IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Automating Computer Science Ontology Extension with Classification Techniques

NATASHA C. SANTOSA[1], JUN MIYAZAKI[2], AND HYOIL HAN.[3]

[1]Department of Computer Science, School of Computing, Tokyo Institute of Technology (e-mail: santosa@lsc.c.titech.ac.jp)
[2]Department of Computer Science, School of Computing, Tokyo Institute of Technology (e-mail: miyazaki@lsc.c.titech.ac.jp)
[3]School of Information Technology, Illinois State University (e-mail: hyoil.han@acm.org)

Corresponding author: Jun Miyazaki (e-mail: miyazaki@lsc.c.titech.ac.jp).

This paragraph of the first footnote will contain support information, including sponsor and financial support acknowledgment. For example, “This work was supported in part by .....”

ABSTRACT In information technology, an ontology is a knowledge structure consisting of the definitions and relations of information within one or even multiple domains. This semantically represented information is helpful for tasks such as document classification and item recommendation in recommender systems. However, as big data prevails, manually extending existing ontologies with up-to-date terminologies becomes challenging due to the tedious and time-consuming process and the expensive cost of expert manual labor. This study aims to achieve a fully automatic ontology extension. We propose a novel "Direct" approach for extending an existing Computer Science Ontology (CSO). This approach consists of two steps: initially extending the CSO with new topics and using this extended graph to obtain the new topic’s node embeddings as inputs for training classifiers. However, this initial extension still contains many noisy links; therefore, the classifier later acts as a filter and a link predictor. We experiment with various traditional machine learning and recent deep learning models and then compare them using our Direct approach. We also propose two evaluation procedures to decide the best-performing model and approach: the novel Wikipedia-based $F1_w$ score and the total number of resulting links. Furthermore, manual evaluation by four human experts is conducted to conclude the reliability of our proposed approach and evaluation procedure. This study concludes that the Direct approach’s *Gaussian Naive Bayes* model produces the most valid and reliable links, and we, therefore, use it to further extend the CSO with hundreds of new CS topics and links.

INDEX TERMS Classification Methods, Concept Classification, Deep Learning, Link Prediction, Ontology Extension

I. INTRODUCTION

Organizing information is becoming more crucial due to the rapid growth of information. Proper information organization allows for easier access to certain information from a vast set of information. Ontology is one of the popular data structures used to store information efficiently; it is used to store concepts and the relationships among them. Our long term goal is to integrate ontologies into recommendation systems; which would be beneficial in enhancing semantic reconciliation among data sources and improving recommendation explainability. However, a major problem here is the maintenance of the ontology itself, that is, we need to keep ontologies up-to-date while minimizing human intervention. Maintenance means updating an existing ontology with new information. This includes the action of adding new topics and relationships to an existing ontology. Several methodologies exist for ontology maintenance, which often times include corpus annotation, training of information extraction engine, and validation by human experts.

There has been a lot of research on ontology construction in the last two decades, and diverse methods to construct ontology have been proposed. In the early 2000s,[1] various studies tried to produce multiple semiautomatic tools for constructing ontology from raw texts (e.g., news articles, scientific publications, and product reviews) [1], for example, Text2Onto [2] and OntoLing [3]. However, the semi-

[1]When the ontology construction research topic was relatively at its peak of popularity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

automatic approaches for ontology construction still heavily depend on expert human labor, and only decreases the effort for ontology construction by approximately $\leq$50% compared to constructing it entirely in a manual fashion. This is undesirable because methods that require human interventions at any of its sub-process are non-scalable for future changes [4].

Rather than building an ontology from scratch, utilizing an already existing one is preferable as it saves time and effort. In this paper, we propose a "direct" automatic ontology extension approach. Particularly, we experiment with the **Computer Science Ontology**[2] (CSO) [5], that stores information about Computer Science research topics. To the best of our knowledge, this is the most up-to-date and exhaustive Computer Science topic ontology by far. However, the CSO project still relies on semi-manual maintenance. Particularly, it relies on a human-expert to filter-out and cross-check automatic machine suggestion (cf. Section III).

In this paper, we explore and evaluate various approaches to find the most appropriate one for realizing a completely automated ontology maintenance, specifically the ontology extension process. The main contributions of this paper are threefold:

1) We propose a novel "Direct" automatic extension approach. It consists of the direct involvement of new topics during the training of the classification models. This is also the difference between our approach and the existing "Indirect" approach (cf. Section II), which relies on training the classification models on the already-existing ontology before using the selected trained model to classify new topics into the ontology (indirect involvement of new topics during training).
2) We compare the performance of the two approaches via a novel Wikipedia-based evaluation procedure as well as the total number of valid links resulted by the classifiers. In the past, expert human cross-checking was required when updating an existing ontology. However, human labor is time consuming and prone to inconsistency. Wikipedia itself is very comprehensive, and not to mention updated very often due to its open-collaboration system. Furthermore, there have also been multiple works that relied on Wikipedia as their main source of information (see Section II-H). In addition, we conducted a manual meta-evaluation by employing four experts and showed that the experts tend to have high agreement with the links resulted by the top three best performing classifiers (decided based on the number of total links resulted). This shows by using only the Wikipedia-based evaluation procedure and the number of total valid links, we can achieve a conclusion very close to human-involved manual evaluation.
3) As the result of our effort, we present an extended CSO ontology with hundreds of new links and topics.[3]

[2]https://cso.kmi.open.ac.uk/downloads

[3]We plan to make our extended CSO ontology publicly available.

A. RESEARCH QUESTIONS

This paper aims to answer the following research questions:

1) Is a complete automated ontology extension approach feasible?
2) Which approach, direct or indirect, is better at recognising new topics and properly linking them into the CSO ontology?

B. PAPER STRUCTURE

The remainder of this paper is organized as follows. Section II describes a literature review that includes the general definition of ontology and several related works within the field of ontology extension. In Section III, we describe the ontology data used and how this ontology is obtained. The methodology of each approach proposed in this study is explained in Sections IV and V. Section VI is the presentation of the details of the experiments and an in-depth discussion of performances. Finally, the conclusions of this study and further study are explained in Section VII.

II. RELATED WORKS

A. ONTOLOGY IN COMPUTER SCIENCE

Man [6] defined ontology in Computer Science as a formal representation of knowledge consisting of a set of concepts under a domain and the relationships between these concepts. In other words, an ontology is a description of a domain. This method of knowledge representation has been used in several fields including Artificial Intelligence, Semantic Web, Systems Engineering, Software Engineering, Biomedical Informatics, Library Science, Enterprise Bookmarking, and Information Architecture. In this paper, ontology is used as a knowledge representation of scientific paper topics (i.e., relationships between topics) under the Compute Science field.

Domain-specific ontology is usually built depending on the goal of the builder, and this type of ontology would focus only on one specific field and/or domain. In the case of preexisting ontologies, **ontology maintenance** such as extending ontology with updated concepts should be done as well. For the simplicity of referring the idea of maintaining existing ontology with new up-to-date information, ontology maintenance is referred to as **ontology extension** in our study.

B. TAXONOMY EXTENSION

Before proceeding to the next section, it is important to mention that there has been researches done on methods of taxonomy extension. Taxonomy extension, like ontology extension, is the action of enriching an already existing taxonomy with new information. One example is the work done by Shen et al. [4] called HiExpan. By using HiExpan, users are allowed to provide the taxonomy constructor with a seed taxonomy (in the form of a tree graph) as a guide for further expansion of the same taxonomy either horizontally (i.e., taxonomy width expansion) or vertically (i.e., taxonomy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

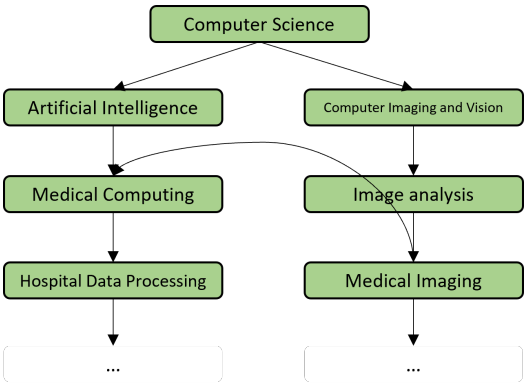


FIGURE 1. Unique structure of Computer Science Ontology.

depth expansion). Another example is TaxoGen, a work by Zhang et al. [7] which used an unsupervised approach to construct a topic taxonomy. Even though their work aims to build a taxonomy from scratch from raw texts, the idea can be further extended to an extension task as well by involving a seed taxonomy in some way just like what is done in HiExpan [4]. Here text embedding is used in a clustering approach to recursively construct the taxonomy from top (more general root) to bottom (more fine-grained descendants) until a specified maximum number of levels is met.

Despite the excellent potential of applying either of the two earlier mentioned taxonomy extension methods, the approaches used in this paper are incompatible to them in a couple of aspects. The goal of extension in this paper is aimed to the CSO which **does not have the traditional format of a taxonomy**. The form of the CSO is a combination of a tree structure and an ordinary graph – hierarchy is present (supertopic-subtopic relationship), but at the same time, it does not adhere to the classic hierarchical rules of a tree or taxonomy. Relationships within the CSO is far more complicated and more like the relationships between nodes in a graph. For example, a certain node in the CSO can have multiple roles: child, grandparent, and even the grandparent of its grandparent. An example of this format taken from the original CSO is shown in Figure 1

In Figure 1, it is shown that the topic "Medical Imaging" is within the same level as "Hospital Data Processing", however, at the same time, it is the parent of "Medical Computing" and therefore, it is the grandparent of "Hospital Data Processing". Furthermore, the aforementioned taxonomy extension methods rely on a key term extractor which is still patchy. Some limitations of using automatic terms extractors are explained in Section II-C.

C. AUTOMATIC TERMS EXTRACTION

Just like the name, an automatic term extractor aims to automatically extract important terminologies from a given corpus [8], usually consisting of raw texts. There has been a lot of research on this topic, and some of the produced automatic term extraction algorithm and tools includes SemCluster [8], AutoPhrase [9], [10], YAKE [11]–[13], and RAKE [14].

The use of terms extractor algorithms or tools is very popular among extensions of any kinds of graph. In the case of HiExpan [4], AutoPhrase [9] is used to extract important terminologies from raw texts.

In this paper, the YAKE [11]–[13] term extractor is used for extracting important keywords from Wikipedia articles. The terms extracted are used for the creation of the **Wikipedia Graph** that is used as a form of ground truth for our ontology extension approaches.

D. SEMIAUTOMATIC ONTOLOGY EXTENSION

Research on Ontology Extension mostly focuses on the semiautomatic method. This method essentially consists of two main steps. *First*, the model automatically extracts ontology concepts from an information source (e.g., text documents) while providing the predictions or probabilities of the supposed location of the information within the ontology. *Second*, the model relies completely on human intervention for checking, correcting, and confirming the automatically generated results from the earlier step.

Some works categorized under semiautomatic ontology extension include the work by Ayadi et al. [15] that proposed a semiautomatic ontology extension approach by implementing a deep-learning-based extractor to extract new ontology concepts alongside their relations and attribute instances. After the new information has been achieved, experts are relied on to completely check the outputs of their deep learning extraction model.

Another work is done by Liu et al. [16] where they proposed a semiautomatic ontology extension and refinement. In their work [16], they use WordNet lexical dictionary to create a semantic network from a given seed ontology. This semantic network is then processed by spreading activation, which decides if a concept should be extended to the seed ontology or not.

Finally, Novalija and Mladenic [17] use a combination of text mining and a user-oriented approach to semiautomatically extend a cross-domain Cyc ontology [4]. They followed up this work with another publication [18] in which they extended their previous methodology with the idea of combining ontology content and structure for ontology extension.

E. BASELINE: A RECENT ATTEMPT ON AUTOMATIC APPROACH

Althubaiti et al. [19] claimed to have attempted the realization of a fully automated ontology extension via employing an artificial-neural-network-based unigram classification. In their work, they can automatically classify new disease ontology concepts to their appropriate super-concepts only through the utilization of each concept's unigram vector representations. To the best of our knowledge, the work in

[4]Cyc, https://bit.ly/2WA4GQP

Althubaiti et al. [19] is the only one thus far that claimed an attempt of fully automating an ontology extension task.

In their work, Althubaiti et al. [19] did not use the complete disease ontology terminologies as a reference for their extension. Instead, they only selected a few parent topics as potential labels for the classified new topics. They separated their classification types into four: disease (number of classes = 2), infectious disease (number of classes = 5), anatomical disease (number of classes = 13), and infectious + anatomical disease (number of classes = 17).

Nevertheless, this work by Althubaiti et al. [19] inspired one of the approaches we propose in this paper as well as in our previous paper [20] where we observed the performances of flat and hierarchical classifiers when applied to the method proposed by them. From this, we conclude and decide to only use flat classifier models due to their better performance scores explained in [20]. In general, to extend our previous work, we introduce a new approach called the "Direct" approach. This new approach is then experimented on and compared to the approach proposed by Althubaiti et al. [19] which we call the "Indirect" approach in this paper.

For this **"Indirect"** approach, we adopt the framework proposed by Althubaiti et al. [19], which essentially consist of two major steps: training a topic classifier on the original seed ontology, and using the trained topic classifier to classify new concepts to their appropriate super-concepts within the seed ontology. New topics are not yet introduced during the training process, preventing a direct involvement of these new topics.

F. TOPIC CLASSIFICATION

Topic classification is a classification process that categorizes topics into singular or multiple set of predefined labels or supertopics in the case of our study. Unlike text classification, topic classification differs in the length of the text document used as input with most documents consisting of only at most five-worded topics. In ontology extension, inspired by the idea of the work done by Althubaiti et al. [19] who implemented a text classification method, the topic classification method may also allow automation of ontology extension. In this case, new topics are considered as text and their most probable supertopic(s) are considered as their corresponding labels. In our work, to determine the best topic classifier, two types of text classifiers are implemented on a topic data consisting of very short texts: flat and hierarchical topic classifiers. For the features of each topic, GloVe word embedding is used to represent the topics as topic vectors (topic embedding).

G. LINK PREDICTION

The goal of link prediction is to use a trained classifier for predicting whether the presence of a link between two nodes in a network is true or false. The majority of link prediction algorithms rely on the use of similar information between two nodes for concluding the final prediction result. Aside from text classification, link prediction is also expected to be able to allow for the automation of the ontology extension process.

For link prediction, Node2Vec's node embedding is used as the node information as well as the input features for the classifier. This is different from using word embeddings as data features, because instead of using the information only from a single node, Node2Vec takes into consideration the information of a node and its neighbors within its neighborhood.

Link prediction has been popular amongst Social Networks [21], [22]. For instance, Murata and Moriyasu [23] in their work, used weighted proximity measures of the social networks along with the social network's existing links' weights in Question-Answering Bulletin Boards to predict the evolution of social networks. Another use of link prediction in social media is done by Papadimitriou et al. [24], here they proposed a faster and more accurate link prediction approach for friend recommendations. Fire et al. [25] proposed structural features that are simple and easy to compute for identifying missing links. Aside from these three works, there are still many other works out there regarding link prediction in social media.

We believe the extension of ontology works in a very similar manner to the link prediction in the field of social networks. In ontology extension, we consider that the ontology nodes and new information nodes are the same as the user nodes in a social network. In a social media network, two users within the network may not be connected, however, by using link prediction, it can be predicted if there may be a connection between the two users in the future because of their similarities. This illustration can be applied to ontology networks – link prediction can discover hidden links within an existing ontology network, therefore allowing the enrichment of ontology with more relationships – which is the aims of our study. We describe our approach of using link prediction for ontology extension in Section IV.

H. WIKIPEDIA IN THE RESEARCH COMMUNITY

Wikipedia has been used widely within the research community due to its easily accessible data, as well as the information the website can provide through their seemingly unlimited article collections. By using Wikipedia, different fields of research produced several interesting ideas. A few examples include the work of Boyd [26] who developed a grammatical error correction system for the German language using a gold corpus obtained from the Wikipedia revision history. An enriched Industry 4.0 dictionary[5] has also been produced by Chiarello et al. [27] who in their work use Wikipedia to extract information regarding important connections between technologies and allow real-time update of descriptions of their dictionary.

More examples of the use of Wikipedia for studies closely related to the proposed study can be seen in the work of Kim et al. [28] where they constructed a technological ecol-

[5]Industry 4.0 dictionary: www.industria40senzaslogan.it

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

ogy network using Wikipedia hyperlinks obtained from the Wikipedia database. In their work, they also applied link prediction methods for developing three predictive indicators of technological convergence. Still in the scope of Wikipedia for graph, Ni et al. [29] implement Wikipedia alongside the Yahoo! knowledge graph to improve an entity recommendation task. Finally, Azad et al. [30] use Wikipedia and WordNet as a new approach for Query Expansion.

In our work, Wikipedia articles are used to create a ground truth knowledge base called *Wikipedia graph*, which is specifically used for evaluation. Section VI-B4 explains in more detail regarding this method of evaluation.

III. COMPUTER SCIENCE ONTOLOGY

The final outcome of this paper is an extended and enriched Computer Science Ontology (CSO) with up-to-date topics and relationships obtained from various classification techniques. The CSO itself is a topic ontology containing scientific topics of Computer Science publications. Salatino et al. [5] automatically generated the CSO by applying the Klink-2 algorithm [31] to the Rexplore dataset [32]. But despite being automatically generated, Salatino et al. [5] claimed that there are also manual revisions done by experts, therefore categorizing the CSO as a semiautomatically generated ontology. The CSO consists of several root topics including Computer Science, Linguistics, Geometry, and Semantics. Despite these various root topics, the main root of their ontology is Computer Science. Following Salatino et al.'s [5] description of the CSO as a large-scale *taxonomy* of research areas, this ontology's contents are occasionally explained with taxonomy terminologies (e.g., children, parent, and descendants).

A. STRUCTURE OF CSO

The CSO in general, like any other graph knowledge structure, consists of triples. With source nodes (subjects) representing the parent topic, predicates (Pred.) representing the relationship between the source nodes and the target nodes, and target nodes (object) represent the direct child of the parent topic. Table 1 is a sample of the "parent of" relationship obtained from the CSO's website[6]. In our study, we focus on the "parent of" relationship in the CSO.

Subject	Pred.	Object
computer science	parent of	artificial intelligence
computer science	parent of	bioinformatics
computer science	parent of	computer aided design
computer science	parent of	computer hardware
computer science	parent of	computer imaging and vision
computer science	parent of	computer networks

TABLE 1. First five *"parent of"* predicate of the CSO's "computer science" topic.

For illustration of the actual format of the ontology, the CSO is illustrated in its website as a tree structure. However, different from a tree, its structure is more a combination of a graph and a tree, where hierarchy exists, but with the enabling of object nodes being both children of their respective subject nodes, as well as grandparents of other nodes within the same level of their respective subject nodes. This situation often occurs in the lower levels of the ontology.

[6]More details on the CSO's "computer science" topic: http://cso.kmi.open.ac.uk/topics/computer_science

B. USING CSO AS DATASET

In this study, we mainly focus on the "parent of" relationship. The CSO's Computer Science root is used along with all of its direct children and each of their descendants. The entirety of the CSO topics are used and processed in two different manners, as follows:

1) For the first processing, the ontology is automatically reformed in a way that is appropriate for the text classification models by applying a **limitation rule** explained in Section IV-A1.
2) For the second processing, an **initial extension** of the CSO is given to the models. This is done by combining the CSO with the Topics Graph. The steps taken to obtain the Topics Graph is explained in detail in Section V-B2).

IV. METHODOLOGY: THE INDIRECT APPROACH

The idea of this first approach is to: (1) train a multilabel text classifier on data obtained purely from the seed ontology, CSO, and (2) select the best performing classifier to classify the direct parents, or supertopics of the new topics, based on the characteristics of the seed ontology learned by the classifier.

The indirect approach is largely influenced by the work of Althubaiti et al. [19]. In their work, they showed the potential of complete automation of ontology extension. They applied a text classification method for extending a disease ontology with new links between the existing disease categories (existing topics) to the new diseases (new topics).

A. SEED ONTOLOGY AND DATA PREPARATION

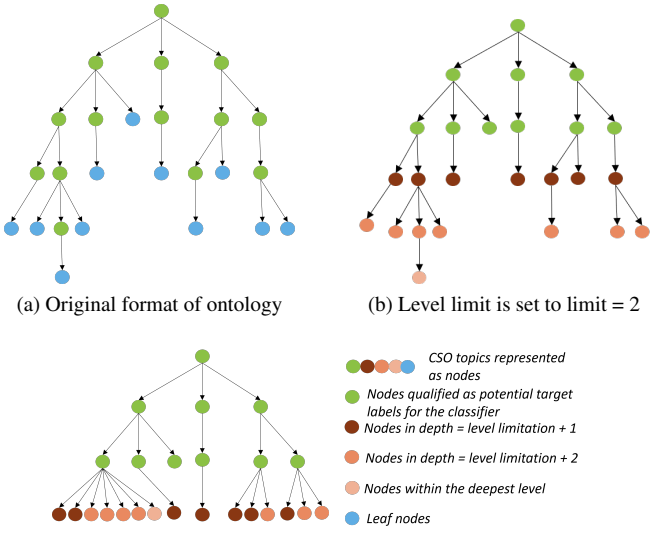
Usually, a text classification task uses vector representations of documents consisting of hundreds of words as features. This task of our study differs in terms of the documents used. Instead of representing a complete text document as a vector, we use vector representation of topic phrases obtained from the CSO. These topic vectors are used as the experiment dataset for both training and testing each topic classification model. This is because the aim of our study is to classify only new topic terms extracted from up-to-date scientific publications (instead of larger text documents consisting of more than ten words) into their appropriate position within the original CSO. This subsection describes the CSO preparation and the steps taken to obtain the vector representation of the CSO topics.

1) Classifier Friendly Ontology Format

We employ a *level limitation rule* to the original ontology format: **we set the maximum level of potential parents to**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

1, this means only the root topic (Computer Science) and all topics within one level under the root topic are considered as potential parents. The rest, the descendants of the topics from level two or one, are set as direct children of the topics within level one. Figure 2 illustrates how level limitation works when *limit* is set to 2.



(a) Original format of ontology (b) Level limit is set to limit = 2

(c) Final level-limited ontology format

FIGURE 2. *Level limitation rule* process applied to the CSO with *limit* = 2.

For a more detailed explanation, each of the three sub-figures in Figure 2 are described as follows; (1) Figure 2a illustrates the original format of the ontology. Here, the green nodes represent all potential target labels for the classifier, since in the case of this study, as long as a node has at least one child, then it will be considered as a parent node, which are set as labels in the experiment dataset. (2) In Figure 2b, the *level limitation rule* is applied. Here, the brown, orange, and lighter orange colors represent the depth of the nodes, and the green color represents the nodes that are within the limitation rule, in this case limit = 2. (3) Figure 2c shows the final form of the ontology after it is flattened based on the set *level limitation rule*. Here, the descendants of the nodes in the second level are all flattened and become the direct children of the second level nodes.

We use the level limitation rule to increase the performance of the machine learning classifier. A problem with the CSO is in some cases, a parent topic only has one child topic (sub-topic), which later becomes only one training example for this specific parent topic (label). While in other cases, a parent topic can have hundreds of children. This results in a skewed data and machine learning methods do not perform well with this kind of extreme data. In the original CSO, topics nearing the leaf level are more repetitive and specific. These topics are more appropriately placed underneath an existing parent topic which are more general instead of being made into a new set of potential parents. The features used in this approach are topic embeddings, where information is not as rich as a larger text (containing hundreds of words per document) embedding. Complication classification targets (too many potential parents) and limited information from the data is a poor combination for training a machine learning classifier.

2) Topic Dictionary

We build a topic dictionary based on the CSO. This dictionary includes the connections between a subtopic with its supertopics (or parent topics), purely obtained from the CSO. Before applying the level limitation rule to the CSO, the topic dictionary consists of 32,043 topic connections. This topic dictionary is used as a guide for deciding the classification examples and labels among topics. In our experiment, the subtopics are set as classification examples, and the supertopics are used as classification labels.

3) Topic Embeddings

From the topic dictionary, any multi-word topics will be split and individually represented with their corresponding word vectors before finally combined. We use a pretrained GloVe model to obtain the word representation of each word within the multi-word topics. We build the topic embeddings by combining multiple word vectors of the words a topic contains.

Specifically, for the framework applied in this study, it requires the 840B pre-trained GloVe model [33] [7]. For each of the topics, they will be represented by their word vectors obtained from the pre-trained GloVe model. In the case of n-gram topics (n-worded topics e.g., 'Artificial Intelligence'), each word will be represented by its corresponding word vectors (e.g., word vectors of 'Artificial' and Intelligence' separately) before they are combined by vector averaging. The output of this step is the list of topic vectors ready to be classified.

B. FLAT CLASSIFICATION

We adopted the classification framework proposed by Althubaiti et al. [19] and propose our Flat Classification Framework shown in Figure 4. As described in this section, we create topic vectors for topics in the topic dictionary and feed the topic vectors to a classifier we chose for experiments. For the classifier, multiple machine learning models (*Logistic Regression, Gaussian Naive Bayes, K Nearest Neighbor, Decision Tree, Random Forest, Support Vector Machine, and Multilayer Perceptro*) are experimented with.

The final step of the framework is to classify topics into their appropriate direct parents in the CSO.

V. METHODOLOGY: THE DIRECT APPROACH

We propose the **"Direct"** approach that is opposite to the "Indirect" approach. Here, hundreds of new Computer Science topics are first obtained from paper abstracts and then directly connected to the seed ontology through intersecting topics as

[7] Pre-trained GloVe, https://stanford.io/3cAbqY5

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

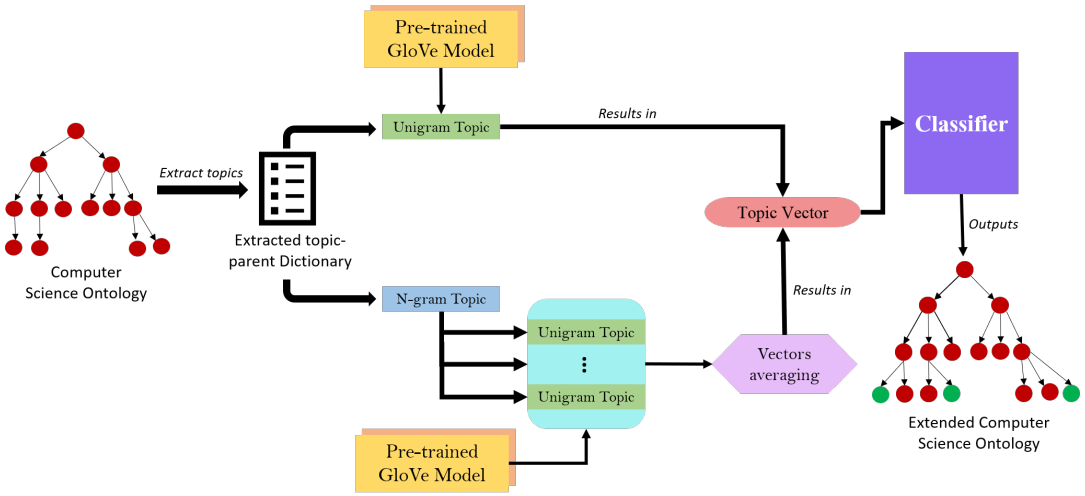


FIGURE 3. Architecture of our Flat Classification Framework

an *initial extension* of the ontology. This undoubtedly results in many noisy links, which is why for this approach, the use of classifier is slightly different than the "Indirect" approach. Here classifiers have two functions: to filter noisy links, and to predict hidden links. Additionally in this approach, we obtain the topic vectors for training and testing the classifiers from a node2vec node embedding model trained on the initially extended ontology.

Several classification models are experimented on using this approach, and all of these classifiers are categorized into the *flat classification* category in the "Indirect" approach since level information is not taken into consideration. The reason is because by initially extending the CSO Graph, level information has been implicitly taken into consideration. Section V-D explains more about this matter.

The Direct approach consists of two modules: a *Filter Module* and a *Evaluation Module*. The initial performance of the *Filter Module* can be measured using *Filter degree*, followed by a further evaluation in the Evaluation module (see Section VI-B2) by using the *Wikipedia graph-based precision, recall, and F1* metrics. The process flow from the input ontology to the output of the Filter Module is illustrated in Figure 4. All data preparation methods used for the Direct approach are explained in the subsections below and details of the Filter Module are explained in Section V-E.

A. NEW TOPICS DATA FROM PAPER ABSTRACTS

To obtain as many new Computer Science topics as possible, a total of 5,798 paper abstracts from multiple proceedings in the ACM digital library[8] published from 2018 to 2020 (occasionally year 2021 if available) are scraped. The reason for choosing this time range is since, to the best of our knowledge and during the time of working on this study, the CSO's most recent major update was done in 2018. From each of the scraped paper abstracts the *title, abstract,* and *tags* information are obtained. These three types of information of the papers are called the *sources* of either new topics or CSO topics. Out of these sources, the *tags* information is the main source used to obtain the actual new topics for extending the CSO, while the other sources are used to obtain topics that already exist within the CSO. This action helps find as many *initial* connections as possible of the new topics to the CSO. Section V-B2 explains collecting new topics in detail.

[8] ACM digital library, https://dl.acm.org/

B. GRAPH PREPARATION

1) Initial Extended CSO

In the Direct approach, the original CSO is initially modified by giving it direct connections (i.e., direct links) to new Computer Science topics obtained from thousands of recently published papers (from 2018 until 2020). To prepare the data, two graphs are used, namely, the original CSO Graph and a new graph consisting of new topics called **Topics Graph**. These two graphs are then combined to create a larger initial graph.

2) Topics Graph

For simplicity of explanation, starting from this section, topics that have already existed within the original CSO are to be called **CSO topics**, while **new topics** are used to represent the topics that are entirely new or do not exist within the original CSO Graph. The Topics Graph contains all possible links between CSO topics and new topics.

The new topics are obtained from recently published papers' titles, abstract, and tag sections. As explained in Section V-A, selecting new topics is relied on the *tags* of each paper pages in the ACM Digital Library since authors tag their papers manually and therefore we can avoid unnatural topic terms generated automatically.

Aside from the tags, abstracts and titles are used to find the existing CSO topics that may appear in the paper together with the new topics. Once we find the existing CSO topics

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

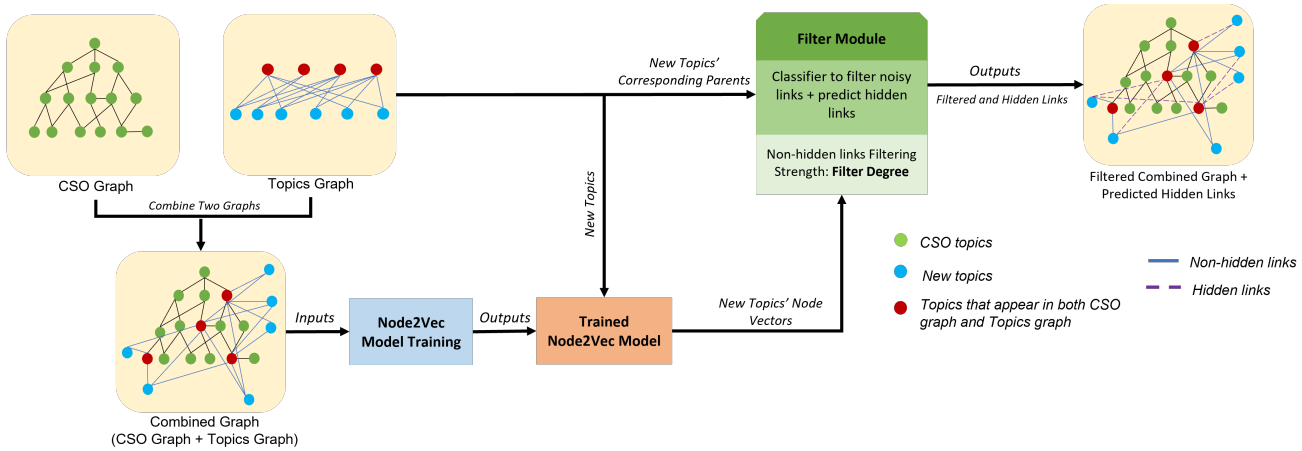


FIGURE 4. Process of the Direct Approach's main module, the Filter Module.

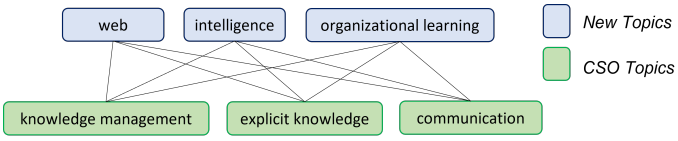


FIGURE 5. Format of Sub-topic Graph.

and new topics together in the same paper, we connect them to build sub-topic graphs.

A simple real example of the format of one sub-topic graph from a paper is illustrated in Figure 5. In this graph, every new topic is connected to the existing CSO topics. Essentially, we connect (i.e., link) the new topics to the existing CSO topics as long as both topics appear together in the same paper. The complete Topics Graph consists of multiple sub-topic graphs obtained from multiple papers.

Figure 6 illustrates the steps taken to initially extend the Computer Science Ontology with new topics extracted from newly published papers. The idea is to simply combine the Topics Graph with the CSO Graph via mutual topics. Since the Topics Graph already contains connections between new topics and CSO topics, the process of combining both graphs is easy to do and the combined CSO–Topics Graph is then used to obtain new topic features that are to be inputted into the *Filter Module*.

C. TOPICS GRAPH'S TOPIC DICTIONARY

Like the topic dictionary used in the "Indirect" approach, we build a topic dictionary to connect (i.e., link) between new topics with their corresponding CSO supertopics. In the "Direct" approach, the topic dictionary is built based on the Topics Graph. Therefore, instead of using all connections within the original CSO, only the CSO topics related to new topics are used as the classification labels. This strategy significantly decreases the number of potential labels compared to the "Indirect" approach, which uses all CSO topics for labels.

D. IMPLICIT LEVEL INFORMATION

As shown in Figure 6, combining the Topics Graph with the CSO Graph involves placing the new topics within the appropriate levels under their corresponding CSO supertopics in the CSO Graph. Since the experiment data are based on the Topics Graph, which consists of both the new topics and the CSO topics, the final classification results would automatically be placed within the appropriate levels as well, under their corresponding CSO supertopics. Therefore, the level information is implicitly taken into consideration in the "Direct" approach, and only flat classification models are implemented.

E. FILTER MODULE

The filter model is the main module of the *"Direct"* approach. The Direct approach is named after *"Direct New Topic"*. The filter module includes a classifier that has two roles: **Classifier to filter noisy links** and **Classifier to predict hidden links**. Figure 4 shows the process of the filter module. The input of the filter module is a set of node vectors obtained from the initially extended CSO Graph along with each of these nodes' supposed parent(s). The output of this module is the filtered CSO–Topics Graph, which consists of the remainder links after filtering, as well as predicted hidden links.

1) Classying Filter Noisy Links

As explained in Section V-B2, the method used for creating the Topics Graph introduces noisy links in the new topic data, i.e., some of the links in the Topics Graph are probably not for connecting to new topics in Computer Science. Due to the noisy condition of the data, we introduce a module to filter out as many of these unnecessary links as possible by using the characteristics of the existing topics in the CSO Graph.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

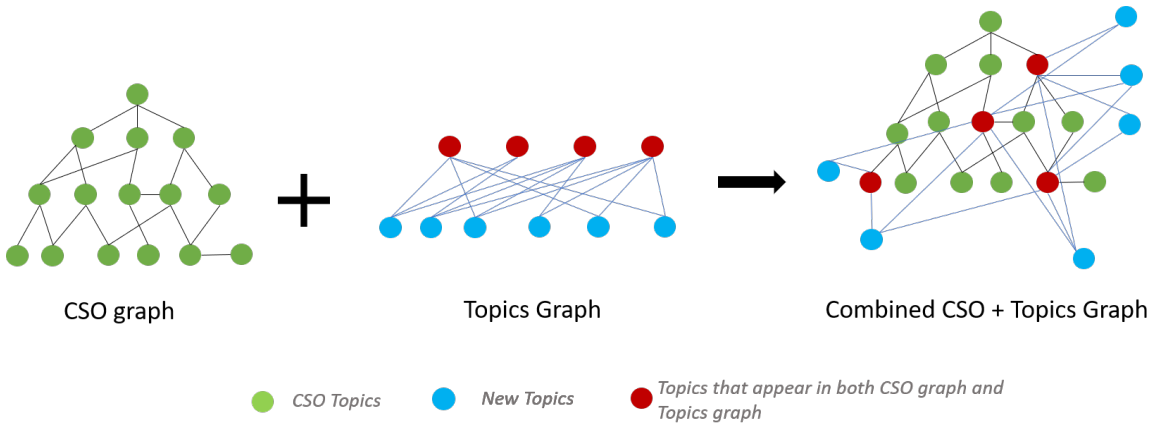


FIGURE 6. Initial Extension of the Computer Science Ontology with new topics obtained from newly published papers.

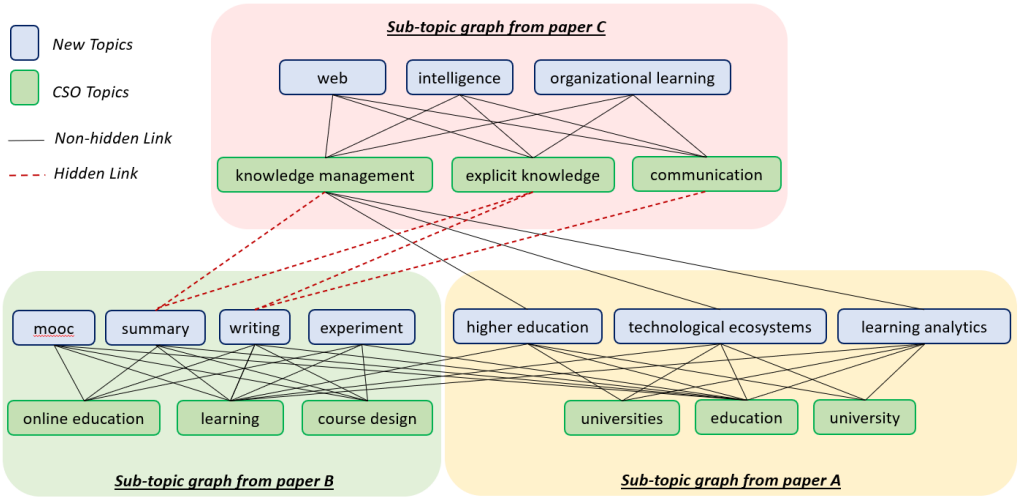


FIGURE 7. Illustration of Hidden Links in the Topics Graph.

Given a link data with their features, we use a classifier to decide whether or not a link is a noise. In this paper, a link prediction approach is implemented and Node2Vec [34] is used to obtain the features of each node from the combined CSO–Topics Graph. If the classifier classifies the presence of a link between two nodes as *False*, then this link can be assumed as noise and can be removed from the combined CSO–Topics Graph.

We experiment with various classifiers and select the best model suited for this task from how well it can filter out the noise links. In other words, the best model is the model that can filter out links moderately, (i.e., not filtering out too many or too few, and is able to smartly select the most links with the most potential to be valid), and achieve the best Wikipedia-based $F1_w$ score in the final *Evaluation Module*.

In Figure 4, it is shown that the filter moderation is initially measured by a degree called *Filter Degree*, which indicates the percentage of the links filtered out by a chosen classifier.

2) Classifying Predict Hidden Links

Aside from filtering out noisy links from the combined CSO-Topics graph, we define hidden links that come from neither the Topics Graph nor the CSO Graph and can represent the connection between one topic to another topic in an entirely different paper, as shown Figure 7 for the visual illustration of hidden links amongst sub-topic graphs. We determine the validity of hidden links by comparing them with the links within the Wikipedia Graph (see Section VI-B3) and describe how it works in the **Evaluation Module**. Figure 8 shows the relationships among various links that appeared in our study.

VI. EXPERIMENTS AND RESULTS

For our study, the latest version of the CSO is used (v.3.2, updated by the creators of the CSO on 17 of June 2020) and downloaded from the CSO portal website[9] in **November 2020**. The CSO is used as a seed ontology for both the "Indirect" and "Direct" approaches. Slight modifications are

[9]CSO Portal, https://cso.kmi.open.ac.uk/downloads

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

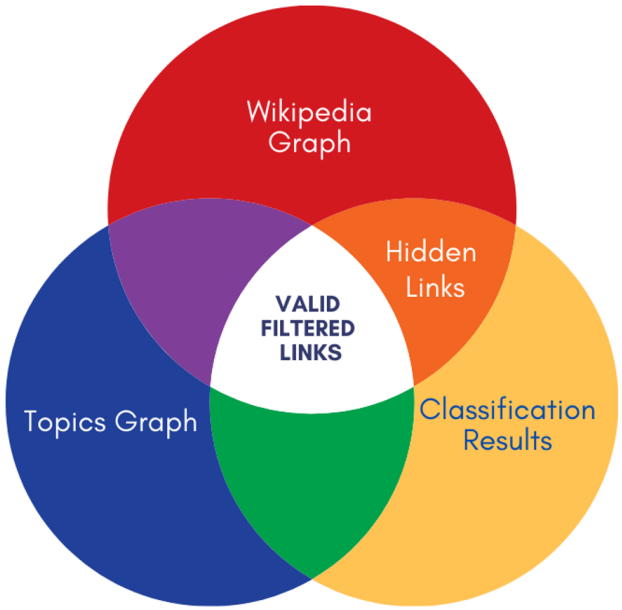


FIGURE 8. Relationship between the links in the Topics Graph, Wikipedia graph, and classifier's predicted links.

done to the original CSO format, specifically for the "Indirect" approach where the *level limitation* rule is applied (level limitation = 1 and level limitation = 2).

A. INDIRECT APPROACH

1) Data Preparation

Inspired by the work of Athubaiti et al. [19], we adopted and implemented a topic classification using the CSO. To make the data suitable for this approach, all topic terms from the ontology are first extracted. For topics that consist of multiple words, they are separated into individual words, and each of these words is represented with their corresponding word vector obtained from the pre-trained GloVe model. Finally, all word vectors are combined into one using vector averaging to obtain the final *topic vectors*. For example, the topic 'Computer Science' are split into 'Computer' and 'Science', then the vectors for each 'Computer' and 'Science' are obtained before combined to represent 'Computer Science' as a whole.

These topic vectors are used for training and testing the data, and in the case of flat classification, the data also consists of each of the topic phrase's direct parents as classification labels. Table 2 shows a sample of the data format where **Vector_rep** is the 300 dimension vector representation of the **Topic**. Since all of this study's topic classification models are multi-class and multi-label, the data also includes information regarding topics with multiple supertopics.

2) Evaluation Metrics

Three evaluation metrics are employed, *precision, recall, and F-measure* (F1 score) for this experiment.

Topic	Vector_rep	Supertopics
robotics	-0.744,..., 0.169	computer science

TABLE 2. A sample of the data format.

While the Direct approach uses the Topics Graph and the Wikipedia-based performance measure, the Indirect approach does not. To have fair comparison to the Direct approach, a specific evaluation scheme is applied to the Indirect approach. To achieve the Wikipedia-based performance measure, the Indirect approach's classification models classify a set of entirely new topics obtained from the Topics Graph, and the resulting new links from this classification task are validated using the Wikipedia-based performance metrics; $precision_w$, $recall_w$, and $F1_w$. Each of these metrics is explained in Section VI-B4. The score as well as the resulting new links and new topics comparison of the two approaches are presented in Section VI-C.

3) Performance Scores of Flat Classification Models

For the Indirect approach, several machine learning models are used, i.e., *Logistic Regression (LR), Gaussian Naive Bayes (GNB), K Nearest Neighbor (KNN), Decision Tree (DT), Random Forests (RF), Support Vector Machine (SVM), and four versions of Multilayer Perceptron (using 10 (MLP_10), 50 (MLP_50), 100 (MLP_100), and 200 (MLP_200) hidden layers)*.

Three types of data are also used in this study: one is based on the original structure of the CSO, and the rest are based on the modified structure of the CSO. The reasons why we modified the structure of CSO is to obtain the number of classes that is as close as possible to the largest number of classes used in the work of ALthubaiti et al. [19] for fair comparisons. In their work, they used a maximum 17 number of classes.

For our experiment **without the level limitation**, the CSO's original structure is used. The vector representations of each word in the topic phrase are combined using a simple vector averaging approach for the input format. Table 3 shows the performances of each model trained on the dataset based on the original CSO structure without applying the *limitation rule* (*no_limit*). For this CSO format, there are 4,241 potential parents (classes) for this version of CSO.

For our experiment **with the level limitation** explained in Section IV-A1, the same input format as the previous experiment is also used in this experiment. There are two versions of the modified CSO, namely, CSO with the level limitation set to 1 without using any synonyms (*limit1_19*), and CSO with the level limitation set to 1 with synonyms included (*limit_30*).

Table 3 shows each implemented classification models' performance on the data obtained from the modified CSO structure. The items in **bold** show the best performing model and its score for each of the classifier's versions.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

Model	CSO Versions								
	Limit1_19			Limit1_30			no_limit		
	precision	*recall*	*F1-macro*	*precision*	*recall*	*F1-macro*	*precision*	*recall*	*F1-macro*
LR	**0.451**	0.194	0.264	**0.338**	0.126	0.179	0.010	0.005	0.006
KNN	0.427	0.220	**0.279**	0.300	0.141	0.187	0.007	0.003	0.004
DT	0.147	0.112	0.123	0.099	0.068	0.079	0.007	0.005	0.005
RF	0.124	0.048	0.066	0.073	0.024	0.035	0.000	0.000	0.000
GNB	0.171	**0.650**	0.242	0.125	**0.648**	0.189	**0.022**	**0.163**	**0.032**
SVM	0.446	0.209	0.272	0.324	0.119	0.167	0.018	0.014	0.014
MLP_10	0.280	0.241	0.254	0.215	0.172	**0.189**	0.002	0.001	0.001
MLP_50	0.315	0.238	0.261	0.206	0.165	0.180	0.000	0.000	0.000
MLP_100	0.284	0.236	0.254	0.212	0.164	0.180	0.000	0.000	0.000
MLP_200	0.295	0.238	0.260	0.217	0.165	0.183	0.000	0.000	0.000

TABLE 3. Performances of flat classification with different models and data types.

B. DIRECT APPROACH

1) Training and Testing Data Preparation

For the "Direct" approach, link prediction method is used. Different from the previous approach, this time the topic features are obtained directly from the graph instead of individual words within the topics. Here, each topic, either from CSO or new topics, are represented in the form of nodes positioned in a large initial combined CSO–Topics Graph (result of combining Topics Graph and CSO Graph, as described in Section V-B2).

To obtain the topic node representation, a Node2Vec [34] model is trained on the combined CSO–Topics Graph, and later used to extract the vector representation of the new topics' position within its neighborhood and its neigborhood information in the graph. To obtain the training and test data, **only topics within the Topics Graph** are used (see section V-C). The Topics Graph consists of both the new topics and CSO topics, as well as their initial connections to one another. In total, we scraped 5,798 paper abstracts from ACM digital library. These abstracts are used to obtain all possible links that are used to the Topics Graph. For the Direct approach, CSO topics are the topics set as labels for the classifier, and the node vectors of each new topic are the features inputted to the multilabel classifier.

2) Evaluation Module

The Evaluation Module is the last module of the Direct approach. The filtered graph, the output graph from the *Filter Module*, is compared to the links in the Wikipedia graph. This is illustrated in Figure 9, with the process from the Filter Module (explained in Section V-E1 and Figure 4) colored in grey to represent processes that have been done in a previous step. Here, there are two types of links that are considered as *valid links*: (1) valid non-hidden links which are remaining links within the filtered graph that also exist within the Topics Graph and the Wikipedia graph. These valid non-hidden links appear in Figure 8 with the name titled "Valid Filtered Links."; (2) the valid hidden links, or the remaining links within the filtered graph that do not exist within the Topics Graph, but exist within the Wikipedia graph.

The performance of the classifier is measured by the Wikipedia-based $F1_w$ score on only the non-hidden links.

3) Wikipedia Graph

The Wikipedia Graph is the ground truth used in this last step of the "Direct" approach. From the Topics Graph explained in Section V-B2, unique topics from the graph, consisting of both the new topics and CSO topics, are listed. This list of unique topics is then used as **search queries** for the Wikipedia API [10]. For example, "Recommender Systems" is used as an example search query for the Wikipedia API. The Wikipedia API returns a list of titles of related Wikipedia articles from the given **search query**. We apply the following two rules before proceeding to the next step:

1) If there is **no exact match** within the returned list of Wikipedia article titles (search results) to the **search query**, extract the complete Wikipedia articles from all of the search results.
2) If an **exact match exists** within the returned list of search results, only extract one complete Wikipedia article that has the same title as the **search query**.

After obtaining all of the needed Wikipedia Articles, important topic terms from each of these articles are extracted by the *YAKE* keyword extractor tool [11]–[13]. All of the obtained keywords are then combined. All of the extracted topic terms are linked to the **search query**.

4) Wikipedia-based Evaluation Metric

Snce the validity of the resulting links are determined by their presence in the Wikipedia Graph, slightly different metric is used for measuring the performance of the implemented classifiers. The evaluation metrics used for the final performance scores are called the Wikipedia-based $precision_w$ (4), $recall_w$ (5), and $F1_w$ (6). For the Filter module, a performance measure called *filter degree* (1) is chosen along with a Wikipedia-based *precision* and *recall*. *Filter degree* in general is used to explain what percentage of links are removed from the input graph (Topics Graph). For the *Filter Degree*, the lower the score the better. However, if an extreme *Filter Degree* score is achieved, i.e., 100% or 0%, then it would also not be acceptable because then the links are either

[10]Wikipedia API, https://bit.ly/3oorGzz

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

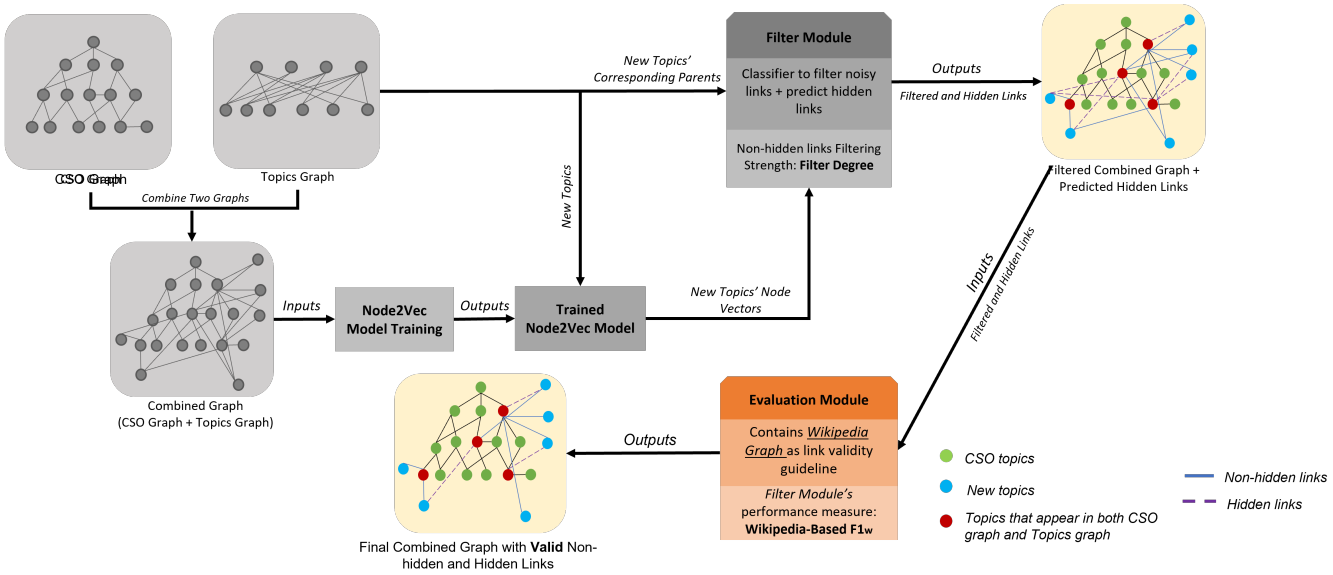


FIGURE 9. Process of the Direct Approach's Evaluation Module.

being completely filtered out or none of the links are filtered out.

$$filter\ degree = 1 - \frac{|noisy\ true\ positives|}{number\ of\ test\ data} \quad (1)$$

$$noisy\ true\ positives = result\ links \cap test\ data \quad (2)$$

Further evaluation of the Filter Module is **done within the Evaluation Module** which accepts the already filtered Topics Graph from the Filter Module as input. The performance of the Filter module is based on how many *valid filtered links* are produced by the link prediction classifier. To obtain the *valid filtered links* (3), the *noisy true positive* links obtained from equation (2) should be intersected again with the set of Wikipedia graph links ($WG\ links$).

In the case of $precision_w$ (4), the *valid filtered links* are divided by the number of links within (or the length of) the set of *result links* obtained from a classifier model in the Filter Module. For $recall_w$ (5), the *valid filtered links* are divided by the number of valid links within the test data, or the intersection of *test data* with $WG\ links$.

$$valid\ filtered\ links = noisy\ true\ positives \cap WG\ links \quad (3)$$

$$precision_w = \frac{|valid\ filtered\ links|}{|result\ links|} \quad (4)$$

$$recall_w = \frac{|valid\ filtered\ links|}{|test\ data \cap WG\ links|} \quad (5)$$

$$F1_w\ score = 2 \times \frac{precision_w \times recall_w}{precision_w + recall_w} \quad (6)$$

For the proposed approaches, the final performance score of a classifier is decided by the $F1_w$ (6) score. However, aside from the links limited only to the Topics Graph, ***valid remainder prediction results*** is also taken into consideration. *Valid remainder links* are contained in a set of links predicted by the classifier that **appear in the Wikipedia Graph** and **do not appear in the Topics Graph**. In other words, for simplicity, *valid remainder links* are the *hidden links* explained in Section V-E2. See Figures 7 and 8 to get a better idea of what *hidden links* are.

C. PERFORMANCE COMPARISON OF TWO APPROACHES

From this section the filtering strength, Wikipedia-based $F1_w$ score, number of new links and topics, as well as manual evaluation scores from the "Direct" and "Indirect" approaches are compared. For fair comparison, both the classification models are retrained using **the same complete data obtained from the "Direct" approach's Topics Graph**. In this experiment, 5-Fold Cross Validation is used to evaluate each of the models from each approach. For each of the approaches, three types of CSO are used: CSO with level limit = 1 without including synonyms, CSO with level limit = 1 with synonyms, and CSO without level limit (original format of CSO).

We compare two types of classifiers: (1) classifiers trained as *topic classifiers* from the "Indirect" approach and (2) classifiers trained as *link predictors* from the "Direct" approach. The same ten classification models mentioned in Section VI-A3 are used. However, we only show the six best performing models, i.e., GNB, SVM, and four versions of MLP for evaluation and performance comparisons.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

Model	CSO Versions					
	limit1_19		**limit1_30**		**no_limit**	
	Indirect	*Direct*	*Indirect*	*Direct*	*Indirect*	*Direct*
GNB	0.995	0.246	0.994	0.246	0.988	0.245
SVM	0.999	0.165	0.999	0.164	0.999	0.178
MLP_10	0.999	0.190	0.999	0.190	1.000	0.200
MLP_50	0.999	0.227	0.999	0.227	1.000	0.238
MLP_100	0.999	0.237	0.999	0.237	1.000	0.247
MLP_200	0.999	0.243	0.999	0.242	1.000	0.254

TABLE 4. Filter Degree score comparison between models of the two approaches.

Model	CSO Versions					
	limit1_19		**limit1_30**		**no_limit**	
	Indirect	*Direct*	*Indirect*	*Direct*	*Indirect*	*Direct*
GNB	**0.032**	0.274	**0.034**	0.273	**0.077**	0.273
SVM	0.004	0.267	0.003	0.267	0.002	0.267
MLP_10	0.004	0.271	0.004	0.271	0.001	0.270
MLP_50	0.004	0.275	0.002	0.276	0.001	0.275
MLP_100	0.005	**0.278**	0.004	0.278	0.001	**0.277**
MLP_200	0.006	**0.278**	0.002	**0.279**	0.000	**0.277**

TABLE 5. The $F1_w$ score for models of the two approaches.

Model	CSO Versions					
	limit1_19		**limit1_30**		**no_limit**	
	Indirect	*Direct*	*Indirect*	*Direct*	*Indirect*	*Direct*
GNB	0.509	0.164	0.474	0.164	0.527	0.164
SVM	0.272	0.157	0.358	0.157	0.202	0.157
MLP_10	0.266	0.160	0.264	0.160	0.001	0.160
MLP_50	0.266	0.164	0.138	0.165	0.200	0.164
MLP_100	0.276	0.166	0.286	0.166	0.200	0.165
MLP_200	0.310	0.166	0.196	0.167	0.133	0.166

TABLE 6. The $precision_w$ score for models of the two approaches

Model	CSO Versions					
	limit1_19		**limit1_30**		**no_limit**	
	Indirect	*Direct*	*Indirect*	*Direct*	*Indirect*	*Direct*
GNB	0.017	0.837	0.017	0.837	0.042	0.836
SVM	0.002	0.890	0.890	0.001	0.001	0.877
MLP_10	0.002	0.882	0.002	0.880	0.001	0.868
MLP_50	0.002	0.860	0.001	0.863	0.000	0.849
MLP_100	0.003	0.859	0.002	0.858	0.000	0.845
MLP_200	0.003	0.855	0.001	0.855	0.000	0.839

TABLE 7. The $recall_w$ score for models of the two approaches

1) Filtering Strength

The filtering strength is the *filter degree* (1) scores obtained after classifying the links of new topics and CSO topics in the Topics Graph. In this case, classifiers are considered as filters since classifiers are used to filter noisy links out (as explained in Section V-E1). The filtering strength represents the ability of a classifier to keep the valuable links between two topics ("True" according to the classifier) and remove noisy links ("False" according to the classifier). Table 4 shows filter degree scores on each model.

Table 4 shows that for the "Direct" approach, the classifiers are better at picking the good links or removing the noisy links in the Topics Graph. The "Indirect" approach's classifiers show weakness as filters since the classifiers filtered out more than 98% of the links within the Topics Graph as shown in Table 4. This is not desirable as it is essential to keep correct links in the Topics Graph and only remove the noisy ones.

2) Wikipedia-based Performance Scores

Wikipedia-based $F1_w$ score is obtained by using equation (6). The scores shown in this section are average $precision_w$, $recall_w$, and $F1_w$ scores after a 5-fold cross validation.

According to the $F1_w$ scores shown in Table 5, the scores of models under the "Direct" approach drastically are larger than scores of the models under the "Indirect" approach. Here, the best scores are shown in bold. For the "Indirect" approach, *Gaussian Naive Bayes* achieved the best $F1_w$ score for all CSO level limitation versions. The *Gaussian Naive Bayes*'s scores for the Direct approach are also close to the highest score in the "Direct" approach.

Tables 6 and 7 show the $precision_w$ and $recall_w$ scores. In Table 6, the "Indirect" approach shows greater precision compared to all of the scores obtained from the "Direct" approach. However, their $recall_w$ scores shown in Table 7 are very low. In real-world applications, a higher recall score is also important for the model to produce more valid links for the final extension of the ontology.

We also selected the best performing approach and models based on the number of actual valid links and topics produced by the model. The following section (Section VI-C3) provides more information on this issue.

3) Potential New Links and New Topics

This section presents the number of potential new links and topics resulting from each of the models after processing the filter module. We consider two types of resulting links: (1) **Non-Hidden Links** - the resulting links that also appear in the Topics Graph, and (2) **Hidden Links** - the resulting links that do not appear in the Topics Graph. Furthermore, we include the number of total links produced by each of the classifier. **Total Links** is the sum of non-hidden and hidden links. In our experiment, we used a 5-fold cross validation using the CSO.

To compare non-hidden links for each model, we present Tables 8 and 9, which present the average number of resulting **Non-hidden** links and valid **Hidden** links, respectively.

From both Tables 8 and 9, *Gaussian Naive Bayes* is the

Model	CSO Versions					
	limit1_19		**limit1_30**		**no_limit**	
	Indirect	*Direct*	*Indirect*	*Direct*	*Indirect*	*Direct*
GNB	**23.2**	1179.2	**24.8**	1178.6	**58.8**	1178.0
SVM	2.2	**1254.2**	1.8	**1254.2**	1.2	**1234.6**
MLP_10	3.2	1241.6	2.8	1239.2	1.0	1223.0
MLP_50	3.2	1121.0	2.0	1215.0	0.4	1195.0
MLP_100	3.6	1208.8	0.4	1190.2	4.2	1203.8
MLP_200	4.2	1203.8	1.8	1205.0	0.4	1181.8

TABLE 8. Approach comparison: Average valid non-hidden links

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

Model	CSO Versions					
	limit1_19		**limit1_30**		**no_limit**	
	Indirect	*Direct*	*Indirect*	*Direct*	*Indirect*	*Direct*
GNB	0.0	**416.8**	0.0	**420.4**	0.0	**538.0**
SVM	0.0	23.2	0.0	23.8	0.0	32.4
MLP_10	0.0	25.8	0.0	29.0	0.0	36.4
MLP_50	0.0	29.2	0.0	27.2	0.0	36.6
MLP_100	0.0	28.0	0.0	27.4	0.0	33.0
MLP_200	0.0	30.4	0.0	29.2	0.0	35.4

TABLE 9. Approach comparison: Average valid hidden links

Model	CSO Versions					
	limit1_19		**limit1_30**		**no_limit**	
	Indirect	*Direct*	*Indirect*	*Direct*	*Indirect*	*Direct*
GNB	**23.2**	**1596.0**	**24.8**	**1599.0**	**58.8**	**1716.0**
SVM	2.2	1277.4	1.8	1278.0	1.2	1267.0
MLP_10	3.2	1267.4	2.8	1268.2	1.0	1259.4
MLP_50	3.2	1240.2	2.0	1242.2	0.4	1232.4
MLP_100	3.6	1238.2	2.6	1236.2	0.4	1223.2
MLP_200	4.2	1234.2	1.8	1234.2	0.4	1217.2

TABLE 10. Approach comparison: Average totals of valid links

best performing model for the "Indirect" approach. It can be seen that most links, both non-hidden and hidden, are resulted more when the *level limitation rule* is not applied. This is a good sign because with the level limitation rule, the resulting links would not be able to extend fine-grained information to the original CSO. However, no matter how good the performance of *Gaussian Naive Bayes* for the "Indirect" approach, its performance is still significantly worse when compared to the hidden links resulted by the same model under the "Direct" approach.

The "Indirect" approach results in almost zero number of links in average for the valid non-hidden links (except for the *Gaussian Naive Bayes* model), and completely zero number of valid hidden links. This is caused by the generality of the results given by the topic classifiers. Since the topic classifiers are trained only on CSO data (without new topics), the classifiers are limited to only 19 (for limit1_19) to 30 (for limit1_30) target classes which are all located in the first level of the CSO. This too-general links resulted from the classifiers may not exist in the Wikipedia graph, where connections between topics are more specific. Since we decide the validity of a link through its existence within the Wikipedia graph, majority, if not all of the links are automatically not considered as valid. Additionally, text classification models will not perform well when they are trained to data with thousands of target classes (for no_limit), this common limitation of any machine learning models is also a possible cause for the very low resulting valid non-hidden and hidden links by the "Indirect" approach.

The limitations of the "Indirect" approach, however, does not happen to the "Direct" approach. This is because our approach ensures the model are trained with data equipped with information obtained from CSO which has been initially extended with the Topics graph. The Topic graph allows the data obtained for training to include very similar or even the same node information of new topics for all versions of CSO (limit1_19, limit1_30, and no_limit). This is because no matter how the CSO format is transformed, the new topics will be initially connected to their corresponding parent topics based on their appearance in the same paper abstract (see Section V-B2). Because of this, the final resulting links given by the "Direct" approach for each model are similar despite the different versions of CSO used.

Further comparison of the two approaches can be done by looking at the number of total valid links produced by each of the classification models. The total links can be obtained by adding the number of valid non-hidden links with the number of valid hidden links. It can be seen from Table 10 that *Gaussian Naive Bayes* is capable of producing the highest total of links for both the "Indirect" and "Direct" approaches. Only looking at the total valid links, *Gaussian Naive Bayes* is the most reliable model and "Direct" is the most reliable approach for production.

From the resulting links, new topics can be obtained. We consider two types of new topics: (1) new topics obtained from the valid non-hidden links, and (2) new topics obtained from the valid hidden links.

Table 11 shows the number of new topics obtained from the resulting **non-hidden links**. As with the number of resulting valid non-hidden links in Table 8, the same models produced the most number of topics under both the "Indirect" and "Direct" approaches. For the "Indirect" approach, *Gaussian Naive Bayes* is the best model for producing the most new topics from the non-hidden links, while for the "Direct" approach, the best model is *Support Vector Machine*.

The same situation also applies to the number of topics obtained from the resulting **hidden links**. Table 11 shows the number of resulted topics from the resulting hidden links. In parallel to the number of resulting valid hidden links shown in Table 9, the best performing model with the highest resulting number of hidden links and topics is *Gaussian Naive Bayes* for both the "Indirect" and "Direct" approaches.

In particular, for the "Direct" approach, taking into consideration the number of both resulting topics and links of the *Gaussian Naive Bayes* model, it appears that even though it results in fewer new non-hidden topics than *Support Vector Machine*, *Gaussian Naive Bayes* is still a preferable model as it produces significantly more hidden topics, which in turn drastically increases the number of final new topics.

Table 12 shows eight randomly sampled topics from each model of the "Direct" approach (from the 5th fold of the 5 fold cross validation with the no_limit CSO version) and their respective parent topics, supertopics, or topics that appear in the original CSO. In the real-world applications, each of these new topics are to be placed directly under their respective CSO supertopics for the CSO extension. In Table 12, the "New Topic" column contains completely new topics obtained from Computer Science papers that have not appeared in the original CSO. The "CSO Topic" column contains topics that exist within the original CSO.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

Model	CSO Versions											
	Limit1_19				*Limit1_30*				*No_limit*			
	Indirect		*Direct*		*Indirect*		*Direct*		*Indirect*		*Direct*	
	Non-hidden	Hidden	Non-hidden	Hidden	Non-hidden	Hidden	Non-hidden	Hidden	Non-hidden	Hidden	Non-hidden	Hidden
GNB	**23.2**	0	766.2	**278.4**	**24.8**	0	767	**284.4**	**52**	0	765.6	**338**
SVM	3.2	0	**805.8**	22.6	1.8	0	**804.8**	59	1.2	0	**800.6**	31.4
MLP_10	3	0	799.6	24.6	2.8	0	800.2	28.2	1	0	796.6	34.2
MLP_50	3.4	0	790.2	28	2	0	792.8	25.6	0.4	0	787.2	35
MLP_100	3.4	0	789.6	26.8	2.6	0	789.2	25.6	0.4	0	786.8	33.4
MLP_200	4.2	0	785	28.8	1.8	0	787.4	27.4	0.2	0	780.4	34.2

TABLE 11. Approach comparison: average number of topics from valid non-hidden links and valid hidden links

The "Description" column explains the relationship between the new topics and their CSO supertopics. In this study only the "subtopic" (for new_topic – cso_topic relationship) or "supertopic" (for cso_topic – new_topic relationship, or the inverse of "subtopic" relationship) relationships are considered.

4) The Paired *t*-test Significance Test

A paired *t*-test is usually used when there is an interest in the difference between two variables of the same subject. Dietterich [35] proposed a version of the paired *t*-test called 5x2cv *t*-test for observing two **Machine Learning** classification models which seemingly have little difference in their performance, but in reality may produce highly different results. 5x2cv *t*-test addressed the shortcomings of other methods, i.e., the re-sampled *t*-test and the K-fold cross-validated paired *t*-test which according to Diettrich [35] has many drawbacks. Specifically, a drawback of the k-fold cross-validated paired *t*-test is overlapping training sets which is not recommended to be used in practice, and the 5x2cv Paired *t*-test is recommended instead.

The 5x2cv paired *t*-test splits the dataset into two equal parts (50% and 50%) for training and testing each, and this action is repeated five times in total. In other words, "5x2cv paired" means a 2-fold cross-validation done five times.

For this statistical test, only the top three performing models are selected based on the total resulting links as shown in Table 10. According to the total links, *Gaussian Naive Bayes* is the best performing, followed by *Support Vector Machine*, and, finally, *Multilayer Perceptron with 10 hidden layers* as last.

Table 13 shows the resulting *p*-values of every model pair. Here, Null Hypothesis is used as an initial assumption, which is **the performance difference between two models are not significant**. In order to reject the Null Hypothesis, a *p*-value should be smaller than the determined α value: α = 5%. According to the *p*-values shown in Table 13, all of the model pairs are different. This means even though the classification models have little difference in $F1_w$ score, their classification results may be significantly different.

D. MANUAL EVALUATION

The purpose of having human evaluators to manually assess or evaluate the classification results is to achieve the level of agreement between the evaluators to the results of the classification. All graphs (Topics Graph, Wikipedia graph) are automatically extracted from the web and automatically generated. In the case of complete automation, the reliability of the final results of each classifier are naturally doubtful.

The human evaluators are asked for their opinions regarding the connections between two topics – one topic being the the new topic and the other is the CSO topic. The opinions of the human evaluators are measured by Inter-Annotator Agreement metrics, **Fleiss' Kappa** [36] and **Krippendorff's alpha** [37] to find out the level of reliability of the classifiers. However, during the calculation of these agreement scores, we noticed the problem of Kappa paradoxes [38], [39] occurring in our evaluation results. Therefore, to counter these paradoxes, we also used **Gwet's** AC_1 **and** AC_2 **scores** [40] to obtain reliability scores that are more suitable for our evaluation results.

The average numbers of new links and topics in Section VI-C3 of the "Direct" approach show promising extension to the original CSO. However, the actual quality of the links produced automatically still need to be evaluated. For the purpose of judging the reliability of the classifiers, a set of randomly sampled links from each model's results are manually assessed by four different human evaluators.

The four evaluators tasked with the evaluation are Computer Science experts. The evaluation consists of simply asking for the evaluators' opinions regarding the connection of two Computer Science topics. Aside from obtaining the reliability of classifiers for producing links, by manually judging these links, the efficiency and possibility of human labor's inconsistency can also be observed via the opinion difference.

The classifiers evaluated in this section are the top three classifiers: *Gaussian Naive Bayes*, *Support Vector Machine*, and *Multilayer Perceptron with 10 Hidden Layers*. From each of the classification models, two types of links are evaluated: 30 randomly selected **Hidden Links** and 60 randomly selected **Non-hidden Links**.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

Model	New Topic	Description	CSO Topic
GNB	context-aware recommendation	sub_topic	recommender systems
	visual recognition	sub_topic	computer vision
	hash code	sub_topic	hashing
	causal inference	sub_topic	inference
	neural architecture search	sub_topic	neural networks
	representation learning	sub_topic	neural networks
	link recommendation	sub_topic	recommendation
	supervised learning	sub_topic	classification methods
SVM	metric learning	sub_topic	machine learning
	learning-to-rank	sub_topic	learning
	educational robot	sub_topic	artificial intelligence
	swap operation	sub_topic	hardware
	graph cycle	sub_topic	cycle
	academic performance	sub_topic	university
	pre-training	sub_topic	neural network
	text linking	sub_topic	software
MLP_10	link recommendation	sub_topic	social network
	meta learning	sub_topic	learning
	item recommendation	sub_topic	collaborative filtering
	supervised learning	sub_topic	learning
	educational robot	sub_topic	robots
	social recommendation	sub_topic	social media
	deep bayesian learning	sub_topic	bayesian
	item-item product	sub_topic	recommender system
MLP_50	learning to rank	sub_topic	search engine
	neural embeddings	sub_topic	learning
	neural architecture search	sub_topic	neural network
	usability	sub_topic	user interface
	representation learning	sub_topic	learning
	meta learning	sub_topic	learning
	supervised learning	sub_topic	neural networks
	educational robot	sub_topic	artificial intelligence
MLP_100	hash code	sub_topic	hashing
	empirical study	sub_topic	mobile
	conterfactual learning	sub_topic	learning
	usability	sub_topic	interaction design
	neural architecture search	sub_topic	optimization
	dual-view sequential learning	sub_topic	learning
	museum robot	sub_topic	robots
	social recommendation	sub_topic	data mining
MLP_200	meta learning	sub_topic	machine learning algorithms
	metric learning	sub_topic	machine learning
	supervised learning	sub_topic	support vector
	skill recommendation	sub_topic	recommendation
	educational robot	sub_topic	robots
	deep bayesian learning	sub_topic	bayesian
	neural factorization machines	sub_topic	neural network
	community hri	sub_topic	robots

TABLE 12. Samples of new topics per the "Direct" approach's models and their respective CSO supertopics.

1) Manual Evaluation Metrics

To evaluate the manual evaluation results, several evaluation metrics are used. The first metric, **Fleiss' Kappa** [36] measures the reliability of agreement between multiple annotators regarding the given sample evaluation questions. Fleiss' Kappa is an extension of Cohen's Kappa [41], which is usually strictly used for two evaluators. Different from the Cohen's Kappa, Fleiss' Kappa can calculate the agreement score of more than just two evaluators. The second metric used is **Krippendorff's alpha** [37], and the last metric used for the evaluation is **Cohen's Kappa** [41]. This last metric specifically is used to measure the level of agreement between pairs of human evaluators.

Furthermore, we also used Gwet's AC_1 and AC_2 scores [40]. The AC_1 score (7), similar to Cohen's Kappa, was specialized to obtain reliability scores from two evaluators, where P_a is the ratio of overall agreement and $P_{e\gamma}$ represents the chance agreement probability.

$$AC_1 = \frac{P_a - P_{e\gamma}}{1 - P_{e\gamma}} \tag{7}$$

Model Pairs	Statistics	$pvalue$	Accept/Reject Null Hypothesis
GNB and SVM	-40.622	1.700×10^{-7}	Reject, performance difference is real
GNB and MLP_10	-34.895	3.600×10^{-7}	Reject, performance difference is real
SVM and GNB	40.622	1.700×10^{-7}	Reject, performance difference is real
SVM and MLP_10	9.620	2.058×10^{-5}	Reject, performance difference is real
MLP_10 and GNB	49.054	7.000×10^{-8}	Reject, performance difference is real
MLP_10 and SVM	-16.892	1.330×10^{-5}	Reject, performance difference is real

TABLE 13. The 5x2cv paired t-test scores of GNB, SVM and MLP_10. (α = 5%)

For the case of multiple (more than two) evaluators, AC_2 score (8), inspired by Krippendorff's alpha, can be used.

$$AC_2 = \frac{P_a - P_e}{1 - P_e} \tag{8}$$

We consider two types of agreements calculated: (1) agreement among all evaluators, and (2) agreement between two evaluators.

2) Agreement Scores

The agreement scores are all based on the opinions of human evaluators regarding the set of sample links that are given to them from each of the selected classification models. Essentially, the evaluators are asked to classify a given link as "True" or "False". However, since deciding an actual connection of two Computer Science topics is not an easy task, and since everyone might have different opinions regarding the closeness of two Computer Science Topics due to their personal Computer Science knowledge networks in each of their minds, this evaluation task was made into a non-binary task, i.e., there are more than two options to decide whether the evaluator "agrees" or "disagrees" with the topic connections.

In this paper, we only show agreement scores obtained from AC_1 and AC_2 because of Kappa Paradoxes. There are two types of Kappa paradoxes [38] and both came from the same type of problem: a large probability by chance P_e score. The high P_e can transform a situation with a high observed agreement P_o into having a very low Kappa score. In our case, the second Kappa paradox occurs. More specifically, our evaluators when paired, tend to have relatively high agreement on the "True" category. This situation allows either symmetrical or asymmetrical imbalanced marginal totals in their 2×2 agreement (confusion) matrix.

As a result, despite the relatively high agreement score for each of the evaluators to the classifiers' resulting links[11], all Cohen's Kappa, Fleiss' Kappa, and Krippendorff's Alpha scores show minimum agreement. These resulting low scores therefore do not represent the actual level of agreement between our expert evaluators.

To counter the Kappa paradox, we include alternative agreement coefficients, introduced by Kilem L. Gwet [40] called AC_1 and AC_2. These agreement coefficients are suitable for data with imbalanced agreement (e.g., 85% to 95% agreement for "True" and only 5% to 15% agreement for "False"), and they are capable in handling categorical, ordinal, interval, and missing data. The formula of AC_2 is very similar to Krippendorff's alpha with only the different definitions of P_a and P_e. Tables 14 and 15 show the AC_1 and AC_2 scores of the human evaluation respectively.

[11]We have calculated Evaluator–Classifier agreement scores using our own metric called "Evaluator-classifier Ratio" ECR, which in general divides each evaluator's positive opinions with the total questions given to them during the manual evaluation.The result shows majority of high agreement between each evaluator to each classifier

E. DISCUSSION

In this section, the positive and negative aspects of the two approaches are explained, along with the reasons as to why some performances are unsatisfactory.

1) Indirect Approach

- **Low $F1$ Scores on Flat Classification**
  Some factors that may have affected the final $F1$ scores (see Section VI-A3) of the **Flat Classification** approach:
  1) **Overwhelming number of classification targets**, which is the case for both the "Indirect" and "Direct" approach. For the "Indirect" approach specifically, there are at most a total of 4,241 number of potential parents for the new topics to be classified to.
  2) The data used in this study is **imbalanced data** as explained in section IV-A1. Similar to the above factor, imbalanced data is commonly known to affect classification methods negatively. An example of an extreme data imbalance problem is that in one case, a data sample does not have enough examples for the classifier to learn well, and in another case, a data sample might have so many examples that the classifier will become over-fitted to that specific information.
- **Positive and Negative Affects of Minimizing Classification Targets**
  According to the experiment results shown in Table 3, the lesser the ontology level, the better the $F1$ score is. Significant increase in $F1$ score is obtained by *Support Vector Machine* on data with *level limitation* set to 1. Without applying the limitation rule, there are a total of 4,241 potential parents in total. After applying the limitation rule, the number of potential parents decreased drastically. In the case of level limit1_19, there are a total of 20 potential parents at most (if topic "Computer

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3131627, IEEE Access

Model	Evaluator Pairs											
	1 & 2		1 & 3		1 & 4		2 & 3		2 & 4		3 & 4	
	Non-hidden	*Hidden*	*Non-hidden*	*Hidden*	*Non-hidden*	*Hidden*	*Non-hidden*	*Hidden*	*Non-hidden*	*Hidden*	*Non-hidden*	*Hidden*
GNB	**0.731**	0.597	**0.756**	0.546	**0.642**	**0.627**	**0.909**	0.723	**0.820**	0.608	**0.843**	0.562
SVM	0.121	**0.706**	0.077	**0.706**	0.204	0.608	0.642	**1.000**	0.233	**0.803**	0.364	**0.803**
MLP_10	0.568	0.481	0.627	0.510	0.478	0.520	0.890	0.848	0.890	0.790	0.837	**0.803**

TABLE 14. AC_1 scores of the evaluator-pairs for the resulting non-hidden and hidden links from the "Direct" approach's three best models.

Model	Non-hidden	Hidden
GNB	**0.788**	0.610
SVM	0.263	**0.783**
MLP_10	0.731	0.674

TABLE 15. AC_1 scores based on the opinion of four evaluators on the resulting links from the "Direct" approaches three best models.

Science" is included) and for level limit1_30, there are only 30 potential parents in total. Nevertheless, it should always be remembered that the more the limitation rule is applied, the less fine-grained (or more general) the classification results are. This is because the classifier would focus more on the final parent levels when they are classifying the new topics. For example, if the level limitation is set to limit = 1, then this means the parent nodes would only be located in the first level of the ontology. The parents in this case become the classes during the training process, and therefore the classification models would only assume these first level nodes as potential parents. Finally, when classifying new topics, the classification model would not be able to classify them under a node within a deeper level than 1, resulting in a general connection instead of a more direct fine-grained one.

2) Direct Approach

- **Low $F1_w$ Scores**
  Different to the reasons explained in Section VI-E1, the reason of the "Direct" approach's low Wikipedia-based $F1_w$ score is because most of the resulting links are not valid links. A side note to remember: the $F1_w$ score is the score to measure the strength of a classifier in picking quality links within the Topics Graph during the filtering process. Since the idea of valid links are final Topics Graph links that also appear in the Wikipedia graph, the theory for now is the current classifiers do not work well enough as link filters. The classifiers are still picking noisy links and consider them as important links, while in reality, the classifiers should be able to differentiate noisy links and important links within the noisy Topics Graph.
- **Classifier Works Better as Filters**
  According to the resulting links shown in Tables 8 and 9, despite the low $F1_w$ score, on average, the classifier works better as a **filters** than a **hidden links predictor**. It can be seen from the aforementioned tables that the resulting valid non-hidden links are a lot more compared to the valid hidden links.
- **Significantly More Hidden Links**
  Compared to the number of valid hidden links produced by the "Indirect" approach, the "Direct" approach can produce significantly more valid hidden links. This can be seen in Table 9 where all models of the "Indirect" approach produce 0 valid hidden links on average. From this result, the topic classifier is shown to be not suitable for automatically extending an ontology.
- **Difference in Data to the Baseline**
  The input features used in this study differ from the ones used in the reference study [19]. In the case of flat classification, the same approach as Althubaiti et al. [19] is used. However, the seed ontologies we used are different. In their case, they are using a partial disease ontology, while in the case of this study the complete Computer Science Ontology is used. Experiments of [19] are faced with only at most 17 classes. In the case of both the proposed approaches, the complete Computer Science Ontology terminologies are used to train all of this study's classification models. This results in a drastically increased number of classes in the case of this study: a total of 4,241 classes for the "Indirect" approach and 1,991 total classes for the "Direct" approach.
  Additionally, the data used is also different. Instead of using the same disease ontology used in the work of Althubaiti et al. [19], in this study, Computer Science Ontology is used as data for the experiments. To the best of our knowledge, the work of Althubaiti et al. [19] and this study are two of the few, if not the only ones that attempted in the automatic extension of CSO topic ontology with a large scale of potential parents (or classification categories).
- **Connection of Filtering Strength and $F1_w$ Score**
  The connection of Filtering Strength and $F1_w$ score can be used as support information of a model's capability by showing the level of its intelligence when applied to the specific link-filtering task. More specifically, a combination of moderate *filter degree* score (30% to 70%) and large $F1_w$ score concludes a reliable classification model. This is because the model is able to filter out most of the unnecessary links and leaving behind only the important links.
  In this section, based on the best and worst performing $F1_w$ scores in our previous experiments (purposely not shown in this paper due to different settings and worse performance) the "Direct" approach's *Gaussian*

Naive Bayes and *Random Forest* are used to make an example. Table 16 shows the *filter degree* and $F1_w$ scores obtained from this experiment specifically only to illustrate the connection between the two evaluation metrics. Here the **Batch** column represents the data batches which include different numbers of paper abstracts from which the new topics are obtained.

Batch	Gaussian Naive Bayes		Random Forest	
	filter degree	$F1_w$	*filter degree*	$F1_w$
1	0.702	0.274	0.979	0.044
2	0.529	0.286	0.980	0.056
3	0.390	0.296	0.952	0.141
4	0.335	0.269	0.968	0.099

TABLE 16. *Filter degree* and $F1_w$ score combination of *Gaussian Naive Bayes* and *Random Forest* from our older experiment

Scores shown in Table 16 show the *Gaussian Naive Bayes* model is more reliable compared to the *Random Forest* model. It can be seen from the connection of *filter degree* and $F1_w$ scores of each of the data batches. For the first data batch (Batch 1), the *Gaussian Naive Bayes* model obtained a rather high *filter degree* score of 0.702 with an $F1_w$ score of 0.274. This means by removing 70% of the links within the Topics Graph, *Gaussian Naive Bayes* could still achieve the most valid links compared to the other models. In other words, *Gaussian Naive Bayes* is the most intelligent thus far, compared to the other models, in picking up important links and removing unimportant ones. In comparison, *Random Forest*'s *filter degree* score is 0.979 with an $F1_w$ score of 0.044. This means, *Random Forest* struggles in differentiating which links in the Topics Graph that are important, and which ones are not important.

Since *filter degree* (Equation (1)) is based on the *noisy true positives* (Equation (2)), which is the intersection between the classification results and the test data (obtained from Topics graph), many noisy links are still being taken into consideration in this case. Therefore, **only the $F1_w$ score should be used for primarily determining the best model** and *filter degree* can only be used later as support information. For example, selection of the best classification model should be based on first, the $F1_w$ score and only after that, the model's corresponding *filter degree* score can be used for supporting the model's capability further.

To conclude, to know a classification model's strength or intelligence for this specific task, *filter degree* can be used as an additional measurement **after** selecting the best model via the overall $F1_w$ performance score.

- **Human Evaluation for Reliability**

  We conclude that overall, the "Direct" approach is consistently reliable from only looking at the AC_1 and AC_2 scores[12]. The reliability of the "Direct" approach can be seen from the results in Tables 14 and 15 where there are moderate to high agreement scores between the evaluator pairs and the overall evaluator. In Table 14, highest agreement scores from all evaluator pairs for the non-hidden links all obtained by the *Gaussian Naive Bayes* modal. However, for the hidden links, the evaluator pairs have mostly high agreement for the *Support Vector Machine* model. The same conclusion is shown in Table 15 with *Gaussian Naive Bayes* as the model with the most agreed upon valid non-hidden links and *Support Vector Machine* for the hidden links.

[12]Our defense for only using these two agreement coefficients is due to the issue of Kappa paradox (see Section VI-D2)

F. OBSERVATION

In our experiment, the *Gaussian Naive Bayes* showed potential in realizing a fully automated ontology extension, however not alone. The "Direct" approach's *Gaussian Naive Bayes* model produced reliable non-hidden links while the *Support Vector Machine* produced reliable hidden links. However, the "Direct" approach's *Gaussian Naive Bayes* also produced the most number of valid hidden links, and therefore, this classifier is more reliable in terms of production. Aside from these two classification models, the remaining *Multi Layer Perceptron* model also showed promising performance and agreement between evaluators. The confidence in selecting the "Direct" approach's classification models also comes from their stable ($F1_w$) scores and total produced links (See Section VI-C) where *Gaussian Naive Bayes* is always (one of) the best performing model(s) with *Support Vector Machine* always following closely.

The "Indirect" approach can also be used, however, from only looking at the number of resulting new links of its classification model, it can be seen that only the *Gaussian Naive Bayes* model is reliable. The other models tend to result in few new links and topics.

Additionally, most of the performance scores of the "Indirect" approach's classification models are unsatisfactory and inconsistent. Despite the potential of "Indirect" approach's *Gaussian Naive Bayes* model, the "Indirect" approach itself is still too risky to be called reliable due to their low $F1_w$ scores. (Table 5 shows the comparison of $F1_w$ scores between the "Indirect" and "Direct" approaches are compared.)

1) Achieved Research Goals

After conducting the experiments, the following research goals are successfully achieved:

1) Extending the original CSO with as many new up-to-date Computer Science topics obtained from newly published papers (from year 2018 to 2020) as possible.
2) Enriching the original CSO with new relationships between topics.
3) Obtain an extended CSO which is most appropriate for future scientific paper recommender system project.

2) Answered Research Questions

1) **Is an automated ontology extension approach feasible?** Depending on the purpose of the ontology

extension, an automated ontology extension can be either reliable or unreliable. Since we plan to use the extended ontology for a scientific paper recommender, the proposed automatic ontology extension approach is reliable for our purpose. This is because the resulting extended ontology consists of the enriched information (i.e., new Computer Science Topics and ther links) and can aid the recommender system in recommending more related items to the users.

On the other hand, the manual link evaluation presented in this paper shows that our human experts tend to have high agreement with the resulting links produced by the top classifiers, as well as high agreement with each other. Therefore, this automation may be reliable enough for extending a task-neutral ontology. However, we must always keep in mind that not all tasks are as flexible as a recommendation task. There might be cases where information given to users need to be more than 90% accurate. When this level of accuracy is required, then perhaps changing the source of information from the Wikipedia graph to a more reliable knowledge source can be useful.

2) **Which approach, direct or indirect, is better at recognising new topics and properly linking them into the CSO ontology?** The best performing classification model for both the "Direct" and "Indirect" approaches is the *Gaussian Naive Bayes* models according to their respective $F1_w$ scores. Taking into consideration the real results of the *Gaussian Naive Bayes* models, i.e., the number of new topics and links, the "Direct" approach is the better one that can extend the original CSO than the Indirect approach.

VII. CONCLUSION AND FUTURE WORKS

This study focused on automating ontology extension for Computer Science Ontology.

For extending the Computer Science Ontology, we proposed the "Direct" approach and adopted an "Indirect" approach. We compared both approaches with various classification techniques and evaluated their performances.

Several experiments were done on the classification models and the results of each approach, including comapring the number of valid resulting links, statistical significance test for machine learning classification models with similar $F1_w$ scores, and finally an evaluation by four human experts.

The experiment results show that the "Direct" approach has an overall better and more reliable performance out of our two approaches: Direct and Indirect approaches. This statement is supported by the higher $F1_w$ scores, number of valid resulting links of either non-hidden or hidden links, and finally, the more consistently agreeable links resulted according to the human evaluators.

From the $F1_w$ scores to the evaluator agreement scores, the "Direct" approach shows the same best classification models, which are *Gaussian Naive Bayes*, *Support Vector Machine*, and *Multilayer Perceptron with 10 hidden layers*. Despite this, only *Gaussian Naive Bayes* was selected to extend the CSO with both its resulting non-hidden and hidden links. This is because *Gaussian Naive Bayes* could produce more valid links and has only slightly lower agreement scores by the human experts compared to the rest of the best models in regards to the hidden links.

To conclude, even though the $F1_w$ scores of both approaches are not significantly high, the real resulting links are promising, especially the links resulting from the "Direct" approach. This can be seen from the consistent agreement scores between the human evaluators when evaluating the "Direct" approach's resulting links. This situation is, therefore,ble to also accept the hypotheses stated at the beginning of the study by providing from the test data alone, approximately 1,200 new perfectly written topics (around 8.6% topic increase from the CSO's original 14K topics) along with approximately 1,557 new links to the existing CSO topics.

Our study verifies that automating ontology extension is feasible for Computer Science Ontology. We will further apply the Direct approach to enrich the Computer Science Ontology to make the full CSO publicly available. We also plan to investigate multi-label and hierarchical classifications further.

REFERENCES

[1] A. Konys, "Knowledge repository of ontology learning tools from text," *Procedia Computer Science*, vol. 159, pp. 1614–1628, 2019.

[2] P. Cimiano and J. Völker, "text2onto," in *International conference on application of natural language to information systems*. Springer, 2005, pp. 227–238.

[3] M. T. Pazienza and A. Stellato, "The protégé ontoling plugin-linguistic enrichment of ontologies in the semantic web," in *poster proceedings of the 4th International Semantic Web Conference (ISWC-2005)*, 2005.

[4] J. Shen, Z. Wu, D. Lei, C. Zhang, X. Ren, M. T. Vanni, B. M. Sadler, and J. Han, "Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2180–2189.

[5] A. 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*. Springer, 2018, pp. 187–205.

[6] D. Man, "Ontologies in computer science," pp. 43 – 46, 2013.

[7] C. Zhang, F. Tao, X. Chen, J. Shen, M. Jiang, B. Sadler, M. Vanni, and J. Han, "Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2701–2709.

[8] H. H. Alrehamy and C. Walker, "Semcluster: unsupervised automatic keyphrase extraction using affinity propagation," in *UK Workshop on Computational Intelligence*. Springer, 2017, pp. 222–235.

[9] J. Shang, J. Liu, M. Jiang, X. Ren, C. R. Voss, and J. Han, "Automated phrase mining from massive text corpora," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 10, pp. 1825–1837, 2018.

[10] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han, "Mining quality phrases from massive text corpora," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1729–1744.

[11] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt, "Yake! keyword extraction from single documents using multiple local features," *Information Sciences*, vol. 509, pp. 257–289, 2020.

[12] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt, "A text feature based automatic keyword extraction method for single documents," in *European conference on information retrieval*. Springer, 2018, pp. 684–691.

[13] ——, "Yake! collection-independent automatic keyword extractor," in *European Conference on Information Retrieval*. Springer, 2018, pp. 806–810.

[14] S. Rose, D. Engel, N. Cramer, and W. Cowley, "Automatic keyword extraction from individual documents," *Text mining: applications and theory*, vol. 1, pp. 1–20, 2010.

[15] A. Ayadi, A. Samet, F. d. B. de Beuvron, and C. Zanni-Merk, "Ontology population with deep learning-based nlp: a case study on the biomolecular network ontology," *Procedia Computer Science*, vol. 159, pp. 572–581, 2019.

[16] W. Liu, A. Weichselbraun, A. Scharl, and E. Chang, "Semi-automatic ontology extension using spreading activation," *Journal of Universal Knowledge Management*, no. 1, pp. 50–58, 2005.

[17] I. Novalija and D. Mladenic, "Semi-automatic ontology extension using text mining," in *Conference on Data Mining and Data Warehousing (SiKDD 2008)*, 2009.

[18] I. Novalija and D. Mladenić, "Content and structure in the aspect of semi-automatic ontology extension," in *Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces*. IEEE, 2010, pp. 115–120.

[19] S. Althubaiti, Ş. Kafkas, M. Abdelhakim, and R. Hoehndorf, "Combining lexical and context features for automatic ontology extension," *Journal of biomedical semantics*, vol. 11, no. 1, pp. 1–13, 2020.

[20] N. C. Santosa, J. Miyazaki, and H. Han, "Flat vs. hierarchical: Classification approach for automatic ontology extension," in *Proceedings of the 13th Forum on Data Engineering and Information Management*. DEIM, 2021.

[21] N. N. Daud, S. H. Ab Hamid, M. Saadoon, F. Sahran, and N. B. Anuar, "Applications of link prediction in social networks: A review," *Journal of Network and Computer Applications*, p. 102716, 2020.

[22] P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: the state-of-the-art," *Science China Information Sciences*, vol. 58, no. 1, pp. 1–38, 2015.

[23] T. Murata and S. Moriyasu, "Link prediction of social networks based on weighted proximity measures," in *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*. IEEE, 2007, pp. 85–88.

[24] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, "Fast and accurate link prediction in social networking systems," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2119–2132, 2012.

[25] M. Fire, L. Tenenboim, O. Lesser, R. Puzis, L. Rokach, and Y. Elovici, "Link prediction in social networks using computationally efficient topological features," in *2011 IEEE third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*. IEEE, 2011, pp. 73–80.

[26] A. Boyd, "Using wikipedia edits in low resource grammatical error correction," in *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, 2018, pp. 79–84.

[27] F. Chiarello, L. Trivelli, A. Bonaccorsi, and G. Fantoni, "Extracting and mapping industry 4.0 technologies using wikipedia," *Computers in Industry*, vol. 100, pp. 244–257, 2018.

[28] J. Kim, S. Kim, and C. Lee, "Anticipating technological convergence: Link prediction using wikipedia hyperlinks," *Technovation*, vol. 79, pp. 25–34, 2019.

[29] C.-C. Ni, K. Sum Liu, and N. Torzec, "Layered graph embedding for entity recommendation using wikipedia in the yahoo! knowledge graph," in *Companion Proceedings of the Web Conference 2020*, 2020, pp. 811–818.

[30] H. K. Azad and A. Deepak, "A new approach for query expansion using wikipedia and wordnet," *Information sciences*, vol. 492, pp. 147–163, 2019.

[31] F. Osborne and E. Motta, "Klink-2: integrating multiple web sources to generate semantic topic networks," in *International Semantic Web Conference*. Springer, 2015, pp. 408–424.

[32] F. Osborne, E. Motta, and P. Mulholland, "Exploring scholarly data with rexplore," in *International semantic web conference*. Springer, 2013, pp. 460–477.

[33] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[34] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[35] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural computation*, vol. 10, no. 7, pp. 1895–1923, 1998.

[36] J. L. Fleiss, "Measuring nominal scale agreement among many raters." *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[37] K. Krippendorff, "Computing krippendorff's alpha-reliability," 2011.

[38] A. R. Feinstein and D. V. Cicchetti, "High agreement but low kappa: I. the problems of two paradoxes," *Journal of clinical epidemiology*, vol. 43, no. 6, pp. 543–549, 1990.

[39] D. V. Cicchetti and A. R. Feinstein, "High agreement but low kappa: Ii. resolving the paradoxes," *Journal of clinical epidemiology*, vol. 43, no. 6, pp. 551–558, 1990.

[40] K. L. Gwet, "Computing inter-rater reliability and its variance in the presence of high agreement," *British Journal of Mathematical and Statistical Psychology*, vol. 61, no. 1, pp. 29–48, 2008.

[41] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

NATASHA C. SANTOSA is a Ph.D. student in the Computer Science department in Tokyo Institute of Technology, Japan. She has a B.Sc. degree in Computer Science from Gadjah Mada University, Indonesia and a M.Eng. degree in Artificial Intelligence from Tokyo Institute of Technology, Japan. Her research interests include Recommender Systems, Natural Language Processing, Data Mining, and Machine Learning.

JUN MIYAZAKI received a B.E. degree from Tokyo Institute of Technology (Tokyo Tech.), Tokyo, Japan, in 1992, and both M.S. degree and Ph.D. from Japan Advanced Institute of Science and Technology (JAIST), Japan, in 1994, 1997, respectively. He joined Tokyo Tech. in 2013, where he is currently a professor of School of Computing and a Deputy Director of Global Scientific Information and Computing Center. Before joining Tokyo Tech., he worked at Nara Institute of Science and Technology (NAIST) as an associate professor. His research interests include database systems, cloud computing, and information recommendation. He is a member of IEEE, ACM, IEICE, IPSJ, and DBSJ.

HYOIL HAN received the Ph.D. degree in Computer Science and Engineering from the University of Texas at Arlington in 2002. She is currently an Associate Professor in the School of Information Technology at Illinois State University, USA. She worked for Samsung Electronics and Korea Telecom before obtaining her Ph.D. Her research interests include machine learning, natural language processing, big data management, and applying AI to security.