Flat vs. Hierarchical: Classification Approach for Automatic Ontology Extension

Natasha Christabelle SANTOSA[†], Jun MIYAZAKI[†], and Hyoil HAN[††]

[†] Department of Computer Science, School of Computing, Tokyo Institute of Technology
2-12-1 Oookayama, Meguro-ku, Tokyo 152-8550, Japan
[††] School of Information Technology, Illinois State University
100 N University St, Normal, IL 61761, United States
E-mail: †{santosa,miyazaki}@lsc.c.titech.ac.jp, ††hyoil.han@acm.org

Abstract Originally, a branch of philosophy, ontology shares a similar purpose within information technology, where it is a knowledge structure regarding definitions and relations of information under the same or even multiple domains. This information representation is convenient when applied in real-life problems such as document classification and recommender systems which value the importance of semantics. However, with big data becoming more relevant, manual extension of existing ontologies with the most up-to-date information is challenging due to the tedious manual process and the cost of expert human labor. Therefore, researchers have come up with several ideas of automatic approaches in ontology extension. In this paper, we implemented and measured the performances of several text classification methods, categorized into two categories: flat classification and hierarchical classification, dependent on the hierarchical position/location of the concept in the ontology.

Key words Ontology Extension, Classification Methods, Hierarchical Classification, Concept Classification, Deep Learning

1 INTRODUCTION

With information rapidly growing in today's era, keeping this large amount of information orderly is necessary and can be achieved by storing extracted knowledge within an ontology. The concept of ontology has existed for decades, especially amongst the people of Philosophy, and in Information Technology, ontology serves the same base goal: organizing concepts by the structure of relationships connecting them. Ontology is useful as it can store knowledge that is readable by machines, and due to its structure, applying ontology can eventually provide explainable information for human users. Furthermore, ontologies are compatible in single or multi-domains (also known as cross domains) environments due to their general concepts and properties, which are highly reusable.

Research on constructing ontologies has been around for more than a decade. Diverse methods have been proposed for constructing ontology. In popular cases, ontologies were constructed from raw texts (e.g. news articles, scientific publications, product reviews, etc.), and ontology has been constructed mainly in a semiautomatic manner at best. In the early 2000s, research on ontology construction tools became popular, and diverse ontology construction tools from the raw text have been produced [20], e.g., Text2Onto [21], OntoLT [22], and OntoLing [23]. Nonetheless, the semi-automatic ontology construction approaches only decreased expert human labor by approximately 50% or less than complete manual construction, and it still remains dependent on expert human labour. With the massive growth of information, it is essential to minimize human labor as much as possible, as it is costly to construct a new ontology manually or simply to maintain an ontology. This situation aroused interests in new research topics regarding an entirely automatic ontology creation and population.

Ontology is often used where structures of information are important. Domain-specific ontologies are used in specific domains. For example, Gene Ontology [11] [12] in bioinformatics, WordNet [9] [10] in natural language processing, Computer Science Ontology [7] [8] in information retrieval and recommender systems, Disease Ontology [13] in health and medicine, and many more. In this paper, we focus on existing ontology in a specific domain, Computer Science, and chose to use a preexisting ontology called Computer Science Ontology (CSO) [7] that stores information of the Computer Science field's topics. CSO is thus far the most complete and up to date Computer Science topic ontology. However, CSO the most complete and automatically generated Computer

Science ontology, is still missing some recent concepts (e.g. new topic terms introduced in papers published recently before time of writing this paper). It is very possibly due the difficulty of maintenance. This problem inspired us to experiment with one of the most important ontology maintenance processes — ontology extension.

For extending the existing ontology (CSO), we are especially interested in the idea of applying text classification approaches to automatically categorize new concepts extracted from recent publications. This paper aims to apply, observe, and conclude if text classification approaches are an appropriate method in realizing a completely automated ontology maintenance, specifically the ontology extension process. Hence, this paper's contributions includes but is not limited to implementing text classification methods for ontology extension and experimenting with two types of text classification methods (flat and hierarchical) on the existing CSO. Performance-wise, our experimental result showed flat text classification method performs better than the hierarchical classification method and more suitable for automatic ontology extension. Nevertheless, we believe there are potentials to both approaches and many improvements are needed before we can conclude with confidence the possibility of an entirely automatic ontology extension method.

2 RELATED WORK

2.1 Ontology in Computer Science

Man [4] 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. Ontology can be considered as a description of an entire domain. This approach 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 our current work, we used ontology as a knowledge representation that describes scientific paper topics specifically under the Compute Science field and the connections between them. There are two types of available topic ontologies — Cross domain ontologies and domain-specific ontologies. In the next two sections, cross-domain and domain-specific ontologies are explained, along with their corresponding problems at the end of each section.

2.1.1 Cross Domain Ontologies

When it comes to scientific paper topics, Ruiz-Iniesta and Corcho [5] mentioned in their ontology review, there have been several ontologies created for the sole purpose of describing bibliography and citations such as the well-known BIBO(注1), which is an ontology that describes bibliographic information (authorships, chapters, issues, etc.) of journals, web pages, books, and other sources. Another ontology of this kind is FaBiO [6] which describes bibliographic entities and their groupings (e.g. a book and its series). Finally, the Citation Typing Ontology (CiTO)(注2) essentially provides types of scientific paper citations. CiTO includes in total 41 object properties that can enrich a citation information (e.g. agrees with, corrects, likes, uses method in). Due to the nature of cross-domain ontologies that hold various knowledge from different fields, the current approach to building this type of ontology depends on a group of experts that understand in an expert level what happens behind each of the field within the ontology. This is why many ontologies from this category is not always up-to-date or available for use because the maintenance or construction process is usually done manually by a large group of experts or an organization.

2.1.2 Domain Specific Ontologies

Unlike the cross-domain ontologies, domain-specific ontologies are more commonly found available on the web. Creation of this type of ontology is more achievable by individual workers or smaller group of people due to the smaller range of domains needed for the ontology's contents. 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. For example, a small and simple ontology consisting of knowledge regarding hotel information in a specific area can easily be constructed by a researcher that wants to do a research on a hotel recommendation system. However, despite the simplicity of the example, if there is a construction of ontology, population of ontology should always follow. In the case of preexisting ontologies, maintenance such as ontology extension with updated concepts should be done as well.

2.2 Ontology Learning for Ontology Extension

Ontology Learning (OL) is to create and populate an ontology with concepts and relationships between each of these concepts in an automatic or semi-automatic manner. Lehmann and Volker [3] in their book classified OL into four approaches: **ontology learning from text, data mining, concept learning in description logics and OWL, and crowdsourcing**. However, since the aim of our paper is to apply and observe the performance of several existing text classification methods to an existing topic ontology extension, and to conclude whether an entirely automatic approach to ontology extension is feasible, we decided to focus

(注1)：BIBO, https://bit.ly/2LKpfYD

(注2)：CiTO, https://bit.ly/3nEtao7

entirely on **Ontology Learning from Text** approach.

2.2.1 Ontology Learning from Text

Existing OL methods under this approach adhere to common Natural Language Processing (NLP) techniques such as the generation of lightweight taxonomies via text mining and information extraction approaches. Inspired by previous works produced by the computational linguistics field, many methods used under this approach in general hold one aim — acquiring important facts from text corpora useful for populating new or premade ontologies. Several works have followed the idea of OL from text from which multiple ontologies of diverse domains have been produced either semi-automatically or automatically.

2.3 Semi-Automatic Ontology Extension

Research on Ontology Extension mostly focuses on the semi-automatic 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 its position predictions or probabilities 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 semi-automatic ontology extension include the work by Ayadi et al. [25] who have attempted a semi-automatic ontology extension using a deep learning approach for extracting new ontology concepts alongside their relations and attribute instances, before later depending on experts to check the outputs of their extraction model. Another work was done by Liu et al. [26] where they attempted a semi-automatic ontology extension and refinement. In their work [26], they used WordNet lexical dictionary to create a semantic network from a given seed ontology. This semantic network was then processed by spreading activation to determine whether or not it is to be included to the seed ontology for extension. Finally, Novalija and Mladenic used text mining combined with a user-oriented approach to semi-automatically extend a cross-domain Cyc ontology[注3] [27]. They followed up this work with another publication [28] that presented an extension of their previous methodology with the idea of combining ontology content and ontology structure for ontology extension.

Understandably, many would trust human intervention in ontology extension. Ontology is a naturally complicated structure consisting of many concepts connected by different types of relationships and attributes, making its knowledge structure impossible for many to understand unless they are experts in the ontology's domain. This difficulty made many ontology extension approaches choose not to completely trust a machine.

2.4 Recent Attempts on Automatic Approaches

During the time of conducting this research, we have noticed a recent publication by Althubaiti et al. [17] in which they claimed to have attempted the realization of a fully automated ontology extension via employing an artificial neural network unigram classification. In their work, they were able to automatically classify the position of new ontology concepts only through the utilization of unigram vector representations as input to the classifier. To the best of our knowledge, this publication is the only one thus far that attempted a fully automatic ontology extension approach.

2.5 Text Classification

Text classification is the process of categorizing textual data into singular or multiple set of predefined labels. In ontology extension, we believe the use of text classification approaches may allow its automation where new concepts are considered as text and their super-class(es) are considered as their corresponding labels. In our work, we divided text classification into two types, flat classification and hierarchical classification.

2.5.1 Flat and Hierarchical Classifications

The major difference between flat classification and hierarchical classification lies on the presence of hierarchically structured targets, i.e., taxonomic representation of the classification labels, and the consideration of level information. In the case of flat classification, level information is completely ignored and usually classification of the inputs are done only on the final leaf-level labels. Hierarchical classification on the other hand, is the opposite as it considers all level information during the classification process. Typically, a hierarchical classifier has the same number of classifiers as the number of levels of its taxonomy, or in a more extreme case, its classifiers amount to the same number as its number of nodes within the taxonomy [24].

From the above definition, it is clear that flat classification is very simple and its implementation is certainly quicker. However, in the case of ontology extension, we are assigned with the task of placing new ontology concepts under its appropriate super-concepts that are equipped with level information. Therefore, we implement both of these classification methods to observe the importance of level information in ontology extension.

3 DATASET

The experiment dataset is different from the dataset used in either of the text classification approaches explained before. Instead of using an entire text, we use topic phrases from the ontology for training the classification models. The reason is that we aim to classify new topic phrases or ter-

[注3]：Cyc, https://bit.ly/2WA4GQP

minologies extracted from up-to-date scientific publications into their appropriate position within the existing CSO. This section describes the CSO and the text corpus we used for the topic vectorization step in detail.

3 1 Computer Science Ontology

Our interest is to extend the existing CSO with up-to-date terminologies using various text classification approaches. CSO itself is a topic ontology containing scientific papers' topics under Computer Science. Salatino et al. [7] automatically generated CSO by applying Klink-2 algorithm [14] to the Rexplore dataset [15]. Despite being automatically generated, Salatino et al. [7] claimed that there were also manual revisions done by experts. CSO consists of several roots which include Computer Science, Linguistics, Geometry, and Semantics. However, the main root of their ontology is Computer Science. Since Salatino et al. [7] described CSO as a large-scale taxonomy of research areas, we explain this ontology's contents with taxonomy terminologies as well (e.g., children, parent, and descendants).

In this paper, we used the Computer Science root and all of its direct children (Artificial Intelligence, Bioinformatics, Computer Aided Design, Computer Hardware, Computer Imaging and Vision, Computer Networks, Computer Programming, Computer Security, Computer Systems, Data Mining, Human Computer Interaction, Information Retrieval, Information Technology, Internet, Operating Systems, Robotics, Software, Software Engineering, Theoretical Computer Science), and their descendants. We used the entire CSO topics and automatically reformed the ontology in a way it is appropriate for the text classification models by applying a **limitation rule**. The structure of CSO in our version and the limitation rule are explained in the following subsection.

3 1. 1 Classifier Friendly Ontology Format

In our experiment with the flat classification model, we noticed the model would perform poorly using the original format of CSO. We assumed this is because there were too many categories present within the ontology, and each category contains extremely imbalanced numbers of members. In some cases, categories (labels in terms of classification) will have hundreds of member, whereas other categories contain only one member. For classification, this type of extreme data imbalance is not desirable as it prevents the model from performing well during the training process.

To counter this problem, we employed a limitation rule for the ontology format: **We set the maximum level of categories to 2 and 1**, this means only the root topic (Computer Science) and all topics within two levels or one level under the root topic are considered as categories. The rest of the descendant topics are set as sub-topics of the topics within level two or one.

To further explain why we determined the aforementioned rule, we listed some reasons as to why we believe this rule is the right approach to our goal:

（1） We are extending an existing ontology using text classification approaches, which requires a machine learning process. In many cases, a category topic only has one child topic (sub-topic) which later becomes only one training sample. This provides a skewed data and Machine Learning methods do not perform well with a skewed dataset.

（2） Similar to the previous problem, machine learning methods do not perform well with extremely imbalanced data. In CSO, many category topics have hundreds of sub-topics, whereas many category topics have only approximately 1 to 10 sub-topics.

（3） In the original CSO, topics near the leaf level are more repetitive and specific. We believe these topics are more appropriately placed underneath an existing category topic (super-topic) which are more general instead of being made into a new set of category.

3 2 Topic Dictionary

We built a topic dictionary based on CSO. This dictionary includes the topics and their super-topics obtained from CSO. Before applying the level limit rule, our dictionary consists of 32,043 topics alongside their direct parents (super-topics). Then, any multi-worded topics will be split and individually represented with their corresponding word vectors before finally combined. Section 5 1 describes this approach in detail.

4 METHODOLOGY

We aim to compare the performances between flat classification and hierarchical classification to decide which one is potentially more suitable for automating ontology extension. In this section, we describe each of the aforementioned classification approaches.

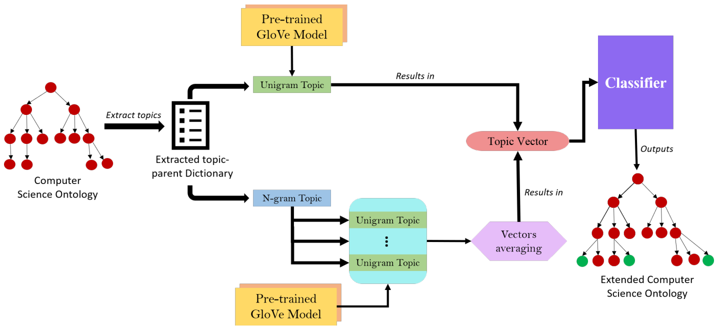

Figure 1: Architecture of the flat classification framework modified to suit our data

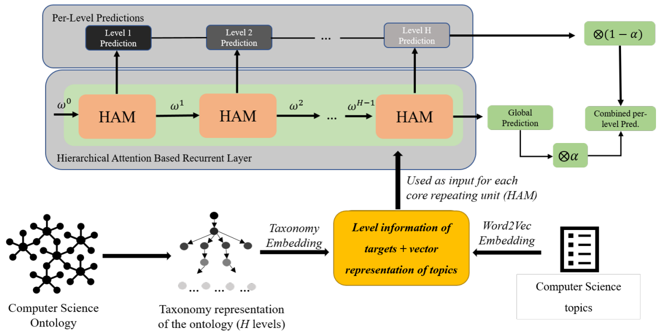

Figure 2: Architecture of HARNN modified to suit our data

4 1 Flat Classification

We implemented the flat classification proposed in [17] by slightly modifying (1) its preprocessing step due to the incompatibility of the Whatizit annotation tool(注4) to our data, and (2) the classification step since we experimented with multiple machine learning models. Essentially, the method proposed in [17] involved three major steps. Figure 1 illustrates the modified framework to suit our data.

Specifically, for our framework, it requires the 840B pre-trained GloVe model [29] (注5). For each of the topics, they will be represented by their word vectors obtained from the pre-trained GloVe model. In the case of ngram topics (n-worded topics e.g., 'Artificial Intelligence'), each word will be represented by its corresponding word vectors before they are combined by vector averaging. The output of this step is the list of topic vectors ready to be classified.

Finally, in the classification step, topics are classified into their appropriate direct super-topics. We noticed problems with a flat classification approach for classifying ontology concepts during this framework's implementation. These problems and our reasons for implementing a hierarchical classification approach are explained in detail in Section 6 of this paper.

4 2 Hierarchical Classification

Aside from the flat classification, we also implemented a hierarchical classification approach based on a multilabel hierarchical text classifier proposed by Huang et al., HARNN [18]. This model uses text documents as input and produce a multilabel output in the final step. Additionally, this model considers level information per output label during the entire classification process. In this setting, the model requires two types of information being pushed into the model as inputs. Therefore, we modified our data to satisfy this requirement. In figure 2, we illustrated the process of the original HARNN model simplified and modified to suit our data.

As shown in Figure 2, the HARNN model takes two types of input information in the first level: the taxonomy and the topic document. This topic document consists of CSO topics alongside their respective direct super-topics regarding the taxonomy information (i.e. the super-topic's level and position in the ontology). The second level of the HARNN architecture is where the classifiers are in use. The number of classifiers depends on the number of levels in the taxonomy. Since our input ontology consists of two levels in total, we implemented two classifiers. In the last level is where the results obtained from each classifier are combined to finalize the final output.

5 EXPERIMENT & RESULTS

5 1 Data Preparation

Inspired by Athubaiti et al. [17] that use text classification to classify words instead of a complete text, we first extracted all topic terms from the ontology. For topic terms that consist of multiple words, we first represent each of this topic's words with its corresponding word vectors obtained from the pre-trained GloVe model. Finally, we combined all word vectors into one using a vector averaging method 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 gathered before combined to represent 'Computer Science' as a whole.

Training and testing data consists completely of the topic phrase information which is their topic vectors, and in the case of flat classification, the data also consists of each of the topic phrase's direct parents information. Table 1 shows a sample of our data format where **Vector_rep** is the 300 dimension vector representation of the **Topic**. Since all of our models are multi-class and multi-label classifiers, our data also includes information regarding topic with multiple super-topics.

Topic	Vector_rep	Super-topics
grid_workflow	-0.08472,..., 0.1793	distributed_computer_systems

Table 1: Sample of our data format

5 2 Evaluation Metric

We employed three evaluation metrics, *precision, recall, and F-measure*(F1 score) for this experiment. *Precision*, also described as $\frac{TruePositive}{TruePositive+FalsePositive}$ is essentially the percentage of how many positive predictions are correctly predicted. *Recall*, also described as $\frac{TruePositive}{TruePositive+FalseNegative}$ is the percentage of the correctly predicted positive instances over all of the positive instances within the dataset. Finally, *F1 score*, also described as $2 \times \frac{precision \times recall}{precision+recall}$ is a measurement that combines precision and recall — a method to calculate the weights of precision and recall in a balanced way.

(注4) : Whatizit tool, https://bit.ly/3phtlWV

(注5) : Pre-trained GloVe, https://stanford.io/3cAbqY5

5 3 Flat Classification

As previously explained, for flat classification, we implemented the classification framework proposed by Althubaiti et al. [17]. We adapted our dataset into a format compatible with the framework in [17]. However, we slightly modified the classifier part of the framework. We used several machine learning models, i.e., *Logistic Regression (LR), Gaussian Naive Bayes (GNB), K Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF), and Support Vector Machine (SVM)* that are different from the Multilayer Perceptron in [17]. We also used two types of data, one is based on the original structure of CSO, and the other is based on the modified structure of CSO. The following two subsections explain the actions taken before and during each of the experiments.

5 3. 1 Without Level Limit Rules Application

We used CSO's original structure. We combined the vector representations of each word in the topic phrase using a simple vector averaging approach for the input format. Table 2 shows the performances of the implemented model using different machine learning models on the dataset based on the original CSO structure.

5 3. 2 With Level Limit Rules Application

Different from the previous experiment, in this experiment, we applied the rules explained in Section 3 1.1. We also applied the same approach for the input format as the previous experiment. Table 2 the implemented model's performances using different machine learning models on the dataset based on the modified CSO structure.

Level Limit	Model	Precision	Recall	F1-macro
Without	LR	0.0198	0.0237	0.0193
	GNB	**0.0238**	**0.1713**	**0.0342**
	KNN	0.0171	0.0216	0.0166
	DT	0.0079	0.0058	0.0060
	RF	0.0001	0.0000	0.0000
	SVM	0.0256	0.0248	0.0227
Limit = lvl 2	LR	0.0787	0.0734	0.0693
	GNB	**0.0569**	**0.4265**	**0.0853**
	KNN	0.0538	0.0511	0.0483
	DT	0.0204	0.0152	0.0159
	RF	0.0070	0.0028	0.0038
	SVM	0.0680	0.0859	0.0708
Limit = lvl 1	LR	0.2473	0.2231	0.2321
	GNB	0.1251	0.6809	0.1908
	KNN	0.2665	0.2098	0.2214
	DT	0.0957	0.0651	0.0762
	RF	0.0824	0.0255	0.0377
	SVM	**0.2504**	**0.2420**	**0.2423**

Table 2: Performances of flat classification with different models and data types

5 4 Hierarchical Classification

5 4. 1 HARNN

We implemented a hierarchical text classification model HARNN proposed by Huang et al. [18] for our experiment using a hierarchical classification approach. For this experiment, there is a slight difference between our data and the data implemented by the original author. In our data, instead of regarding document as a large collection of words, we treat a document as a very small collection of words (less than five words per document) due to our goal of topic terms classification.

Since the application of level limitation majorly increased the performance of the flat classifier models, we used this finding to determine the type of data for HARNN. We used the type of data with level limitation set to 2 only. Even though using data with level limitation set to 1 achieved the best results for the flat classification approach, we did not use this data as setting the level limitation to 1 would defeat the purpose of hierarchical classification — the requirement of multi-level class/category taxonomy.

With HARNN, we trained the model using a 5-Fold cross validation approach. Tables 3 shows the performance of the HARNN model on our dataset in each fold.

5 4. 2 HiLAP

Aside from HARNN, we also implemented another hierarchical text classification model proposed by Mao et al. [19]. However, due to unsatisfactory performance results to our data, we decided not to include it in this paper.

KFold	Precision	Recall	F1-macro
Fold 1	0.0045	0.0117	0.0038
Fold 2	0.0039	0.0216	0.0032
Fold 3	0.0051	0.0125	0.0038
Fold 4	0.0041	0.0188	0.0039
Fold 5	0.0043	0.0178	0.0031
Average	0.00438	0.0165	0.0035

Table 3: Performances of hierarchical classification using the HARNN Model

6 DISCUSSIONS

6 1 Low Performance on Flat Classification

Since the flat classification does not consider the hierarchy of the ontology, we flattened all ontology levels into one, and this action results in an overwhelming number of classification targets or classes because all ontology concepts become classes except for the leaf level concepts. Another problem is how imbalanced our data is. Classification methods do not perform well with too many classification targets as well as with extremely imbalanced datasets.

6 1. 1 Minimizing Classification Targets

According to our experiment results presented in Table 2, the lesser the level will result in better performance. Significant increase in performance was obtained by SVM on data with level limitation set to 1. Without applying the limitation rule, there are a total of 4,241 classes in total. After applying the limitation rule, the numbers of categories decreased drastically. In the case of level limit = 2, there are a total of only 388 classes and for level limit = 1, there are only 30 classes in total. Nevertheless, it should always be remembered that the more the limitation rule is applied, the less fine-grained the classification results are.

6 2 Difference in Data

The input features we used differ from the ones used in our reference works [18] [17]. In the case of HARNN hierarchical classifier [18], hundreds of word vectors are combined to represent the information of a large text document (consisting of more than 100 words). However, in our case, we applied word vector combinations obtained from topic terms which consist of less than 10 words (only around 1 to 5 words), with the fewest being unigrams. Due to this, our data is less rich in information compared to the one applied in HARNN [18] and therefore this may be another cause of far from perfect results in the hierarchical classification, as well as the flat classification. To the best of our knowledge, our work is one of the few, if not the only one that attempted in the automatic extension of CSO topic ontology.

6 3 Flat or Hierarchical

Opposed to our experiment results, we believe Hierarchical classification is a more appropriate approach for extending an existing ontology in terms of theory. First, it can lessen the number of classification targets. The reason is, a classifier exists in every level of the ontology, and it allows a more even spread of targets. Second, the hierarchical method can output multiple labels for each data while considering their most appropriate ontology levels. This suits the characteristic of concepts in ontologies — a concept in an ontology may be a sub-concept of multiple super-concepts, and a sub-concept can be a super-concept of other concepts of a lower level (with root level = 0) within the ontology.

As shown in tables 2 and 3, the performance of HARNN is similar to some of the performance of several models in flat classifier without level limitations applied and with level limitation set to 2. From the results, we believe that despite the decrease of classes in each classifier in a hierarchical classifier structure, and since HARNN architecture considers both local and global predictions [18], an extremely large number of classes in the original data will still make the hierarchical classifier focus on large final classification targets.

7 CONCLUSIONS & FUTURE WORK

We experimented with and presented different performances of two types of text classification approaches (the flat and hierarchical classification approaches) to automate ontology extension. We used CSO as the seed ontology but the data extracted from it is highly imbalanced due to its large scale. Thus, we applied a limitation rule to the original ontology to determine the maximum ontology level from where topics are made into categories (labels or classification targets) for the classifier inputs. It was shown in Table 2 that performances increased if we imposed level limitation on our data. Despite basing our model on [17], we implemented a different preprocessing idea: instead of transforming ngrams into unigrams, we combined word vectors of each word into one to represent a topic as a whole. Finally, our experimental showed that the flat classification approach has better performance, thus making it more appropriate for automatic ontology extension compared to the hierarchical classification approach performance-wise. Nevertheless, theory-wise, the hierarchical classification remains the best approach to ontology extension because it considers the hierarchy properties of the data, which is essential in any tree-format data structure. Furthermore, the flat classifier performs best on data which has been given a limitation rule to level 1. This means the extension can only be performed on the first level of the ontology at best, and therefore preventing a fine-grained classification result.

In the future, we will input new updated research terminologies into the best performing classification model with the best performing parameters to achieve the final extended version of CSO. Aside from this, we will utilize this extended ontology into a creation of user profiles which will later be implemented to a scientific paper recommender system. Finally, an entirely automatic ontology maintenance seems unlikely, but we believe it is possible, hence we would like to explore further into this problem in the future — especially for the hierarchical classification approach.

Acknowledgement

This work was partly supported by JSPS KAKENHI Grant Numbers 18H03242, 18H03342, 19H01138A.

References

[1] Jung, J., Oh, K. and Jo, G., 2009, "Extracting Relations towards Ontology Extension", *In Agent and Multi-Agent Systems: Technologies and Applications* (pp. 242 - 251).

[2] Cruanes J., 2011, "Ontology Extension and Population: An Approach for the Pharmacotherapeutic Domain", *In 16th International Conference on Applications of Natural Language to Information Systems, NLDB 2011 Alicante,*

Spain, June 2011, Proceedings (pp. 342 - 347)

[3] Lehmann, J. and Volker, J., 2014, “ An Introduction to ontology learning. ” , *In Perspectives on Ontology Learning.*

[4] Man, D., 2013, ”Ontologies in Computer Science”, *In Didactica Mathematica, Vol. 31(2013), No. 1* (pp. 43 - 46).

[5] Ruiz-Iniesta, A. and Corcho, O., 2014, ”A Review of Ontologies for Describing Scholarly and Scientific Documents”, *In the 4th Workshop on Semantic Publishing.*

[6] Peroni, S. and Shotton, D., 2012, ”FaBIO and CiTO: Ontologies for Describing Bibliographic Resources and Citations”, *In Web Semantics: Science, Services and Agents on the World Wide Web, 17(0)* (pp. 33 - 43).

[7] Salatino, A. A., Thanapalasingam, T., Mannocci, A., Osborne, F. and Motta, E., 2018, ”The Computer Science Ontology: A Large-Scale Taxonomy of Research Areas”, *In: ISWC 2018: The Semantic Web (Proceedings, Part II), Lecture Notes in Computer Science 11137, Springer* (pp. 187 - 205).

[8] Salatino, A.A., Osborne, F., Thanapalasingam, T. and Motta, E., 2019, ”The CSO Classifier: Ontology-Driven Detection of Research Topics in Scholarly Articles”, *In TPDL 2019: 23rd International Conference on Theory and Practice of Digital Libraries, Springer* (pp. 296 - 311).

[9] Miller, G. A., 1995, ”WordNet: A Lexical Database for English”, *In Communications of the ACM Vol. 38, No. 11* (pp. 39 - 41).

[10] Fellbaum, C., 1998, ed., ”WordNet: An Electronic Lexical Database”, *In Cambridge, MA: MIT Press.*

[11] Ashburner, M., Ball, C. A., Blake, J., Botstein, D., Butler, H. and Cherry, J. M., 2000, ”Gene Ontology: Tool for the Unification of Biology”, *In The Gene Ontology Consortium. Nature genetics, 25(1)* (pp. 25 - 29).

[12] The Gene Ontology Consortium, 2019, ”The Gene Ontology Resource: 20 Years and still GOing Strong”, *In Nucleic Acids Research, January 2019, Vol. 47, Database Issue D330-D338.*

[13] Schriml, L. M., Mitraka, E., Munro, J., Tauber, B., Schor, M., Nickle, L., Felix, V., Jeng, L., Bearer, C., Lichenstein, R., Bisordi, K., Campion, N., Hyman, B., Kurland, D., Oates, C. P., Kibbey, S., Sreekumar, P., Le, C., Giglio, M. and Greene, C., 2018, ”Human Disease Ontology 2018 update: classification, content and workflow expansion”, *In Nucleic Acids Resaerch, January 2019, Vol. 47* (pp. D955-D962).

[14] Osborne, F. and Motta, E., 2015, ”Klink-2: Integrating Multiple Web Sources to Generate Semantic Topic Networks”, *In International Semantic Web Conference, 2015, Bethlehem, Pennsylvania, USA.*

[15] Osborne, F., Motta, E. and Mulholland, P., 2013, ”Exploring Scholarly Data with Rexplore”, *In International Semantic Web Conference, Boston, MA* (pp. 460 - 477).

[16] Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P. and Chi, Y., 2017, ”Deep Keyphrase Generation”, *In 55th Annual Meeting of Association for Computational Linguistics, 2017* (pp. 582-592).

[17] Althubaiti, S., Kafkas, Ş., Abdelhakim, M. et al., 2020, ”Combining Lexical and Context Features for Automatic Ontology Extension”, *In Journal of Biomedical Semantics.*

[18] Huang, W., Chen, E., Liu, Q., Chen, Y., Huang, Z., Liu, Y., Zhao, Z., Zhang, D. and Wang, S., 2019, ”Hierarchical Multi-label Text Classification: An Attention-based Recurrent Network Approach”, *In CIKM ’19: Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 1051–1060).

[19] Mao, Y., Tian, J., Han, J. and Ren, X., 2019, ”Hierarchical Text Classification with Reinforced Label Assignment”, *In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).*

[20] Konys, A., 2019, ”Knowledge Repository of Ontology Learning Tools from Text”, *In 23rd International Conference on Knowledge-Based and Intelligent Information and Engineering Systems* (pp. 1614 - 1628).

[21] Cimiano, P. and Volker, J., 2005, ”Text2Onto: A Framework for Ontology Learning and Data-driver Change Discovery”, *In Natural Language Processing and Information Systems, 10th International Conference on Applications of Natural Language to Information Systems, NLDB* (pp. 227-238).

[22] Buitelaar, P., Olejnik, D. and Sintek, M., 2003, ”OntoLT: A Protégé Plug-In for Ontology Extraction from Text” *In Proceedings of the International Semantic Web Conference(ISWC).*

[23] Pazienza, M. T. and Stellato, A., 2005, ”The Protégé Ontoling Plugin - Linguistic Enrichment of Ontologies in the Semantic Web”, *In Poster and Demo Proceedings of the 4th International Semantic Web Conference (ISWC).*

[24] Weiss, N., 2019, ”The Hitchhiker ’ s Guide to Hierarchical Classification” *Available at: https://bit.ly/3avR95e* (Accessed: 21 December 2020).

[25] Ayadi, A., Samet, A., Bertrand, D. B. D. B. and Zanni-Merk, C., 2019, ”Ontology Populaiton with Deep Learning-based NLP: A Case Study on the BIomolecular Network Ontology”, *In 23rd International Conference on Knowledge-Based and Intelligent Information Engineering Systems* (pp. 572-581).

[26] Liu, W., Weichselbraun, A., Scharl, A. and Chang, E., 2005, ”Semi-Automatic Ontology Extension Using Spreading Activation”, *Journal of Universal Knowledge Management, vol. 0, no. 1.*

[27] Novalija I. and Mladeni D., 2009, ”Semi-automatic Ontology Extension Using Text Mining”, *In Proceedings of the 11th International Multi-conference on Information Society IS.*

[28] Novalija I. and Mladeni D., 2010, ”Content and Structure in the Aspect of Semi-Automatic Ontology Extension”, *In Proceedings of the ITI 32nd International Conference on Information Technology Interfaces* (pp. 115-120).

[29] Pennington, J., Socher, R., and Manning, C. D., 2014, ”GloVe: Global Vectors for Word Representation.” *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*(pp. 1532–1543).