Dependency Formalisation for Improved Learning Processes

Konrad Voelkel
Mathematisches Institut der Universität Freiburg
konrad@konradvoelkel.de

Abstract: This paper describes dependencies between knowledge artefacts and how their formalisation can improve some kinds of extremely goal-oriented learning processes.This kind of learning is already performed by researchers and students, but without formalisation. The paper proceeds with an analysis of requirements for tools for dependency formalisation. As a very domain-specific example, it discusses the tools available for formalisation of dependencies in mathematics.

1 The problem of dependency resolution in learning

Dependency resolution processes in learning situations: Given an unknown concept or problem X, the learner[1] originally interesting in a different field Y, wants to get a good overview of the basic notions and the current state of research about X, without wasting too much time better spent on proper research in the field Y. He therefore asks himself: "What should I learn next, to get to X as fast as possible?".

The current state: The dependency relationship between knowledge artefacts, such as scientific documents, is encoded in the literature by natural language. To extract this information, one has to read the entire text as well as all explicitly and implicitly referenced documents.

Missing the opportunity of re-usage: During the process of learning a concept X, the learner implicitly creates a network of dependencies between the different concepts involved to understand X. This network of dependencies is often in memory, or written in personal notes. In either way, it is not available for re-use.

In fact, the very same network of dependencies is created during the process of writing an article or textbook. It is then thrown away, when linearising the dependency network to create a document from the internal knowledge model [Völ07].

[1] A learner in this setting is someone who wants to acquire knowledge about something. This may be a student or a researcher.

1.1 An example of formalised dependencies among knowledge artefacts

Consider the following set of formalised dependencies[2]

```
graphs -> incidence matrices of graphs
matrices -> incidence matrices of graphs
inc. matrices of graphs -> flow network optimization
matrices -> efficient matrix operations
eff. matrix op. -> eff. flow network opt.
flow network opt. -> eff. flow network opt.
```

Imagine you are in the position of someone who tries to solve a problem X via efficient flow network optimization, but doesn't know anything about it. Then you have to decide the next step in the learning process. If you have some knowledge about graphs, then the dependency analysis shows clearly, that you have to learn the concept of a matrix next.

1.2 Benefits of formalised dependencies

Apart from being able to re-use knowledge that has already been created, formalisation of dependencies has more benefits.

The big picture: Teaching knowledge details becomes obsolete, since time is limited and knowledge is always growing and becoming obsolete at the same time. Teaching the big picture instead of technical details will become more and more important, since filling in details can be done individually with the literature.[3]

In many situations, it is not necessary to know every related concept to a certain problem in detail. It is crucial to have some clues, which concepts are required to be analysed in depth and which concepts may be treated as a black box. This is already standard practice, but the knowledge about which concepts are usable as black boxes, is hidden.

Recommendation engines: If there is a formalised dependency network available for some set of concepts, the learning process may be improved by computers. Given a set of already learned knowledge artefacts, the computer can print a list of possible next steps. Ideally, one would also get for each distant goal a recommendation for the preferred next step. According to Schunk, this would have a psychological impact on the learner, too [Sch90].

A formalised dependency network may be used to produce a curriculum. In fact, developing a curriculum requires thinking about dependencies beforehand, which is easier if a previous curriculum is available - and thus easier if a dependency analysis has been done before. Explicitly stated dependencies then remove the need to get the information directly from a previous teacher. Currently, this is already done in some textbooks, where the authors recommend different curricula for courses using their books, some even using flow diagrams relating the chapters of their books.

[2] where the arrow means that the left hand side is required to understand the right hand side.

[3] Of course, to master a field, in-depth exploration is still necessary.

1.3 Costs of formalisation

Since formalisation of knowledge is a costly process, which is complex and hard to change radically, it is important to keep costs low and enable immediate benefit, when proposing changes to current working processes.

It is possible to separate dependency formalisation from other knowledge formalisation processes (such as the creation of documents), to enable the consumers of knowledge artefacts to further refine the formal structure. The learner has direct benefit from dependency formalisation, so he is more likely willing to invest time, than the author of a document.

2 Solution approaches: requirements and related tools

Implementation requirements: Knowledge artefacts have to be formally represented for the purpose of dependency analysis. Concepts are already represented by the text snippets that discuss and explain them, so a pragmatic approach could use text snippets and literature references used nowadays to implement formalisation of dependencies.

Using this perspective, every ordinary reference management system can be used for dependency formalisation, given that it supports internal cross-references (resp. relations of any form).

Ontology: Three relations are necessary to formalise dependencies as sketched above[4].

`Same-as` is required to identify various text snippets discussing the same concept.

`Contained-in` refers to one text snippet (representing a concept) being contained in a bigger document (representing a usual reference in a reference management system).

`Depends-on` is the inverse relation of the arrow used in the example 1.1.

Related tools: BibTeX is a reference storage system as described in 2 [Fed]. Together with any BibTeX-editor supporting custom fields, it is easy to implement dependency formalisation without touching any programming language. This would be the approach with the least costs, but also with the least improvements. Further improvements can be made by exporting the BibTeX data to RDF and then using semantic web tools to manipulate and visualize the data.

Other approaches choose to formalise more than just dependencies. In what follows, we focus on the field of mathematics, which is particularly affine to formalisation.

[4]In addition to these relations, many compatible formalised relations seem plausible for usage with reference management systems, such as the relation *see-also*.

2.1 Domain-specific solutions in mathematics

During authorship of mathematical texts, the inner structure of the knowledge model can be revealed with the usage of special LaTeX tools like SALT (semantically annotated LaTeX) [GHMD07]. Although SALT is not yet domain-specific for mathematics, its techniques are easily adaptable for such purposes. Given the LaTeX-source of a document, a third person can add meta-data like explicit dependencies. De Jong showed how to extract the dependency structure of a mathematical proof out of LaTeX-reference information [DJ].

Going away from the document paradigm, semantic wikis have taken a lot of interest [KV09]. Domain-specific approaches like SWiM can handle domain-specific advantages even better, thus providing more benefit at the same cost [Lan08].

Similar in spirit to SWiM, there is a highly developed computer-aided learning system with recommendation engine for mathematics at the high-school level, called ActiveMath [MAB+01]. While the benefit of such a system is high, the same can be said of the costs.

Mathematics also allows very strong formalisations, unknown in other sciences, namely computer-checked theorem databases like MIZAR [Rud92].

References

[DJ] Johan De Jong. *Stacks Project*. http://math.columbia.edu/algebraic_geometry/stacks-git/.

[Fed] Alexander Feder. BibTeX Format Description. http://www.bibtex.org/Format/.

[GHMD07] T. Groza, S. Handschuh, K. Möller, and S. Decker. SALT-Semantically Annotated LaTeXfor Scientific Publications. *The Semantic Web: Research and Applications*, pages 518–532, 2007.

[KV09] M. Krötzsch and D. Vrandecic. Semantic wikipedia. In *Social Semantic Web*, pages 393–421. Springer, 2009.

[Lan08] Christoph Lange. SWiM–a semantic wiki for mathematical knowledge management. *The Semantic Web: Research and Applications*, pages 832–837, 2008.

[MAB+01] E. Melis, E. Andres, J. Büdenbender, A. Frischauf, G. Goguadze, P. Libbrecht, M. Pollet, and C. Ullrich. ActiveMath: A generic and adaptive web-based learning environment. *International Journal of Artificial Intelligence in Education*, 12(4):385–407, 2001.

[Rud92] P. Rudnicki. An overview of the Mizar project. In *Proceedings of the 1992 Workshop on Types for Proofs and Programs*, pages 311–332, 1992.

[Sch90] D.H. Schunk. Goal setting and self-efficacy during self-regulated learning. *Educational Psychologist*, 25(1):71–86, 1990.

[Völ07] Max Völkel. From Documents to Knowledge Models. In *Proc. of the 4th Conference on Professional Knowledge Management*, pages 209–216, 2007.