Identification of Dependencies between Learning Outcomes in Computing Science Curricula for Primary and Secondary Education – On the Way to Personalized Learning Paths

Yelyzaveta Chystopolova, Stefan Pasterk, Andreas Bollin, and Max Kesselbacher

University of Klagenfurt, 9020 Klagenfurt, Austria
{Yelyzaveta.Chystopolova, Stefan.Pasterk, Andreas.Bollin, Max.Kesselbacher}@aau.at
https://www.aau.at/en/informatics-didactics

Abstract. The multitude of curricula and competency models poses great challenges for primary and secondary teachers due to the wealth of descriptions. Defining optimal (or personalized) learning paths is thus impeded. This paper now takes a closer look at 7 curricula from 6 different countries and presents an approach for the identification of learning outcomes and dependencies (requires and expands) between them in order to support the identification of learning paths. The approach includes different strategies from natural language processing, but it also makes use of a refined and simplified version of Bloom's Taxonomy to identify dependencies between the learning outcomes. It is shown that the identification of similar learning outcomes works very well compared to expert opinions. The identification of dependencies, however, only works well for detecting learning outcomes that refine other learning outcomes (expands dependency). The detection of learning outcomes which build on each other (requires dependency) is, on the other hand, still heavily dependent on the definition of dictionaries and a computing science topics ontology.

Keywords: Primary and Secondary Education · Learning Outcomes · Computing Science · Natural Language Processing

1 Introduction

Every year the field of Computer Science becomes a bigger part of everyday life not only of scientists but also of all the people. Schools pay more attention to teaching the basics of Computer Science starting already from primary school. However, teachers often face the problem of choosing the optimal learning path, which will be the most suitable for every exact group of students. There are lots of curricula, which describe different standards, and different options of achieving a certain learning goal. In some aspects they differ, but also similarities can be found. Using the collected knowledge from some of them can help to improve the level of Computer Science education in general and be more flexible to the current needs of students as well as to new improvements in the field. Merging of Computer Science curricula rises the question: "How to represent the collected knowledge from several curricula?".

Pasterk and Bollin present a graph-based approach for analysis of Computer Science curricula, where they map the Learning Outcomes (LO) to a graph by connecting them via dependency relations which can be of two types "EXPANDS" and "REQUIRES" [10]. Relation type "EXPANDS" shows that 2 LO share same main topic extending each other, while the relation of the type "REQUIRES" assume that in the pair of 2 LO (LO1 and LO2), LO1 cannot be reached without LO2 and at the same time they do not form a pair, with the relation type "EXPANDS" [9]. Showing a wide range of new possibilities, this model requires considerable effort for dependencies identification between LO.

To identify all the dependencies, the experts need to work with curricula, which are presented in PDF files. Some of them store LO in the form of a table, however, in most cases, they are presented as lists or plain text. Even looking for dependencies inside one curriculum, the experts need to keep in mind dozens of LO. Adding new curricula makes the situation even more complicated. Besides the identification of dependencies inside one curriculum, the experts need to find relations between LO from different curricula. Thus the number of LO to work with rapidly increases to hundreds, which makes the task too complicated for human experts.

The goal of this paper is to describe a semiautomatic approach for dependencies identification between LO among Computer Science curricula for primary and secondary education. We also describe a possibility to transfer the available curricula, which are stored in PDF files, to a directed graph, and also to simplify the addition of new LO in the future.

To introduce the approach, this paper gives answers to the following questions:

- To which extent is it possible to identify dependency relations of the type "EXPANDS" between learning outcomes?
- To which extent is it possible to identify dependency relations of the type "REQUIRES" between learning outcomes?
- To which extent is it possible to identify directions for dependencies between learning outcomes?
- How similar is the semiautomatic approach for dependencies identification to the experts' opinion?

In order to answer these questions we use seven curricula from six countries. This way, we have a diverse set of learning outcomes concentrating on the Primary and Secondary education.

This paper is structured as follows. After a motivation and an overview of related work in first two sections, two approaches for determination of dependencies of two types, together with the approach for direction determination are presented. Section 4 shows first results from the comparison of semi-automatic determination to the precision of experts. The paper concludes with the section for discussion and future work.

2 Related Work

With the rise of the research interest for computer science education, the amount of published literature also increases. More and more papers discuss the analysis of curricula content in different options (e.g. relevance of the topics [4]). Different approaches using graph representations of curricula for analysis and comparison can be found as well (see e.g. [7]). Pasterk and Bollin also propose to present curricula as graphs [10] which opens new opportunities for further analysis. LO in such graphs are presented as nodes, and dependencies as edges between these nodes. Dependent relations can be of two types: "EXPANDS" and "REQUIRES" [9]. Those mentioned approaches are based on expert opinions who add relations between courses, knowledge areas, or LO.

Sekiya, Matsuda, and Yamaguchi [13] use statistical methods from natural language processing (NLP) and text analysis to identify relations between topics and to generate maps of curricula. With the method called *latent Dirichlet allocation (LDA)* topics from curricula are extracted and their relations are calculated. The results from this process using *LDA* are interpreted as coordinates for the generation of maps of curricula. Similar techniques are used by Badawy, El-Aziz, and Hefny [2] to analyze textbooks for higher education based on included LO. They follow their aim to identify important chapters in these textbooks according to intended LO of a curriculum. The steps they take in their research are comparable to a standard process in NLP which includes *data preparation*, *synonym identification*, *data preprocessing*, and the analysis of LO based on *frequency of the occurring words* [2].

Pasterk, Kesselbacher, and Bollin present an approach to semi-automatically categorize LO into *computer science* or *digital literacy* which is also based on NLP [11]. Asking experts to also categorize the LO to produce a validation corpus, they found out that experts focus on keywords, especially on nouns, during categorization, and that the experts' opinions are often diverse. In the best cases the semi-automated system matched in 70% of the LO categorization with the data from the experts [11].

Based on the approach of Pasterk and Bollin [10] the present contribution describes different approaches to semi-automatically determine dependencies of different types between LO and the directions of the dependencies. These approaches are based on the textual analysis of the LO and NLP techniques, and are described in the following section.

3 Methodology

3.1 Background and Process Description

As already mentioned Pasterk and Bollin define the two types of dependencies ”EXPANDS” and ”REQUIRES” [10]. Each of these types needs its own approach for relation determination. Relations of the type ”EXPANDS” connect learning outcomes on the same topic, those which share some similar context. As every learning outcome is presented by a short sentence, we can use sentence similarity measures [1].

In this paper we concentrate on Jaccard Similarity Coefficient. In its original version this measure compares the size of the intersection of the words occurring in two sentences (A, B) to its union (Equation 1).

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

Being rather simple, compared to other Natural Language Processing methods for similarity determination between sentences, the Jaccard Similarity Coefficient includes all the needed features for identifying relations of the type ”EXPANDS”. It is not only easy to implement but can also serve as a good basis for different modifications. Hamedani and Kim propose a few ways of using unweighted and weighted options for link-based similarity measure in graphs [12].

Identification of dependent relations of the type ”REQUIRES” is a more complicated task. Pairs of learning outcomes, which have a connection of this type, can belong to different topics. They mostly do not share common lexis, thus similarity metrics are not suitable for their determination. Nevertheless, even without common lexis, they do have a connection, which can be found by human experts based on knowledge of the field.

Such a knowledge base can be created with the help of relation extraction (RE) technics which give a possibility not only to extract dependent pairs of keywords, but also some background knowledge [5]. It is obvious that for such knowledge extraction we need to have data, where dependencies between LO will be already defined. With such a goal a group of experts was working with 7 curricula from 6 countries to create a validation corpus (VC). As a result of their work, we got a document, which includes identified dependencies of two types (”EXPANDS” and ”REQUIRES”) including directions between LO withing one or several curricula.

Together with identifying pairs of dependent LO and type of relations, we need also to identify directions. It will be easy if we know the level of each LO in the pair. It is a well-known approach in the field of education to use Bloom’s Taxonomy for such goals. A revised version of Bloom’s Taxonomy [8] is popular nowadays, and for computer science a 2-dimensional version is suggested by Fuller et al. [3]. Both taxonomies are based on action verbs which are separated in levels. In the case of revised Bloom’s taxonomy, there are 6 cognitive levels: remembering, understanding, applying, analyzing, evaluating, creating.

In the case of two-dimensional Bloom's Taxonomy, the same categories got transferred into two dimensions [3]: the ability to understand and interpret the existing product, and the ability to design and build a new product.

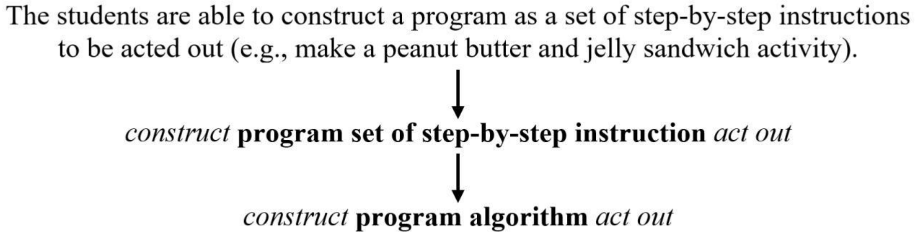
Even though Bloom's Taxonomy is very popular, Johnson and Fuller showed that it is not perfect for Computer Science education [6]. In the course of our study, we noticed that it includes only about 36% of action verbs, which we met in LO from Computer Science curricula for primary and secondary education. Nevertheless, Bloom's Taxonomy served as a ground base for our own approach for direction determination.

3.2 Preprocessing and Standardization

The task of dependencies identification is complex and requires a few preprocessing steps. Besides manual transferring of the curricula that were stored in PDF to CSV files, it includes cleaning and standardization. Firstly we remove all the irrelevant information, such as punctuation, stop words, and also irrelevant phrases such as text in brackets and the phrase ”The students are able to...” (or its equivalents). This step finishes with lemmatization, which helps to present all the words in their base (dictionary) form.

Working with a variety of Curricula we found out that many concepts are presented by different synonyms and it influences the relation determination. Thus the Standardization step aims to reduce the number of diverse synonyms in learning outcomes saving the semantic context. As an example, we met four synonyms for the term **”algorithm”**: *”sequence of events”*, *”sequence of instructions”*, *”sequence of steps”*, and *”set of step-by-step instructions”*.

Figure 1 shows how a learning outcome changes during the preprocessing phase.



The students are able to construct a program as a set of step-by-step instructions to be acted out (e.g., make a peanut butter and jelly sandwich activity).

construct **program set of step-by-step instruction** *act out*

construct **program algorithm** *act out*

Fig. 1. Preprocessing of the learning outcome.

3.3 Weighted Jaccard Similarity

For relation identification of the type ”EXPANDS” we use modified Jaccard Similarity Coefficient. The main difference compared to the original method is that words with different parts of speech have different levels of influence on the result.

If we take a deeper look on the pair of learning outcomes, which are connected, we will see that nouns play the most important role in similarity determination. They show the object (what exactly the student should learn). The next level of importance goes to action verbs. They show what exactly the student should do with the object. Most learning outcomes include also auxiliary words (adjectives, adverbs, prepositions), which are not as important as nouns and verbs, but still have influence on calculations, helping to calculate similarity more precisely.

Modifying the original Jaccard Similarity Coefficient we add weights to it. It means that instead of contributing equally, some parts of speech contribute more than others. The default weight distribution is: nouns - 50%, verbs - 30%, and auxiliary words - 20%.

However, it can change, depending on the presence of different parts of speech in LO. Thus, if LO does not contain any adverbs, adjectives or prepositions, the weight of auxiliary words will be equally distributed between Nouns and Verbs.

3.4 Relation Extraction between keywords

As mentioned earlier, nouns are the most meaningful for relation determination. Even though nouns in the pairs of learning outcomes connected with the type "REQUIRES" in most cases belong to different categories, we can still see the connection between them.

With the validation corpus, it was possible to create a knowledge base, which shows related pairs of keywords. Based on the expert evaluation, we extracted pairs of keywords from the pairs of LO with the relation type "REQUIRES". Thus we got pairs of keywords (which include nouns and verbs) in the form *requires(K1, K2)*, where the term *K1* requires *K2*.

The problem of such an approach was that automatic extraction gave us not only words which are relevant for Computer Science, thus we needed to edit the table manually.

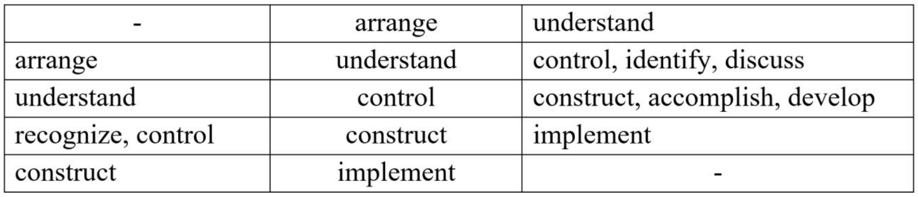
Having such knowledge as a basis, we can use it for dependency identification between LO. If two LO contain an extracted earlier pair of related keywords, we can preliminary see that such LO might be connected. However, to check it more precisely, we need to look for the percentage of keywords, which have a pair in the opposite LO. If the result is more than 37%, it is most likely that LO we are currently checking are connected, otherwise, they are not.

3.5 Action Verbs Triples

In the field of education, Bloom's taxonomy is a very popular tool for level determination of learning outcomes. Knowing the level we can easily determine the direction of relation for the connected pair of LO. Nevertheless, our try to apply it on praxis did not show satisfying results. Less than 5% of directions were determined. The problem was, that having a variety of action verbs divided in 6 levels (in the revised Bloom's Taxonomy), it is still missing most action verbs specific for the field of Computer Science.

Giving better results in those cases, when action verbs were included, the Revised Bloom's Taxonomy served as a basis for the next idea: using triples of Action Verbs, extracted from the Validation Corpus (Figure 2).

The first step in this process is to extract all the action verbs (AV) and to add them into the middle column of the three-column table. It is known from VC, that directions between LO go from lower to higher levels. Applying this knowledge for AV, two other columns can be added. For each AV in the middle column, we extracted possible AV of lower and higher levels by investigating the pairs of related LO. For a pair of related LO (LO1, LO2), with LO1 being of a lower level compared to LO2, the AV of LO1 can be added to the left column of the respective AV of LO2. On the other hand, the AV of LO2 can be added to the right column of the respective AV of LO1. To avoid cases when the same AV appear in one row in the left and right column, we use a count (how many times the AV was met in this position). If the count of the AV from the left column is higher, it stays there, otherwise in the right one.



-	arrange	understand
arrange	understand	control, identify, discuss
understand	control	construct, accomplish, develop
recognize, control	construct	implement
construct	implement	-

Fig. 2. Automatically extracted triples of Action Verbs.

The table shows action verbs from easier to harder levels (left to right). With its help we know that for example before being able to *"control"* something, the student should *"understand"* it.

With the help of such a table, we cannot identify the exact level of the LO, however, we can check which of two LO to compare are of the higher level. Thus if LO1 includes the action verb *"understand"* while LO2 includes *"discuss"*, we can see that LO2 is of a higher level than LO1.

Applying it for relation determination, we increase the percentage of determined directions between pairs of learning outcome to 83%. However there were still not-determined directions. The problem was that in some pairs both learning outcomes had the same action verb. An improvement of the identification of the dependency direction between such LO is difficult using Action Verb Triples, or Bloom's Taxonomy. The possible solutions of this problem are to be discussed in Section 5.

In the following section the results of applying the methodology are presented.

4 Trial Results

4.1 Relation determination

Each of two types of relations between LO needs its own approach for dependency identification:

- for the type ”EXPANDS” we use weighted Jaccard Similarity Coefficient, where weights are distributed depending on part of speech,
- for the type ”REQUIRES” we use extracted pairs of keywords *requires(K1, K2)*, where the term K2 requires the understanding or knowledge of the term K1.

As we are using a weighted Jaccard Similarity Coefficient (the metric, which gives us a probability of how close are two sentences) for relation identification of the type ”EXPANDS”, the results depend on the probability boundary (Figure 3).

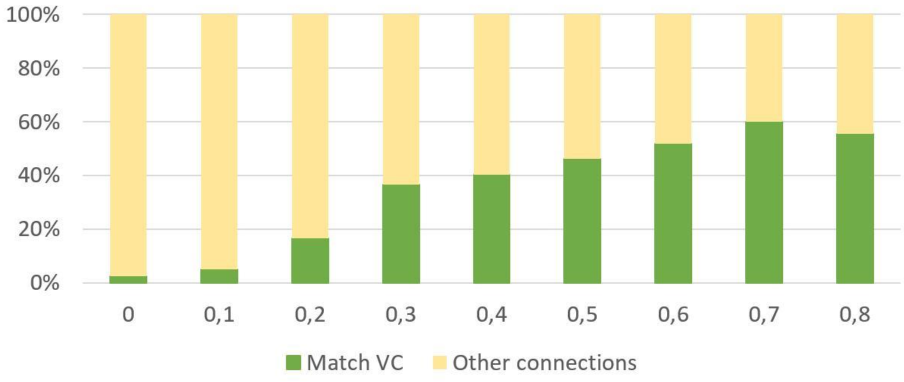


Fig. 3. Automatically determined relations of the type ”EXPANDS”.

The x-axis shows the probability of how close the LO are in the range from 0 to 1. The closer the value is to 1, the more similar are the LO. The y-axis shows with the green color percentage of determined relations which match VC comparing to all the determined dependencies (yellow color) of the type ”EXPANDS”.

We can see, that the higher the probability is, the higher is the percentage of identified dependencies that match VC. However, Figure 4 shows us, that the higher the probability is, the lower is the percentage of identified dependencies from VC.

Analyzing the results, presented on the Figures 3 and 4, we choose the boundary of 0.4, as it maximally reduces the percentage of noise (which is now 54.02%

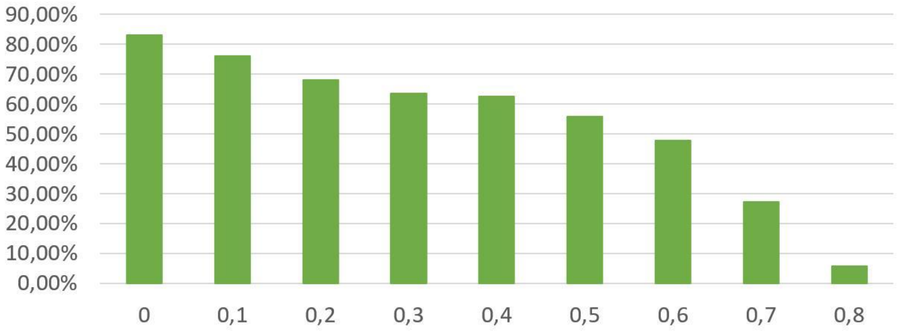


Fig. 4. Automatically determined relations of the type “EXPANDS” which match VC.

compared to the lower probability boundaries) and at the same time helps us to identify 62.5% of dependencies from VC.

Besides comparing the gained results to VC we asked experts for a new evaluation. The goal was to check whether all the pairs of LO which were not met in VC were really identified wrongly, or if they were simply missed by experts during the first evaluation. The results of Precision and Recall for dependency identification of the type ”EXPANDS” showed that the actual level of noise is much lower (Figure 5). Having identified 41% (recall) of dependencies from VC, the second evaluation done by experts showed that 96.4% (precision) of automatically determined relations of the type ”EXPANDS” were identified correctly.

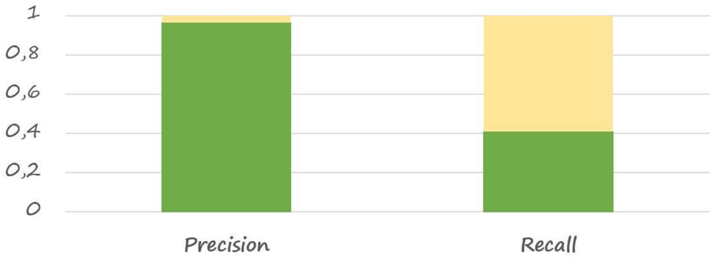


Fig. 5. Results of the Precision and Recall for “EXPANDS” relations

As for dependencies identification of the type ”REQUIRES”, the results were less satisfying. Related pairs of keywords gave us a chance to identify a high amount of dependencies from VC, but at the same time, the level of noise was too high even after excluding pairs of keywords that are irrelevant for Computer Science (Figure 6).

Together with reducing the level of noise (Figure 6), also the percentage of identified relations from VC was cut in half.

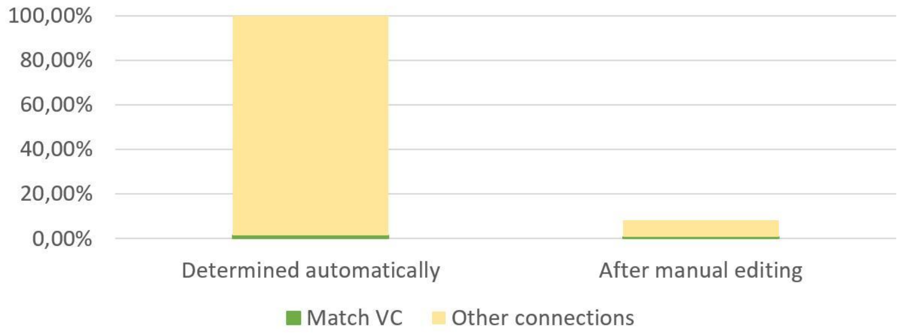


Fig. 6. Automatically determined relations of the type ”REQUIRES”.

Asking experts for reevaluation, we found out, that our approach helped to identify dependencies of the type ”REQUIRES” with a precision of 52.7% (which includes related pairs of LO which were missed by experts during the first evaluation) and a recall of 10.3% (Figure 7). It shows that, in order to identify relations missed by experts, our approach needs improvement, as the level of noise is too high.

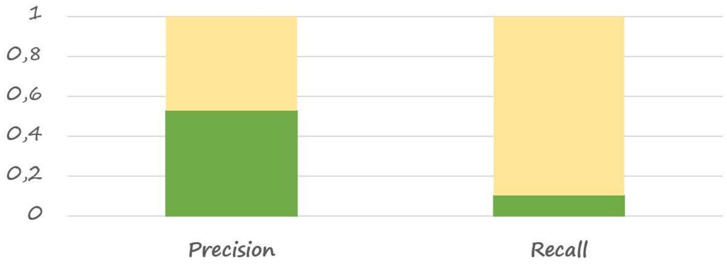


Fig. 7. Results of the Precision and Recall for “REQUIRES” relations.

4.2 Direction Determination

Direction determination is needed for the relations of the type ”EXPANDS”. Jaccard Coefficient shows us that Learning Outcomes are connected, but does not show the order. That is why we are using Triples of Action Verbs, to identify the relative level between two LO which have a dependency of the type ”EXPANDS”.

In the case of direction determination, we were able to get satisfying results using triples of Action Verbs. This approach gave us 88.5% of correctly determined directions.

The main problem of such an approach is, that it is impossible to identify the direction, if both LO from the pair share the same Action Verb. That is why we still get 11.5% of not identified directions.

5 Discussion of Findings

Dependency identification between LO which are presented as short sentences is a rather complicated task. While dependency as well as direction identification of relations of the type "EXPANDS" works quite well, it is still pretty weak for relations identification of the type "REQUIRES".

It is easier to identify relations of the type "EXPANDS" as they share similar topics. Nevertheless, some related LO share terms, which belong to the same topic, but are not synonyms or that obviously connected.

The approach for dependencies identification of the type "REQUIRES" still needs improvement. While a high percentage of dependent LO from VC can be identified, the level of noise still has to be reduced. Nevertheless, it can serve as a good basis for example for developing a Computer Science Ontology of key terms. Such an ontology might be helpful both for the identification of relations of the type "EXPANDS" and direction identification.

Triples of action verbs show positive results for direction determination (88.5%). As Bloom's Taxonomy served as a basis of our approach, the range of action verbs can be extended. The problem of defining directions between those LO which share the same action verb still remains a problem which we also hope to solve in the future with the help of ontology.

6 Conclusion

In the current paper, we show the approach, which will serve as a helping instrument for experts in dependencies identification between LO from Computer Science curricula. It includes relation identification of two types: "EXPANDS" and "REQUIRES". We can identify the relations of the type "EXPANDS" with the precision of 96.4%. As the weighted Jaccard Similarity Coefficient, the method for relation identification of the type "EXPANDS" gives us results only on the probability of how close are the two LO, we use also triples of action verbs for direction identification. Such an approach gives us a possibility to identify directions with the precision of 88.5%.

The task of dependency identification of the type "REQUIRES" is more complicated. It requires a knowledge base, which includes dependencies between key terms. With such knowledge, we were able to identify relations of this type with a precision of 52.7%. Our approach gives a high level of noise, which we plan to decrease by developing and using an ontology of Computer Science terms.

Approaches for dependency identification of both types still can be improved, however already on this point they showed, that human experts miss the radical amount of relations. For example, only half of the relations of the type "EXPANDS" was identified by experts during primary evaluation.

The presented approach, for now, cannot serve as an independent tool for dependency identification, nevertheless, it is a helping hand for experts. It makes a process of relation identification faster and easier, showing related pairs of LO, it gives an opportunity for experts to work with a bigger amount of computer science curricula, and simplify the task of adding new ones.

References

1. Achananuparp, P., Hu, X., Shen, X.: The evaluation of sentence similarity measures. In: International Conference on data warehousing and knowledge discovery. pp. 305–316. Springer (2008)
2. Badawy, M., El-Aziz, A.A.A., Hefny, H.A.: Analysis of learning objectives for higher education textbooks using text mining. In: 2016 12th International Computer Engineering Conference (ICENCO). pp. 202–207 (2016)
3. Fuller, U., Johnson, C., Ahoniemi, T., Cukierman, D., Hernán-Losada, I., Jacková, J., Lahtinen, E., Lewis, T., Thompson, D., Riedesel, C., Thompson, E.: Developing a computer science-specific learning taxonomy. ACM SIGCSE Bulletin **39**, 152–170 (2007)
4. Gupta, S., Dutta, P.K.: Topic objective and outcome: performance indicators in knowledge transfer through in-depth curriculum content analysis. Procedia Computer Science **172**, 331 – 336 (2020), 9th World Engineering Education Forum (WEEF 2019) Proceedings
5. Ji, G., Liu, K., He, S., Zhao, J.: Distant supervision for relation extraction with sentence-level attention and entity descriptions. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
6. Johnson, C.G., Fuller, U.: Is bloom's taxonomy appropriate for computer science? In: Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006. pp. 120–123 (2006)
7. Lightfoot, J.M.: A Graph-Theoretic Approach to Improved Curriculum Structure and Assessment Placement. Communications of the IIMA **10**(2), 59–73 (2010)
8. LW, A., DR, K., PW, A., KA, C., Mayer, R., PR, P., Raths, J., MC, W.: A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives (01 2001)
9. Pasterk, S.: Competency-Based Informatics Education in Primary and Lower Secondary Schools. Ph.D. thesis, University of Klagenfurt - Department of Informatics Didactics (2020)
10. Pasterk, S., Bollin, A.: Graph-based analysis of computer science curricula for primary education. In: 2017 IEEE Frontiers in Education Conference. pp. 1–9 (2017)
11. Pasterk, S., Kesselbacher, M., Bollin, A.: A semi-automated approach to categorise learning outcomes into digital literacy or computer science. In: Passey, D., Bottino, R., Lewin, C., Sanchez, E. (eds.) Empowering Learners for Life in the Digital Age. pp. 77–87. Springer International Publishing, Cham (2019)
12. Reyhani Hamedani, M., Kim, S.W.: Jacsim: An accurate and efficient link-based similarity measure in graphs. Information Sciences **414**, 203–224 (2017)
13. Sekiya, T., Matsuda, Y., Yamaguchi, K.: Analysis of computer science related curriculum on lda and isomap. In: Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education. pp. 48–52. ITiCSE '10, ACM, New York, NY, USA (2010)