khazanah informatika
Jurnal Ilmu Komputer dan Informatika

Aggregate Functions in Categorical Data Skyline Search (CDSS) for Multi-keyword Document Search

Mardiah Mardiah [1], Annisa Annisa [2]*, Shelvie Nidya Neyman [3]
*annisa@apps.ipb.ac.id
[1, 2, 3] Department of Computer Science
IPB University
Bogor, Indonesia

Abstract - Literature review is the first step in starting research for a deep understanding of the research interest. However, finding literature relevant to research interests is difficult and takes time. Skyline query is a method that can be used for filtering. An object p is said to dominate object q if p equals q on all of its attributes, and p is at least better than q on one attribute. Categorical Data Skyline Search (CDSS) is an algorithm that can filter skyline objects in categorical data types such as documents. CDSS uses Extended Distance Wu and Palmer (DEWP) to calculate the distance between the user query and document keywords. The document keywords and user queries are represented as nodes in the ACM CCS ontology, and documents are assumed to be represented by a single keyword. This study aims to use the CDSS algorithm to search for skyline documents represented by more than one keyword by adding an aggregate function (average, minimum, maximum) to the CDSS algorithm, especially in calculating DEWP. This study used the thesis documents from the IPB University computer science department. Document keywords will be extracted using the Term Frequency-Inverse Term Frequency (TF-IDF) method. The collected keywords will be mapped in a mixed ontology tree that refers to the Association of Computing Machinery Computing Classification System 2012 (ACM CCS 2012) and Computer Science Ontology (CSO) as ontology standards in computer science. The skyline query algorithm for determining skyline documents is Block Nested Loop (BNL). The evaluation method uses the skyline ratio of each aggregate function in the CDSS. Based on the ratio value, CDSS using the maximum DEWP has the most relevant skyline results compared to the average DEWP and minimum DEWP.

Keywords: categorical data skyline search, aggregate function, ontology, skyline query, term frequency inverse term frequency

Article info: *submitted April 19, 2022, revised November 3, 2022, accepted November 3, 2022*

1. Introduction

The topics to be studied in a study cannot be separated from the issues and themes discussed in previous studies. Deepening understanding of topics or issues discussed in previous research is one of the goals of a literature review [1]; this makes a literature review an essential part of research [2]. A literature search takes a lot of time, because the topics discussed are very diverse, and the amount of literature that must be read is substantial. The time to carry out research has certain limitations.

A skyline query can select a small amount of data from a large number of data sets. Skyline query is a method that can select a small number of desired data objects, which are not dominated by other objects (dominant). An object m is said to dominate object n if m is as good as n in all its attributes and at least better than n in one attribute [3].

Skyline queries are more often used for filtering numerical data, as in research [3], [4], and [5]. Unlike other studies that use numerical data, research [7] uses categorical data, namely the Association of Computing Machinery Computing Classification System (ACM CCS) ontology and the ACM CCS literature. Each node in the ontology is a keyword that represents literature. This study applies a skyline query by proposing the Categorical Data Skyline Search (CDSS) algorithm for filtering or selecting literature. To determine the relationship or similarity between the literature and the query as input, a calculation is performed between the query and keywords from the literature using the Distance Extended Wu and Palmer (DEWP) calculation method. DEWP is the "extended" version of Distance Wu and Palmer (DWP) [8]. The results of the similarity calculation will be the basis for the selection criteria for skyline objects/skyline literature.

Figure 1 illustrates a skyline query on a selection of literature.

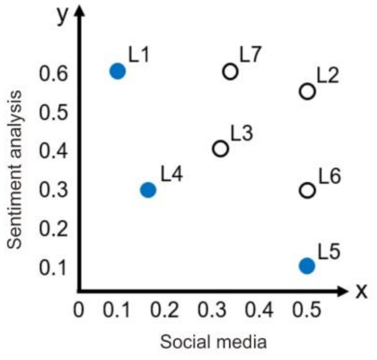

Figure 1. Skyline query on literature filtering

L1 to L7 is literature, and the values on the x-axis and y-axis are the results of calculating the distance using CDSS between query 1: sentiment analysis, and query 2: social media for each keyword contained in the literature L1 to L7. The smaller the distance value, the higher the relationship between the literature and query.

L1, L4, and L5 are skyline literature (illustrated with blue dots) because they are not dominated by other literature on both dimensions, namely the keyword distance from the literature to query 1: sentiment analysis, and query 2: social media. In Figure 1 it can be seen that L1 dominates L7 in the query "social media", L4 dominates L2, L3, L6, and L7 in both dimensions, while L5 dominates L2 and L6 in query 1: sentiment analysis.

Research [7] has applied skyline queries for filtering literature using categorical data, namely ontology and literature. The result is skyline literature, which dominates based on the similarity value between query users and keywords. Thus, CDSS can be used for filtering literature so that it can be useful for researchers in finding the dominant literature according to the research topic.

Unfortunately, this study does not include whether CDSS is used to filter the literature represented by multiple keywords in the ontology (multi-keyword document). Journals, conference proceedings, and thesis documents usually have more than one keyword representing the main content or ideas discussed. For this reason, it is important to formulate the calculation of the CDSS distance by considering the document's multi-keyword. Based on these problems, searching for multi-keyword documents with CDSS is the aim of this study.

Research related to the calculation of multi-keyword document spacing has been carried out using the minimum, maximum, and average aggregate functions in [9] and [10]. Research [9] uses a multi-keyword document describing each keyword as a node in the SNOMED ontology. To calculate the distance or similarity, this study uses the minimum distance from the keywords in the document. Research [10] uses maximum and average values to determine whether there are interactions between proteins, each protein is represented by more than one term described as a node in the gene ontology (GO).

Based on research [9] and [10], the multi-keyword document search with CDSS in this study was constructed using minimum, maximum and average aggregate functions. Furthermore, the performance of each aggregate function will be compared using the skyline ratio, which is a comparison between the skyline literature obtained from each aggregate function.

This study used the thesis documents of IPB University Master of Computer Science students from 2007 to 2020 as data sources. This study will use the Term frequency - Inverse document frequency (TF-IDF) method to get multi-keywords from each document that represents the main idea. Each extracted keyword is described as a node in a mixed ontology consisting of the Association Computing Machinery Computing Classification System (ACM CCS 2012) and Computer Science Ontology (CSO) ontologies [11].

2. Methods

In this study, additional aggregate functions were carried out in CDSS [7] and trials were conducted to find the most suitable aggregate function for skyline literature search. Figure 2 shows the stages of this study. The research phase is divided into two parts, starting from ontology learning, namely the stage for building a mixed ontology starting from text preprocessing, keyword extraction, and mapping on mixed ontologies. The next stage is CDSS, namely the stages to build a CDSS with aggregate function modules, and the final stage is testing and analysis.

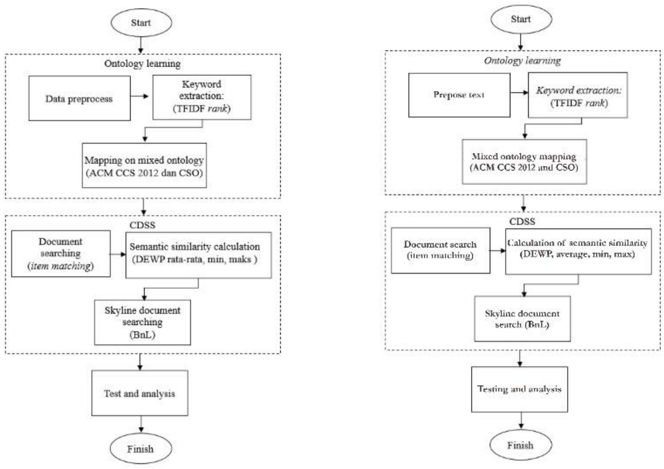
Start
Ontology learning
Data preprocess
Keyword extraction: (TFIDF *rank*)
Mapping on mixed ontology (ACM CCS 2012 dan CSO)
CDSS
Document searching (*item matching*)
Semantic similarity calculation (DEWP rata-rata, min, maks)
Skyline document searching (BnL)
Test and analysis
Finish
Start
Ontology learning
Prepose text
Keyword extraction: (TFIDF *rank*)
Mixed ontology mapping (ACM CCS 2012 and CSO)
CDSS
Document search (item matching)
Calculation of semantic similarity (DEWP, average, min, max)
Skyline document search (BnL)
Testing and analysis
Finish

Figure 2. Research stage

Ontology Learning

Ontology learning is building an ontology from scratch with structured or unstructured data, enriching an ontology or using an existing ontology [12]. This study uses text as a data source; the use of text as a data source in building or enriching ontologies is known as Ontology learning from text [13]. The text used is the thesis document for the Master of Computer Science at IPB University, from which the keywords are extracted. The extracted keywords will be used to build a mixed ontology, namely by enriching the ACM CCS 2012 ontology with the same CSO keywords as the results of the extraction. The development of this mixed ontology is carried out to accommodate all the keywords contained in the thesis document, which are not sufficiently accommodated by only one ontology.

1. Preprocessed Text

Preprocessed text is a step in changing or deleting elements from the original test and the result of preprocessing is the required text [14]. In this study, the preprocessed text consists of tokenization, case folding, post-tagging, and stopword removal. For a simple explanation of preprocessed text data, here is an example sentence and preprocessed steps "Stage 2 processing is to calculate DEWP."

a) Tokenization: separating sentences in a document into words.
Example: “Pemrosesan”, ”Tahap”, ”2”, ”adalah”, “menghitung”,”DEWP”,”.”

b) Case folding: mengubah huruf kapital menjadi huruf kecil.
Example: “pemrosesan”, ”tahap”, ”2”, ”adalah”, “menghitung”,”dewp”,”.”

c) Remove numbers and punctuation.
Example: “pemrosesan”, ”tahap”, ”adalah”, “menghitung”,”dewp”

d) Post tagging: Separates each word into its own group. Example: “pemrosesan” (kata kerja), ”tahap” (taka bantu), ”adalah”(kata keterangan) , “menghitung” (kata kerja), ”dewp”(objek)

e) Remove stop words [15] words in the tags “kata sifat”, “kata kerja”, “kata keterangan”, as well as words that are not contained in the keywords from ACM CCS and CSO.
Example: “dokumen”,”kata”,”kunci”

f) N-gram is a sequence of words with a total of n words. n=2 (2-gram) or known as a bigram is a word order consisting of two words. n = 3 (3-gram) or known as a trigram is a word order consisting of three words [16]. Examples of n-grams (1-4) for the sentence "Ontology to calculate DEWP" are:
1 gram: “ontology”, “for”, “compute”, “DEWP”
2 gram: “ontology for”, “calculate DEWP”
3 gram: “ontology for calculate”, “to calculate DEWP”
4 gram: “ontology for calculating DEWP”

2. *Keyword extraction*

Keyword extraction or keyword extraction is the process of finding a word as a word that represents a document as the main idea or main topic of the document [17]. The method used for keyword extraction is Term Frequency – Inverse Term Frequency (TF-IDF)[18]. Keywords that have been extracted from a document will be checked whether they are included in ACM CCS or CSO, this aims to ensure that keywords that have been extracted from a document are not different from the keywords in ACM CCS or CSO. For keywords that are not included in ACM CCS or CSO, those keywords will be deleted. This refers to research [19] that extracts keywords from BiblioDem corpus documents and equates the extracted terms with terms in the Alzheimer's dictionary so that the extracted terms are only terms that concern Alzheimer's disease.

The TF-IDF calculation consists of calculating the frequency ($TF_{t,d}$), namely the number of occurrences of a term *(t)* in document *(d)*. The frequency of documents (DF_t) is the number of documents in which there is a term *(t)*, the inverse value of frequency of documents in which there is a term *(t)* (IDF_t), the total number of documents in the corpus *(N)*, follows the equation *(1)* as follows [20].

$$IDF_t : log \frac{N}{DF_t}$$
$$TF - IDF_{t,d} : TF_{t,d} \times IDF_t \qquad (1)$$

Research related to keyword extraction from text includes extracting keywords from the "sina news" corpus with the number of keywords from one document including at least 3 keywords and at most 6 keywords contained in [21]. Other research extracts keywords from questionnaire documents with at least 1 keyword and a maximum of 10 keywords from each document [22]. Based on the results of these two studies, it can be concluded that the least number of keywords that can be extracted is 1 keyword and at most 10 keywords. In this study the number of keywords extracted from one document is at least 2 keywords, and a maximum of 6 keywords.

3. *Mixed ontology mapping*

Mixed ontology is the entire ACM ontology and is enriched with keywords from CSO as a new node. Figure 3 shows an illustration of the ontology and also its parts, namely the CCS Node is Root: level 0, namely the root of the ontology where the upper level is more general and the lower it is more specific (levels 1, 2 and so on). An ancestor is a higher-level node; for example, "software and engineering" is an ancestor of an "operating system" node. The “memory management”, “file system management”, “process management” nodes are siblings with the parent node being the “operating system” node.

Mapping on a mixed ontology is the addition of new nodes to the ACM CCS ontology. The new node added is the keyword extracted from the thesis document which comes from the CSO keyword. For example, from the extraction of the thesis documents, the keyword “skyline query” is obtained which is the CSO keyword and is not found in ACM CCS.

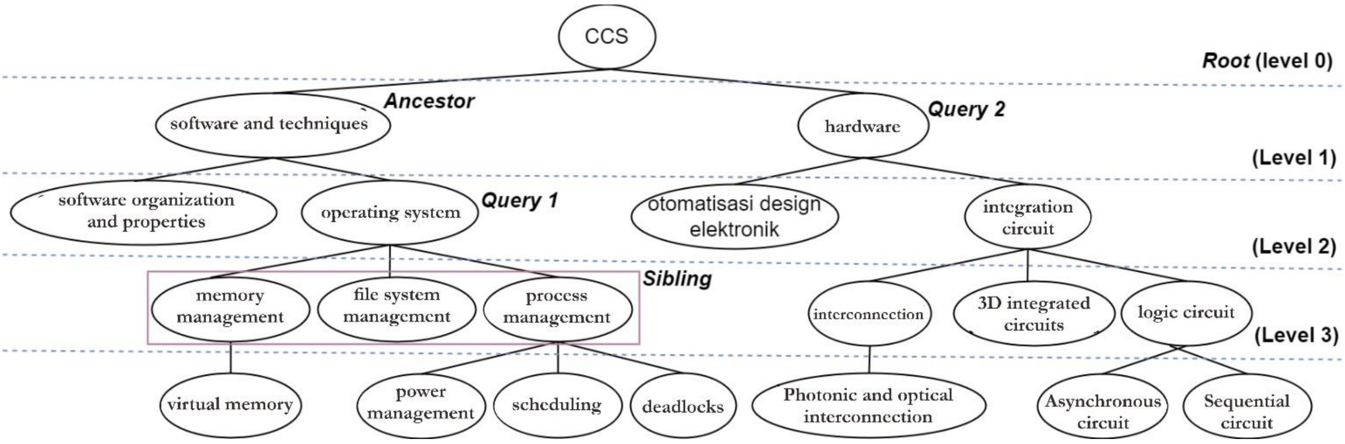

Figure 3. Ontology illustration

The mapping is divided into three steps, the first is checking the ancestor keywords in the CSO hierarchy, the second is matching or equalizing the ancestor of the two ontologies, and the third is adding new nodes in ACM CCS based on the relationships obtained.

The first step is to examine the ancestor keyword in the CSO relationship shown in Figure 4, the ancestor "skyline query" is "query processing", and "query language". The second step is to equate the ancestor of the two ontologies, namely to look for the same ancestor, "query language" or "query processing" in the ACM ontology as shown in Figure 5 where in the ACM ontology there is a "query language" node.

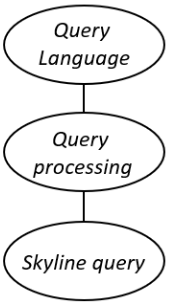

Figure 4. Checking CSO ancestor

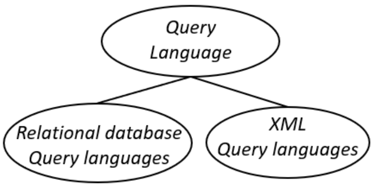

Figure 5. Equalizes the ACM ancestor

The final step is to add these nodes as new nodes as shown in Figure 6. The "query processing" and "skyline query" nodes are added at the level below "Query language".

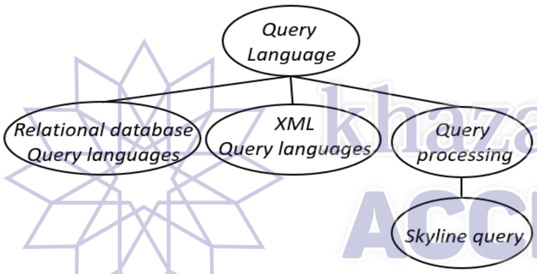

Figure 6. Added new nodes

The steps in the CDSS refer to research [7] consisting of three steps, namely:

1) Document search with item matching

Item matching is a document search that produces documents that have the same words as the user's query [23]. For example, the user enters two queries, namely " sistem operasi " and " perangkat keras ", then the results of the matching items are L1, L2 and L3 as shown in Table 1.

Table 1. Documents and keywords

Id	Full text	*Keyword*
L1	…**perangkat keras** yang digunakan …	**memori virtual**, manajemen daya, sirkuit terintegrasi 3d
L2	**sistem operasi** akan dikembangkan dengan…	**sirkuit asinkron**, manajemen sistem file
L3	berikut **sistem operasi** dan perangk at keras yang digunakan…	**organisasi dan properti perangkat lunak**, interkoneksi fotonik dan optik ,
L4	Tujuan pada rancangan…	Interkoneksi, penjadwalan

In L1, L2, and L3 each has the same word as the query, namely "hardware" and "operating system" while L4 does not have the same word and is not a matching item from both queries.

2) Calculates the DEWP value

Research [7] applies CDSS to searching documents that are assumed to have only one keyword using equation (2). The following defines the DEWP calculation: defines the lowest common ancestor *(z)* between the query node *(x)* and the keyword node *(y)*, calculates the value 1 + the number of child nodes in z (w_z), the level between z and root (n_{zr}), level between x and z (n_{xz}), 1 + number of sibling nodes of y (w_y), level or distance between y to z (n_{yz}):

$$DEWP_{(x,y)} = 1 - [2w_z n_{zr} / (n_{xz} + w_y n_{yz} + 2w_z n_{zr})] \quad (2)$$

Table 2 is an example of the results of calculating DEWP with one keyword from the queries "sistem operasi" and "perangkat keras".

Table 2. DEWP one keyword

Id	Keyword	*Query*	
		Operating system	***Hardware***
L1	Software organization and properties	0.250	0.538
L2	virtual memory	0.143	0.600
L3	Asynchronous circuit	0.700	0.429

This study aims to add an aggregate function to DEWP so that it can be applied to multi-keyword documents. The following in equations (3), (4), (5) is the DEWP which has added the average, minimum and maximum aggregate functions.

$$\overline{DEWP}_{(Q,K)} = \frac{\sum_{i=1}^{n} DEWP_{(q,k)}}{n} \quad (3)$$

$$DEWP_{Min} = Min\{DEWP_{(q,k)} \in DEWP_{(Q,K)}\} \quad (4)$$

$$DEWP_{Max} = Maks\{DEWP_{(q,k)} \in DEWP_{(Q,K)}\} \quad (5)$$

To calculate the aggregate DEWP, it is necessary to calculate the DEWP for all keywords from the item-matching process as well as the query user. Table 3 is the DEWP calculation of all L1, L2 and L3 document keywords with queries. Table 4 shows the aggregate DEWP results with average, minimum, and maximum.

Table 3. DEWP calculation results for all keywords

Id	Keyword	*Query*	
		Operating system	***Hardware***
L1	Virtual memory	0.143	0.600
L1	Power management	0.250	0.739
L1	3d integrated circuits	0.700	0.400
L2	Asynchronous circuit	0.700	0.429
L2	File system management	0.143	0.684

Id	Keyword	*Query*	
		Operating system	*Hardware*
L3	Software organization and properties	0.250	0.538
L3	Photonic and optical interconnection	0.625	0.333

After obtaining the aggregate value, the next step is to search for the skyline documents for each aggregate function. The following is an example of searching a skyline document based on the min DEWP value in Table 4 in bold.

Table 4. DEWP with aggregate

Id	Operating system			Hardware		
	Average	*Min*	*Max*	*Average*	*Min*	*Max*
L1	0.364	0.143	0.700	0.579	0.400	0.739
L2	0.412	0.143	0.700	0.556	0.429	0.684
L3	0.437	0.250	0.626	0.435	0.333	0.538

3) Skyline document search

The skyline query algorithm used in this research is Block Nested Loop (BnL). BnL has good performance when used on small amounts of data [3]. BnL filters by initializing the data in a container list. Each input data is compared with the data in the container list. If the input data is dominated by data in the container list, the input data will be deleted. Figure 7 shows the skyline document based on the DEWP min.

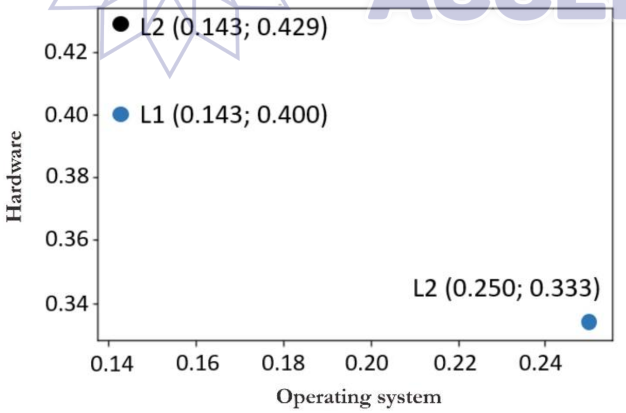

Figure 7. Skyline document DEWP min

The skyline document in Figure 7 is L1 and also L3, while L2 is the dominant document.

The results of each aggregate function will be evaluated by calculating the ratio of skyline documents, following equation (6) [7].

$$\mathcal{Skyline\ document\ ratio} = \frac{number\ of\ skyline\ documents}{The\ number\ of\ documents\ in\ the\ corpus} \quad (6)$$

3. Results and Discussion

In this study, there were two different tests. The first test was to see the effect of the number of keywords on the ratio of skyline documents and computation time. The second test is to see the effect of increasing the number of queries on the ratio of skyline documents and computation time. The following are the results of the two tests:

1. The effect of increasing the number of keywords on the ratio of skyline documents and computation time

In the first test, 2, 3, 4, 5, and 6 keywords were used, with a fixed number of queries of 2. Each test was carried out at least 10 times with different keywords. The results shown are the average results of all tests performed. Figure 8 presents the results of the first test for the skyline document ratio. From each aggregate function, different ratio values are obtained.

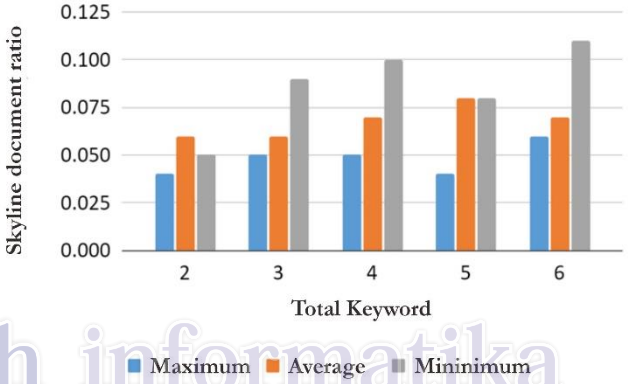

Figure 8. The effect of increasing the number of keywords on the ratio of skyline documents

Based on [24], a small ratio value is better than a high ratio value, and for this test the best ratio is obtained from calculating the maximum aggregate function, followed by the average and minimum.

Figure 9 is the result of testing the increase in the number of keywords for the required computation time. The more the number of keywords used, the longer or increase the computation time. This is because the more keywords, the more keyword nodes that are processed for semantic calculations in the ontology, so the time needed is getting longer [25]. From Figure 9 it can also be seen that the differences in the aggregate functions used have no effect on computation time.

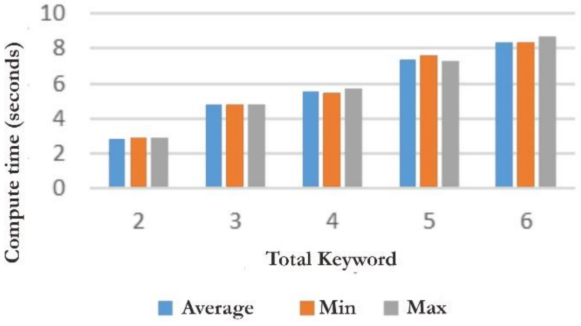

Figure 9. Increasing the number of keywords against computing time

2. The effect of increasing the number of queries on the ratio of skyline documents and computation time

In the second test, the number of queries used was 2,3,4,5, and 6, and the number of fixed keywords was 2 keywords. Figure 10 shows the results of the test, where the increasing number of queries has an effect on the increasing value of the skyline document ratio. This happens because an increasing number of queries means an increasing number of dimensions are taken into account in the skyline search, so the resulting number of skyline objects will increase [26]. From Figure 10 it is also seen that the best ratio value is produced by the maximum aggregate function.

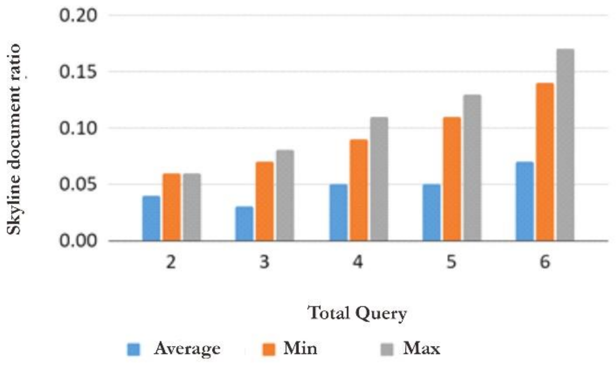

Figure 10. The effect of increasing the number of queries on the ratio of skyline documents

Figure 11 shows the results of testing the query increase in computation time. As the number of queries increases, the computation time increases or takes longer.

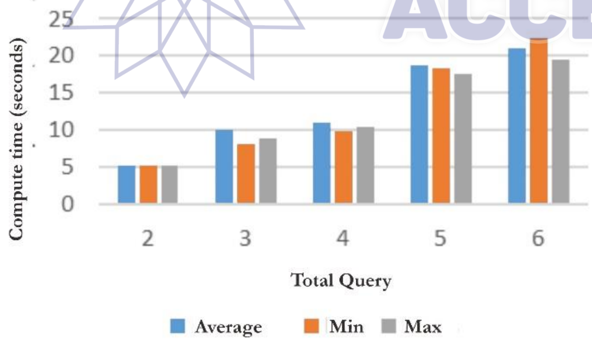

Figure 11. Effect of increasing the number of queries on computation time

The difference in the number of ratios of each aggregate function is caused by several things. The following illustration explains the differences in the performance of the aggregate function in generating skyline literature.

Given input of 3 queries, namely: "clustering", "expert system", and "information system" on documents that have as many as 2 keywords. For example, there are 8 documents item matching results. Table 5 shows the DEWP calculation of the three queries for each keyword contained in the document.

Table 5. DEWP calculation

Id	Keyword	Query 1: Clustering	Query 2: Expert system	Query 3: Information Systems
1	Geographic information system	0.163	0.163	0.429
	Expert system	0.143	0	0.385
5	Geographic information system	0.163	0.163	0.429
	Information systems	0.111	0.111	0
8	Government	0.348	0.348	0.302
	Software	0.375	0.375	0.333
2	Decision support system	0.163	0.024	0.5
	Agriculture	0.483	0.483	0.455
0	Expert system	0.143	0	0.385
	Knowledge acquisition	0.492	0.492	0.464
11	Education	0.483	0.483	0.455
	Data mining	0.015	0.163	0.5
22	Clustering	0	0.2	0.5
	Design	0.571	0.571	0.552
14	Education	0.483	0483	0.455
	Learning model	0.717	0.717	0.709

The value in bold is the maximum distance value of the four keywords based on DEWP calculations for the two given queries. Table 6 shows the DEWP values of each document using the maximum aggregate function. There are three skyline documents from the maximum DEWP, namely documents with ids 1, 5 and 8, and the rest are documents that are dominated.

Table 6. DEWP maximum ratio analysis

Id	DEWP Max			Output
	Query 1: Clustering	Query 2: Expert system	Query 3: Information Systems	
1	0.163	0.163	0.429	Skyline
5	0.163	0.163	0.429	Skyline
8	0.375	0.375	0.333	Skyline
2	0.483	0.483	0.5	Dominated
0	0.492	0.492	0.464	Dominated
11	0.483	0.483	0.5	Dominated
22	0.571	0.571	0.552	Dominated
14	0.717	0.717	0.709	Dominated

Figure 12 shows an illustration of the ontology hierarchy for calculating maximum DEWP. The query node = “clustering” is depicted as a green node, the z node between the clustering query and the geographic information system keyword is “Information system application” in gray, and the z node between the clustering query and the agricultural keyword is “CCS” in gray -ash,. The keyword node for each document in the image is written with "dock id" according to the document id.

Siblings from the node keyword "agriculture" totaling 7 siblings are described with purple nodes, namely "archival and digital libraries" to "personal computers and pc applications", and the distance from the node "agriculture" to node z = 3. The "geographical information system" node has 5 siblings and distance to z = 2. “Geographic information system” node dominates “agriculture” node because it has a smaller number of siblings and closer z distance. Table 7 shows the document keywords with the max DEWP value in the “clustering” query as well as the number of siblings and their distance to the z node. Documents with id = 1 and 5 are skyline documents with a max DEWP value = 0.163 with a "geographical information system" node having a number of siblings = 5 and distance to node z = 2.

Table 7. Number of siblings and distance to node z in “clustering” query

Id	Keyword DEWP max	Number of siblings	Distance to node z	DEWP max from the clustering query
1	geographic information system	5	2	0.163
5	geographic information system	5	2	0.163
2	Agriculture	7	3	0.483

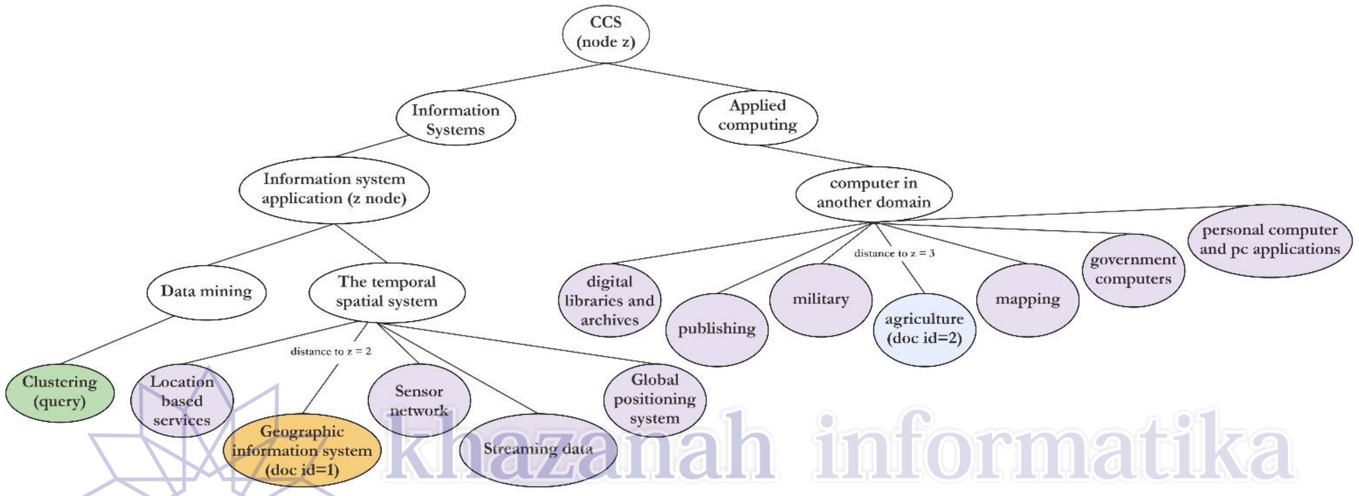

Figure 12. An example of a skyline document hierarchy from a DEWP max

Based on the position in the ontology hierarchy, namely the number of siblings and the distance of the keyword node at node z, the fewer the number of siblings, and the closer the distance of the keyword node to the z node, the smaller the DEWP value and the documents that have these keywords will become skyline documents and dominate many documents. The other has a greater number of siblings and a farther distance to the z node.

Table 8 shows the minimum DEWP calculation results based on the DEWP calculation results in the previous example (Table 5).

Table 8. DEWP calculation results min

Id	DEWP Max			*Output*
	Query 1: Clustering	Query 2: Expert system	Query 3: Information Systems	
0	0.143	0	0.385	Skyline
1	0.143	0	0.385	Skyline
5	0.111	0.111	0	Skyline
11	0.015	0.163	0.455	Skyline
22	0	0.2	0.5	Skyline
8	0.348	0.348	0.302	Dominated
2	0.163	0.024	0.455	Dominated
14	0.483	0.483	0.455	Dominated

The results of calculations with a min DEWP value = 0 mean that the keywords in the document are the same as the words contained in the query (document id = 0, 1, 5, 22). This is not the case for maximum DEWP. DEWP value = 0 makes sure the document becomes a skyline document. This causes the min DEWP calculation to produce a greater number of skyline documents than the maximum DEWP and average DEWP.

Judging from the required computation time, CDSS with maximum, minimum, and average aggregates does not require much different computation time. However, if the number of keywords or queries increases, the computation time required will be longer. Skyline documents resulting from the maximum aggregate are documents that are few in number but relevant to the query given by the user. This is evidenced by the ratio value generated by the maximum DEWP (the lowest ratio value when compared to the min and average DEWP). The documents used in this study are limited within the scope of the Department of Computer Science thesis, further research can use documents within the scope of other fields of science. The attributes used in this study are document keywords which are described in the ontology as nodes, the DEWP calculation in this study only utilizes the ancestor and sibling of the nodes’ keywords. Future research can add descendants of keyword nodes in the DEWP calculation formulation. In addition, future research can compare the results of DEWP with other machine learning algorithms such as decision trees or naïve Bayes.

4. Conclusion

The application of the aggregate function in DEWP calculations in the CDSS algorithm for searching multi-keyword documents has been successfully carried out in this study. The value of the ratio will increase if the number of queries used increases. Computation time will be higher (longer) if there are more queries or keywords (increases linearly). The experimental results show that the maximum aggregate function in the DEWP calculation has the best performance when compared to the minimum and average aggregate functions. This is seen from the ratio value of each aggregate function, the maximum DEWP has the lowest ratio value when compared to the ratio value of the min and average DEWP. Thus, the search for skyline documents using CDSS for multi-keyword documents can be performed using the maximum aggregate function.

References

[1] B. B. L. Penning de Vries, M. van Smeden, F. R. Rosendaal, and R. H. H. Groenwold, "Title, abstract, and keyword searching resulted in poor recovery of articles in systematic reviews of epidemiologic practice," *J. Clin. Epidemiol.*, vol. 121, pp. 55–61, 2020, doi: 10.1016/j.jclinepi.2020.01.009.

[2] J. Brocke, A. Simons, K. Riemer, B. Niehaves, R. Plattfaut, and A. Cleven, "Standing on the shoulders of giants: challenges and recommendations of literature search in information systems research," *Commun. Assoc. Inf. Syst.*, vol. 37, no. 9, pp. 205–224, 2015, doi: 10.17705/1cais.03709.

[3] S. Börzsönyi, D. Kossmann, and K. Stocker, "The skyline operator," in *Proceedings - International Conference on Data Engineering*, 2001, pp. 421–430.

[4] W. Zhang, A. Li, M. A. Cheema, Y. Zhang, and L. Chang, "Probabilistic n-of-N skyline computation over uncertain data streams," *World Wide Web*, vol. 18, no. 5, pp. 1331–1350, 2015, doi: 10.1007/s11280-014-0292-2.

[5] N. Zhang, C. Li, N. Hassan, S. Rajasekaran, and G. Das, "On skyline groups," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 942–956, 2014, doi: 10.1109/TKDE.2013.119.

[6] H. Jaudoin, P. Nerzic, O. Pivert, and D. Rocacher, "On making skyline queries resistant to outliers," *Stud. Comput. Intell.*, vol. 665, pp. 19–38, 2017, doi: 10.1007/978-3-319-45763-5_2.

[7] W. Lee, J. J. Song, and C. K. S. Leung, "Categorical data skyline using classification tree," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, vol. 6612 LNCS, pp. 181–187, doi: 10.1007/978-3-642-20291-9_19.

[8] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, 1994, pp. 133–138, doi: 10.3115/981732.981751.

[9] T. Mabotuwana, M. C. Lee, and E. V. Cohen-Solal, "An ontology-based similarity measure for biomedical data - Application to radiology reports," *J. Biomed. Inform.*, vol. 46, no. 5, pp. 857–868, 2013, doi: 10.1016/j.jbi.2013.06.013.

[10] S. B. Zhang and Q. R. Tang, "Protein-protein interaction inference based on semantic similarity of gene ontology terms," *J. Theor. Biol.*, vol. 401, pp. 30–37, 2016, doi: 10.1016/j.jtbi.2016.04.020.

[11] A. Salatino, T. Thanapalasingam, A. Mannocci, F. Osborne, and E. Motta, "The computer science ontology: a large-scale taxonomy of research areas," in *International Semantic Web Conference*, 2018, vol. 11137 LNCS, pp. 187–205, doi: 10.1007/978-3-030-00668-6_12.

[12] A. Gómez-Pérez and D. Manzano-Macho, "An overview of methods and tools for ontology learning from texts," *Knowl. Eng. Rev.*, vol. 19, no. 3, pp. 187–212, 2004, doi: 10.1017/S0269888905000251.

[13] kaoutar mourchid mohammed Belhoucine, "A survey on methods of ontology learning from text," 2020, no. May, pp. 113–123, doi: 10.1007/978-3-030-38501-9_11.

[14] M. Anandarajan, C. Hill, and T. Nolan, "Text preprocessing," *Handb. Nat. Lang. Process. Second Ed.*, pp. 9–30, 2018, doi: 10.4018/978-1-5225-4990-1.ch006.

[15] F. Z. Tala, "A study of stemming effects on information retrieval in bahasa indonesia," 2003.

[16] D. Jurafsky and J. Martin, "N-Gram Language Models N-Gram Language Models," in *Speech and Language Processing*, 2020.

[17] N. Firoozeh, A. Nazarenko, F. Alizon, and B. Daille, "Keyword extraction: issues and methods," *Nat. Lang. Eng.*, vol. 26, no. 3, pp. 259–291, 2020, doi: 10.1017/S1351324919000457.

[18] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manag.*, vol. 24, no. 5, pp. 513–523, 1988.

[19] D. E. Cahyani and I. Wasito, "Automatic ontology construction using text corpora and ontology design patterns (ODPs) in alzheimer's disease," *J. Ilmu Komput. dan Inf.*, vol. 10, no. 2, p. 59, 2017, doi: 10.21609/jiki.v10i2.374.

[20] C. D. Manning, P. Raghavan, and H. Schütze, "Scoring, term weighting and the vector space model," in *Introduction to information retrieval*, Cambridge University Press, 2009, pp. 120–126.

[21] L. Yao, Z. Pengzhou, and Z. Chi, "Research on news keyword extraction technology based on tf-idF and textrank," in *Proceedings - 18th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2019*, 2019, pp. 452–455, doi: 10.1109/ICIS46139.2019.8940293.

[22] W. Zhang, "Management and plan of undergraduates' mental health based on keyword extraction," *J. Healthc. Eng.*, vol. 2021, 2021, doi: 10.1155/2021/3361755.

[23] H. Bast, B. Buchhold, and E. Haussmann, "Semantic search on text and knowledge bases," *Found. Trends Inf. Retr.*, vol. 10, no. 2–3, pp. 119–271, 2016, doi: 10.1561/1500000032.

[24] Annisa, A. Zaman, and Y. Morimoto, "Area skyline query for selecting good locations in a map," *J. Inf. Process.*, vol. 24, no. 6, pp. 946–955, 2016, doi: 10.2197/ipsjjip.24.946.

[25] W. Sun *et al.*, "Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 3025–3035, 2014, doi:

10.1109/TPDS.2013.282.

[26] C. Kalyvas and M. Maragoudakis, “A skyline-based decision boundary estimation method for binominal classification in big data,” *Computation*, vol. 8, no. 3, pp. 1–22, 2020, doi: 10.1109/SEEDA-CECNSM49515.2020.9221822.