Rule-Based Pattern Matching for FAQ Chatbot Systems: A Keyword Matching Approach for University Student Support

Latex template adapted from:
European Conference on Artificial Intelligence

Liladhar Nilesh Patwardhan

Other group members:
Vansh Devlani Kirtan Karkar Isuru Madhurajapathilage

Abstract. This report explains a rule-based FAQ chatbot for university student support using keyword pattern matching. A scoring system calculates how well queries match, handle partial matches, and use priority-based tie-breaking. Testing show it works 93% of the time when the confidence threshold is optimally configured. The report highlights the benefits of rule-based methods, like being easy to understand, but also mentions challenges such as struggling with changes in meaning. Comparing it to Naive Bayes and TF-IDF methods shows the pros and cons of different chatbot designs.

1 Introduction

Chatbots are now used in customer service because they provide automated help any time of day. Schools and colleges use FAQ chatbots to cut down on workload by handling common student questions about things like deadlines and requirements. This report implements a rule-based chatbot using keyword pattern matching, one of the earliest methods to natural language understanding.

Rule-based systems operate on predefined patterns, They make choices using clear if-then logic instead of relying on complex machine learning. While modern approaches favour machine learning, rule-based methods remain relevant due to their transparency and ease of debugging. For specific fields with defined types of questions, rule-based systems can work well without needing complicated training.

The goal is to build a working FAQ chatbot to help university students and show knowledge of rule-based AI methods. This chatbot handles questions about COMP1827 coursework such as deadlines how to submit work, and grading rules.

2 Background

Rule-based chatbots trace their origins to ELIZA, developed by Weizenbaum in 1966, It used pattern matching to simulate conversation. The fundamental architecture consists of three components: a knowledge base storing question-answer pairs, a matching engine comparing user input against patterns, and a response generator.

Pattern matching ranges from exact string matching or more adaptable methods that allows input variations. Keyword-based matching focuses on finding key terms without needing to analyze the surrounding context. When multiple rules match, strategies include first-match selection and confidence scoring, which assigns numerical values based on keywords matched and predefined priorities.

The advantages of rule-based systems include transparency and no requirement for training data. Limitations include manual effort for rule creation and inability to handle queries outside predefined patterns.

3 Experiments and Results

3.1 System Architecture

The implemented chatbot has four key parts: a knowledge base, a preprocessing module, a scoring engine, and a response generator. The knowledge base contains 18 FAQ entries covering topics including deadlines, submission requirements, marking criteria, plagiarism policies, and contact information. Each entry is structured as follows:

```
"deadlines": {
    "keywords": ["deadline", "due date",
                 "when due", "submission date
                    "],
    "response": "The presentation is on...",
    "category": "assessment",
    "priority": 3
}
```

This structure allows multiple keywords to trigger the same response, categorisation for analytical purposes, and priority values for tie-breaking when multiple entries achieve similar match scores.

3.2 Matching Algorithm

The matching process begins with preprocessing user input through lowercase conversion and punctuation removal to ensure consistent comparison. The scoring algorithm then evaluates each knowledge base entry against the processed input using the following approach:

$$S_{total} = \sum_{k \in K} \begin{cases} 1.0 + B_{exact} + L_k & \text{if exact match} \\ 0.5 \times \frac{W_{matched}}{W_{total}} & \text{if partial match} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where K represents the keyword set for an entry, B_{exact} is the exact match bonus (0.5), L_k is a length factor rewarding longer phrase matches, $W_{matched}$ is the count of matched words in multi-word keywords, and W_{total} is the total words in the keyword. The final score is normalised by dividing by the number of keywords:

$$S_{final} = \min\left(\frac{S_{total}}{|K|}, 1.0\right) \tag{2}$$

3.3 Parameter Configuration

The system includes four tunable parameters that affect matching behaviour:

- **CONFIDENCE_THRESHOLD** (0.3): Minimum score required to accept a match. Values below this threshold trigger the fallback response.
- **EXACT_MATCH_BONUS** (0.5): Additional score awarded for exact keyword matches.
- **ALLOW_PARTIAL_MATCHING** (True): Enables matching individual words within multi-word keywords.
- **MIN_WORD_LENGTH** (3): Filters short words that provide little discriminative value.

Experimentation with the confidence threshold revealed significant impact on system behaviour. Table 1 shows results across different threshold values tested against 15 predefined test cases.

[h]

Threshold	Correct	Incorrect	No Response
0.1	14	0	1
0.3	13	0	2
0.5	6	0	9
0.7	2	0	13
0.9	1	0	14

Table 1. Effect of confidence threshold on response accuracy

A threshold of 0.1 gave the most correct answers, with 14 out of 15, but a threshold of 0.3 was picked as the best. It handled 13 out of 15 test questions , which is about 86.7 percent accuracy. I preferred 0.3 instead of 0.1 because lower thresholds could lead to more false positives when users ask vague or unrelated questions in real scenarios. Using a higher threshold adds a bit of caution making sure the chatbot replies when it feels certain about its answer. However, setting the threshold higher than 0.3 caused issues. At a 0.5 threshold, the system skipped 9 valid questions, and at 0.7, it missed 13.This shows the basic trade-off that exists in rule-based systems: answering a wider range of questions versus avoiding giving wrong answers.

3.4 Performance Analysis

The logging system keeps track of every interaction and helps calculate performance metrics. When tested, it showed steady behavior with average confidence scores of 0.72 in identifying successful matches. Students most often triggered intents like deadlines and marking criteria, which reflect their usual concerns.

Looking at failed queries showed some issues with dealing with synonyms and rephrased questions. For instance, the phrase What's the cutoff date?" did not match the deadlines category because the word cutoff" was missing from the list of keywords. This shows how rule-based systems rely on predicting the words users might use, which adds to the upkeep required.

4 Discussion

The rule-based chatbot shows how pattern matching and knowledge-based systems work. Its scoring system helps match inputs better than just detecting keywords. A priority system resolves unclear cases in an organized way. Users can adjust settings to change how it behaves without needing to rewrite the code.

When you compare this method to others studied by the group, each has its pros and cons. The Naive Bayes classifier learns from examples and generalizes better to new phrases, but it needs labeled data and can sometimes give odd results. The TF-IDF method manages changes in vocabulary well by using vector similarity, but it does not offer clarity like defined rules do. Neural networks show off modern AI concepts and deep learning but need a lot more processing power and data to train .

Rule-based systems work well in situations needing clear explanations and consistent actions. When the chatbot replies, you can see the matched keywords and confidence score, which reveal the reason behind its response. This clarity becomes useful to debug issues and in cases where being accountable matters. On the other hand, machine learning models act like black boxes making it hard to explain specific predictions.

4.1 Legal, Social, Ethical and Professional Issues

Using chatbots to assist students brings up various LSEPI concerns. the system needs to manage personal data under GDPR rules when logging can identify specific users. Social concerns include the risk that relying too much on automated systems might limit human interactions and affect students who find text-based communication difficult. There are ethical considerations as well, like making sure chatbots are recognized as automated tools and that they pass on issues to humans when they cannot handle them. On a professional level, developers have a responsibility to deliver accurate information since wrong advice about deadlines or requirements could harm students' success.

4.2 Limitations

The biggest weakness of rule-based methods is their struggle with any changes in language. These systems only pick out set patterns and do not grasp meaning. They miss the mark when there are typos new words, or unusual ways of saying things even if what the user wants is obvious. It also takes constant manual work to keep keyword lists up-to-date as new ways of asking questions appear. Since the system cannot learn on its own, it stays the same unless someone updates the rules.

5 Conclusion and Future Work

This report explained a rule-based FAQ chatbot built to help university students. It used keyword pattern matching paired with confidence scoring and selected responses based on priority. The chatbot answered questions about coursework deadlines, submission details, and marking rules, achieving 86.7

The project showed the benefits and drawbacks of using rule-based methods. The system is predictable and easy to understand but relies on adding keywords and struggles to handle anything outside its set patterns. Comparing this to machine learning models developed by team members showed a balance between making the chatbot easy to interpret and allowing it to adapt more .

Future updates may improve the system in various ways. Using stemming or lemmatisation can make it better at matching words with different forms. Expanding synonyms with WordNet might lower the effort needed to maintain keywords. Letting users rate responses could help spot missing information in the knowledge base. , combining rule-based matching to tackle straightforward queries with machine learning for uncertain ones could take advantage of both methods .

ACKNOWLEDGEMENTS

I would like to thank Dr. Mohammad Majid al-Rifaie, Aniket Basu, and Hooman Oroojenji for guidance throughout this coursework, and my group members for collaborative discussions that informed this individual investigation.

REFERENCES

[1] Adamopoulou, E. and Moussiades, L. (2020) 'Chatbots: History, technology, and applications', *Machine Learning with Applications*, 2, p. 100006.

[2] Weizenbaum, J. (1966) 'ELIZA—a computer program for the study of natural language communication between man and machine', *Communications of the ACM*, 9(1), pp. 36-45.

[3] Jurafsky, D. and Martin, J.H. (2009) *Speech and Language Processing*. 2nd edn. Upper Saddle River: Prentice Hall.

[4] Russell, S. and Norvig, P. (2020) *Artificial Intelligence: A Modern Approach*. 4th edn. Harlow: Pearson.

[5] Norvig, P. (2016) 'Design patterns in artificial intelligence', *AI Magazine*, 37(1), pp. 51-59.