

Article

AR Search Engine: Semantic Information Retrieval for Augmented Reality Domain

Maryam Shakeri ¹, Abolghasem Sadeghi-Niaraki ^{2,*}, Soo-Mi Choi ²  and Tamer AbuHmed ³ 

¹ Geoinformation Technology Center of Excellence, Faculty of Geomatics Engineering, K. N. Toosi University of Technology, Tehran 19697, Iran

² Department of Computer Science & Engineering and Convergence Engineering for Intelligent Drone, XR Research Center, Sejong University, Seoul 05006, Republic of Korea

³ College of Computing and Informatics, Sungkyunkwan University, Suwon 16419, Republic of Korea

* Correspondence: a.sadeghi@sejong.ac.kr

Abstract: With the emergence of the metaverse, the popularity of augmented reality (AR) is increasing; accessing concise, accurate, and precise information in this field is becoming challenging on the world wide web. In regard to accessing the right information through search engines, semantic information retrieval via a semantic analysis delivers more relevant information pertaining to the user's query. However, there is insufficient research on developing semantic information retrieval methods in the AR domain that ranks and clusters AR-based search results in a fair fashion. This paper develops an AR search engine that automatically organizes, understands, searches, and summarizes web documents to enhance the relevancy scores in AR domains. The engine enables users to organize and manage relevant AR documents in various AR concepts and efficiently discover more accurate results in terms of relevancy in the AR field. First, we propose an AR ontology for clustering AR documents into AR topics and concepts. Second, we developed an ontology-based clustering method using the k-means clustering algorithm, vector space model, and term frequency-inverse document frequency (TF-IDF) weighting model with ontology to explore and cluster the AR documents. Third, an experiment was designed to evaluate the proposed AR search engine and compare it with the custom search engine in the AR domains. The results showed that the AR search engine accessed the right information about 42.33% faster and with a 34% better ranking.

Keywords: augmented reality; ontology; information retrieval; clustering



Citation: Shakeri, M.; Sadeghi-Niaraki, A.; Choi, S.-M.; AbuHmed, T. AR Search Engine: Semantic Information Retrieval for Augmented Reality Domain. *Sustainability* **2022**, *14*, 15681. <https://doi.org/10.3390/su142315681>

Academic Editor: Ilias Gialampoukidis

Received: 4 October 2022

Accepted: 20 November 2022

Published: 25 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Augmented reality (AR) combines digital information (such as texts, images, videos, and 3D models) with the real world [1–4]. AR—as a future user interface—plays a significant role in supporting both consumers and industrial applications. In recent years, the primary demands for AR solutions have come from professionals and enterprises. Several studies have shown growing trends in the research and development (R&D) of AR technology in various applications, e.g., learning [5], health [6,7], tourism [8], and manufacturing [9]. The growth rate of the global AR market is forecasted at 40.29% between 2018 and 2023; thus, the European Union introduced it as one of the nine key enabling technologies in Industry 4.0 [1,10]. The growing trends in research and enterprise AR applications indicate that the number of AR information sources and users (researchers, craftsmen, and managers) who search for AR information in various fields is increasing on the world wide web (web). Therefore, it is essential to develop a specific information retrieval method that enhances the accuracy of retrieving AR documents and, likewise, Google Images, which enhances the accuracy of retrieving image content. Semantic information retrieval techniques and rankings are efficient approaches in this regard since they can understand AR concepts.

There is some research on using ontology to enhance AR development or on the basic concepts of AR, but there is no comprehensive ontology of the basic concepts of AR that can be utilized for the semantic ranking of AR documents on the web. Djordjevic et al. [11,12] developed an ontology for AR application code generation and semantic annotations of building information models. The proposed ontology was limited to the field of building information and did not cover all of the tracking, software, hardware, and interface approaches in AR technology. Hervás et al. [13] defined AR device, user, environment, and service ontologies for personalization visualization. The device and service parts of the ontology do not include specific AR devices and software, such as head-mounted displays and authoring tools. Eckertz et al. [14] developed an ontology-based machine-readable knowledge base to formalize AR knowledge containing AR device specifications, technical and functional components, ambient factors, experiences, and application knowledge. Evangelista et al. [1] classified the basic concepts of AR into display devices, tracking systems, user interaction systems, and applications. However, they did not provide an ontology based on the classified concepts.

Semantic information retrieval methods have been developed in several studies. Methods that have been developed in previous studies for semantic information retrieval include the following: semantic annotation [15] combining fuzzy rough sets, intelligence Latent Dirichlet Allocation (LDA), and ontology [16]; an integrated framework that includes a query expansion method, machine learning-based classification method, and semantic clustering for personalization of e-commerce [17], an LDA-based-DBSCAN algorithm for technical cards recording information on assembly problems [18]; a combination of the term frequency-inverse document frequency (TF-IDF) weighting method, LDA, and the k-means clustering algorithm [19,20]; machine learning-based classification methods [21]; semantic similarity and the single pass clustering (SPC) algorithm [22]; an ontology-based k-means clustering algorithm [23–25]; and ontology-based semantic similarity for cultural heritage [26]. Various methods for semantic information retrieval have been developed, but none have been applied to the AR domain.

In this paper, we developed an AR search engine based on a semantic information retrieval method in the AR domain to perform effective information retrieval from web documents. The proposed method enables users to obtain more relevant results for their search queries in different groups and obtain results very quickly in the AR domain. This research was conducted in three main steps. First, we propose an AR ontology for calculating relevance scores in semantic information retrieval by reviewing key AR concepts, such as software, hardware, tracking, interaction, and interface. Second, we propose semantic information retrieval based on the AR ontology and clustering method to conduct semantic analysis on web documents, retrieve syntactically and semantically relevant information from them, and cluster them according to the AR concepts. According to [27], this is a query result clustering problem, which is conducted to the information retrieval accuracy. Third, we evaluate the proposed approach by comparing it with the Google search engine using two evaluation metrics: time completion and mean reciprocal rank (MRR). The main advantages of the proposed approach are faster retrieval times and semantic retrieval with the hierarchical inheritance of ontology. The main contributions of this research are listed as follows: 1—we introduce an AR ontology to cluster AR web documents into AR topics; 2—we develop an ontology-based clustering method for information retrieval systems; and 3—we prove that the AR search engine (based on the proposed semantic information retrieval method) outperformed the custom search engine through a user study.

The remainder of the paper is organized as follows. In Section 2, we propose the AR ontology as well as the superclasses, subclasses, and relationships according to the AR concepts, which will be used for semantic information retrieval in the AR domain. Section 3 describes the proposed methodology, which is based on the k-means clustering algorithm concept-based vector space model and TF-IDF approaches for the AR search engine. Section 4 explains the design of an experimental study that evaluates the performance

of the proposed approach for semantic information retrieval in the AR domain. Section 5 presents snapshots of the proposed AR search engine and the evaluation results obtained by comparing it with the results of the Google search engine in the AR domain. Section 6 concludes the research and provides future work.

2. AR Ontology

2.1. Development Procedure

Since the 1990s, there have been various approaches to ontology development [28]. We applied the methodology approach proposed by Zheng et al. [29] for building the AR ontology. The AR ontology is built in four steps using the following approach: specification, knowledge acquisition/conceptualization, implementation, and evaluation.

AR ontology specification. Ontology specification is the first step of the ontology development procedure, in order to determine the AR ontology scope and purpose, intended users, and requirements of the ontology [29]. The purpose of AR ontology is to offer an organized representation of augmented reality concepts to support semantic information retrieval from web documents, which is a shared reusable representation of the AR conceptualization. The scope of AR ontology involves the representation of AR and entities related to AR development for various applications according to the AR definition and characteristics. The end users of AR ontologies are developers, researchers, stakeholders, and relevant authorities. The developers can directly use AR ontology to develop various applications, including search engines and AR information retrieval. Other end users can indirectly use the ontology through software and applications that are developed based on it.

The widely accepted definition of AR was proposed by Azuma [30]; it involves a ubiquitous user interface with three characteristics aimed at visualizing digital/virtual information that is directly registered in the real world [31,32]. The characteristics that AR systems must satisfy the combining real and virtual worlds, including interactions in real-time, and registering in 3D [30]. Azuma's AR definition requires specific hardware/devices, spatial registrations, and interactivities. The virtual information must be precisely aligned with the real world, which is handled by the tracking system by dynamically determining its spatial properties [1]. The intuitive interactions between the user and virtual information are created through an appropriate user [33]. Finally, the software that will make the hardware perform what AR systems need to do, regardless of the hardware platform [34].

Knowledge acquisition and conceptualization. This step involves reviewing relevant knowledge regarding AR technology concepts as they are retrieved through search engines for AR system development in various applications. Relevant knowledge of AR ontology is obtained from existing ontologies and previous studies. Existing ontologies that have been developed in the AR domain can be found in work by Djordjevic et al. [11,12], Hervás et al. [13], and Eckertz et al. [14]. Existing ontologies have not covered all AR concepts, i.e., they are mostly developed for specific applications. Therefore, studies that have focused on AR concepts were used to develop AR ontologies, including Evangelista et al. [1]. After collecting knowledge from the existing ontologies and relevant studies, the relevant terms were listed to define class hierarchies, class properties, and individuals. The existing ontologies and AR concepts were used to develop a new hierarchy for AR ontologies, and no links were created to the existing ontologies. According to the AR definition, we defined five superclasses—hardware, software, interaction, tracking, and interface. The relevant classes are defined as the subclasses for the superclasses with the *SubClass Of* property. Figure 1 depicts the main classes of AR ontology based on the concepts.

AR ontology implementation. The AR ontology was implemented in the Protégé software in the OWL format. The OWL file was utilized in the semantic information retrieval process to develop the AR search engine. The OWL, which includes classes, object properties, and data properties [29], is used to cluster web AR documents. The next section describes how AR ontology is used in AR document clustering.

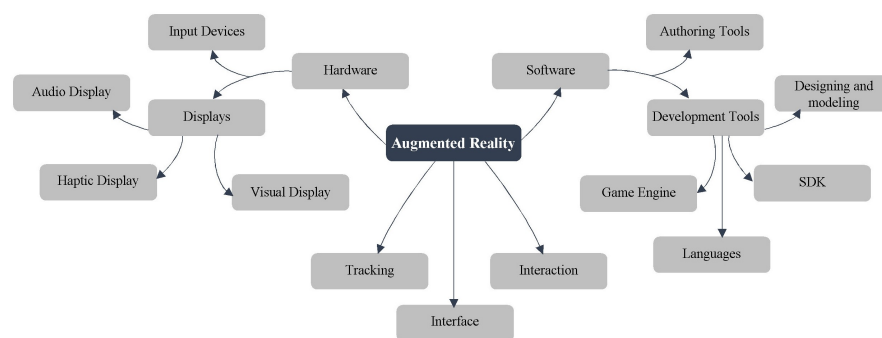


Figure 1. AR ontology.

AR ontology evaluation. Several approaches have been proposed for ontology evaluation [29]. We apply task-based or application-based evaluation approaches to evaluate the AR ontology. This approach evaluates the ontology’s performance within the context of the application by measuring its competency to complete targeted tasks [35]. We evaluate the AR ontology in the semantic information retrieval system.

2.2. Ontology Classes and Individuals

This section explains the classes, subclasses, and individuals of the AR ontology in detail. The software is related to creating and using an augmented reality system that might be used directly in the AR application, creating the AR application, or providing content for the AR application, among other things [34]. We divide the software into two main classes (Figure 2): authoring tools and developing tools. Authoring tools enable users with non-programming backgrounds to create virtual content in AR [36–39]. There are a number of various types of AR authoring tools that can be used to create augmented reality experiences. One tool is the Designers AR Toolkit (DART) [40]. DART was specially created for designers, allowing them to develop AR applications quickly via tools that they are familiar with [37]. Development tools contain software development kits (SDKs), game engines, design/modeling software [41], languages, and libraries. The AR SDK is a set of development tools that provides a variety of functionalities, including AR recognition, AR tracking, and AR content rendering [42]. The SDK reduces the integration and development time for software professionals [43]. Vuforia, Wikitude, ARCore, and ARToolkit are samples of AR SDKs that are widely used to develop AR applications. Vuforia engine’s recognition and tracking features can support different types of targets, both 2D and 3D. The Vuforia SDK supports both Android and iOS platforms. Horizontal surfaces, such as floors and tabletops, can also be detected and tracked [43]. The game engine, by providing an integrated AR framework, can interact with AR sensors and displays. The information that will be sent to the renderer in order to generate the signals for the display device is generated by the game engine, such as the unity development tool. The key feature of the game engine is to provide a simulation loop supporting user interaction [42].

Figure 3 shows the AR hardware classes and their individuals. The main AR devices are displays and input devices [33]. Displays deliver signals to our senses of sight, touch, hearing, and smell, as well as taste in some cases [34]. Therefore, visual and non-visual displays are both involved in the display class. Non-visual displays include audio, haptic, olfactory, and gustatory displays [31,34]. The basic function of a visual display is to generate light signals that our eyes interpret as visual imagery [34]. In AR, there are four types of visual displays: head-mounted displays (HMD)/head-attached display (HAD), head-up display (HUD), handheld display, and spatial display. In addition to displays, input devices help to realize an immersed environment, which enables the user to interact with virtual content, including selecting, manipulating, and creating virtual content [44]. The Vive controller, gloves, pointing device, joystick, and touchpad are samples of the input devices.

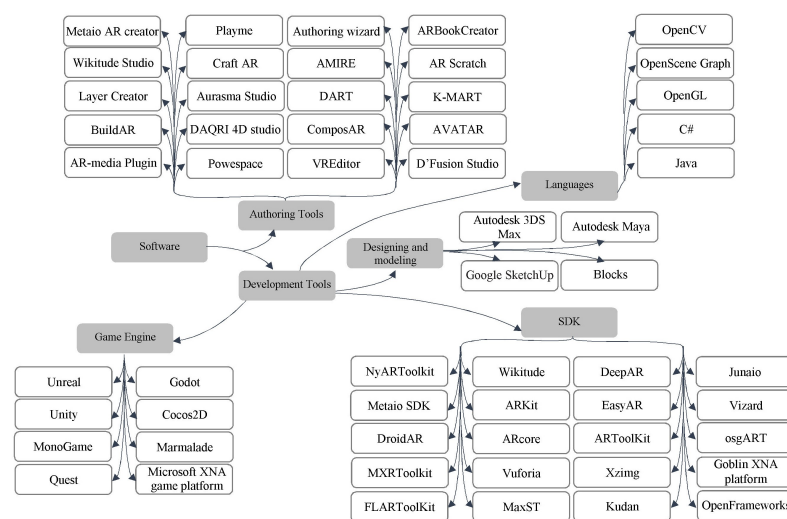


Figure 2. AR software subclasses.

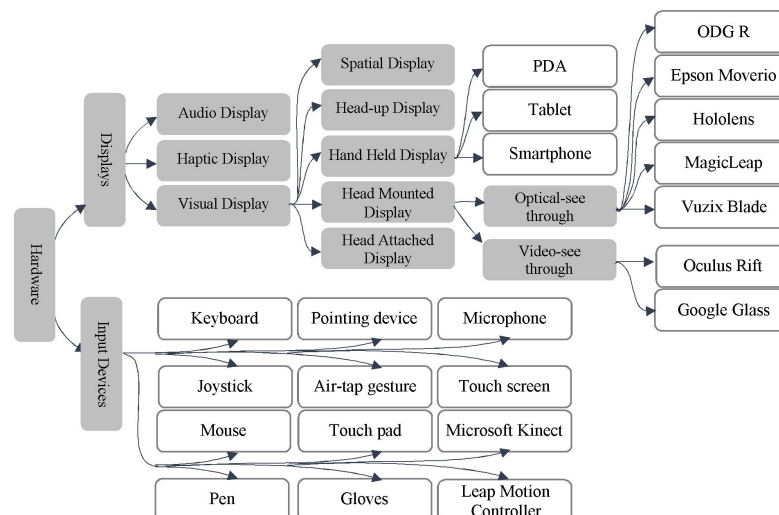


Figure 3. AR hardware subclasses.

An HMD that overlays both real and virtual images in the user's view of the world uses a device on the head or as part of a helmet [33]. Merging real and virtual images can be accomplished by using either a video see-through or an optical see-through HMD system [1]. The video see-through HMDs require two cameras and the processing of both cameras to provide both the "real part" of the augmented scene and the virtual objects with the unmatched resolution, while optical-see-through HMDs employ a half-silver mirror technology to allow views of the real world to pass through the lens and graphically overlay information to be reflected in the user's eyes [33]. HUD is a method of projecting information onto windscreens [45], which is widely used in vehicles. They provide transparent screens that visualize information without requiring drivers to take their eyes away from the road. This kind of display incorporates external environment information into the windshield screen by allowing drivers to focus their attention in a new way [46]. Handheld devices, such as smartphones, tablets, and PDAs, are small computing devices that are equipped with panels of some sort to display on mobile projectors; AR users can hold them in their hands [33,34]. They support mobile AR applications as they have enough computational capability and their projectors are small and portable enough [34]. Finally, spatial displays enable users to experience AR without the need to wear or carry a display by augmenting digital information onto physical objects. This type of display supports the grouping of the users and collaboration between the users [33].

The tracking class, which is one of the most significant classes of AR systems, contains systems that execute virtual object registration in a real environment. The tracking systems are aimed at precisely aligning virtual information with the real world in real time in order to provide an engaging AR experience. As a result, the tracking system is utilized to dynamically identify the spatial attributes (6DOF) of virtual entities (three components for the position and three components for the orientation) in the real world [1,37]. There have been a variety of methods for AR tracking, which are categorized into two main categories according to [47]: marker-based and marker-less methods (Figure 4). Marker-based methods rely on physical markers, such as tags and fiducial markers, to identify objects in the real world. The markers should contain unique visual characteristics, such as lots of corners and edges, which could be easily detected with computer vision techniques. Marker-less methods have two subclasses: natural features and motion sensing. The former includes tracking methods that employ computer vision and image processing techniques to track the AR user based on the natural features that are easily detectable in a scene. The latter are tracking methods that analyze the sensor data to estimate the camera's position and orientation.

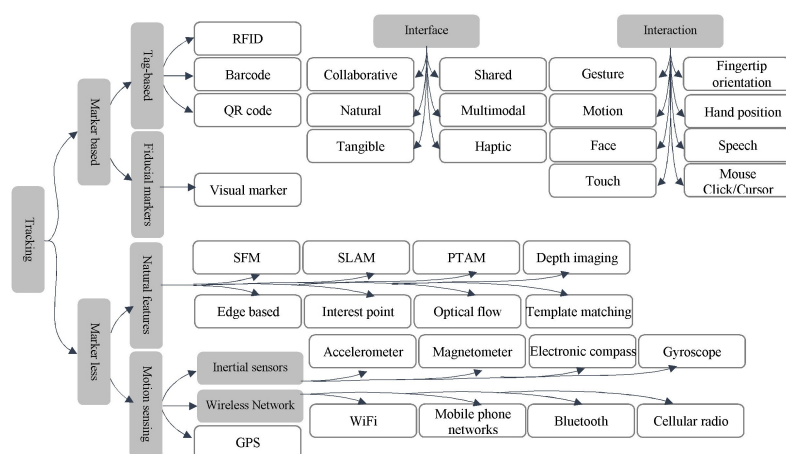


Figure 4. AR tracking, interface, and interaction subclasses.

Finally, AR interfaces and interactions are two main classes in AR systems. AR interfaces apply various techniques to create intuitive interactions between the user and the digital content [33]. The AR interfaces depend on the number and variety of information inputs and outputs that enable users to interact with the AR systems [48]. Tangible AR interfaces, collaborative AR interfaces, shared AR interfaces, haptic AR interfaces, natural interfaces, and multi-modal interfaces are samples of AR interfaces [33,48]. According to [48], the interaction methods include gesture recognition, facial recognition, speech recognition, touching, and pointing and grabbing.

3. Ontology Based AR Information Retrieval

The traditional web crawler results in the fields of AR have scarcely met efficiency and effectiveness performances, even when using an advanced document indexing approach because the amount of information available in AR fields is exponentially growing. For instance, a user desires to find which AR SDKs are utilized in the field of agriculture, but the user searches with dissimilar keywords, such as “AR in Agriculture”. Using the search keywords alone, the crawlers deliver irrelevant results; instead, the crawlers should categorize the results to highlight the results related to the AR SDKs that the user is interested in. Moreover, the user should search using several keywords to achieve the desired results; categorizing the web crawler results into AR fields enables the user to obtain the desired result in an efficient way.

The main goal of the AR search engine is to effectively analyze the user query in the AR concepts with the least amount of effort and time. In order to integrate semantics into

all-out web search tools, we enriched the existing tools with the proposed AR ontology in the previous section. Figure 5 shows the methodology for an ontology-based AR search engine. The proposed methodology consists of three steps: preprocessing, semantic vector space modeling (SVSM), and clustering. In the preprocessing step, the user query and web documents are analyzed to generate a clean and uniform format of data for the following processes [49]. The second step is to represent AR documents as vectors according to the concepts of AR ontology. The clustering step is to be performed by semantically clustering all web documents into AR domains.

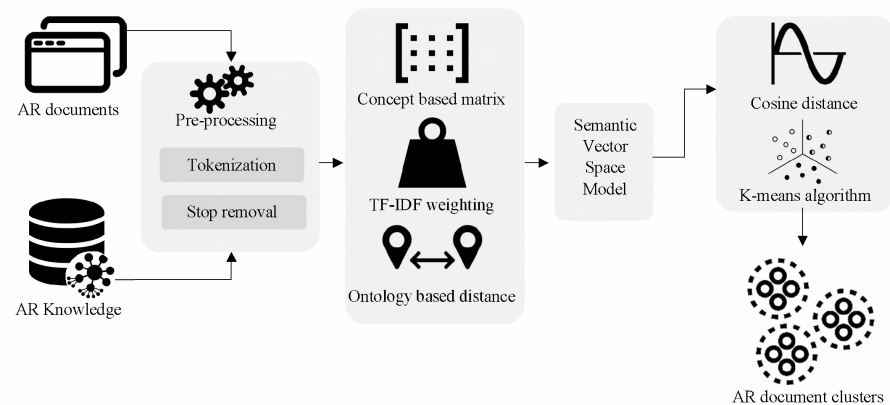


Figure 5. Methodology for an ontology-based AR search engine.

Preprocessing. This step is necessary to obtain efficient results [50]. The preprocessing step is conducted to convert query and AR documents into a sequence of words using NLP techniques including tokenization and stop-word removal. Tokenization is the process of breaking down an input query and AR documents into sentences and then individual words or terms. Stop-word lists are used to erase all non-informative terms.

Semantic Vector Space Model (SVSM). We use the semantic vector space model (SVSM) in this step, in which an AR document A_i is represented by an N -dimensional vector denoted by $A_i = \{w_n, n = 1, \dots, N\}$, where N is the number of main concepts in the AR ontology and w_n is the weight of each concept. To create SVSM, we consider that an ontology element, including a class or an individual, represents a concept in the AR domain. The main concepts are the superclasses that are used to represent each AR document and clustering. The vector of each document is created in three steps, adapted from [51]. First, given a set of L AR documents denoted by $D = \{A_i, i = 1, \dots, L\}$ that each one contains M concepts $\{C_m, m = 1, \dots, M\}$, we create a two-dimensional $(M \times N)$ matrix for each document, where each element of the matrix represents the weight w_{mn} corresponding to superclasses of seed concepts. Second, for each pair of the m and n concepts, we calculate w_{mn} using the semantic similarity between the concepts denoted by s_{mn} and the TF-IDF score of the m^{th} concept denoted by t_m . Figure 6 shows the matrix and the SVSM with an example in the AR ontology domain. In the figure, C'_n was used to represent the main concepts.

$$w_{mn} = \frac{1}{2}(t_m + s_{mn}) \quad (1)$$

Semantic similarity s_{mn} is determined as inversely proportional to the shortest distance d_{mn} between concepts in the AR ontology. According to the ontology hierarchy, the distance between two concepts is the number of links between them. For instance, the distance between *GPS* and *tracking* is three. The lower the similarity score, the further apart two concepts are in the AR ontology. Therefore, semantic similarity is mathematically defined as:

$$s_{mn} = \frac{1}{d_{mn} + 1} \quad (2)$$

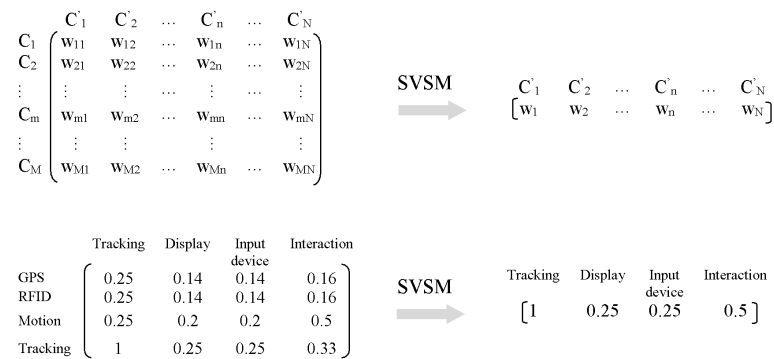


Figure 6. Semantic Vector Space Model.

The TF-IDF score, t_m , represents the importance of the concepts in each AR document, A_i . In this research, the term frequency-inverse document frequency (TF-IDF) technique, which examines the relationship between each concept in a collection of documents, is used to give the concept weight. The technique extracts the keywords from the collection of documents in several steps, including computing similarities among documents and determining the search ranks. The *TF* reflects the importance of a concept in a document by measuring the occurrence of the concept; a high value of *TF* means the high importance of the concept in the document. The importance of a concept across a collection of documents is measured by *IDF*, which is the inverse of how many times the concept appears [19].

$$t_{m,A_i} = TF(C_m, A_i) \cdot IDF(C_m) \quad (3)$$

$$TF(C_m, A_i) = \frac{f_{m,A_i}}{L} \quad (4)$$

$$IDF(C_m) = \log \frac{L}{|A_i \in D : C_m \in A_i|} \quad (5)$$

where f_{m,A_i} represents the number of times the concept C_m appears in the AR document A_i and $|A_i \in D : C_m \in A_i|$ represents the number of AR documents in which the concept C_m appears. After creating the matrix in the second step, we can complete SVSM for each AR document by taking the maximum weight for a given superclass in the third step.

$$w_n = \arg \max_m (w_{mn}) \quad (6)$$

Clustering. Clustering, in which similar terms extracted from the documents are grouped to create clusters, is a kind of text-mining process [52,53]. In this step, we use k-means clustering to cluster AR document vectors to display information retrieval results in clusters. The k-means algorithm is one of the most widely used unsupervised learning algorithms with the following objective function that should be minimized during the iterative process through necessary conditions [54].

$$J(z, B) = \sum_{i=1}^L \sum_{k=1}^K z_{ik} \|A_i - b_k\|^2 \quad (7)$$

where $J(z, B)$ is the objective function, $B = \{b_k\}$ is the K cluster centers, and $z_{ik} \in 0, 1$ is a binary value, which is 1 if A_i belongs to k-th cluster. In the classical K-means algorithm, $\|A_i - b_k\|^2$ is the Euclidean distance between the AR document A_i and the cluster center b_k . As we develop the k-means algorithm for the clustering AR document, we use cosine distance, which provides better results when finding text similarities [55]. By assuming two documents in D , ($A_p \in D$ and $A_q \in D$, in which p and q are the indexes of the documents), the cosine distance between two AR documents is calculated using the following

formula. w_n^p and w_n^q are the weights that are obtained for the main concept Cn' for the A_p and A_q documents, respectively.

$$d_{pq} = 1 - \text{similarity}(A_p, A_q) \quad (8)$$

$$\text{similarity}(A_p, A_q) = \frac{\sum_{n=1}^N w_n^p \times w_n^q}{\sqrt{\sum_{n=1}^N (w_n^p)^2} \times \sqrt{\sum_{n=1}^N (w_n^q)^2}} \quad (9)$$

4. Experimental Design

We designed an experimental study in order to evaluate the proposed AR search engine to facilitate access to the desired information in various AR domains. It was expected that the AR search engine would improve the performance of searching in the AR domains with the least amount of effort and time because it provides a simpler and more efficient tool to obtain the essential information for any users with minimal search keywords. In the experimental study, we compared the results of the Google search engine as a custom search with those of the AR search engine as an AR search in the AR domain. Two evaluation metrics are employed to compare the results: task completion time and mean reciprocal rank (MRR). The task completion time measures the efficiency of the engine, which means the total time that participants require to find the desired results in searching. The MRR measures the mean of the multiplicative inverse of the correct result's rank in the engine's output sorted list of results [56]. In our experimental study, it means how many results the participants check until they obtain the desired result. It calculates using Equation (10) in which $|Q|$ is the number of searches, q is the index of the search, and $rank_q$ represents the rank position of the first correct result for the q^{th} search.

$$MRR = \frac{1}{|Q|} \sum_{q=1}^{|Q|} \frac{1}{rank_q} \quad (10)$$

For the experimental study, the population group consisted of 30 participants. The participants were asked to search for three topics using both the custom search and the AR search. The experimental study was carried out in four steps: (i) the participants were given explanations of the AR search engine and the evaluation scenarios; (ii) three topics were searched using three sets of keywords via both engines. We asked the participants to select three sets of keywords to search for on both engines. The query keywords should be simple, with a maximum of five keywords; (iii) the completion time and the rank of the desired result were written in the designed forms. We used a timer to measure the duration from when the user started to type the query keywords to when the user found the desired result. After the user found the desired result, we checked the rank of the result in the list of results for each query. For the AR search engine, the rank of the desired result in the cluster was used; and (iv) interviews were held with the participants to obtain their feedback on the AR search engine.

We utilized the flask framework to develop a web-based AR engine, the Protégé software for implementing the AR ontology, and other Python libraries, including Owlready to create the interaction between the engine and the AR ontology, and googleapiclient, which is a library used to access the Custom Search JSON API to retrieve and display search results from the Google engine. The googleapiclient returns the document links that correspond to the user's query. The links were used to crawl document pages to obtain the document's text and title using the BeautifulSoup library. It should be noted that we retrieved 100 documents using the google API for each query. Algorithm 1 shows the pseudocode of the proposed methodology for the AR search engine. Flask is a model view controller (MVC) framework for the Python language. It is a light framework that contains a library and a collection of codes for developing websites without requiring them to do everything from scratch [57]. The flask is used to develop a RESTFUL service to execute the semantic information retrieval for the search engine.

Algorithm 1: Semantic information retrieval for AR search engine.**Input:** q_u : User query C' : list of main concepts Ont : AR ontology**Output:** $result$: Classified documents

```

1  $DOC = \text{getRDocuments}(q_u)$  #using the google API: googleapiclient
2 foreach  $doc \in DOC$  do
3    $link = \text{getDocLink}(doc)$ 
4    $text = \text{crawlDocPage}(link)$  #
5    $words = \text{preprocessing}(text)$ 
6    $matrix = \text{generateMatrix}(words, Ont, C')$ 
7    $\{w_n\} = \text{getSVSM}(matrix)$ 
8    $A = \{w_n\}$ 
9    $D.add(A)$ 
10  $result = \text{kMeanClustering}(D)$ 

```

5. Results**5.1. Overview of the AR Search Engine**

Figure 7 shows a snapshot of the AR ontology that was developed in the Protégé software. The left panel shows the class hierarchy of the AR ontology, which contains superclasses and subclasses. As the figure shows, 32 classes and 112 individuals were implemented for the AR ontology. The right panel visualizes the relationships, including subclass, individual, and equivalence, between ontology classes and their individuals using the OntoGraf plug-in.

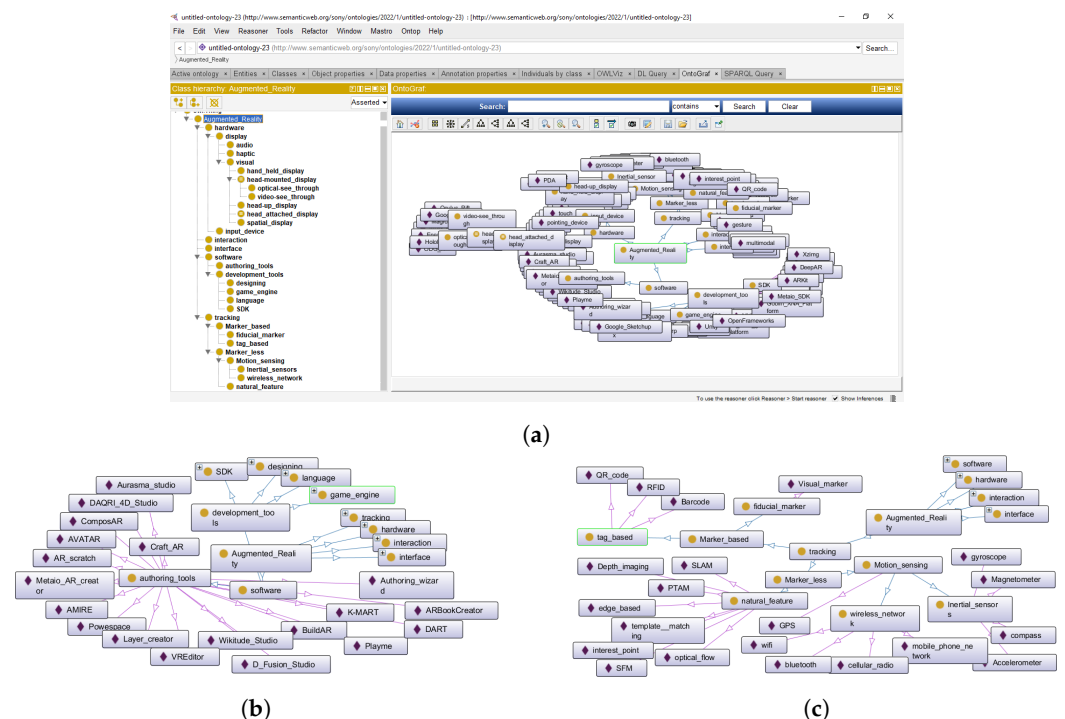


Figure 7. Visualization of the AR ontology classes and individuals in Protégé. (a) Entire. (b) Authoring tools. (c) Tracking.

Figures 8 and 9 show snapshots of web pages of the developed AR search engine using the Flask framework. Figure 8a is the main page that enables users to search in AR domains by typing keywords in the developed box. On this page, there are two options: the custom

search and the AR search. It should be noted that participants do not need to add the AR term as a search keyword in the developed engine. Figure 8b illustrates the page for visualizing the custom search results in an ordered list. This page, similar to the result page of the Google search engine, shows the results' titles, short descriptions (sinpets), and hyperlinks. Figure 9 depicts the page for visualizing the AR search results in some clusters. This page contains two panels: the above panel for the clusters and the below panel for the ordered list of the results in the selected cluster. On this page, the retrieved results from the Google search engine are clustered according to the AR ontology for enhancing the performance of retrieving and ranking information in the AR domains. It should be noted that the k value that is fixed for all queries was considered five in the user study. This value can be set by developers in the code.

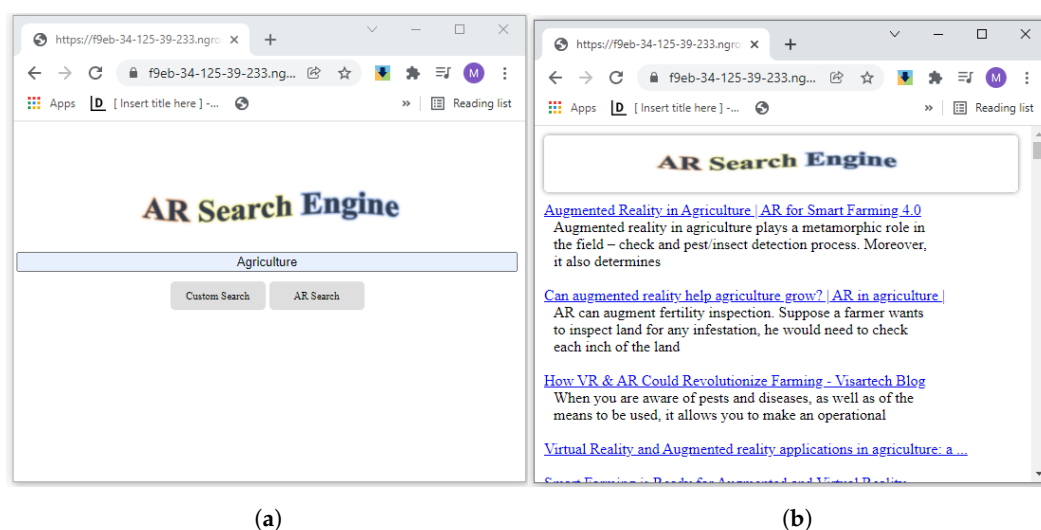


Figure 8. Samples of the interfaces of the developed AR search engine. (a) The AR Search Engine. (b) The custom search.

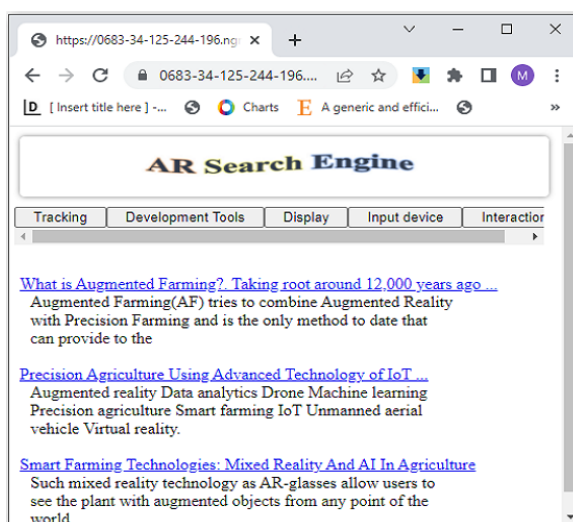


Figure 9. The interface of the AR search.

5.2. Evaluation Results

The participants consisted of 21 females and 17 males, and their average age was approximately 31 years. The participants, who were university students, were from different majors, including civil engineering, geo-informatics engineering, computer science engineering, environmental science, art and graphic design, and agriculture education. Among the participants, 14 participants were familiar with AR technology, and five of

them used an AR application, at least briefly. In addition, five participants already had experience searching for AR in their majors.

The evaluation findings are shown in Figures 10 and 11. We can realize from Figure 10 that the proposed approach, the AR search, outperformed the custom search in the AR domain for all searches. The average completion time for the AR search was 4.21 min, whereas the average completion time for the custom search was 7.30 min for all searches. This means that the AR search takes 42.34% less time to complete the searching task and find the desired result than the custom search. According to Figure 11, the AR search achieved better performance in terms of MRR. The average value of the MRR for the AR search was 0.83 while the average value of the MRR for the custom research was 0.63. Therefore, the participants could find the desired results using the AR search, which was approximately 32% lower rank than the custom search.

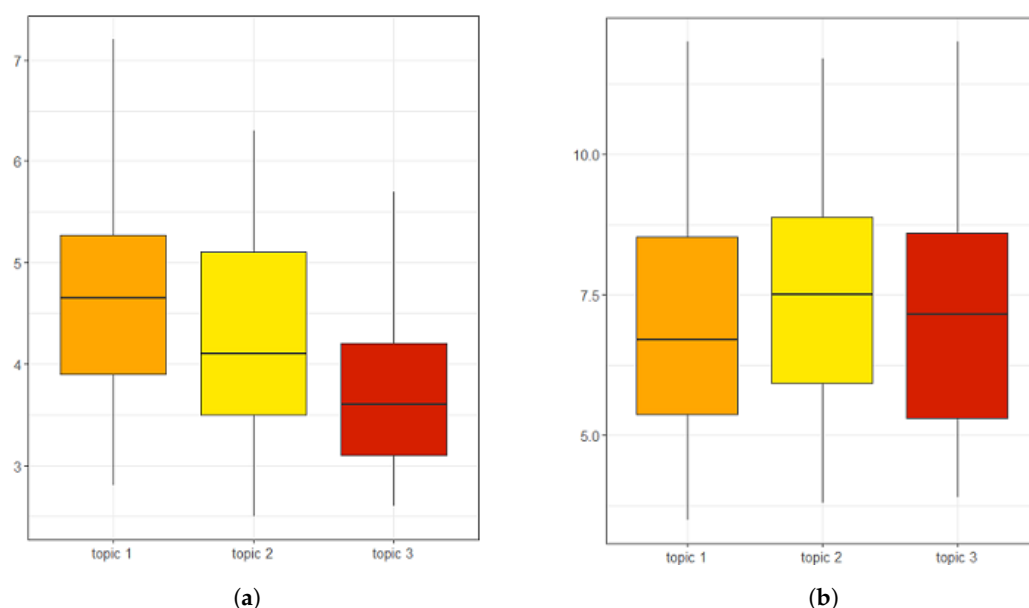


Figure 10. Task completion time results. (a) AR search. (b) Custom search.

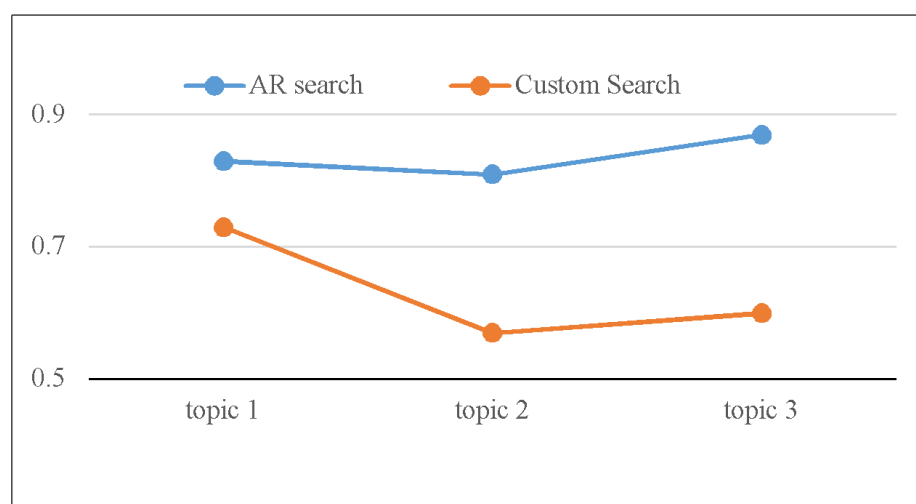


Figure 11. Performance comparison of the MRR metric.

We also compared the completion times between participants who used or were familiar with AR (as group 1) and those who were not (as group 2). The results showed group 1 spent less time than group 2 in both the AR search engine and the custom search engine. The average time of group 1 is 8.2% less than that of group 2 in the AR search

engine, while the average of group 1 is less than 26.3% than that of group 2 in the custom search. Likewise, the average value of the MRR for group 1 outperformed the value for group 2. The MRR value was 9.8% and 31.6% better for group 1 than group 2 in the AR search and custom search, respectively. This might be because the participants (who were familiar) used better keywords to obtain the desired result.

The participants' concerns and expectations that were gathered through interviews include the following: (1) It would be preferable to create an interface that is more user-friendly to interact with. In other words, the interface could be designed, such as Google Image, in which the image categories have a better design on top of the page. (2) The AR search engine should be improved to consider the results that could be categorized into several groups. According to some participants, some web documents contained various AR concepts. (3) The ambiguity in web AR search queries. According to some participants, the quality of the results of the AR search engine should be improved by determining ambiguous queries automatically. As we used the Google API results, the ambiguity still exists in the AR search engine results. For instance, *environment* is an ambiguous query because the API returns the documents that contain AR environments while the user intends to retrieve the AR applications in the natural environment. It is not clear what the user intent is, which can be either an AR environment or a natural environment.

6. Conclusions and Future Works

This paper focused on developing semantic information retrieval in the AR domain in order to access more precise information on the web. To this end, we propose an AR ontology that contains the main concepts of AR, including hardware, software, tracking, interaction, and interface. This ontology not only allows search engines to understand AR concepts, but to cluster search results from AR domains into narrower and more specific groups. Then, we developed the AR search engine based on the ontology-based clustering method, which was developed in this research to analyze the user query and manage the relevant AR documents given the proposed AR ontology. The proposed clustering method is an integrated approach of the concept-based vector space model for AR document modeling based on the AR ontology, the TF-IDF method for weighting the concepts, and the K-means algorithm for clustering AR documents. The developed search engine was evaluated using a user study to determine whether the proposed method effectively enhances the results of the Google search engine in the AR domain.

The results show the better performance of the AR search engine compared to the Google search engine. The AR search engine decreased the time and effort to find the relevant AR documents for the user queries so the participants spent 42.33% less time than the Google search engine. On the other hand, the score of the MRR-based metric revealed that the AR search engine outperformed the Google search engine in the field of AR. The average MRR value for the AR search engine was about 34% higher than the custom search. Finally, the interviews indicated the positive effects of the proposed approach in obtaining a high level of user satisfaction.

Although the proposed approach enhances information retrieval in the AR domain, it can be improved further by considering future developments. Future research could include a disambiguation approach for both syntactic and semantic ambiguities using the proposed AR ontology. The ambiguity in the user queries and web documents created challenges in retrieving accurate results. The AR ontology could be extended to involve other AR concepts, including hand tracking, eye tracking, avatars, and the metaverse. Another future task will be to define more class properties between the classes, such as properties to indicate the relationship between tracking and interaction classes or the relationship between software and hardware classes. In addition, automatic topic labeling for the clusters using the proposed AR ontology can be developed for the search engine to improve efficiency. More evaluations, including how different AR terms help to improve the AR search engine using the ablation experiments, should be conducted to prove the AR search engine's functionalities.

Author Contributions: Conceptualization, M.S. and A.S.-N.; methodology, M.S.; software, M.S.; validation, M.S. and A.S.-N.; investigation, A.S.-N.; resources, M.S.; data curation, M.S.; writing—original draft preparation, M.S.; writing—review and editing, S.-M.C. and T.A.; visualization, A.S.-N.; supervision, A.S.-N.; project administration, S.-M.C. and T.A.; funding acquisition, S.-M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2022-RS-2022-00156354) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation) and the Ministry of Trade, Industry and Energy (MOTIE) and Korea Institute for Advancement of Technology (KIAT) through the International Cooperative R&D program (Project No. P0016038).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the privacy.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Evangelista, A.; Ardito, L.; Boccaccio, A.; Fiorentino, M.; Petruzzelli, A.M.; Uva, A.E. Unveiling the technological trends of augmented reality: A patent analysis. *Comput. Ind.* **2020**, *118*, 103221. [\[CrossRef\]](#)
2. Rezaee, S.; Sadeghi-Niaraki, A.; Shakeri, M.; Choi, S.M. Personalized Augmented Reality Based Tourism System: Big Data and User Demographic Contexts. *Appl. Sci.* **2021**, *11*, 6047. [\[CrossRef\]](#)
3. Park, H.; Shakeri, M.; Jeon, I.; Kim, J.; Sadeghi-Niaraki, A.; Woo, W. Spatial transition management for improving outdoor cinematic augmented reality experience of the TV show. *Virtual Real.* **2022**, *26*, 1059–1077. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Shakeri, M.; Sadeghi-Niaraki, A.; Choi, S.M. Augmented reality-based border management. *Virtual Real.* **2022**, *26*, 1123–1143. [\[CrossRef\]](#)
5. Arici, F.; Yildirim, P.; Caliklar, S.; Yilmaz, R.M. Research trends in the use of augmented reality in science education: Content and bibliometric mapping analysis. *Comput. Educ.* **2019**, *142*, 103647. [\[CrossRef\]](#)
6. Eckert, M.; Volmerg, J.S.; Friedrich, C.M. Augmented reality in medicine: Systematic and bibliographic review. *JMIR mHealth uHealth* **2019**, *7*, e10967. [\[CrossRef\]](#)
7. Han, J.; Kang, H.J.; Kim, M.; Kwon, G.H. Mapping the intellectual structure of research on surgery with mixed reality: Bibliometric network analysis (2000–2019). *J. Biomed. Inform.* **2020**, *109*, 103516. [\[CrossRef\]](#)
8. Yung, R.; Khoo-Lattimore, C. New realities: A systematic literature review on virtual reality and augmented reality in tourism research. *Curr. Issues Tour.* **2019**, *22*, 2056–2081. [\[CrossRef\]](#)
9. Egger, J.; Masood, T. Augmented reality in support of intelligent manufacturing—A systematic literature review. *Comput. Ind. Eng.* **2020**, *140*, 106195. [\[CrossRef\]](#)
10. Paelke, V. Augmented reality in the smart factory: Supporting workers in an industry 4.0. environment. In Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA), Barcelona, Spain, 16–19 September 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–4.
11. Djordjevic, L.; Petrovic, N.; Tosic, M. Ontology based approach to development of augmented reality applications. In Proceedings of the 2019 27th Telecommunications Forum (TELFOR), Belgrade, Serbia, 26–27 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–4.
12. Đorđević, L.; Petrović, N.; Tošić, M. An ontology-based framework for automated code generation of Web AR applications. *Telfor J.* **2020**, *12*, 67–72. [\[CrossRef\]](#)
13. Hervás, R.; Bravo, J.; Fontecha, J.; Villarreal, V. Achieving adaptive augmented reality through ontological context-awareness applied to aal scenarios. *J. Univers. Comput. Sci.* **2013**, *19*, 1334–1349.
14. Eckertz, D.; Möller, M.; Anacker, H.; Dumitrescu, R. Digital Knowledge Base for Industrial Augmented Reality Systems Based on Semantic Technologies. In Proceedings of the 2021 4th International Conference on Information and Computer Technologies (ICICT), Hawaii, GA, USA, 11–14 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 243–252.
15. Sala, H.V.; Leyva, P.R.; Febles, J.P.; Sentí, V.E. *Information Retrieval with Semantic Annotation*; Technical Report; EasyChair: Greater Manchester, UK, 2019.
16. Selvalakshmi, B.; Subramaniam, M. Intelligent ontology based semantic information retrieval using feature selection and classification. *Clust. Comput.* **2019**, *22*, 12871–12881. [\[CrossRef\]](#)
17. Sharma, S.; Mahajan, S.; Rana, V. A semantic framework for ecommerce search engine optimization. *Int. J. Inf. Technol.* **2019**, *11*, 31–36. [\[CrossRef\]](#)
18. Ning, W.; Liu, J.; Xiong, H. Knowledge discovery using an enhanced latent Dirichlet allocation-based clustering method for solving on-site assembly problems. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102246. [\[CrossRef\]](#)

19. Kim, S.W.; Gil, J.M. Research paper classification systems based on TF-IDF and LDA schemes. *Hum.-Centric Comput. Inf. Sci.* **2019**, *9*, 30. [\[CrossRef\]](#)
20. Bukhari, A.; Liu, X. A Web service search engine for large-scale Web service discovery based on the probabilistic topic modeling and clustering. *Serv. Oriented Comput. Appl.* **2018**, *12*, 169–182. [\[CrossRef\]](#)
21. Li, X.; Li, K.; Qiao, D.; Ding, Y.; Wei, D. Application research of machine learning method based on distributed cluster in information retrieval. In Proceedings of the 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), Haikou, China, 5–7 July 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 411–414.
22. Indra, Z.; Adnan, A.; Salambue, R. A Hybrid Information Retrieval for Indonesian Translation of Quran by Using Single Pass Clustering Algorithm. In Proceedings of the 2019 Fourth International Conference on Informatics and Computing (ICIC), Semarang, Indonesia, 16–17 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
23. Logeswari, S.; Premalatha, K. Biomedical document clustering using ontology based concept weight. In Proceedings of the 2013 International Conference on Computer Communication and Informatics, Coimbatore, India, 4–6 January 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–4.
24. Kulathunga, C.; Karunaratne, D. An ontology-based and domain specific clustering methodology for financial documents. In Proceedings of the 2017 Seventeenth International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 6–9 September 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–8.
25. Sandhiya, R.; Sundarambal, M. Clustering of biomedical documents using ontology-based TF-IGM enriched semantic smoothing model for telemedicine applications. *Clust. Comput.* **2019**, *22*, 3213–3230. [\[CrossRef\]](#)
26. Kambau, R.A.; Hasibuan, Z.A. Concept-based multimedia information retrieval system using ontology search in cultural heritage. In Proceedings of the 2017 Second International Conference on Informatics and Computing (ICIC), Jayapura, Indonesia, 1–3 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
27. Janani, R.; Vijayarani, S. Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Syst. Appl.* **2019**, *134*, 192–200. [\[CrossRef\]](#)
28. Zhou, Z.; Goh, Y.M.; Shen, L. Overview and analysis of ontology studies supporting development of the construction industry. *J. Comput. Civ. Eng.* **2016**, *30*, 04016026. [\[CrossRef\]](#)
29. Zheng, Y.; Törmä, S.; Seppänen, O. A shared ontology suite for digital construction workflow. *Autom. Constr.* **2021**, *132*, 103930. [\[CrossRef\]](#)
30. Azuma, R.T. A survey of augmented reality. *Presence Teleoper. Virtual Environ.* **1997**, *6*, 355–385. [\[CrossRef\]](#)
31. Schmalstieg, D.; Hollerer, T. *Augmented Reality: Principles and Practice*; Addison-Wesley Professional: Boston, MA, USA, 2016.
32. Kipper, G.; Rampolla, J. *Augmented Reality: An Emerging Technologies Guide to AR*; Elsevier: Amsterdam, The Netherlands, 2012.
33. Furht, B. *Handbook of Augmented Reality*; Springer Science & Business Media: Berlin, Germany, 2011.
34. Craig, A.B. *Understanding Augmented Reality: Concepts and Applications*; Newnes: Newton, MA, USA, 2013.
35. Haghighi, P.D.; Burstein, F.; Zaslavsky, A.; Arbon, P. Development and evaluation of ontology for intelligent decision support in medical emergency management for mass gatherings. *Decis. Support Syst.* **2013**, *54*, 1192–1204. [\[CrossRef\]](#)
36. Kim, K.; Billingham, M.; Bruder, G.; Duh, H.B.L.; Welch, G.F. Revisiting trends in augmented reality research: A review of the 2nd decade of ISMAR (2008–2017). *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 2947–2962. [\[CrossRef\]](#)
37. Billingham, M.; Clark, A.; Lee, G. A survey of augmented reality. *Found. Trends® Hum.-Comput. Interact.* **2015**, *8*, 73–272. [\[CrossRef\]](#)
38. Palmarini, R.; Erkoyuncu, J.A.; Roy, R.; Torabmostaedi, H. A systematic review of augmented reality applications in maintenance. *Robot. Comput.-Integr. Manuf.* **2018**, *49*, 215–228. [\[CrossRef\]](#)
39. Bae, H.; White, J.; Golparvar-Fard, M.; Pan, Y.; Sun, Y. Fast and scalable 3D cyber-physical modeling for high-precision mobile augmented reality systems. *Pers. Ubiquitous Comput.* **2015**, *19*, 1275–1294. [\[CrossRef\]](#)
40. Chen, M.; Monroy-Hernández, A.; Sra, M. SceneAR: Scene-based Micro Narratives for Sharing and Remixing in Augmented Reality. In Proceedings of the 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Bari, Italy, 4–8 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 294–303.
41. Nebeling, M.; Speicher, M. The trouble with augmented reality/virtual reality authoring tools. In Proceedings of the 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Munich, Germany, 16–20 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 333–337.
42. Tongprasom, K.; Boongsood, W.; Boongsood, W.; Pipatchotitham, T. Comparative Study of an Augmented Reality Software Development Kit Suitable for Forensic Medicine Education. *Int. J. Inf. Educ. Technol.* **2021**, *11*, 10–15. [\[CrossRef\]](#)
43. Vakaliuk, T.A.; Pochtoviuk, S.I. Analysis of tools for the development of augmented reality technologies. In Proceedings of the CEUR Workshop Proceedings, Online, 7–8 January 2021.
44. Pham, D.M.; Stuerzlinger, W. Is the pen mightier than the controller? A comparison of input devices for selection in virtual and augmented reality. In Proceedings of the 25th ACM Symposium on Virtual Reality Software and Technology, Parramatta, Australia, 12–15 November 2019; pp. 1–11.
45. Belyavskii, A.; Sorokin, A.; Khaustov, A. Head-Up Display Systems in Aviation. *Russ. Eng. Res.* **2021**, *41*, 61–63. [\[CrossRef\]](#)
46. Zhang, Y.; Yang, T.; Zhang, X.; Zhang, Y.; Sun, Y. Effects of full windshield head-up display on visual attention allocation. *Ergonomics* **2021**, *64*, 1310–1321. [\[CrossRef\]](#)

47. Sadeghi-Niaraki, A.; Choi, S.M. A Survey of Marker-Less Tracking and Registration Techniques for Health & Environmental Applications to Augmented Reality and Ubiquitous Geospatial Information Systems. *Sensors* **2020**, *20*, 2997.
48. Nizam, S.S.M.; Abidin, R.Z.; Hashim, N.C.; Lam, M.C.; Arshad, H.; Majid, N. A review of multimodal interaction technique in augmented reality environment. *Int. J. Adv. Sci. Eng. Inf. Technol.* **2018**, *8*, 1460–1469. [[CrossRef](#)]
49. Carpineto, C.; Romano, G. A survey of automatic query expansion in information retrieval. *Acm Comput. Surv. (CSUR)* **2012**, *44*, 1–50. [[CrossRef](#)]
50. Avasthi, S.; Chauhan, R.; Acharjya, D.P. Processing large text corpus using N-gram language modeling and smoothing. In Proceedings of the Second International Conference on Information Management and Machine Intelligence, Jaipur, India, 1–2 June 2021; Springer: Berlin/Heidelberg, Germany, 2021; pp. 21–32.
51. Mabotuwana, T.; Lee, M.C.; Cohen-Solal, E.V. An ontology-based similarity measure for biomedical data—Application to radiology reports. *J. Biomed. Inform.* **2013**, *46*, 857–868. [[CrossRef](#)] [[PubMed](#)]
52. Avasthi, S.; Chauhan, R.; Acharjya, D.P. Techniques, Applications, and Issues in Mining Large-Scale Text Databases. In *Advances in Information Communication Technology and Computing*; Goar, V., Kuri, M., Kumar, R., Senjyu, T., Eds.; Springer: Singapore, 2021; pp. 385–396.
53. Avasthi, S.; Chauhan, R.; Acharjya, D.P. Topic Modeling Techniques for Text Mining Over a Large-Scale Scientific and Biomedical Text Corpus. *Int. J. Ambient. Comput. Intell. (IJACI)* **2022**, *13*, 1–18. [[CrossRef](#)]
54. Sinaga, K.P.; Yang, M.S. Unsupervised K-means clustering algorithm. *IEEE Access* **2020**, *8*, 80716–80727. [[CrossRef](#)]
55. Park, K.; Hong, J.S.; Kim, W. A methodology combining cosine similarity with classifier for text classification. *Appl. Artif. Intell.* **2020**, *34*, 396–411. [[CrossRef](#)]
56. Nguyen, T.; Gopalan, N.; Patel, R.; Corsaro, M.; Pavlick, E.; Tellex, S. Affordance-based robot object retrieval. *Auton. Robot.* **2021**, *46*, 83–98. [[CrossRef](#)]
57. Mufid, M.R.; Basofi, A.; Al Rasyid, M.U.H.; Rochimansyah, I.F.; rokhim, A. Design an mvc model using python for flask framework development. In Proceedings of the 2019 International Electronics Symposium (IES), Surabaya, Indonesia, 27–28 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 214–219.