Natural Language Understanding for Computerized Cognitive Behavourial Therapy
PHY-555A Endsem Report

Anirudha Brahma
Dept. Of Physics
anirbrhm@iitk.ac.in

Prof. Nitin Gupta
Dept. Of BSBE
guptan@iitk.ac.in

Prof. Mahendra Kumar Verma
Dept. Of Physics
mkv@iitk.ac.in

Abstract

Depressive and anxiety disorders have a high health burden and a considerable treatment gap worldwide, especially in India. Cognitive Behavioral Therapy (CBT) is an empirically validated form of therapy to treat depression. This form of therapy is quite expensive and inaccessible to most of the population in India. We aim to develop an artificially intelligent system for Computerized CBT which would make this treatment accessible to the majority of the population in India. This work aims to introduce natural language understanding in our system in a way that is compatible with knowledge graphs and finite state machines. Our work would focus on developing a spell correction system using the Needleman Wunsch algorithm from Biophysics and also introducing pretrained tokenizers and pretrained featurizers into the NLU system.

1 Introduction

Major Depressive Disorder is a mental illness that affects how one feels, thinks, behaves, and enjoys various activities. A depressed person is likely to have compromised productivity at work, poor health, and troubled relationships. The proportional burden of mental disorders in the total disease burden of India has doubled since 1990. Depressive and anxiety disorders account for the highest proportional burden among all mental disorders with varying severity in adults. Effectively treating a depressed person has complexities at psychological and socio-economic levels. Countries like India have a treatment gap because of the low availability of trained psychotherapists for the number of depressed patients.

Cognitive Behavioral Therapy (CBT) is a widely accepted and empirically tested form of therapy. Computerized Cognitive Behavioral Therapy (CCBT) is an adapted form of CBT to provide its content through low-cost digital means like websites meet the disparity in therapist to patient ratio.

TreadWill, a conversational AI under development in the Lab of Neural Systems IITK provides an interactive personalized space for the task of CCBT. Research shows that chatbot for mental health has a favourable space amongst users.

2 Methodology

2.1 Knowledge Graphs and Dialog System

The dialog system includes a component of natural language understanding for detecting the user's intent. The intent is a concept in the user's mind while sending a message. For example, again, if asked the question "Describe the situation when you had difficult thoughts", the user with an intent to express the situation responds with the message, "I was spending time together with my friends", while he responds with the message "I don't know" in case of an intent to express doubt.

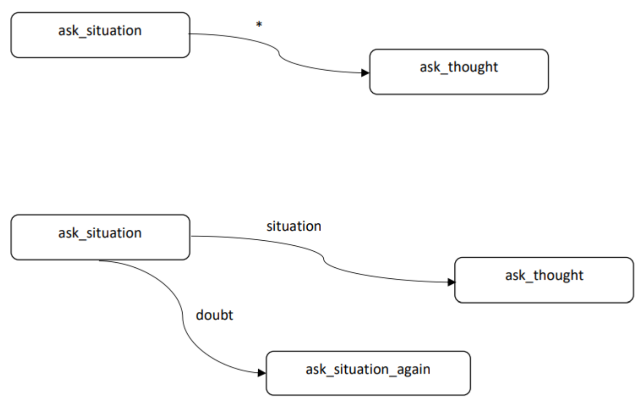

Figure 1: Dialog Sytems based on Knowledge Graphs and Finite State Machines

2.2 Natural Language Understanding

2.2.1 Framework

This work aims to introduce natural language understanding in our system in a way that is compatible with knowledge graphs and finite state machines. Our work would focus on introducing pretrained tokenizers and pretrained featurizers in the system to increase the bot's performance significantly. We use the RASA NLU to classify all user-input sentences into various intents relevant to evaluating the negative automatic thought such as greeting, feeling, situation, and thought.

RASA uses a pipeline of three components – a tokenizer, a featurizer, and a classifier to train a model to process the raw text data and output an intent. A tokenizer breaks a sentence into smaller units called tokens. A featurizer converts the tokens into numerical representations, generally vectors, in a space of features. A classifier uses the vectors to map an intent onto an incoming sentence.

Figure 2: RASA NLU Framework

2.2.2 Training data

A good amount of time was spent trying to collect and clean good quality training data for our AI system. Data was collected using inputs from friends, family, and self. Online resources such as open-source databases and movie quotations were also used to add to our training data.

2.2.3 Configurations

First we experimented with the default pipeline suggested in the RASA docs for The NLU component, which consisted of the Whitespace Tokenizer, Count Vectors Featurizer, and DIETClassifier. Whitespace Tokenizer separates the tokens using whitespaces in the sentences. Count Vectors Featurizer creates a space of known tokens and takes a sentence to make its bag-of-words (BoW) representation, a vector with each component corresponding to the frequency of a token in the sentence. DIET (Duel Intent and Entity Transformer), a neural network transformer-based classifier embeds vectors of sentences and intent labels into a single semantic vector space and classifies a new sentence using a dot product to put an intent label with the minimum dot product loss. Then we attempted to increase the performance of our model using pretrained featurizers and tokenizers from the Spacy NLP library, which uses the Regex Featurizer and Lexical Syntactic Featurizer in its usage. Then we used more complex language models such as Roberta and BERT to increase the performance of our classifier many folds.

2.3 Original Contribution to the Project - Spell Correction System

Even though our transformer-based classification model with pretrained tokenizers and featurizers was successful enough to classify our intents into the 12 classes presented to us, the model had significant limitations when classifying sentences with human spelling mistakes. Our model performed poorly while classifying even straightforward sentences with very basic human errors, which would otherwise have been very obvious to a human. So we need a robust system to mitigate this problem and allow our model to give good results in practice where a human might make similar errors while conversing with our AI system.

A naive approach to solving this problem is to use autocorrect to rectify the words that are not in the English dictionary and then apply our ML model to the corrected sentence. On the surface, this approach looks like the correct one. However, researchers in RASA have shown that applying this autocorrect without any context to the problem at hand decreases the model's accuracy and performance due to the autocorrect system's lack of understanding of the words required in the problem.

We need an autocorrect system that corrects the words from a small vocabulary containing the words in our problem's context and relevant to us. We propose using the famous Needleman-Wunsch algorithm used to find similarities in the amino acid sequences of two proteins in BioPhysics to address this seemingly unrelated problem of spell correction. On top of using this algorithm, we suggest a modification to this algorithm to further increase the efficiency of our system.

The NW algorithm is commonly used to assess whether it is likely that two sequences evolved from the same sequence or to determine which sequences from the database are similar to the sequence at hand. The NW algorithm compares 2 DNA sequences using a metric that calculates a similarity score in relation to the number of steps we would take to convert one sequence into the other with the help of 3 operations given to us, namely addition, deletion, and insertion. We call this number of steps the edit distance to convert sequence 1 to sequence 2.

For example, Assume we want to align two nucleotide sequences, S and T, where S = AGT T = AAGC

The first step is placing the two sequences along the margins of a matrix and initializing the matrix cells. To initialize, we assign a 0 to the first entry in the matrix and then fill in the first row and column based on the incremental addition of gap penalties. Although the algorithm could fill in the first row and column through iteration, it is essential to clearly define and set boundaries on the problem.

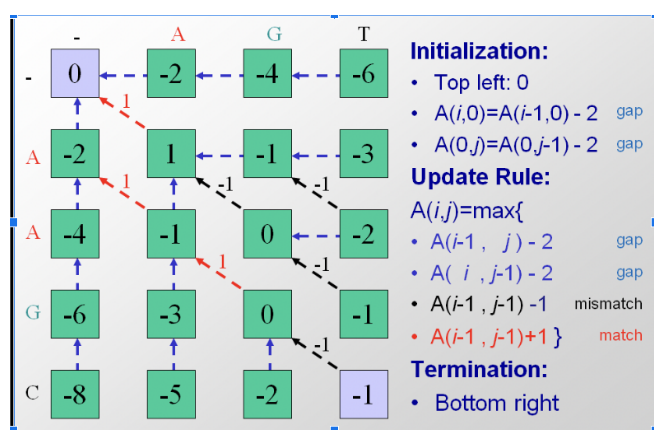

Figure 3: Implementation of the Needleman Wunsch Algorithm to compare the similarities of 2 nucleotide sequence

The next step is iteration through the matrix. The algorithm proceeds either along rows or columns, considering one cell at a time. For each cell, three scores are calculated, depending on the scores of three adjacent matrix cells (specifically the entry above, the one diagonally up and to the left, and the one to the left). The maximum score of these three possible tracebacks is assigned to the entry, and the corresponding pointer is stored. Termination occurs when the algorithm reaches the bottom right corner.

Using this algorithm, we are able to get a metric of how similar or different two nucleotide sequences are based on the similarity scores.

2.3.1 Using Needleman Wunsch Algorithm for spell correction system

We first create a contextual dictionary of words for our spell correction system. We do this by making a set of all the unique words in our training data. Now for any incorrect word that the user inputs, we compare the similarity scores of the input with all the words in our contextual dictionary using the NW algorithm and update the incorrect input word with the correct word.

In this way, we are able to leverage the fantastic algorithms developed in the field of BioPhysics to address our issue of spell correction.

2.3.2 Modifications suggested to the Needleman-Wunsch Algorithm

The knowledge that the user uses the qwerty keyboard to input sentences allows us to exploit this fact and improve our algorithm by many manifolds. We suggest a change in the way we calculate similarity scores to improve the algorithm. In our current system, we give a weightage of 1 to the operation of substitution of a letter by another in the edit distance version of our algorithm. Although this approach seems correct when comparing two nucleotide sequences, that is not how humans make errors while typing sentences. A human actually has a lot more probability of mistakenly typing a letter next to the intended letter on a qwerty keyboard rather than a letter far away from the intended letter on the keyboard.

So instead of assigning a weightage of 1 to the substitution operation, we assign a weightage according to the distances of the two letters in question on the qwerty keyboard. Our algorithm takes the normalized euclidean distance between the two letters on the qwerty keyboard.

2.3.3 Assembling all the parts together

The three approaches we suggested, i.e., classification without spell correction, spell correction with Needleman-Wunsch, and spell correction with modified Needleman-Wunsch, have their pros and cons, and their performance varies from case to case basis. We suggest an ensemble technique to leverage the benefits of all three approaches. If the three approaches agree on the classification, we output the class. If all the three approaches do not agree with each other, we output the class with the maximum classification class at hand.

If two approaches agree, we accept that output except when the output from the third approach is too strong compared to the output from the two approaches. In mathematical terms, if the classification probability from the dissimilar approach is ¿0.8 and the classification probability from the similar approaches are both ¡0.5, then we choose the output from the dissimilar approach.

3 Results and Conclusions

To test the performance of our models and make a robust comparative study of the different configurations and analyze the performance in each of the configurations

Config Name	Accuracy	Efficiency
Default	57 %	High
Spacy	65 %	High
Roberta	67 %	Low
BERT	76 %	High

Table 1: Accuracy comparisons of different configurations

3.1 Testing Methodology

To avoid overfitting of the model we test and train our models on varying degrees of data. We have 5 sets of data namely, 0% exclusions, 25%, 50%, 70%, 90% exclusions. After excluding these percentages we split our data into train and test in the ratio of 4:1. We then train our model on the training data and test on the remaining testing data.

We use accuracy and confusion matrix as metrics to test our model.

3.2 Conclusions

We see that using the BERT pretrained featurizers along with our spell correction system boosts the accuracy of our model from 57% to 76% and provides us with a robust and reliable system to perform real-time on-field Computerized Cognitive Behavioral Therapy.

4 Acknowledgements

I would like to take this opportunity to thank Prof. Nitin Gupta for considering me for this project and for believing that I could carry this work forward and do some impactful work on this project. His lab was a great source of inspiration and positive energy for me and I look forward to working in a lab like this.
I would also like to thank Prof. Mahendra Kumar Verma for agreeing to co-advise me on this project. It means a lot to me to get this help from him, that I was able to do a project in the field I wanted to. Thank you Prof. Verma.
I would finally like to thank Silky Gupta for constantly and smoothly guiding me through this project and always being a constant help throughout the semester. This project would not have been possible without her constant support and positive approach.

5 References

- Beck, J. S. (2020). Cognitive behavior therapy: Basics and beyond. Guilford Publications.
- Richards, D., Richardson, T. (2012). Computer-based psychological treatments for depression: a systematic review and meta-analysis.
- https://rasa.com/docs/rasa/installation/environment-set-up/
- TreadWill. https://www.treadwill.org/