, Marrone S, Moscato V, Piantadosi G, Picariello A, Sansone C. Chatbots Meet eHealth: Automatizing Healthcare. In: WAIAH@ AI* IA; 2017. p. 40–49.
14. van Heerden A, Ntinga X, Vilakazi K. The potential of conversational agents to provide a rapid HIV counseling and testing services. In: 2017 International Conference on the Frontiers and Advances in Data Science (FADS). IEEE; 2017. p. 80–85.

15. Jin L, White M, Jaffe E, Zimmerman L, Danforth D. Combining cnns and pattern matching for question interpretation in a virtual patient dialogue system. In: Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications; 2017. p. 11–21.

16. Rizzo A, Kenny P, Parsons TD. Intelligent virtual patients for training clinical skills. JVRB-Journal of Virtual Reality and Broadcasting. 2011;8(3).

17. Ochs M, De Montcheuil G, Pergandi JM, Saubesty J, Pelachaud C, Mestre D, et al. An architecture of virtual patient simulation platform to train doctors to break bad news. In: Conference on Computer Animation and Social Agents (CASA); 2017.

18. Miner AS, Milstein A, Schueller S, Hegde R, Mangurian C, Linos E. Smartphone-based conversational agents and responses to questions about mental health, interpersonal violence, and physical health. JAMA internal medicine. 2016;176(5):619–625.

19. Abd-alrazaq AA, Alajlani M, Alalwan AA, Bewick BM, Gardner P, Househ M. An overview of the features of chatbots in mental health: A scoping review. International Journal of Medical Informatics. 2019;132:103978.

20. Bresó A, Martínez-Miranda J, Botella C, Baños RM, García-Gómez JM. Usability and acceptability assessment of an empathic virtual agent to prevent major depression. Expert Systems. 2016;33(4):297–312.

21. Shamekhi A, Bickmore T, Lestoquoy A, Negash L, Gardiner P. Blissful agents: adjuncts to group medical visits for chronic pain and depression. In: International Conference on Intelligent Virtual Agents. Springer; 2016. p. 433–437.

22. Fitzpatrick KK, Darcy A, Vierhile M. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (Woebot): a randomized controlled trial. JMIR mental health. 2017;4(2):e19.

23. Guntuku SC, Yaden DB, Kern ML, Ungar LH, Eichstaedt JC. Detecting depression and mental illness on social media: an integrative review. Current Opinion in Behavioral Sciences. 2017;18:43–49.

24. Al-Mosaiwi M, Johnstone T. In an absolute state: Elevated use of absolutist words is a marker specific to anxiety, depression, and suicidal ideation. Clinical Psychological Science. 2018;6(4):529–542.

25. Holtzman NS, et al. A meta-analysis of correlations between depression and first person singular pronoun use. Journal of Research in Personality. 2017;68:63–68.

26. Newell EE, McCoy SK, Newman ML, Wellman JD, Gardner SK. You Sound So Down: Capturing Depressed Affect Through Depressed Language. Journal of Language and Social Psychology. 2018;37(4):451–474.

27. Kshirsagar R, Morris R, Bowman S. Detecting and explaining crisis. arXiv preprint arXiv:170509585. 2017;.

28. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:14090473. 2014;.

29. Gkotsis G, Oellrich A, Velupillai S, Liakata M, Hubbard TJ, Dobson RJ, et al. Characterisation of mental health conditions in social media using Informed Deep Learning. Scientific reports. 2017;7:45141.

30. Benton A, Mitchell M, Hovy D. Multi-task learning for mental health using social media text. arXiv preprint arXiv:171203538. 2017;.

31. Dobson D, Dobson KS. Evidence-based practice of cognitive-behavioral therapy. Guilford Publications; 2018.

32. Barlow DH. Clinical handbook of psychological disorders: A step-by-step treatment manual. Guilford publications; 2014.

33. Cook MN. Transforming Teen Behavior: Parent Teen Protocols for Psychosocial Skills Training. Academic Press; 2015.

34. Schoth DE, Liossi C. A systematic review of experimental paradigms for exploring biased interpretation of ambiguous information with emotional and neutral associations. Frontiers in psychology. 2017;8:171.

35. Lefebvre MF. Cognitive distortion and cognitive errors in depressed psychiatric and low back pain patients. Journal of consulting and clinical psychology. 1981;49(4):517.

36. Pössel P. Cognitive Error Questionnaire (CEQ): Psychometric properties and factor structure of the German translation. Journal of Psychopathology and Behavioral Assessment. 2009;31(3):264–269.

37. Barber JP, DeRubeis RJ. The ways of responding: A scale to assess compensatory skills taught in cognitive therapy. Behavioral Assessment. 1992;.

38. Covin R, Dozois DJ, Ogniewicz A, Seeds PM. Measuring cognitive errors: Initial development of the Cognitive Distortions Scale (CDS). International Journal of Cognitive Therapy. 2011;4(3):297–322.

39. Beck AT, Alford BA. Depression: Causes and treatment. University of Pennsylvania Press; 2009.

40. Zigmond AS, Snaith RP. The hospital anxiety and depression scale. Acta psychiatrica scandinavica. 1983;67(6):361–370.

41. Beck AT. Cognitive therapy of depression. Guilford press; 1979.

42. Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–1543.

43. Wolf MJ, Miller KW, Grodzinsky FS. Why we should have seen that coming: comments on microsoft's tay "experiment," and wider implications The ORBIT Journal. 2017;1(2):1–12.

44. Colombo D, Fernández-Álvarez J, Patané A, Semonella M, Kwiatkowska M, García-Palacios A, Cipresso P, Ri.va G, Botella C. Current state and future directions of technology-based ecological momentary assessment and intervention for major depressive disorder: A systematic review Journal of clinical medicine. 2019;8(4):465–491.

45. Cronin TJ, Lawrence KA, Taylor K, Norton PJ, Kazantzis N. Integrating between-session interventions (homework) in therapy: The importance of the therapeutic relationship and cognitive case conceptualization. Journal of clinical psychology. 2015;71(5):439–450.

46. Young JE, Klosko JS, Weishaar ME. Schema therapy: A practitioner's guide, Guilford Press. 2006.

47. Gutiérrez PA, Perez-Ortiz M, Sanchez-Monedero J, Fernandez-Navarro F, Hervas-Martinez C. Ordinal regression methods: survey and experimental study IEEE Transactions on Knowledge and Data Engineering. 2015;28(1):127–146.

48. Cheng J, Wang Z, Pollastri G. A neural network approach to ordinal regression 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). 2008. 1279–1284.

49. Cramer AOJ, van Borkulo CD, Giltay EJ, van der Maas HLJ, Kendler KS, Scheffer M, et al. Major depression as a complex dynamic system PloS one. 2016;11(12):e0167490.

50. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding arXiv preprint arXiv:1810.04805. 20018.

51. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, et al. Language models are few-shot learners. arXiv preprint arXiv:2005.14165. 2020.

52. Needleman LD. Cognitive case conceptualization: A guidebook for practitioners. Routledge. 1999.

53. Young JE, Klosko JS, Weishaar ME. Schema therapy: A practitioner's guide Guilford Press. 2006.

54. Peeters MMM, van Diggelen J, Van Den Bosch K, Bronkhorst A, Neerincx MA, Schraagen JM, et al. Hybrid collective intelligence in a human–AI society AI & SOCIETY. 2020:1–22.