Analysis Ontology-Based Simple Closed Domains Question-Answer Application

Ade Hodijah, Urip Teguh Setijohatmo, Jonner Hutahaean, Ade Chandra Nugraha, Naufal Dalhats Zhalifunnas

Computer and Informatics Engineering
Politeknik Negeri Bandung
Bandung, Indonesia
ade.hodijah@polban.ac.id

Abstract. Time performance and precision became crucial factors in question-answer applications. This research compares two ontology-based models of the simple question-answer (QA) application. The domain of the application is movie information. The first model used lightweight natural language processing (NLP) and a detail movie ontology combination (shallow NLP movie ontology). The other designed lightweight movie ontology was suitable with the IndoBERT mechanism (IndoBERT movie ontology). The research used a run-time experiment to get the best performance model using the IMDB dataset. The query processing experiment controlled the amount of data, while the precision experiment was based on the output generated by the Shallow NLP movie ontology and the IndoBERT movie ontology. The results show that the Shallow NLP movie ontology was better for the execution times and more precision (as long as supported by valid ontology instances for all of the questions) than the IndoBERT movie ontology. In contrast, the precision of IndoBERT movie ontology was more powerful NLP than the Shallow NLP movie ontology. This research is expected to provide an empirical contribution regarding building and developing a simple QA application and ontology for closed domains.

Keyword: Question-answer application; movie ontology; IndoBERT.

1 Introduction

A way to find the desired information is to utilize the Question-Answer (QA) application. Generally, a QA system consists of three main modules: question processing, document retrieval, and answer processing. A QA application receives queries in the form of natural language questions. Then, look for answers in a set of documents or the knowledge base of a domain. Finally, formulate a concise response [1].

Most QA application group questions are based on the type of question [2]. If the kind of question can be determined, then the type of answer can also be determined. For example, if the question is "Who...", the desired response is a person or organization. If the question is "When…" the expected answer is a time or date. This research focuses on data mode analysis of a QA application that can provide relevant information from an ontology-based knowledge base in response to queries provided by users in natural language (NL).

M. U. H. Al Rasyid and M. R. Mufid (eds.), *Proceedings of the International Conference on Applied Science and Technology on Engineering Science 2023 (iCAST-ES 2023)*, Advances in Engineering Research 230,
https://doi.org/10.2991/978-94-6463-364-1_111

Utilizing an ontology to represent its knowledge base was necessary to improve the quality of a QA application. Ontology connects symbols that humans understand with their forms that machines can process. Thus, ontology bridges the gap between humans and machines [3].

The current trend in QA research is towards open domain QA, which is based on many documents on the web using IndoBERT. In contrast to this trend, several studies focus on closed-domain QA research [4], [5][6]. The choice of a closed domain is based on several reasons, including, first, exploitation of information in web documents is often faced with the problem of the reliability of that information. The information provided may need to be updated or corrected. Second, using formal knowledge in closed domains can increase the accuracy of the QA system because both questions and answers are analyzed based on this knowledge base. Third, an institution may have and manage a limited knowledge base and only use it within its scope.

Table 1 shows the research gap in this paper. The similarities lie in the use of movie domain [7], closed QA [8], [4], IndoBERT in NL processing (NLP) [9], [4], NLP framework [8], and the way of representing the knowledge base (ontology) with data management [8]. Otherwise, the differences lie in expressing the knowledge base of movie-ontology and IndoBERT combination using data management solutions for QA closed domains.

TABLE I. SIMILAR SCIENTIFIC WORKS.

Year	Author	Natural Language	Knowledge Representation
2010	Kristoforus Jawa Bendi	Indonesian	Ontology
2019	M Syarief et al.	Indonesian	Ontology
2021	M. Indah Rahajeng et al.	English	Knowledge graph, IndoBERT
2022	Thiffany	Indonesian	Sequence labeling tasks, IndoBERT

2 Methods

This paper applies a comparative study using two prototypes having different ontology-based data models, as seen in Fig. 1. A simple QA application was built using Python to compare the precision and the performance of the Shallow NLP and the IndoBERT movie ontology with the Django framework, the Owlready2 library to import SPARQL function [8], and the transformers library for IndoBERT. The movie ontology was managed using Protégé.

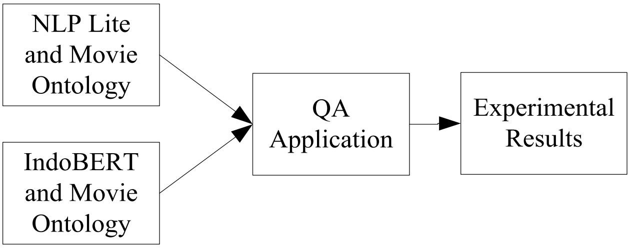

Fig. 1. Research methods.

This paper used one movie ontology for a simple question-answer application, which used two models called Shallow NLP movie ontology and IndoBERT movie ontology. The application follows the models and undergoes the NLP preprocessing. The movie ontology used was from [7], adding the rangkuman data property in Movie classes. In this research, the movie ontology ignores inference cases that require changing the SPARQL query.

The movie ontology consisted of the Person and Movie concept, as seen in Fig. 2. The Movie class was related to the Person class through four properties: director, cast, writer, and producer. These four relationships (Object property) explain the Persons who act as directors, actors, screenwriters, and producers of a Movie. The Person has a name and gender, while the Movie has the title, location, and rangkuman as Data property. Generally, BERT works on one source text (summaries put together) [11], but it will be slow. Hence, the IndoBERT movie ontology speeds it up by organizing through the formation of an Ontology, which requires summary property data as a container for the source text to answer questions. So, the preprocessing results are mapped to an entity (Movie class), and BERT only works on a specific summary based on the name of the film title from the ontology created. Question processing follows this model and undergoes preprocessing such as tokenizing, etc.

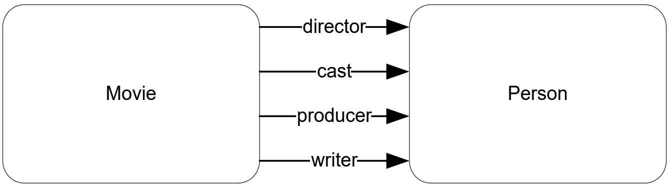

Fig. 2. Movie ontology snippet.

The two models were tested by two parameters, which were execution times and precision using the IMDB dataset [10] and a question list from [7]. The total execution times were the average of 10 times testing measured in ms. The experiment was tested in a single server environment using a PC with Intel (R) Core (TM) specifications i5-8250U CPU @ 1.60GHz 1.80 GHz uses 12.0 GB RAM and uses the operating system MS Windows 10.

The five-question closed domain QA application used question words of who and where (location) as single question sentences to limit the ontology of the Shallow NLP. Meanwhile, the IndoBERT processed a compound sentence, not only a single sentence [11]. The five questions are as follows [7]:

1. Who produced the Titanic film? (object property: producer)
2. Who is the director of the Harry Potter films? (object property: director)
3. Who wrote the screenwriter for the Titanic films? (object property: writer)

4. Who are the players in the Slumdog Millionaire films? (object property: cast)
5. Where the Saving Private Ryan was made? (data property: location)

The execution times of query processing are critical in question-answer applications because the longer the process takes, the higher the user's possibility of abandoning the application [12]. Besides the execution times, the important thing about a search engine is its usefulness. A search engine's effectiveness level is seen by how relevant the search results produced are [13]. The story of relevance of search results can be determined by calculating their precision. Two factors influencing accuracy in this paper are the movie ontology and the NLP library for the Bahasa language.

Precision is the ratio of the number of relevant documents retrieved to the total number of documents retrieved [14]. This precision is used to measure the quality of search results [15]. The following is the formula for calculating the percentage of search engine precision [14]:

$$\text{Precision} = \frac{\textit{Number of relevant news retrieved}}{\textit{Number of news retrieved in search}} \ldots (1)$$

Assume that the application yields 100 responses, but only one is relevant (a fact-based response to the question) answer. The precision value in that scenario is 0.01 or 1% [14]. The higher precision % determined the higher question-answer application quality.

3 Results And Discussion

Fig. 3 shows the movie ontology used by the Shallows NLP movie ontology [7], while Fig. 6 shows the movie ontology used by the IndoBERT movie ontology.

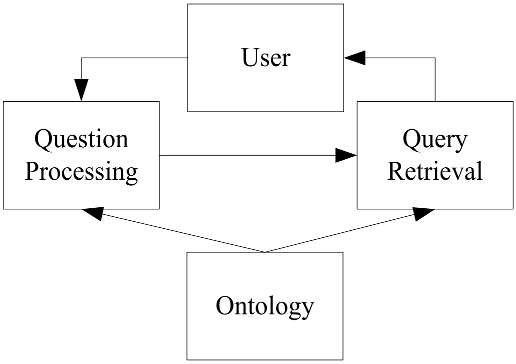

Fig. 3. Ontology-based QA application model of the Shallow NLP movie ontology [7].

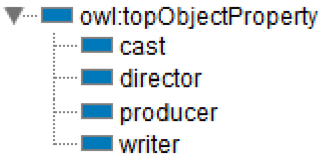

Fig. 4. Protégé object properties.

The difference between the two models is in the class hierarchy. The NLP Lite movie ontology used a Question Processing module to check question sentence validity and get question keywords [14] after stopping word removal using a shallow NLP library. Afterwards, the Query Retrieval module used the keywords to generate SPARQL query statements [7]. The Query Retrieval module used the keywords to generate SPARQL query statements [7]. Fig. 4 shows the object properties of the Shallow NLP movie ontology. However, to answer the IndoBERT movie ontology, the data property called rangkuman is needed to save all movie information in text description form, as seen in Fig. 5.

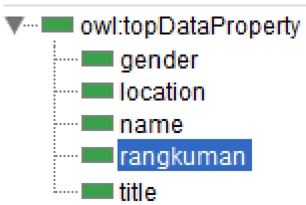

Fig. 5. Protégé data properties

Each movie had one the rangkuman consisting of all movie information: director, cast, producer, and writer. Because IndoBERT would give the answer questioned by the user obtained from one collected text description [9], while in the Shallow NLP movie ontology, the answer is divided into pieces of text located separately in each movie information.

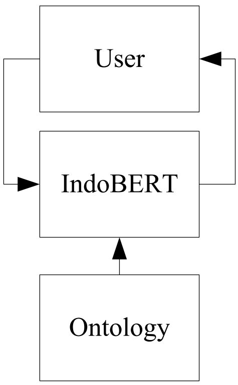

Fig. 6. Ontology-based QA application model of the IndoBERT movie ontology.

Resources for rangkuman were generated from ChatGPT [16]. The examples of the rangkuman are seen in Fig. 7.

Fig. 7. The ChatGPT example asserted in Protégé data property

Table 2 shows the experimental results for all questions tested by the two models. The execution times of the Shallow NLP movie ontology was in the tens, mostly less than 20, while the IndoBERT movie ontology needed thousands of time. Otherwise, the IndoBERT movie ontology was 100% more precise than the Shallow NLP movie ontology. It was because the movie ontology has the wrong description for the movie director, as seen in Fig. 8, and the invalid answer result, as seen in Fig. 10, where four directors were retrieved by application (25% precision). It should only get one director (100% of precision), Chris Columbus, as seen in Figure 11.

TABLE II. EXPERIMENTAL RESULTS.

Question Number	Shallow NLP Movie Ontology		IndoBERT Movie Ontology	
	Precision (%)	Execution Times (ms)	Precision (%)	Execution Times (ms)
1	100	13.96	100	1640.61
2	25	17.95	100	533.57
3	100	15.55	100	2016.13
4	100	16.95	50	894.61
5	100	27.90	100	1695.47

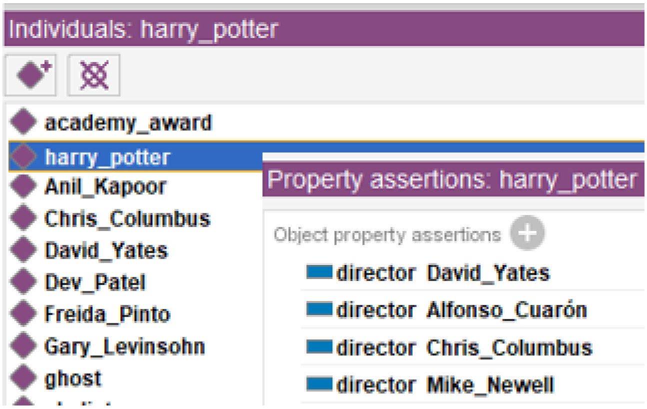

Fig. 8. The unwell defined property assertion for question number two of the Shallow NLP movie ontology.

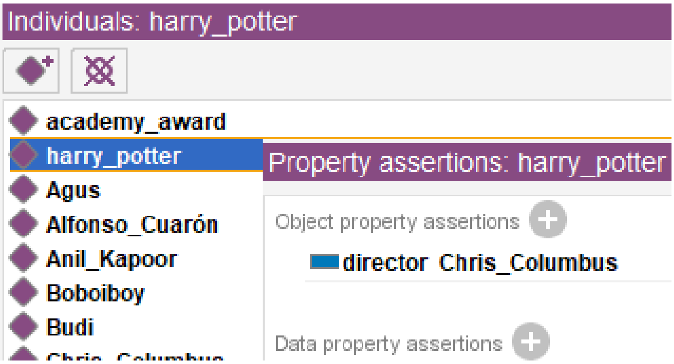

Fig. 9. The well defined property assertion for question number two of the Shallow NLP movie ontology.

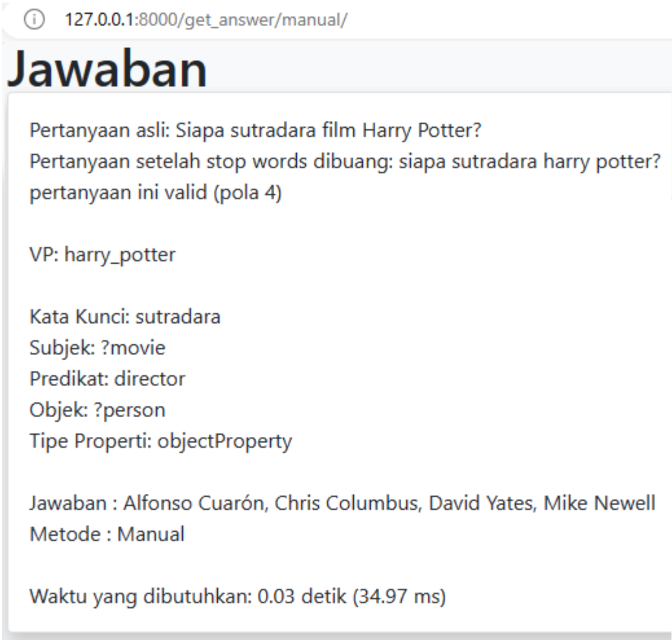

Fig. 10. The invalid answer result for question number two of the Shallow NLP movie ontology.

After redefining the movie director in the movie ontology, as seen in Fig.9, the answer was valid, as seen in Fig. 11, which was identical to the IndoBERT movie ontology result, as seen in Fig. 12.

To get the answer to the question “Who is the director of the Harry Potter films?”, the Shallow NLP movie ontology divided the sentences into three keywords: (1). Who; (2). Director; (3) Harry Potter. Based on these keywords, the SPARQL statement was created, matched each derived keyword, and searched manually from the ontology structure provided. Whereas, the IndoBERT movie ontology only needed a keyword (value property or VP or title of movie or individuals in protégé), which was Harry Potter, to get its rangkuman. Therefore, there was only VP in the IndoBERT movie ontology (Fig. 12 and Fig. 14). There were VP, keywords, subject, predicate, object, and property types in the Shallow NLP movie ontology (Fig. 11 and Fig. 13). For example:

- Question: Who is the director of the Harry Potter films?
  - Keywords: who the director of Harry Potter
  - Subject: Movie
  - Predicate: director
  - Object: Person
  - Property types: object property
- Question: Where the Saving Private Ryan made?
  - Keywords: where was Saving Private Ryan made
  - Subject: Movie

- Predicate: location
- Object: Movie
- Property types: data property

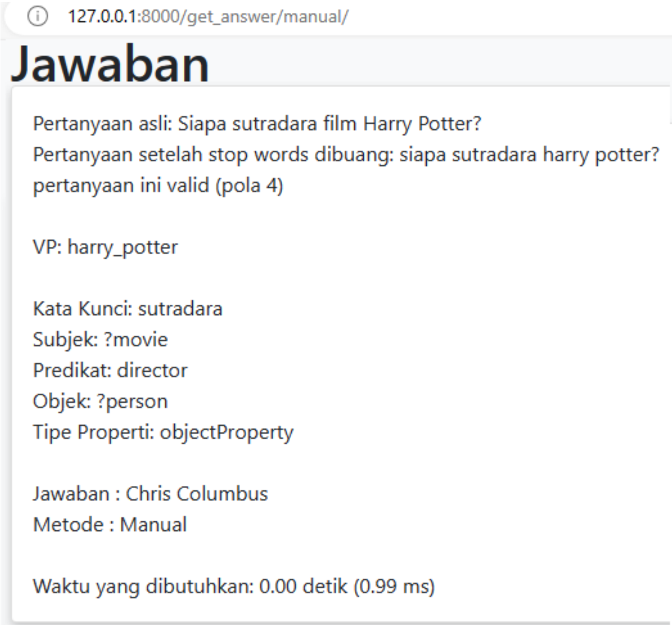

Fig. 11. The valid answer result for question number two of the Shallow NLP movie ontology.

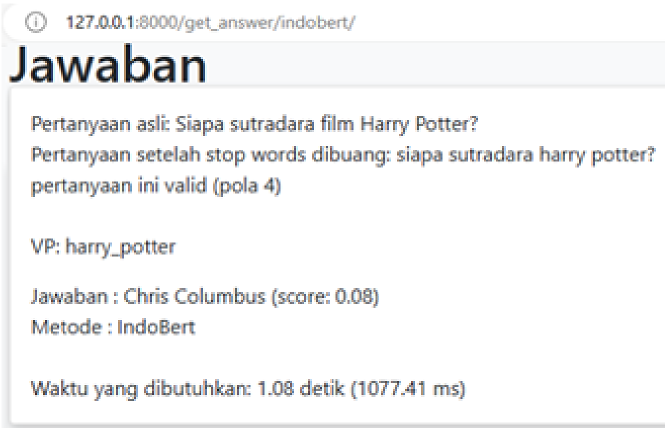

Fig. 12. The answer result for question number two of the IndoBERT movie ontology.

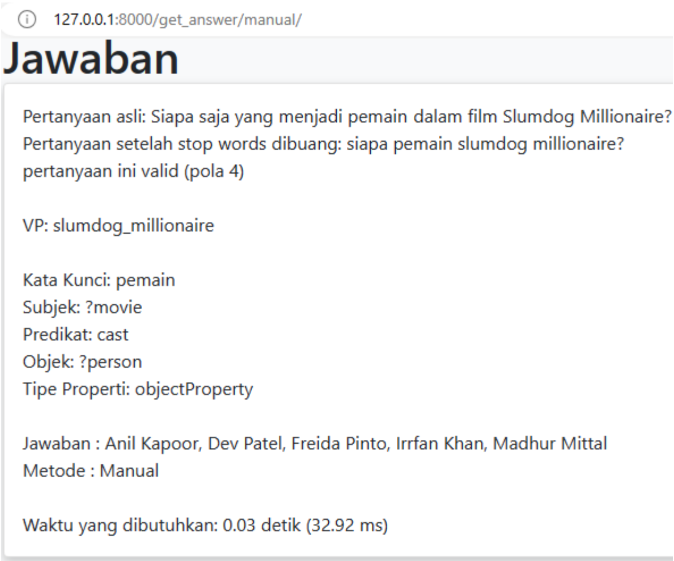

Fig. 13. The answer result for question number four of the Shallow NLP movie ontology.

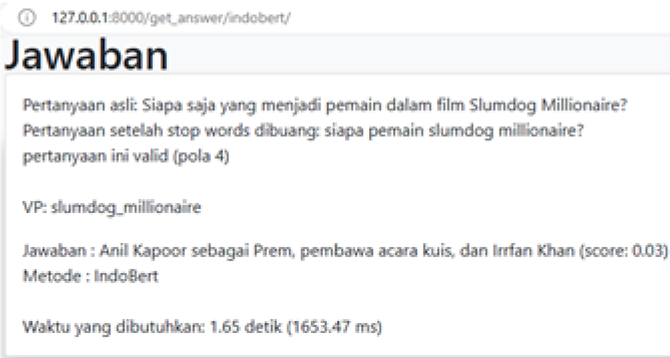

Fig. 14. The answer result for question number four of the IndoBERT movie ontology.

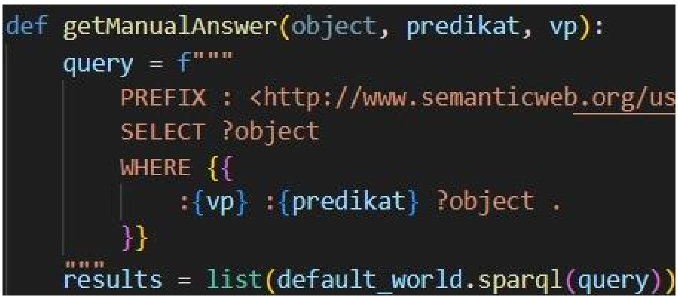

Fig. 15. A code snippet for the Shallow NLP movie ontology.

However, the IndoBERT resources were generated by ChatGPT. As long as the data was complete (all answers to all questions are available) and valid, the answer would also be correct. It happened because of the question, "Who are the players in the Slumdog Millionaire?" [10]. The Shallow NLP movie ontology had 100% precision (Fig. 10). In comparison, the IndoBERT movie ontology had 50% of accuracy (Fig. 11). It would have 100% of accuracy if the rangkuman had valid information about the movie players [17].

```
def getAutomaticAnswer(vp):
    query = f"""
        PREFIX : <http://www.semanticweb.org/us

        SELECT ?object
        WHERE {{
            :{vp} :rangkuman ?object .
        }}
    """
    results = list(default_world.sparql(query))
```

Fig. 16. A code snippet for the IndoBERT movie ontology

A threshold for confidence score in IndoBERT had to be determined in supporting the question's relevant answer [11]. For example, the threshold for response to question two was 0.08 (Fig. 12), while for question number four was 0.03 (Fig. 14). The result was 100% for question two and 50% for question four. Nevertheless, sometimes answers considered less relevant still had a confidence score that exceeded the threshold value [17]. Moreover, each BERT model has a different level of accuracy [18]. It causes the model to be trained independently to be more accurate. It would be better if an in-depth analysis were carried out regarding how a threshold was obtained and what its relationship is with the use of the ontology that had been created.

4 Conclusions

This research experimented with an ontology-based QA application using a movie dataset that can process single-question sentences. The Question Processing and Query Retrieval algorithm required further research for compound sentence questions. The study result showed that the Shallow NLP movie ontology for closed domain QA application was a hundred times faster than the IndoBERT movie ontology. Print the properties (object and data) after the node was found in the Shallow NLP movie ontology. Meanwhile, in the IndoBERT movie ontology, there was still an IndoBERT set process from the summary (the rangkuman data property) after the node met.

The IndoBERT movie ontology and the Shallow NLP movie ontology could have a similar precision. As long as the Individuals of protégé were valid in the Shallow

NLP movie ontology and the more question keywords arranged (to be a deep NLP movie ontology) in the Classes, Object properties, and Data properties of Protege, then the more the number of precision answers obtained (Individuals in Protégé). Otherwise, the more complete, valid, and well-managed the data set in the rangkuman data property, the more errors in the answers to IndoBERT movie ontology will be avoided. Therefore, it is necessary to develop a movie ontology by creating special classes (placed in the specified data property) to handle two types of data in one question, for example, "When (Date) was the film Titanic released and how long (Integer) is the duration of the film Titanic?". It is because IndoBERT movie ontology only processes the rangkuman data property in the form of descriptive text (String) and will display the first answer obtained after question keyword valid checking. Since the Individuals inputed manually in the Shallow NLP movie ontology using Protégé, how to map the data to ontology dynamically would support the data validity. Meanwhile, the IndoBERT movie ontology used ChatGPT to create data dynamically. Thus, the IndoBERT could be improved by the NLP retraining.

References

1. A. Dhandapani and V. Vadivel, “Question Answering System over Semantic Web,” *IEEE Access*, vol. 9, pp. 46900–46910, doi: 10.1109/ACCESS.2021.3067942.(2021).
2. FT, Mhaisen, and A. Et, “A Simple Question Answering System,” *Angew. Chemie Int. Ed. 6(11), 951–952.*, vol. 13, pp. 10–27, (2018).
3. D. Melo, I. P. Rodrigues, and V. B. Nogueira, “Cooperative question answering for the semantic web,” *KMIS - Proc. Int. Conf. Knowl. Manag. Inf. Shar.*, no. November, pp. 258–263, 2011, doi: 10.5220/0003691202580263.(2011)
4. M. Indah Rahajeng and A. Purwarianti, “Indonesian Question Answering System for Factoid Questions using Face Beauty Products Knowledge Graph,” *J. Linguist. Komputasional*, vol. 4, no. 2, p. 59, doi: 10.26418/jlk.v4i2.62.(2021)
5. V. López, M. Fernández, E. Motta, M. Sabou, and V. Uren, “Question Answering on the Real Semantic Web,” pp. 2–4.
6. P. Atzeni *et al.*, “Ontology-based question answering in a Federation of University Sites: The MOSES case study,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3136, no. July, pp. 413–420, doi: 10.1007/978-3-540-27779-8_40.(2004)
7. K. J. Bendi, “Sistem Question Answering Sederhana Berbasis Ontologi Sebagai Aplikasi Web Semantik,” *Call Pap. Munas Aptikom*, no. January 2010, pp. 187–191, 2010,[Online].Available:https://www.researchgate.net/publication/262923898_SISTEM_QUESTION_ANSWERING_SEDERHANA_BERBASIS_ONTOLOGI_SEBAGAI_APLIKASI_WEB_SEMANTIK(2010).
8. M. Syarief, W. Agustiono, A. Muntasa, and M. Yusuf, “A Conceptual Model of Indonesian Question Answering System based on Semantic Web,” *J. Phys. Conf. Ser.*, vol. 1569, no. 2, 2020, doi: 10.1088/1742-6596/1569/2/022089(2020).
9. T. Pratama and Suharjito, “IndoXLNet: Pre-Trained Language Model for Bahasa Indonesia,” *Int. J. Eng. Trends Technol.*, vol. 70, no. 5, pp. 366–380, 2022, doi: 10.14445/22315381/IJETT-V70I5P240(2022).

10. I. Developer, “IMDb Non-Commercial Datasets.” https://developer.imdb.com/non-commercial-datasets/.
11. F. Koto, A. Rahimi, J. H. Lau, and T. Baldwin, “IndoLEM and IndoBERT: A Benchmark Dataset and Pre-trained Language Model for Indonesian NLP,” no. November, pp. 757–770, 2021, doi: 10.18653/v1/2020.coling-main.66(2021).
12. A. I. Solutions, “The Importance of Website Loading Speed & Top 3 Factors That Limit Website Speed.” https://www.aykira.com.au/2014/04/importance-website-loading-speed-top-3-factors-limit-website-speed/(2014).
13. “Search engine - Wikipedia.” https://en.wikipedia.org/wiki/Search_engine (accessed Aug. 20, 2022)(2022).
14. N. P. Lestari and U. Airlangga, “Uji Recall and Precision Sistem Temu Kembali,” (2016).
15. B. Robert and E. B. Brown, “Google’s PageRank and Beyond: The Science of Search Engine Rankings,” no. 1, pp. 1–14, (2004).
16. OpenAI, “ChatGPT.” https://openai.com/chatgpt.
17. N. D. Z. Chikita Salsabiil Cendikia Negarawati, Citra Mutiara Budiman, “Pengembangan Knowledge Management System Kopi Berbasis Semantic Web Dengan Penerapan IndBERT Pada Mesin Pencarian,” (2023).
18. KDnuggets, “Which flavor of BERT should you use for your QA task?” https://www.kdnuggets.com/2020/10/flavor-bert-use-qa-task.html(2020).