Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230554

Movie information AI question answering system based on knowledge graph

Bohan Wang

Software College, Shandong University, Jinan City, Shandong Province, 250000, China

201900303004@mail.sdu.edu.cn

Abstract. Knowledge is essential for understanding. The ongoing explosion of information highlights the need for electronic texts that allow machines to better understand human language. The explosion of information in various fields has greatly changed our lives. Among them, the information explosion of the film is particularly prominent. Nowadays, people's requirements for high quality of spiritual life are increasing, and people's demand for movies is also increasing. With the extensive coverage of the Internet, the amount of movie-related information is also increasing, so how to design and build a question-and-answer system about movies is more and more urgent and important. At the same time, the original intention of the knowledge graph is to express various entities, concepts that exist objectively in the real world, and various relationships between them. It can better understand the abstract components in the language, and better to confirm the information through the interaction of the ambiguous parts of the language. In addition, the upgrade of deep learning natural language processing technology allows us to better understand and analyse language. In view of the above advantages, the movie question answering system in this paper is constructed based on the knowledge graph and uses natural language processing technology.

Keywords: Question and Answer System, Knowledge Graph, Natural Language Processing.

1. Introduction

The development of the Internet and the development of the film industry have led to more and more film information, and the way people obtain film information has also changed from offline to online, and there are more and more channels. At present, the wide application of knowledge graphs provides a more detailed way to build a question answering system for movies to help the movie industry. [1]

As an advanced version of the information retrieval system, the question answering system can give a concise and accurate answer based on the user's natural language questions, so that people can obtain information that meets their needs faster and more accurately. At present, the question answering system integrating natural language processing and artificial intelligence technology is a popular research direction. As a large-scale knowledge base, knowledge graph can query some complex and related information with higher quality, and at the same time deeply understand the user's intention, so as to achieve the purpose of improving search quality. The rapid expansion of information and data, and the advantages of knowledge graph-based expressions that are more similar to human cognition and management of massive data, make knowledge graphs more widely used in various scenarios.

The definition of the term knowledge graph is unanimously recognized that it was proposed by Google in 2012, and its purpose is to provide a knowledge base for search engines to expand functions and improve performance. Today, the purpose of knowledge graph is no longer limited to search engines, but to represent concepts, entities, events and the relationships between them that really exist in the objective world.

Knowledge graphs can be roughly divided into two categories based on knowledge fields, one is common sense knowledge graphs, and the other is encyclopaedic knowledge graphs. Common knowledge graphs are NELL, KnowItAll, and WordNet. Encyclopaedia knowledge graphs include Yago, Freebase, Wikipedia, Probase, etc.

This system aims to build an AI question answering system based on movie knowledge graph. The system is divided into three parts as a whole: (1) Data acquisition: The open source and widely used IMDB database is used. (2) Construction of knowledge graph: Store the IMDB movie data in the Neo4j graph database according to the relationship. (3) Question and answer processing: It is divided into text segmentation, related word search and question and answer matching. Use two pre-trained models for natural language processing, one for text segmentation and the other for finding related words, matching question templates, data from a graph database, and naive Bayesian classification to find the answer. The realization of the system adopts Python language.

This AI question answering system solves the current demand for a large number of movie problems based on the movie knowledge map, provides a more intelligent and convenient way to give feedback, and reduces the cost of manual response and manual query.

2. Introduction of Key Technologies

This part is an overview of the key technologies required during the development of the system. This question answering system is an automatic WEB system based on the movie knowledge graph. In terms of data, it is based on the open-source graph database Neo4j to build and store the movie knowledge graph. At the same time, as a Chinese natural language processing system, it uses the TF-IDF model to extract keywords and process text. The similarity algorithm is cosine similarity, and the classifier in the machine learning required by this system is implemented using Naive Bayes.

2.1. Knowledge Graph

The official definition of knowledge graph is a knowledge base that it and Google use to improve the quality of their search engines. Therefore, it can be seen that the earliest definition of knowledge graph is Google. In essence, the knowledge graph is not only limited to a certain company, but the main purpose is to use a cognitive method similar to that of human beings to represent various and various types of entities that actually exist in the objective world. concepts and the relationships that exist within them. It can be visualized that the knowledge graph is a very large semantic network, and the nodes in this network represent concepts or entities, and the relationships in these entity concepts constitute the edges in this large network. At present, due to the massive expansion of information and the development of the Internet, the definition of knowledge graph is also somewhat different, and it has already represented various large-scale knowledge bases in a broad sense. [2]

A knowledge graph consists of some interconnected entities and their properties. In other words, the composition of the knowledge graph is a piece of knowledge, and each of the knowledge can be represented as an SPO triple.

However, the construction of a large-scale application knowledge base requires the support of various technologies, including (1) knowledge extraction technology, which can be used to obtain information including knowledge elements including entities, relationships, attributes, etc.;[3] (2) knowledge representation technology, which uses some means to represent the knowledge elements that make up the knowledge graph;[4] (3) knowledge fusion, which aims to eliminate and reduce ambiguity, thereby producing high-quality Knowledge base;[5] (4) Knowledge reasoning is based on the existing knowledge base, digging the hidden knowledge, and then obtaining a more comprehensive and valuable knowledge base.[6]

2.2. Neo4j Graph Database

A graph database is a NoSQL database designed and implemented on the basis of graph theory in mathematics. Unlike traditional relational databases, which store data in database table fields, graph databases store data and relationships between data in nodes and edges. This is the concept of "nodes" and "relationships" in graph databases.

The biggest advantage of graph database is reflected in the retrieval of data relationships. If the relationship between the data is very complex, the data exists in multiple tables, and there are some intermediate tables, the traditional database wants to query some data only through various join table operations, and the SQL will be very complicated to write. Conducive to maintenance, while performance is not high. The graph database can realize the query function only through a simple cypher statement.

2.3. Naïve Bayes Classifier

The Naive Bayes algorithm is a commonly used algorithm in probability problems. The key is: the basis for category determination is the size of the posterior probability. Naive Bayes classification is based on Bayes' theorem, and assumes that the features are conditionally independent of each other. [7] The premise is that on a given training set, the feature attributes are independent, and the joint from X to Y is learned. Probability distribution, and then use the learned model to input the predicted variables to obtain the output with the largest posterior probability.

3. System Analysis

3.1. System Requirements Analysis

The development of the film industry has made people pay more and more attention to movies, and the question and answer for the film industry is gradually increasing. Simple question and answer can no longer provide users with a fast and comfortable experience. Due to the complexity of natural language, how to correctly and accurately understand the user's intention and how to obtain the correct answer from the huge data is still a challenge.[8]

This system designs an automatic question answering system. Through the user's manual input of questions, the algorithm will gradually decompose the user's question text, match the question template, generate query statements, and then match them in the graph database, and then get the correct answer.

3.2. Overall System Design

3.2.1. System Architecture Design. According to the demand analysis, the system needs to build a movie knowledge graph and process user requests from the browser, so the system framework uses the B/S mode. It is divided into presentation layer, business layer and database layer. The presentation layer is mainly to design and display the interface of the system. The business layer is aimed at the user's request, and corresponds to different business modules and handles different businesses, so that the system functions can be performed normally. The database layer is to import the IMDB movie data into the Neo4j database.

3.3. System Main Function Module Design

The whole system is divided into six modules.

3.3.1. Movie Knowledge Graph Storage Module. The movie knowledge graph storage module mainly builds a graph database of knowledge graphs based on IMDB movie data.

3.3.2. Web Module. The Web module completes the function of obtaining the question text input by the user from the front-end page and passing it to the back-end, and feeding the generated answer back to the front-end interface. This module is divided into two major functions: front-end interface and back-end service.[9]

3.3.3. Question Preprocessing Module. The main function of the problem preprocessing module is to preprocess the problem content passed from the front end, and perform typo correction, word segmentation tagging, and key information extraction on the problem input by the user, so as to better understand the problem text later. This module is mainly divided into the following functions: typo correction, Chinese word segmentation, part-of-speech tagging, and abstract questions.

The Chinese word segmentation module is crucial. The problem preprocessing module needs to uniformly encode the original natural language processing. It needs to extract the key information and remove the worthless content. Before this, the text needs to be segmented and long sentences are cut into short and meaningful words.[10] After that, it is tagged with part of speech to better understand the meaning of the word.

Secondly, part-of-speech tagging is also an essential part of natural language processing. There are two common ways of part-of-speech tagging: one is a dictionary query based on string matching, and the other is based on statistics. Generally, the sentence needs to be segmented first, and then the part-of-speech tagging is performed.[11]

After the word segmentation of the question content, it is necessary to further mark the part of speech to mark information such as people, movie names, etc., so as to facilitate the later semantic understanding.

After a series of word segmentation, part-of-speech tagging and other operations are performed on the question text, key information needs to be extracted for subsequent question classification. This process can be called feature extraction, which provides runnable attributes for subsequent classification and query statements. Key information extraction can be analogized to keywords in paper abstracts. For keyword extraction, supervised learning and unsupervised learning can be adopted. This system uses the TF-IDF model to sort according to the TF-IDF value, and selects the first few words of the corresponding number as keywords.[12]

3.3.4. Problem Intent Understanding Module. The main function of the question intent understanding module is to classify the text that has been preprocessed by the question, generate a trained model according to the question training set, and then classify the successfully trained model. The main functions of this module include designing and constructing problem templates, generating problem training sets, and using Naive Bayes classifiers to predict user input problems.[13]

3.3.5. Query Statement Module. The function of the query statement module is mainly to understand the problem classification generated by the problem intent module, corresponding to the problem template, and generate a query Cypher statement in the graph database.

3.3.6. Answering Generation Module. The answer generation module is mainly to connect to the graph database Neo4j, generate a query statement before input, match the data in the knowledge graph, and get the answer. At the same time, the answer is fed back from the backend to the frontend.

4. System Test

For a complete system implementation process, software testing is a key tool for system maintenance and inspection. Therefore, before the system is put into use, it is essential to conduct sufficient and complete testing of the system. When testing the system, some test cases are artificially designed. Detect whether the system meets user needs from the perspective of function and non-function, and correct the problems in time to achieve the original needs.

4.1. System Test Content

This system test mainly starts from two aspects. On the one hand, functional testing is mainly aimed at the main functions proposed in the requirements to test whether the system can operate normally, including the testing of functions such as input, query, and text correction. On the other hand, the non-functional testing of the system mainly covers performance testing such as accuracy and reliability testing.

4.2. Test Results

System requirements are fully tested from both a functional and non-functional perspective. The first is the page responsiveness performance test. The user enters text, presses the send button, and the response time to the request should be within 1.5s, which meets the predetermined requirements. The second is the system reliability test. The user enters text in the text box and clicks the send button, and the correct answer is returned, which meets the requirements. Then there is the accuracy test, which tests 100 questions, and the accuracy reaches more than 90%, which meets the requirements.

5. Summary and Outlook

5.1. Summarize

This paper constructs a question answering system about movie knowledge graph. According to the typical development process, the system shows the realization process of an automatic question answering system based on knowledge graph in the film field. The system adopts B/S mode WEB architecture, developed in Python language, designed and implemented an interactive and concise user interface. Taking into account the performance issues of knowledge graph query accuracy and efficiency, the database adopts the open source Neo4j graph database, which can not only meet the requirements of data storage, but also meet the requirements of performance.

To sum up, this paper uses the related technologies of constructing and designing knowledge graphs, and uses knowledge related to natural language processing to design and implement an automatic question answering system. According to the implementation specifications of software projects, this paper presents a specific example of the design and implementation of an automatic question answering system.

5.2. Outlook

（1） The system does not support speech recognition, and related technologies can be added in the future to design a more diverse system.

（2） To expand the data to more databases, you can try to build an encyclopaedia question answering system.

References

[1] Xianling Mao，Xiaoming Li. A Survey of Question Answering System Research. Computer Science and Exploration，2012（3）

[2] Wei Xia，Shanlei Wang，Zidu Yin，et al. Modeling and Completion of Entity Relationships in Knowledge Graph Based on Mutual Information. Computer Science and Exploration，2018，v.12;No.118(7):49-59

[3] Steiner T, Verborgh R, Gabarro J, et al. Adding real-time coverage to the Google knowledge graph[C]//The International Conference on Posters & Demonstrations Track. 2012

[4] Weiguo Zheng, Hong Cheng, Jeffery Xu Yu, Lei Zou, Kangfei Zhao. Interactive Natural Language Question Answering Over Knowledge Graphs. Information Sciences, 2018, 12(32):41-42

[5] Junwei Bao. Research on knowledge-based automatic question answering and question generation. Harbin：Harbin Institute of Technology，2019.

[6] Bo Zhang，Yupeng Jin，Qian Zhang，et al. On a new type of online educational resources big data organization framework. China Electronic Education，2018

[7] Mishra A, Jain S K. A survey on question answering systems with classification. Journal of King Saud University – Computer and Information Sciences, 2016, 28(3): 345-361

[8] Peiyan Li，Lu Zhu，Duosheng Wu. A review of question answering systems. Digital Technology and Applications, 2015(4):69-69

[9] Yassine EI Adlouni, Horacio Rodriguez, Mohammed Meknassi, et al. A multi-approach to community question answering. Expert Systems with Application, 2019, 7(13):432-442

[10] Zhao Yan, Nan Duan, Junwei Bao et al. DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents. ACL, 2016

[11] Template based Question Answering over RDF Data. UNGER C, BUHMANN L, LEHMANN J, et al. Proceedings of the 21st International Conference on World Wide Web. 2012

[12] Sang-Woon Kim, Joon-Min Gill. Research paper classification systems based on TF-IDF and LDA schemes. Human-centric Computing and Information

[13] Hien T. Nguyen, Phuc H. Duong, Erik Cambria. Learning short-text semantic similarity with word embeddings and external knowledge sources. Elsevier Journal, 2019, 7(13)