A New Approach in Query Expansion Methods for Improving Information Retrieval

Lasmedi Afuan[1], Ahmad Ashari[2], Yohanes Suyanto[3]

[1]*Department of Informatics, Engineering Faculty, Universitas Jenderal Soedirman, Purwokerto*
[2,3]*Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta*
[1]lasmedi.afuan@unsoed.ac.id, [2]ashari@ugm.ac.id, [3]yanto@ugm.ac.id

Abstract - This research develops a new approach to query expansion by integrating Association Rules (AR) and Ontology. In the proposed approach, there are several steps to expand the query, namely (1) the document retrieval step; (2) the step of query expansion using AR; (3) the step of query expansion using Ontology. In the initial step, the system retrieved the top documents via the user's initial query. Next is the initial processing step (stopword removal, POS Tagging, TF-IDF). Then do a Frequent Itemset (FI) search from the list of terms generated from the previous step using FP-Growth. The association rules search by using the results of FI. The output from the AR step expanded using Ontology. The results of the expansion with Ontology use as new queries. The dataset used is a collection of learning documents. Ten queries used for the testing, the test results are measured by three measuring devices, namely recall, precision, and f-measure. Based on testing and analysis results, integrating AR and Ontology can increase the relevance of documents with the value of recall, precision, and f-measure by 87.28, 79.07, and 82.85.

Keywords: query expansion, association rules, ontology, recall, precision, f-measure.

I. INTRODUCTION

The number of documents on the Internet has increased exponentially, users uploaded document to the Internet every minute and even seconds. It raises problems for users, namely how to find documents that are relevant to the query. The use of Information Retrieval (IR) is an alternative that can use to overcome these problems. There are two important developments carried out in IR Research, namely, how to index documents and how to retrieve documents relevant to user queries[1]. IR research carried out at different levels but with the same goal of increasing the relevance of the document retrieval. IR research carried out at different levels, but with the same goal of increasing the relevance of documents retrieval, such as [2] adapt the classic VSM model to IR based on ontology, researchers [3] propose the use of ontology in the indexing process which believed to be able to increase the relevance of documents, besides, researchers [4] proposed ontology-based IR, [5] proposed IR using PSO and IR with query expansion [6][7][8][9][10][11]. IR research carried out generally uses keywords in searching document content. Often users are less able to represent the information needs needed in the form of queries. Thus, documents generated by IRs are not relevant to the user's wishes. The number of relevant documents produced depends very much on the query entered by the user. The user query vocabulary that mismatches with documents also causes no documents to be retrieved [8].

A good IR must be able to bridge the potential distance between documents and user queries. [8], to overcome this, research at IR proposes many solutions, one of which is by query expansion[12]. Query expansion (QE) believed to be able to overcome problems related to user query representation. This approach used to overcome problems in the ineffectiveness of document retrieval by expanding queries to improve the accuracy of user queries, which believed that inaccurate queries are the main problems related to document relevance in IR[13]. Research on query expansion carried out with several methods, including relevant feedback [14] by modifying words in a query based on the distribution of words in relevant and irrelevant documents taken by the initial query. This method can improve document retrieval results. However, this method relies on the top relevant documents. So, if the top document is not good, relevant feedback gives poor results. Another method is Local Co-occurrence, which is a probabilistic method. This method based on the frequency of words appearing in the training corpus. This method has proven to be useful in IR [15], but this method has not been able to understand the connectedness of the appearance of words simultaneously. For example, the word software and hardware, often appear together, because there is no co-occurrence between the two words. Then the relationship between the words is considered non-existent. The relationship problem between words can overcome by the Latent Semantic Indexing (LSI) method [16]. This method is robust, implemented on two types of algorithms, namely in SVD (Singular Value Decomposition) and Probabilistic LSI. LSI builds semantic spaces, maps each term into that space, and groups them automatically based on the meaning of the terms. It's just that with the LSI method, it is difficult to control the degree of query expansion, and it could be that many expanded queries contain irrelevant terms. Research[7] proposes the use of association rules that can overcome the problem of connectedness with the appearance of words simultaneously by mining the appearance of words in the document. However, the use of association rules still chooses limitations in displaying the meaning of words in the document. Researchers [9][17] using WordNet or Thesaurus. WordNet or Thesaurus uses synonyms, homonyms to look for words in common. It is just that using WordNet on some user queries can reduce the performance of IR. In addition to some of the methods mentioned [18][19][20] [10][21] using ontology to expand queries. Ontology can display the meaning of words semantically and is proven to use for query expansion, but it is less able to display the concurrent appearances between words in a document. Based on the previous description and after a literature review, the problem in this study is the limitations of the user in representing the query, which often results in mismatch and miss concept. One solution to this problem can be to use query expansion. In query expansion, several approaches have been proposed. The approach use of the method separately shows the results of query expansion that is over-expansion resulting in queries containing irrelevant words, already displaying the meaning of words but lacking in displaying the interrelationships of joint appearances between words. Therefore, this study proposes a query expansion model on information retrieval in documents by integrating association rules and ontology that believed to be able to overcome the problems in query expansion. The main contribution of this research is the development of query expansion models by integrating the Association Rules and Ontology.

The remainder of this paper is organized as follows in section 2 overview of related work in QE. In section 3, we explain a new approach for QE using Association Rules and Ontology. Discussion and Results in section 4. A small summary and further study of this area conclude in section 5.

II. METHOD

There are several stages in this research, including data collection, ontology development, document retrieval, query expansion using AR, query expansion using ontology, testing, and reporting. The stages are shown in Fig. 1.

The first stage of this research is data collection which is used as input for the developed model. At this stage, the development of Ontology is carried out. The ontology development will be used in two stages, namely (1) document retrieval for the indexing process; (2) The query expansion model uses Ontology. The document retrieval stage is to search the document collection. This stage aims to obtain the initial document to be used as input in the query expansion model. There are several activities carried out at this stage, namely (a) Preprocessing (b) indexing; (c) querying; (d) searching; (e) ranking. In the query expansion model using the AR method, there are three stages in making this model, namely (a) Model building; (b) Model testing; (c) Analysis of the results of model testing. In the query expansion model using Ontology, there are three stages in making this model, namely (a) Model building; (b) Model testing; (c) Analysis of the results of model testing. Model testing is done to measure and analyze the performance of the model that has been developed. The test uses 10 queries which are used as initial input for the model. While the test data is in the form of 100 documents in the field of informatics (journals, proceedings, teaching materials, etc.). This test is carried out on three stages of the model that has been developed, namely (1) the stages of document retrieval; (2) Testing the query expansion model with ontology; (3) Testing the query expansion model using Ontology. The results of the tests performed were analyzed using three measurement methods, namely: recall, precision, and f-measure. In general, recall is the success rate of the system in recovering information.

In Fig. 1, in general the developed model consists of three main sub-models, namely: Document Retrieval, Association Rules, and Ontology. At the document retrieval step, eight processes carried out, namely (1) Document extraction, (2) Tokenization; (3) Stopword Removal, (4) POS-Tagging; (5) Indexing; (6) Querying; (7) Searching; (8) Ranking.

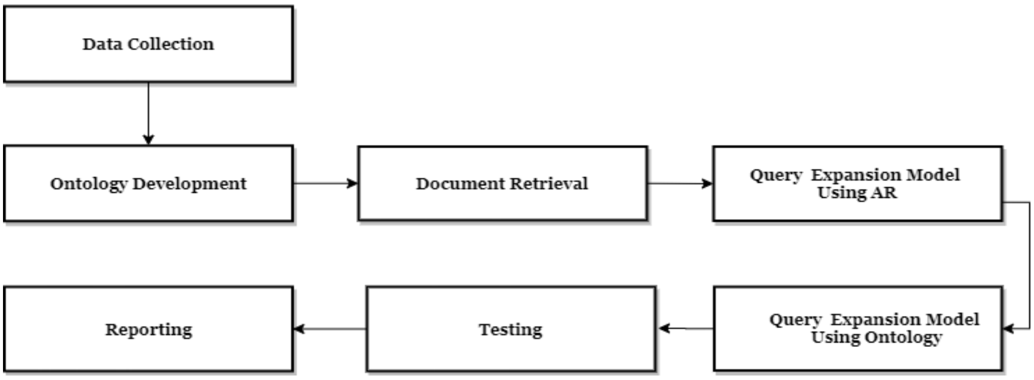


Fig. 1 The stages of the research

III. RESULTS AND DISCUSSION

This study develops a query expansion model that integrates the Association Rules and Ontology shown in Fig. 2. Besides, an overview of the steps of the retrieval of the eight processes shown in Fig. 3.

The document extraction process aims to extract each document into a text format (.txt). It aims to facilitate the pre-processing of documents. The extraction process carried out using third-party software, namely PDFBox, PDFBox embedded into the Python programming. In this process, perform extraction of 100 documents in a text file format with the extension (.txt). Then each document is named DOC001.txt up to DOC100.txt. Tokenization is Process removes punctuation, characters, and numbers in the text. The tokenization process carried out automatically. Tokenization replaces the characters in the token list with spaces when the token list founded in the document collection. The next step is the stopword removal process aims to delete words or terms that often appear in large numbers on the document but do not give meaning to the document. This study uses 910 stopword lists. If the word in the document is in the stopword list, then the word deleted, but if it is not found in the word list in the stopword list, it not deleted. This step repeated until the end of the document. The POS-Tagging process aims to label the class of words in the text, such as nouns, verbs, adjectives, and others. In this study, we using the library of PEBAHASA. In the POS-Tagging process, it uses 35 tag sets. The indexing process aims to build up an index list from documents. The resulting index list serves to represent a collection of documents. As mentioned at the beginning of the chapter, this indexing process includes contributions in this study. The indexing process used an ontology approach to build index lists. Index lists constructed using a contextual approach to semantic information. In general, Fig. 4 shows the steps taken in the indexing process.

The indexing process maps the semantics in the ontology to the words/phrases in the document collection. The ontology structure developed follows the steps proposed by [29]. Semantic entities obtained by a semantic knowledge extraction process carried out in ontology. Semantic entities obtained from ontology extraction processes, each semantic entity is given more than one label. Existing semantic entities, then the semantic annotation process is carried out. Examples of semantic entities shown in Table I. After the semantic annotation performed, the next step of the indexing process is weighting. This weighting using (1).

$$dx = \frac{freqx,d}{maxyfreqy,d} . log \frac{|D|}{nx} \tag{1}$$

The searching process aims to search a collection of documents by matching the query with the collection of documents that carried out in the indexing process. At this step, SPARQL generated from the querying step carried out on the built ontology. The Searching process mapping between semantic entities extracted from user queries mapping into SPARQL with semantic indexes resulting from the indexing process. The ranking process uses the Vector Space Model (VSM) and TF-IDF based on cosine similarity. The ranking carried out on documents generated in the searching process. Each document represented as a vector, where the elements of each vector are the weights of the semantic entity annotations of the documents [2]. The query also represented as a vector, where the element of the vector is the semantic weight of the entity associated with the query variable [4][2] defines the size of the similarity between a document d with the query q as cosine similarity with (2).

$$sim(d, q) = \frac{d.q}{|d||q|} \tag{2}$$

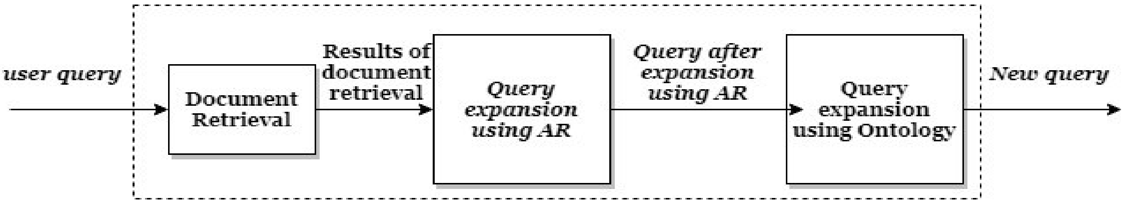

user query
Document Retrieval
Results of document retrieval
Query expansion using AR
Query after expansion using AR
Query expansion using Ontology
New query

Fig. 2 A query expansion model that integrates Association Rules and Ontology

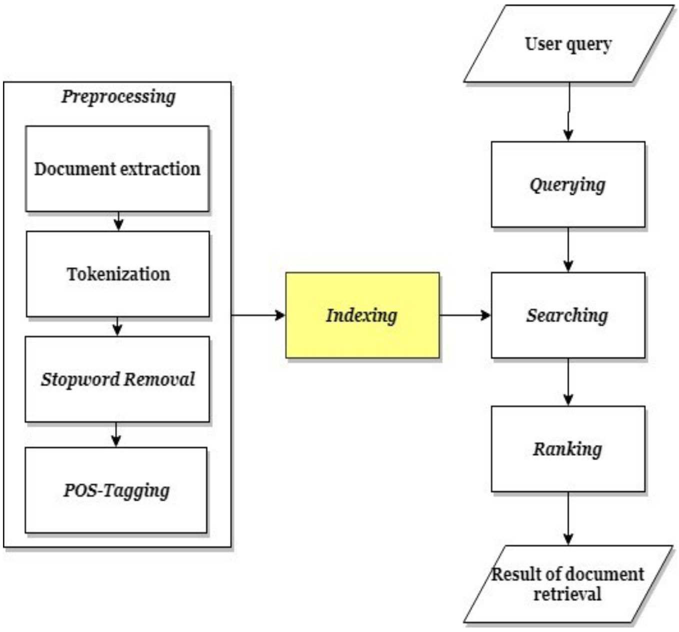


Fig. 3 Document retrieval process

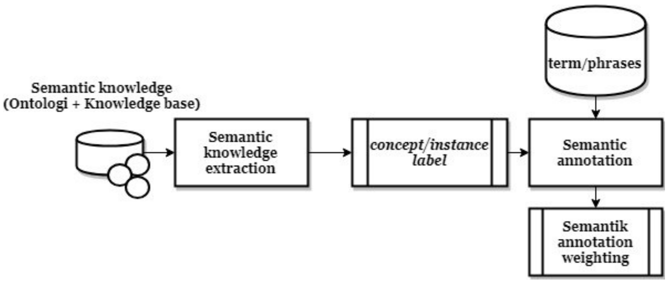

Semantic knowledge (Ontologi + Knowledge base)
Semantic knowledge extraction
concept/instance label
term/phrases
Semantic annotation
Semantik annotation weighting

Fig. 4 Steps of indexing using ontology

TABLE I
EXAMPLE OF SEMANTIC ENTITIES

No	Entity	Label
1	Ai	ai
2	Ajax	ajax
3	Android	android
4	Animation	animation
5	Application	application
6	Artificial_intelligence	artificial intelligence
7	Assembly	assembly
..	..	..
143	Xss	xss

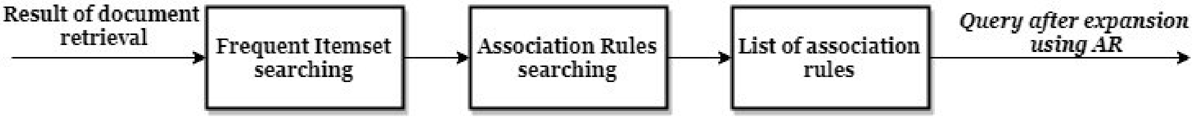

Result of document retrieval
Frequent Itemset searching
Association Rules searching
List of association rules
Query after expansion using AR

Fig. 5 Steps of query expansion using AR

Fig. 5 shows an illustration of the query expansion model using AR. The model consists of three main components, namely (1) the Frequent Itemset Search Component which functions to search for frequent itemset; (2) Association Rules Search Component that functions to search for association rules; (3) A component of the association rules list containing the association rules, the component is the result of the association rules search process.

The initial step of the model expansion query using AR is frequent itemset (FI) searches. The user's initial query entered into the retrieval process produces an ordered document based on the ranking results. The top-ranked document used as input for the model. The document is assumed to be the transaction data needed to be able to carry out the FI search process. The document pre-processed, the results of pre-processed documents stored in a database. It is ready to be used in the FI search process. The initial process is to find an FI. FI obtained using the FP-Growth algorithm. There are several steps to obtain FI using the FP-Growth algorithm. The top document produced at the retrieval process then extracted; each sentence in each document is considered a transaction and given a sentence ID (KID) for identification. The words/phrases that make up the sentence are assumed to be items making up the transaction. As an illustration for the FI search step, the Database query used as the initial query for the user. Documents are generated from the retrieval process using the Database query 9 (nine), as shown in Table II. From the extracted top document, then each word/phrase in the document labelled with letters a, b, c, d, e (Table III).

Transaction data generated as shown in Table III, and then frequent item searches performed. Frequent items are single items that have several occurrences more than or equal to specified minimum support. Frequent items are words or phrases that make up a sentence. At the frequent item search process, all itemset in the selected transaction data scanned to find frequent items. Itemset takes the form of items involved in a transaction. The itemset will be broken down into items and then counted the number of times the item appears. From Table III the itemset is split to obtain frequent item list as shown in Table IV.

If the number of items is more than or equal to the minimum support, then how many items will remain on the list of frequent items. However, if it is less than the minimum support, then the item will be removed from the frequent item list. For example, with the use of minimum support of 20%, the minimum number of times an item will appear is as follows:

Minimum number of occurrences = number of transactions x minimum support

Minimum number of occurrences = 10 X (20/100)

Minimum number of occurrences = 2

The minimum number of occurrences rounded down, the minimum number of occurrences will be 2 or 20% of the total transactions. In addition to saving frequent items, the frequent item list also stores the number of times that item appeared. It is necessary because items in the list sorted by the number of occurrences. The results of the calculation of the number of items appearing with transaction data shown in Table V.

TABLE II
DOCUMENTS GENERATED FROM THE DOCUMENT RETRIEVAL STEPS

No	DocID	Documents
1	33	DOC32.TXT
2	34	DOC33.TXT
3	1	DOC01.TXT
4	94	DOC93.TXT
5	3	DOC03.TXT
6	69	DOC68.TXT
7	9	DOC08.TXT
8	99	DOC98.TXT
9	67	DOC66.TXT

TABLE III
ILLUSTRATION OF TRANSACTION DATA

KID	Term/Pharases
1	{a, b}
2	{a, b, c}
3	{d, e, j}
4	{b, e, f}
5	{c, d, f}
6	{a, b, g}
7	{b, d}
8	{b, c, e}
9	{c, g}
10	{d, c}

TABLE IV
FREQUENT ITEM

KID	Term/Phrases
1	{a, b}
2	{a, b, c}
3	{d, e, j}
4	{b, e, f}
5	{c, d, f}
6	{a, b, g}
7	{b, d}
8	{b, c, e}
9	{c, g}
10	{d, c}

TABLE V
FREQUENCY OF FREQUENT ITEM OCCURRENCE

Term/Phrase ID	Frequent	Nilai Support
b	6	60%
c	5	50%
d	4	40%
a	3	30%
e	3	30%
f	2	20%
g	2	20%
h	0	0%
j	1	10%
i	0	0%

From Table V, set minimum support of 20%, so the items selected include items (b), (c), (d), (a), (e), (f), (g). FP-Tree is a tree scheme used in the FP-Growth algorithm. The process of forming FP-Tree begins with selecting the transaction data. From the itemset of each transaction data, items that are not in frequent items will be ignored. The root in the tree is the document id that is the reference. The reference item is the id of the document selected. In the FP-Tree process, it displays a restricted dataset using a predetermined support count, then the dataset is built into a tree. Examples of ordered and selected transaction data shown in Table VI.

In addition to forming FP-Tree, this process also forms a prefix path, which is the first node address pointer for frequent items in FP-Tree. The making of FP-Tree for reading KID 1 namely {b, a} as shown in Fig. 6. After reading KID 1, then reading KID 2 is {b, c, a}. The results of reading KID 2 are shown in Fig. 7, then KID 3 is read which is {d, e, j} as shown in Fig. 8.

After reading KID 1, KID 2, and KID 3, the reading is done in the same way for the next KID up to KID 10, the results of reading KID 1 to KID 10 are shown in Fig. 9.

The formation of FP-Tree conditional is a process for mining FI. Conditional FP-Tree is a fraction of mining FP-Tree according to the item. If KID readings 1 to KID 10 carried out, it would be continued by searching for paths ending in support counts (b), (c), (d), (a), (e), (f), (g). The results of the FI shown in Table VII.

TABLE VI
ORDERED AND SELECTED TRANSACTION DATA

KID	Term/Phrase
1	{b,a}
2	{b,c,a}
3	{d,e,j}
4	{b,e,f}
5	{c,d,f}
6	{b,a,g}
7	{b,d}
8	{b,c,e}
9	{c,g}
10	{c,d}

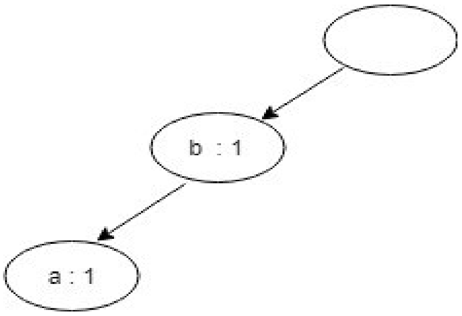


Fig. 6 Formation of FP-Tree for KID 1

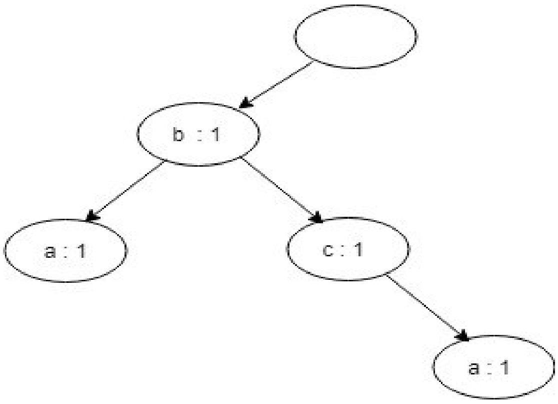


Fig. 7 Formation of FP-Tree for KID 2

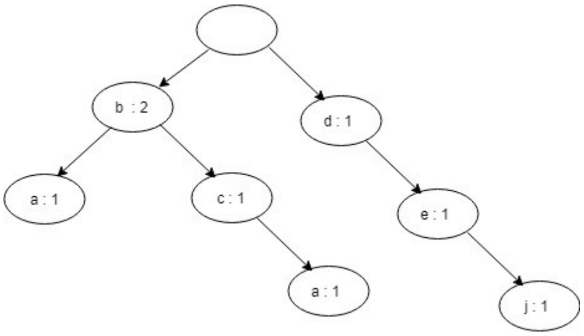


Fig. 8 Formation of FP-Tree for KID 3

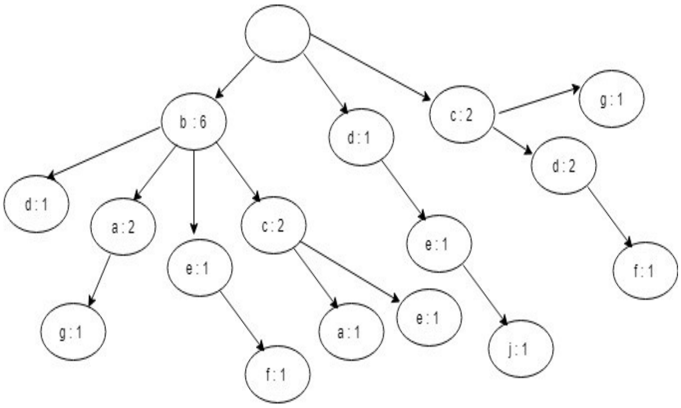


Fig. 9 Formation of FP-Tree for KID 10

The next process searching association rules using AR. The association rule search uses the FI generated at the FI search step, as shown in Table VI. For searching association rules, it is necessary to set the minimum support value and the minimum confidence value. For example, a minimum confidence value of 10%. The association rules search results shown in Table VIII.

The calculation of the value of confidence in most of the frequent items above shows that it meets the minimum confidence value requirement of 10%, as well as other calculations.

The output of the query expansion model using AR is a new query that expanded using AR. The output is the initial entry for MODEL-2. In the expansion query model with Ontology, the initial step is to process the query results from the expansion using AR (Fig. 10). In query processing, the query that results from the AR Query step is performed by word separation. The separation of words changes sentence patterns into words, in which each word consists of only one word. Also, the query separated into words, each consisting of two words. It is to maximize the search process in Ontology because a core meaning is not necessarily not consist of one word but two words. For more details, the process of separating words shown in Fig. 11. As an illustration, queries resulting from expansion with AR are "database concepts". After processing the query, words and phrases are obtained including "concept", "base", "data", "concept base", "database".

In Ontology, in general, there is a relationship between classes, relations between subclasses, and relationships between classes with subclasses or vice versa, namely the relationship of synonyms, hypernym, and hyponyms. Synonym means a word that has the same or similar meaning, which can replace one another. For example, a "database" equals "db" and "operating system" equals "SO". While hyponym is a derivative word from hypernym. For example, the relationship between database and data models. "Data model" is a subclass of "database" which is a hypernym of "data model". In Ontology, the synonym is a class/subclass which is equivalent to other classes/subclasses, while the hyponym on Ontology is a subclass of a class, which is a hypernym of that subclass. Words or phrases generated in the query separation process used as keywords for searching synonyms and hyponyms or hypernym in the ontology. In the initial steps of searching for synonyms and hyponyms or hypernym is done using SPARQL. Searches are done for classes, subclasses. If a keyword is a class name, then a query expansion is relative to the class, such as a synonym expansion and hypernym or hyponym expansion. If the keyword is the instance name, then the query expansion instance is performed. Meanwhile, a class name or instance name searched according to a particular property of the keyword. After an expansion of synonyms and hypernym or expansion hyponyms, the system begins to calculate the similarity to decide whether the results should be returned to the user, according to the threshold similarity. The next process is to calculate the semantic similarity. Currently, research on the method of calculating semantic similarity based on ontology carried out. The hierarchical tree describes the structure of ontology concepts, where Ontology nodes are concept words, and the end is the relationship between Ontological concepts. In general, the broad conceptual domain located in a higher position in the tree structure, and its node density is low. In contrast, the specific conceptual domain located in a lower position in the tree structure, and the node density is relatively high. Therefore, the concept of calculating semantic similarities in tree structures mainly influenced by factors such as node depth, node density, and distance between nodes. The developed model tested using 100 (one hundred) test documents, and 10 (ten) queries as shown in Table IX.

Based on the query in Table IX, testing carried out for each step of the model. From this test, the recall, precision, and f-measure values of the document retrieval steps, query expansion using AR, and query expansion using Ontology shown in Table X.

TABLE VII
RESULTS OF FREQUENT ITEMSET

Suffix	Frequent itemset
b	{-}
c	{c}
d	{c,d}
a	{b,a}, {b.c.a}
e	{b,e}, {b,c,e}, {d,e}
f	{b,e,f}, {c,d,f}
g	{b,a,g}, {c,g}

TABLE VIII
THE CONFIDENCE VALUE RESULTS IN FREQUENT ITEMSET

Frequent itemset	Nilai confidence
b→a	100%
d→a	100%
b→f	50%
d→e	25%
c→g	20%
a→bc	30%
b→ac	100%
c→ab	66.7%
b→a	50%
ab→c	100%

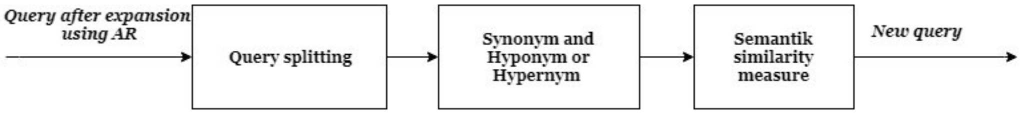
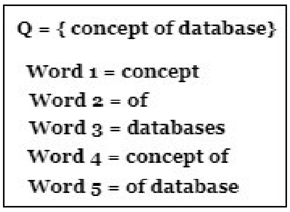

Query after expansion using AR
Query splitting
Synonym and Hyponym or Hypernym
Semantik similarity measure
New query

Fig. 10 Expansion query using Ontology


Q = { concept of database}

Word 1 = concept
Word 2 = of
Word 3 = databases
Word 4 = concept of
Word 5 = of database

Fig. 11 Example of query splitting

TABLE IX
QUERY FOR MODEL TESTING

No	Query	Symbol
1	DATABASE	Q1
2	MYSQL	Q2
3	NETWORK	Q3
4	DML	Q4
5	WEBSITE	Q5
6	HTML	Q6
7	JAVA	Q7
8	PROTOCOL	Q8
9	TOPOLOGY	Q9
10	PHP	Q10

TABLE X
COMPARISON OF RECALL VALUES

	Information Retrieval	Query Expansion using AR	Query Expansion using Ontology
DATABASE	87.50	87.50	88.89
MYSQL	85.71	87.50	87.50
NETWORK	85.71	87.50	88.89
DML	80.00	80.00	83.33
WEBSITE	83.33	85.71	85.71
HTML	80.00	83.33	85.71
JAVA	85.71	87.50	87.50
PROTOCOL	85.71	88.89	88.89
TOPOLOGY	85.71	88.89	88.89
PHP	85.71	87.50	87.50
AVG	84.51	86.43	87.28

Based on Table X, it can seem that an increase in the value of recall, in other words, the number of documents taken from each step has increased, with an increase in the value of 0.85. The average recall value for the IR step was 84.51, the expansion step using AR was 86.43, and the final result of document recall with expansion using an ontology was 87.28. It is presented in graphical form, as shown in Fig. 12.

The calculation of precision values for query expansion using AR and Ontology shown in Table X. From Table X, it can see that the value of precision has increased, by comparing the average value between the retrieval steps by 69.55, the query expansion model with AR by 73.87, and after expansion with the expansion query model with Ontology equal to 79.07. If seen in Table X, anomalies occur for "Network" and "Java" queries, which produce better precision compared to AR expansion and query expansion models with Ontology. It is due to the number of relevant documents retrieved using "networks" and "Java" that are indeed more numerous, but when viewed from the recall value in Table XI, the documents retrieve are fewer when compared to the query expansion model using AR and Ontology.

Table XI is represented in graphical form, as shown in Fig. 13. In Fig. 13, query expansion models that use AR and ontology are colored green, for expansion using AR using red, and blue represent graphs for document retrieval. In Fig. 13 an increase in the value of precision when compared with the steps of retrieval and expansion using AR.

To ensure the expansion of the query model with Ontology does show good performance, then the calculation used f-measure, which combines recall and precision using the mean harmonic weight. The results of the f-measure calculation for query expansion with AR and Ontology were shown in Table XII. From Table XII in general, the f-measure value for the expansion query model with Ontology produces an average value of 82.85. This value has increased when compared to the retrieval step and the results of the query expansion model with AR that is 75.86 and 79.61.

Table XII is represented in the graph to facilitate the reading process, as shown in Fig. 14. In Fig. 14, the query expansion model that uses AR and ontology is colored green for expansion with AR using red, and blue represents the graph for meeting documents.

TABLE XI
COMPARISON OF PRECISION VALUES

	Information Retrieval	**Query Expansion using AR**	**Query Expansion using Ontology**
DATABASE	77.78	77.78	88.89
MYSQL	60.00	70.00	70.00
NETWORK	85.71	70.00	80.00
DML	80.00	66.67	83.33
WEBSITE	62.50	75.00	75.00
HTML	57.14	71.43	85.71
JAVA	85.71	70.00	70.00
PROTOCOL	60.00	80.00	80.00
TOPOLOGY	60.00	80.00	80.00
PHP	66.67	77.78	77.78
AVG	69.55	73.87	79.07

TABLE XII
COMPARISON OF F-MEASURE VALUES

	Document retrieval	**Query expansion using AR**	**Query expansion using AR and Ontology**
DATABASE	82.35	82.35	88.89
MYSQL	70.59	77.78	77.78
NETWORK	85.71	77.78	84.21
DML	80.00	72.73	83.33
WEBSITE	71.43	80.00	80.00
HTML	66.67	76.92	85.71
JAVA	85.71	77.78	77.78
PROTOCOL	70.59	84.21	84.21
TOPOLOGY	70.59	84.21	84.21
PHP	75.00	82.35	82.35
AVG	75.86	79.61	82.85

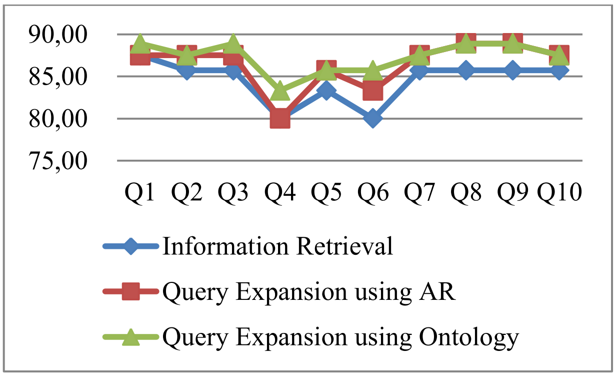


Fig. 12 Comparison graph of recall values

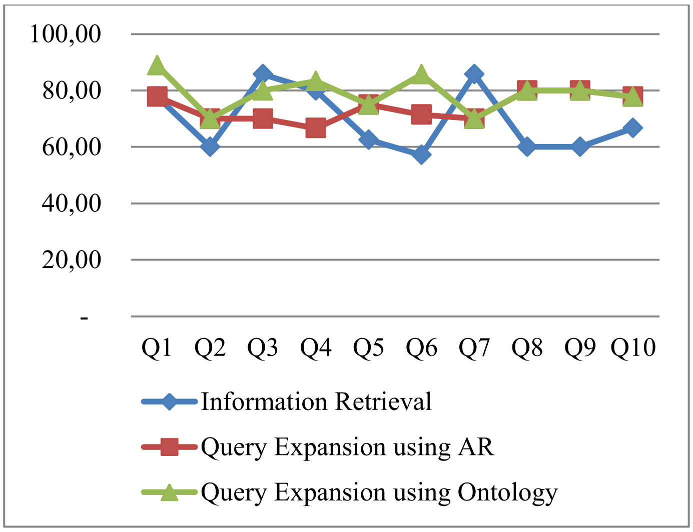


Fig. 13 Comparison graph of precision values

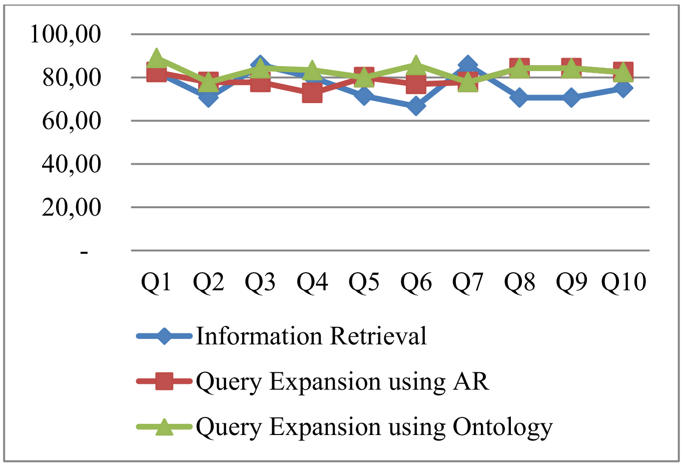


Fig. 14 Comparison graph of F-Measure values

IV. CONCLUSION

In this research, the development of the query expansion model carried out by integrating AR and Ontology to expand the user's initial query. Based on the steps carried out, this study obtained several conclusions, namely the use of ontology in the indexing process in the document retrieval can increase the value of recall, precision, and f-measure. There is an increase in the average value of recall, precision, and f-measure of 9.27, 15.93, and 13.62. With a final average value for recall, precision, and f-measure of 84.51, 69.55, and 75.86. The use of AR (Association Rules) can expand the user's initial query. AR can display the connectedness of the appearance of words simultaneously. Based on test results from query expansion with AR has increased the value of recall, precision, and f-measure of 1.92, 4.31, and 3.75. With the recall, precision, and f-measure values obtained at 86.43, 73.87, and 79.61. The use of Ontology can increase the relevance of documents taken by increasing the value of recall, precision, and f-measure by 0.85, 5.21, and 3.24. The average values of recall, precision, and f-measure were 87.28, 79.07, and 82.85. The results of the study have supported the initial hypothesis, where the proposed model can be used to expand the user's initial query. To produce better recall, precision, and f-measure values. For further research, this model needs to be able to expand queries in the open domain. Also, it needs to be developed at the step of query expansion using Ontology by utilizing the SWRL.

ACKNOWLEDGMENT

We would like to thank LPPM UNSOED for providing funding support for competency improvement research for the 2020 fiscal year. We also express our gratitude to fellow informatics lecturers who have helped to complete this research properly. Hopefully, the results of this research can be useful for all parties.

REFERENCES

[1] B. M. Sanderson and W. B. Croft, "The History of Information Retrieval Research," *IEEE*, vol. 100, pp. 1444–1451, 2012.

[2] P. Castells, M. Fernandez, D. Vallet, M. Fernández, and D. Vallet, "An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 2, pp. 261–272, 2007.

[3] L. Afuan, A. Ashari, and Y. Suyanto, "THE ONTOLOGY APPROACH FOR INFORMATION," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 7, 2019.

[4] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta, "Semantically enhanced Information Retrieval: An ontology-based approach," *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 9, no. 4, pp. 434–452, 2011.

[5] A. Gomathi, J. Jayapriya, G. Nishanthi, K. S. Pranav, and P. K. G, "Ontology Based Semantic Information Retrieval Using Particle Swarm Optimization," *Int. J. Appl. Inf. Commun. Eng.*, vol. 1, no. 4, pp. 5–8, 2015.

[6] D. Zhou, S. Lawless, J. Liu, S. Zhang, and Y. Xu, "Query Expansion for Personalized Cross-Language Information Retrieval," *Int. Work. Semant. Soc. Media Adapt. Pers.*, 2015.

[7] M. Amina, L. Chiraz, and Y. Slimani, "Short Query Expansion for Microblog Retrieval," *Procedia - Procedia Comput. Sci.*, vol. 96, pp. 225–234, 2016.

[8] D. Pal, M. Mitra, and S. Bhattacharya, "Exploring Query Categorisation for Query Expansion : A Study," *CoRR*, pp. 1–34, 2015.

[9] M. Lu, X. Sun, S. Wang, D. Lo, and Y. Duan, "Query Expansion via Wordnet for Effective Code Search," *IEEE*, pp. 545–549, 2015.

[10] M. C. Di. Galiano, M. . M. Valvidia, and L. . U. Lopez, "Query expansion with a medical ontology to improve a multimodal information retrieval system," *Comput. Biol. Med.*, vol. 39, pp. 396–403, 2009.

[11] J. Choi, Y. Park, and M. Yi, "A Hybrid Method for Retrieving Medical Documents with Query Expansion," in *Big Data and Smart Computing (BigComp)*, 2016, pp. 411–414.

[12] A. Abbache, F. Meziane, G. Belalem, and F. Z. Belkredim, "Arabic Query Expansion Using WordNet and Association Rules," *Int. J. Intell. Inf. Technol.*, vol. 12, no. 3, 2016.

[13] J. Ooi and H. Qin, "A Survey of Query Expansion , Query Suggestion and Query Refinement Techniques," *Int. Conf. Softw. Eng. Comput. Syst.*, pp. 112–117, 2015.

[14] M. Mataoui, F. Sebbak, F. Benhammadi, and K. B. Bey, "Query Expansion in XML Information Retrieval A new Approach for terms selection M'hamed," in *Modeling, Simulation, and Applied Optimization (ICMSAO)*, 2015, pp. 4–7.

[15] M. Mitra, C. Buckley, and F. Park, "Improving Automatic Query Expansion," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998.

[16] S. Deerwester, S. T. Dumais, and R. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[17] M. R. A. Nawab, M. Stevenson, and P. Clough, "An IR-based Approach Utilising Query Expansion for Plagiarism Detection in MEDLINE," *J. Comput. Biol. Bioinforma.*, vol. 5963, no. APRIL 2015, pp. 1–9, 2015.

[18] Q. Jin, J. Zhao, and B. Xu, "Query expansion based on term similarity tree model," *Int. Conf. Nat. Lang. Process. Knowl. Eng.*, pp. 400–406, 2003.

[19] R. Mandala, T. Tokunaga, and H. Tanaka, "Combining Multiple Evidence from Different Types of Thesaurus for Query Expansion," *Proc. 22Nd Annu. Int. Acm Sigir Conf. Res. Dev. Inf. Retr.*, pp. 191–197, 1999.

[20] M. Farhoodi, M. Mahmoudi, A. Mohammad, Z. Bidoki, A. Yari, and M. Azadnia, "Query Expansion Using Persian Ontology Derived from Wikipedia," *World Appl. Sci. J.*, vol. 7, no. 4, pp. 410–417, 2009.

[21] H. Al-chalabi, S. Ray, and K. Shaalan, "Semantic based Query Expansion for Arabic Question Answering Systems," *First Int. Conf. Arab. Comput. Linguist. Semant.*, pp. 131–136, 2015.

[22] L. Afuan, A. Ashari, and Y. Suyanto, "A study : query expansion methods in information retrieval," in *International Conference On Engineering, Technology*

and Innovative Researches, 2019, pp. 1–7.

[23] L. Afuan, A. Ashari, and Y. Suyanto, "Query Expansion in Information Retrieval using Frequent Pattern (FP) Growth Algorithm for Frequent Itemset Search and Association Rules Mining," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 2, pp. 263–267, 2019.

[24] A. Babu and S. L, "An Information Retrieval System for Malayalam Using Query Expansion Technique," *Int. Conf. Adv. Comput. Commun. Informatics*, pp. 1559–1564, 2015.

[25] A. Noroozi and R. Malekzadeh, "Integration of Recursive Structure of Hopfield and Ontologies for Query Expansion," *Int. Symp. Artif. Intell. Signal Process.*, 2015.

[26] A. Bouziri, C. Latiri, E. Gaussier, and Y. Belhareth, "Learning Query Expansion from Association Rules Between Terms," 2012.

[27] F. Wang and L. Lin, "Domain Lexicon-based Query Expansion for Patent Retrieval," *Int. Conf. Nat. Comput.*, pp. 1543–1547, 2016.

[28] A. Boubacar and Z. Niu, "Concept Based Query Expansion," *Int. Conf. Semant. Knowl. Grids*, 2013.

[29] N. F. Noy and D. L. Mcguinness, "Ontology Development 101 : A Guide to Creating Your First Ontology," pp. 1–25, 2000.