Performance comparison ensemble classifier's performance in answering frequently asked questions about psychology

Vy Thuy Tong[1], Hieu Chi Tran[1], Kiet Trung Tran[1*]

[1]Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam
[*]Corresponding author: kiet.tt@ou.edu.vn

ARTICLE INFO

DOI:10.46223/HCMCOUJS.tech.en.14.1.2921.2024

Received: August 21st, 2023

Revised: February 03rd, 2024

Accepted: February 07th, 2024

Keywords:

classification; KNN; Naive Bayes; psychology; SVM

ABSTRACT

In today's era of digital healthcare transformation, there is a growing demand for swift responses to mental health queries. To meet this need, we introduce an AI-driven chatbot system designed to automatically address frequently asked questions in psychology. Leveraging a range of classifiers including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Naïve Bayes, our system extracts insights from expert data sources and employs natural language processing techniques like LDA Topic Modeling and Cosine similarity to generate contextually relevant responses. Through rigorous experimentation, we find that SVM surpasses Naïve Bayes and KNN in accuracy, precision, recall, and F1-score, making it our top choice for constructing the final response system. This research underscores the effectiveness of ensemble classifiers, particularly SVM, in providing accurate and valuable information to enhance mental health support in response to common psychological inquiries.

1. Introduction

Overall, in the period of digital transformation in health and application (Wei et al., 2019), improving the quality of mental health is gradually being focused and there are positive changes in healthcare activities. The Ministry of Health promotes the deployment of online and remote medical consultation and supportive treatment platforms, connecting hospitals, medical examination, and treatment facilities to faraway islands.

This project is aimed at assisting patients suffering from psychological disorders. Individuals wondering that they may experience psychological symptoms and seek answers, as well as anyone suffering from inquiries related to psychological issues. To achieve this goal, an AI-driven chatbot system has been developed.

According to the point above, the chatbot system has been designed to address common questions in the field of psychology. We utilized a classification algorithm to predict and answer these frequent questions, based on data collected from experts in the field. By integrating high-quality data and significant contributions from experts, this chatbot system is capable of providing accurate and valuable information about psychology. It helps users gain a clear understanding of their psychological well-being and offers guidance on self-managing their mental health.

The research objective aims to develop an automated chatbot system capable of addressing various psychological issues, such as depression, insomnia, and anxiety, through real-time conversations with users. Within the chat framework, the application allows users to input custom responses and optimizes the system to provide the most effective responses. The research also explores the comparison of machine learning algorithms like SVM (Maraoui, Haddar, & Romary,

2021), KNN (Mohammed & Omar, 2020), and the integration with the Cosine similarity algorithm to assist in user queries and deliver contextually suitable automatic responses. Additionally, we employ natural language processing techniques, specifically utilizing the LDA Topic Modeling model, to identify common themes in user input.

The rest of this paper is organized as follows. Section 2 presents the related works. Section 3 presents methods used for automatic psychological response systems and automatic support of psychological exercises and section 4 presents the experimental results. Finally, the conclusions and future works are presented in section 5.

2. Related works

In our project, we focus on developing a Question and Answer (QA) system specifically tailored to psychological inquiries. These questions and answers are curated and collected by experts in the field of psychology, we draw inspiration from various research proposals, such as the one by (Wei et al., 2019), which leverages the BERT method in conjunction with the Anserini open-source information retriever, Sarkar and Singh (2023) take a pure Natural Language Processing (NLP) approach for QA, (Cai, Wei, Zhou, & Yan, 2020) focus on integrating domain-specific information for Restricted Domains QA model. Maraoui et al. (2021) utilize SVM, CS, and LCS to narrow down the search scope of Hadith documents based on various topics and question types, efficiently analyzing query needs using NLP methods. Mohammed and Omar (2020) use KNN and SVM for the QA system by Question Answering System, which summarizes a tagged datastore and provides summary answers.

In this QA model, we propose a novel approach by calculating the similarity between the question and the extracted questionnaire when the patient answers, then apply a supervised learning machine learning model to estimate the classification score for the patient's answer, the final result is the answer with the highest score. In the realm of psychology, our initial task is to create a dataset question-answer by extracting data from reputable websites such as library.rochester.edu, vietcetera.vn, figshare.com, and more. To accomplish this, we use several libraries, including Python Scrapy, Selenium, and others.

We use tf×idf (Term Frequency-Inverse Document Frequency) to characterize the documents. This method effectively assesses word importance effectively by considering the frequency of a word in a document (tf) and the number of documents containing that word (df). The idf component is inversely related to df. Words with high tf values indicate potential keywords, whereas words frequently appearing in other documents (e.g., "an", "the", "in", "and", etc.) are considered less meaningful. By multiplying tf by idf, we reduce the weight of these common words. Experimental results from (Cai et al., 2020) demonstrate that tf×idf consistently achieves high scores and excels in text classification, aligning closely with the objectives of our problem.

3. The approach

To streamline this process, we utilize the TF-TDF word bag method for constructing feature vectors that capture the semantic essence of the text. Following this, we conduct experiments involving a variety of classification algorithms, including Multi-Class SVM, Naive Bayes, and KNN algorithm in our model, and perform accurate training in our model and classification tasks. Through these experiments, we aim to assess the performance of each algorithm. Subsequently, we employ a machine learning-based voting system to select the most suitable answer based on these experimental results.

3.1. Preprocessing and data cleaning

In linguistic analysis and information retrieval, data preparation plays a crucial role in uncovering the underlying meaning and emotions embedded in textual information. To achieve this, it is essential to eliminate any distortions or abnormal patterns present in the data. Special characters and repeated characters in phrases are removed to ensure a clean and standardized input for analysis. Additionally, stemming and lemmatization are employed to transform words into their base or root forms, thereby generalizing the sentences and revealing the intended meaning. The Porter-Stemmer technique, a widely used stemming algorithm, calculates the suffix of a word to obtain the correct sense of the data. For instance, words like "running", "run", and "ran" are all reduced to the stem "run", facilitating better understanding and interpretation of the content.

In the context of QA data processing, both questions and answers benefit from the application of the Porter Stemmer. By grouping related words, the algorithm enhances the matching and retrieval process, enabling the extraction of relevant answers for a given question. Moreover, reducing words to their root forms increases the chances of finding pertinent information despite minor variations in word endings or inflections. Another important step in the data preprocessing technique is removing stop words, commonly occurring words that add little or no meaningful information to the sentences. Words like "the", "is", "and", "in", etc., are typically discarded from the text to focus on the essential content, leading to improved efficiency in natural language processing tasks. In conclusion, employing the Porter Stemmer and eliminating stop words are fundamental techniques in the preprocessing of linguistic data. They contribute to a clearer representation of the content, enabling more accurate linguistic analysis and facilitating extracting valuable insights and emotions from the textual information.

3.2. KNN

K-Nearest Neighbors (KNN) is a widely recognized supervised machine learning algorithm. To classify a data point, d⟩ into a specific group, KNN identifies its K-nearest data points based on certain similarity or distance metrics. Subsequently, is assigned to the same group as its closest data points. This method is straightforward to implement and often exhibits good performance. However, it comes with the drawback of high computational cost, especially when dealing with large datasets. Additionally, KNN is sensitive to noise data, especially when the value of K is small.

3.3. Naive Bayes

Na¨ıve Bayes is another supervised machine learning algorithm that relies on probabilities and Bayes' theory to make predictions on a given dataset. It assumes that the features are conditionally independent given the class label. In other words, it assumes that each feature contributes independently to the probability of a certain class.

To classify a new data point xx into a particular class using the Na¨ıve Bayes classifier, the probability is calculated as follows:

$$P(C_k|X) = \frac{P(X|C_k).P(C_K)}{P(X)} \quad (1)$$

The Na¨ıve Bayes classifier will then assign the data point to the class with the highest posterior probability *P(class|x)*. Since it assumes independence between features, Na¨ıve Bayes can be computationally efficient and perform well in certain classification tasks, especially when the dataset is relatively small. However, its performance may degrade if the independence assumption does not hold well or when dealing with complex dependencies between features.

3.4. SVM

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm primarily used for classification tasks. The objective of SVM is to find an optimal hyperplane that effectively separates different classes in the feature space. This hyperplane maximizes the margin between the two classes, which leads to better generalization and robustness of the model.

Given a training dataset consisting of labeled samples (x_i, y_i), where x_i represents the feature vector and y_i is the corresponding class label (either +1 or -1), SVM aims to find the hyperplane represented by the equation $w \cdot \mathbf{x} + b = 0$, where w is the weight vector and b is the bias term. The goal is to find w and b such that the margin between the positive and negative classes is maximized.

The distance between a data point x_i and the hyperplane is given by, $d_i = \frac{|w.x_i+b|}{||x||}$, where ‖w‖ is the norm of the weight vector w. The margin is defined as the minimum distance from any data point to the hyperplane.

To maximize the margin, SVM seeks to solve the following optimization problem using Lagrange multipliers:

$$Maximinze \frac{2}{||w||} subjecttoy_i(w.x_i + b) \geq 1 \ (for\ all\ i\ =\ 1,\ldots,n) \quad (2)$$

The constraints ensure that each data point is correctly classified and lies outside the margin region. The Lagrange multipliers α_i are introduced to convert the inequality constraints into equality constraints.

By solving the optimization problem, the Lagrange multipliers α_i are obtained. The support vectors are the data points for which $\alpha_i > 0$, and they lie on the margin or within the margin region. The weight vector w can be calculated as a linear combination of the support vectors, and the bias term b is computed from the support vectors that lie directly on the margin.

In the end, the SVM classifier can be represented by the hyperplane $\boldsymbol{w} \cdot x + b = 0$. To classify a new data point x, we simply compute $\boldsymbol{w} \cdot x + b$ and assign it to the positive class if $\boldsymbol{w} \cdot x + b > 0$, otherwise to the negative class.

3.5. Our proposals

We will choose the highest-rated model out of three models, KNN, Naive Bayes, and SVM, to use the QA System for the model. The dataset consists of paired student questions and expert answers. Full-Text Search (FTS) is used to find relevant questions, and then vectorization with *tf*×*idf* weights is applied for efficient processing and matching of user queries. This approach ensures accurate responses based on expert knowledge. The system calculates each dimension (wi) using the *tf*×*idf* weight. This weight is derived from term frequency (*tf*) and inverse document frequency (*idf*), ensuring an effective representation of the data for improved information retrieval and matching:

$$tf_{w_i} = freq(w_i) log \frac{N}{1 + df} \quad (3)$$

4. Experiments

Before conducting the experiments and testing the results, the model was split into training and testing data with a split ratio of 75%. The random state used for the split was set to 42. Then, we applied three algorithms, namely SVM, Naive Bayes, and KNN, to predict the outcomes. In the SVM algorithm, we used the SVC with a Linear Kernel. To find the best combination of

hyperparameters, we utilized GridSearchCV, which led to the optimal parameters of 'C': 2, 'gamma': 0.1, and 'kernel': 'linear'. For Naive Bayes, we employed the Multinomial Naive Bayes algorithm, and for KNN, we set the number of neighbors (n neighbors) to 5. After analyzing the results, it was evident that the SVM algorithm outperformed both Naive Bayes and KNN algorithms, showing higher accuracy. Following the experimentation phase, we tested all three algorithms with the processed Question tokens data to identify the properly labeled data. The classification report for the algorithms is presented below in Table.

Table 1

Confusion matrix of SVM, Naive Bayes, KNN

	KNN	**Naive Bayes**	**SVM**
Accuracy	72.69	71.60	67.07
Precision	59.11	59.11	59.20
Recall	63.03	71.60	67.07
F1-score	59.96	56.13	51.10

The results obtained from the application of three different algorithms in our system are as follows: SVM achieved an accuracy of 72.69%, with precision and recall scores of 59.11% and 72.69%, respectively. Naive Bayes exhibited an accuracy of 71.60%, with precision and recall scores of 59.11% and 56.02%, respectively. KNN, on the other hand, had an accuracy of 67.07%, with precision and recall scores of 59.20% and 50.02%, respectively.

These results indicate that SVM outperforms both Naive Bayes and KNN in terms of accuracy, achieving nearly 73%. However, it's noteworthy that Naive Bayes also performs reasonably well with an accuracy of 71.60%. In terms of precision, SVM and Naive Bayes exhibit similar performance, with SVM having a slight edge. Meanwhile, KNN lags behind in precision. When it comes to recall, SVM stands out with a significantly higher score, suggesting its effectiveness in correctly identifying relevant answers.

Overall, the comparison graph visually illustrates that SVM boasts the highest accuracy, albeit marginally higher than Naive Bayes, which in turn outperforms KNN. These results provide valuable insights into the performance of these algorithms in our system, with SVM emerging as the top-performing choice for achieving a high level of accuracy in question-answering tasks.

5. Conclusions

The dataset for this system is created by pairing questions and answers together. When a new question is asked, the system performs English word segmentation to remove stop words, which helps the data and meaningful content. Subsequently, Full-Text Search (FTS) is employed to identify questions that are relevant to the user's query, narrowing down the search space for more efficient processing. To provide accurate answers, the system employs three classifiers: K-Nearest Neighbors (KNN), Na¨ıve Bayes, and Support Vector Machine (SVM). The highest-rated model is selected to build the final response model, ensuring effective question answering even for complex queries, as evident from the test results. The versatility of this approach allows it to be applied in various domains.

References

Cai, L. Q., Wei, M., Zhou, S. T., & Yan, X. (2020). Intelligent question answering in restricted domains using deep learning and question pair matching. *Ieee Access, 8*(8), 32922-32934.

Maraoui, H., Haddar, K., & Romary, L. (2021). Arabic factoid question-answering system for Islamic sciences using normalized corpora. *Procedia Computer Science*, *192*(2021), 69-79.

Mohammed, M., & Omar, N. (2020). Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec. *PloS One*, *15*(3), Article e0230442.

Sarkar, S., & Singh, P. (2023). Combining the knowledge graph and T5 in question answering in NLP. *Sentiment Analysis and Deep Learning: Proceedings of ICSADL 2022*, 405-409.

Wei, Y., Yuqing, X., Aileen, L., Xingyu, L., Luchen, T., Kun, X., … Jimmy, L. (2019). End-to-end open-domain question answering with BERTserini. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 72-77.