Module Interface Specification for Natural Language Processing for Mental Health Risk Prediction

Team 13, The Cognitive Care Crew
Jessica Dawson
Michael Breau
Matthew Curtis
Benjamin Chinnery
Yaruo Tian

January 17, 2024

1 Revision History

Date	Version	Notes
January 17, 2024	1.0	Revision 0

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at https://github.com/MichaelBreau/nlp-mentalhealth/blob/main/docs/SRS/index.pdf

Contents

3 Introduction

The following document details the Module Interface Specifications for Natural Language Processing for Mental Health Risk Prediction

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at https://github.com/MichaelBreau/nlp-mentalhealth.

4 Notation

The structure of the MIS for modules comes from Hoffman and Strooper (1995), with the addition that template modules have been adapted from Ghezzi et al. (2003). The mathematical notation comes from Chapter 3 of Hoffman and Strooper (1995). For instance, the symbol := is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1|c_2 \Rightarrow r_2|...|c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by Natural Language Processing for Mental Health Risk Prediction.

Data Type	Notation	Description
character	char	a single symbol or digit
list	[item, item,...]	a list of objects of the same type
dict	$< key : T, val : T >$	a dictionary of key value pairs
integer	$\mathbb{Z}$	a number without a fractional component in (-∞, ∞)
natural number	$\mathbb{N}$	a number without a fractional component in [1, ∞)
real	$\mathbb{R}$	any number in (-∞, ∞)

The specification of Natural Language Processing for Mental Health Risk Prediction uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, Natural Language Processing for Mental Health Risk Prediction uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	**Level 2**	**Level 3**
Hardware-Hiding	None	
Behaviour-Hiding	Eating Disorder IO	ED Input ED Output
Software Decision	Depression Module Anorexia Module	
	Eating Disorder Pipeline	ED Pipeline Manager ED Tokenizer ED Representation Model ED Prediction Model ED Transformation 1...n (represents multiple modules)

Table 1: Module Hierarchy

6 MIS for Search for Symptoms of Depression Module

6.1 Module

Task 1 - Search for Symptoms of Depression for User Sentences

6.2 Uses

Used to rank each input sentence by its correlation to a given depression symptom and returns the 1000 sentences with the highest correlation to said depression symptom. Each sentence will receive a level of correlation for each of the 21 given depression symptoms (totaling an output of 21,000 sentences). These results for user sentences will be tested against the results each user gave when filling out the BDI Questionnaire regarding there depression symptoms.

6.3 Syntax

6.3.1 Exported Constants

None

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
path_address	String	String	-
parse_data	String	Dictionary	-
filter_sentences	Dictionary	Dictionary	-
write_top_sentences	List, String	-	-
detect_depression_symptom	String, String	Integer	-
normalize_symptom_scores	Dictionary	List	-

6.4 Semantics

6.4.1 State Variables

None

6.4.2 Environment Variables

None

6.4.3 Assumptions

- Data for each users sentences are in there own .trec file and the jsonl file acts as a master list of sentences containing all sentences across all users in one jsonl file.

- The users answered the BDI questionnaire honestly so we can take that data as being accurate when we use it.

6.4.4 Access Routine Semantics

path_address():

- transition: None
- output: A string representing the address's of all trec files
- exception:

parse_data():

- transition: None
- output: A dictionary containing every sentence where the key is the sentence id and the value is the sentence itself.
- exception: None

filter_sentences():

- transition: None
- output: A dictionary containing all sentences that were not filtered out by the filter function due to being deemed as low quality or irrelevant sentences for testing. The key for each dictionary entry is the sentence id and the value is the sentence itself.
- exception: None

write_top_sentences():

- transition: None
- output: None
- exception: None

detect_depression_symptom():

- transition: None
- output: A integer that represents the correlation of the inputted sentence to the inputted symptom.
- exception:

normalize_symptom_scores():

- transition: None
- output: A list of lists where each inner array represents a sentence. The first element in the inner list is the sentence id and the second is the score for the given depression symptom from 1-10. The outputted list of lists will be sorted by there depression symptom score from highest to lowest.
- exception:

6.4.5 Local Functions

None

7 MIS for Early Detection of Signs of Anorexia Module

7.1 Module

Task 2 - Early Detection of Signs of Anorexia using User Posts

7.2 Uses

Used for predicting early signs of anorexia using aggregates of user posts trained on data from users who were to known to not have or have anorexia symptoms.

7.3 Syntax

7.3.1 Exported Constants

None

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
pathaddress	String	String	-
parsedata	String	Dictionary	-
goldentruth	String	Dictionary	-
bert	Dictionary	List, List	-
predictor	List, List	Dictionary	-
calculateaccuracy	Dictionary, Dictionary	None	-
gettopicdistribution	List, List	None	-

7.4 Semantics

7.4.1 State Variables

None

7.4.2 Environment Variables

None

7.4.3 Assumptions

- Data from user posts are in an xml file and the ground truth file is in a txt file and are present in the current working directory.
- The model is trained on accurate data with correctly linked user-truth values.

7.4.4 Access Routine Semantics

pathaddress():

- transition: None
- output: The path address for the data files
- exception: None

parsedata():

- transition: None
- output: Dictionary with usernames as keys connected to a list of strings containing user posts
- exception: None

goldentruth():

- transition: None
- output: Dictionary with usernames as keys connected to an integer representing their anorexia status (0 or 1)
- exception: None

bert():

- transition: None
- output: List 1: List of usernames, List 2: List of topics generated from the bertopic model with the indexes corresponding to the usernames from the first list
- exception: None

predictor():

- transition: None
- output: Dictionary with usernames as keys and the predicted 0 or 1 integer linked
- exception: None

calculateaccuracy():

- transition: None

- output: Accuracy scores comparing golden truth to predictor values

$$Precision(P) = \frac{|u \in U : d = g = 1|}{|u \in U : d = 1|}$$

$$Recall(R) = \frac{|u \in U : d = g = 1|}{|u \in U : g = 1|}$$

$$F - Score(F) = \frac{2 * P * R}{P + R}$$

- exception: None

gettopicdistribution():

- transition: None
- output: List with index representing the topic distribution of users with sepperation into golden truth of 0 and 1s
- exception: None

7.4.5 Local Functions

None

8 MIS for Measuring the Severity of the Signs of Eating Disorders

8.1 Module ed_input

8.2 Uses

None

8.3 Syntax

8.3.1 Exported Constants

None

8.3.2 Exported Access Programs

Name	In	Out	Exceptions
read_file	*String*	*Dict*	*IncorrectFileFormat*

8.4 Semantics

8.4.1 State Variables

None

8.4.2 Environment Variables

None

8.4.3 Assumptions

The input to read_file will be a path to a file on the disk.

8.4.4 Access Routine Semantics

read_file($pathaddress : String$):

- transition: None
- output: $Dict\ of < username : String,\ posts : [post : String | post \in user's\ posts] >$
- exception: If the input file is not correctly formatted

8.4.5 Local Functions

None

8.5 Module ed_output

8.6 Uses

None

8.7 Syntax

8.7.1 Exported Constants

None

8.7.2 Exported Access Programs

Name	In	Out	Exceptions
write_file	*Str*, *Dict*	None	None

8.8 Semantics

8.8.1 State Variables

None

8.8.2 Environment Variables

None

8.8.3 Assumptions

The rest of the pipeline up to this point will produce a $Dict\ of\ <username : String,\ scores : [s_1, s_2, ..., s_{22}]>$ where $s_1, s_2, ..., s_{22}$ are the predicted scores for each of the eating disorder questionnaire's 22 questions

8.8.4 Access Routine Semantics

write_file($file : Str,\ scores : Dict\ of\ <username : Str,\ scores : [s_1, s_2, ..., s_{22}]>$):

- transition: writes *scores* to the file specified by $file$ where each line is of the format "username: s1, s2, ..., s22" for each user.
- output: None
- exception: None

8.8.5 Local Functions

None

8.9 Module ed_pipeline

8.10 Uses

ed_input, ed_tokenizer, ed_rep_model, ed_pred_model, ed_output

8.11 Syntax

8.11.1 Exported Constants

None

8.11.2 Exported Access Programs

Name	In	Out	Exceptions
run_pipeline	Str, Str	None	None

8.12 Semantics

8.12.1 State Variables

None

8.12.2 Environment Variables

None

8.12.3 Assumptions

The first input to run_pipeline is a path to a file on the disk

8.12.4 Access Routine Semantics

write_file($in\_file : Str,\ out\_file : Str$):

- transition: calls ed_input to read from in_file, runs the full pipeline, and calls ed_output to write user scores to out_file.
- output: None
- exception: None

8.12.5 Local Functions

None

8.13 Module ed_tokenizer

8.14 Uses

None

8.15 Syntax

8.15.1 Exported Constants

None

8.15.2 Exported Access Programs

Name	In	Out	Exceptions
tokenize	*Str*, [*Bool*]	*List of Str*	None

8.16 Semantics

8.16.1 State Variables

None

8.16.2 Environment Variables

None

8.16.3 Assumptions

None

8.16.4 Access Routine Semantics

write_file(*text* : *Str*, *options* : *List of Bool*):

- transition: None
- output: A list of words representing *text* split by whitespace after *text* is processed in a number of ways specified by *options*.
- exception: None

8.16.5 Local Functions

None

8.17 Module ed_rep_model

8.18 Uses

None

8.19 Syntax

8.19.1 Exported Constants

None

8.19.2 Exported Access Programs

Name	In	Out	Exceptions
represent	$[Str]$, $[Bool]$	$[Float]$	None

8.20 Semantics

8.20.1 State Variables

None

8.20.2 Environment Variables

None

8.20.3 Assumptions

The first input to represent will be the output of ed_tokenizer: a processed list of the words in the input text split by whitespace.

8.20.4 Access Routine Semantics

represent($text : List\ of\ Str$, $options : List\ of\ Bool$):

- transition: None
- output: A list of floats representing *text* encoded by some method into a numerical format, the details of the conversion are specified by *options*.
- exception: None

8.20.5 Local Functions

None

8.21 Module ed_pred_model

8.22 Uses

ed_transform

8.23 Syntax

8.23.1 Exported Constants

None

8.23.2 Exported Access Programs

Name	In	Out	Exceptions
predict	$< Str, [Float] >,\ [Bool]$	$< Str, [Float] >$	None

8.24 Semantics

8.24.1 State Variables

None

8.24.2 Environment Variables

None

8.24.3 Assumptions

The first input to predict will be a dictionary of the outputs of ed_rep_model for each user.

8.24.4 Access Routine Semantics

represent($user\_data :< username : text, representation : [Float] >, options : List\ of\ Bool$):

- transition: None
- output: A dictionary of $< username : String, scores : [s_1, s_2, ..., s_{22}] >$ where $s_1, s_2, ..., s_{22}$ are the predicted scores for each of the eating disorder questionnaire's 22 questions for each user *username*.
- exception: None

8.24.5 Local Functions

None

8.25 Module ed_transform

8.26 Uses

ed_transform

8.27 Syntax

8.27.1 Exported Constants

None

8.27.2 Exported Access Programs

Name	In	Out	Exceptions
predict	$< Str, [Float] >,\ [Bool]$	$< Str, [Float] >$	None

8.28 Semantics

8.28.1 State Variables

None

8.28.2 Environment Variables

None

8.28.3 Assumptions

None

8.28.4 Access Routine Semantics

represent($user\_data :< username : text, representation : [Float] >, options : List\ of\ Bool$):

- transition: None
- output: A dictionary of $< username : String,\ representation : [Float] >$ where each of the output's *representation*s is a transformation of the input *representation*s.
- exception: None

8.28.5 Local Functions

None

References

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 2003.

Daniel M. Hoffman and Paul A. Strooper. *Software Design, Automated Testing, and Maintenance: A Practical Approach*. International Thomson Computer Press, New York, NY, USA, 1995. URL http://citeseer.ist.psu.edu/428727.html.

9 Appendix

9.1 Reflection

9.1.1 Limitations of the proposed solution.

One of the primary limitations recognized in our design is that the performance of the various risk detectors are all turtlenecked with the spread of training data available to them. This primarily being enforced by biases present in the data which may skew results incorrectly, a prominent issue with all projects that rely on artificial intelligence to help predict an outcome. This bias can present itself as a decreased effectiveness for variations in cultural and linguistic backgrounds of the analyzed user data.

Additionally, this ties into the overall limitation of the system of how this tool is intended to be assistive and supplementary to trained mental health professionals, it is not meant to be the sole source of a diagnosis. Biases and limitations of the code structure will always offer an amount of machine error, which is why this program is to be used under the guidance of professionals who can help corroborate the results using their situation context and emotional intelligence, an ability that cannot ever truly be given to a computer.

9.1.2 Benefits and trade-offs of other potential solutions.

Early in the design and planning process, the team experimented with the idea of using large language models similar to ChatGPT for diagnosis. This would allow us to leverage well documented and existing technologies in our efforts, providing a strong jumping off point for our workflow. Unfortunately, the team was able to soon discover research which showed that models like ChatGPT tend to draw undesirable parallels with their training data, resulting in the system unknowingly fabricating information in the input data which would provide incorrect results.