Automatic Concept Relationships Discovery for an Adaptive E-course

Marián Šimko and Mária Bieliková
{simko, bielik}@fiit.stuba.sk
Institute of Informatics and Software Engineering,
Faculty of Informatics and Information Technology,
Slovak University of Technology

Abstract. To make learning process more effective, the educational systems deliver content adapted to specific user needs. Adequate personalization requires the domain of learning to be described explicitly in a particular detail, involving relationships between knowledge elements referred to as concepts. Manual creation of necessary annotations is in the case of larger courses a demanding task. In this paper we tackle a concept relationship discovery problem that is a step in adaptive e-course authoring process. We propose a method of automatic concept relationship discovery for an adaptive e-course. We present two approaches based on domain model graph analysis. We evaluate our method in the domain of programming.

1 Introduction

Authoring an adaptive educational system consists of several steps and differs among particular methodology employed. Nevertheless, one step is common: underlying domain model creation. Its purpose is to describe the domain area that is the subject of learning. The description is most commonly provided in the form of a concept map [1]. Interlinked concepts resemble lightweight ontology (refer to Figure 1) where relationship types are limited to those relevant for educational process. Typical example is a *prerequisity* relationship determining that two concepts have to be learned in a given order. Similarly, a *similar-to* relationship represents the fact that concepts are similar to a certain extent (e.g., they represent topics that often appear together in learning objects). Concepts are also connected with educational material. A weighted relationship determines the degree of concept's relatedness to (or "containess" in) a learning object – educational material portion. Learning objects are not limited to explanatory text (such as chapters or sections from books), but represent also exercises, examples, etc.

Accurate identification of concepts and their relationships is crucial to adaptation quality (e.g., intelligent concept recommendation). However, when authoring a course, a suitable domain model is often not available. Although some standardized domain ontologies exist, they suffer from excessive generalization and only exceptionally fit to the author's needs at the desired level of granularity. In such cases the domain model has to be created "from scratch". Unfortunately, manual construction is a tedious and time-consuming task even for small domains. If there are dozens of concepts identified, relations are counted by hundreds. Adaptive e-course authoring and maintainability complexity is a major bottleneck of adaptive educational systems.

Our research goal is to support teachers (content authors) in adaptive e-course authoring process. In this paper, we tackle automatic concept relationship discovery problem. We aim at concept similarity computation that is a core step in a relationship creation process. We propose two approaches based on graph algorithms processing underlying domain model portion.

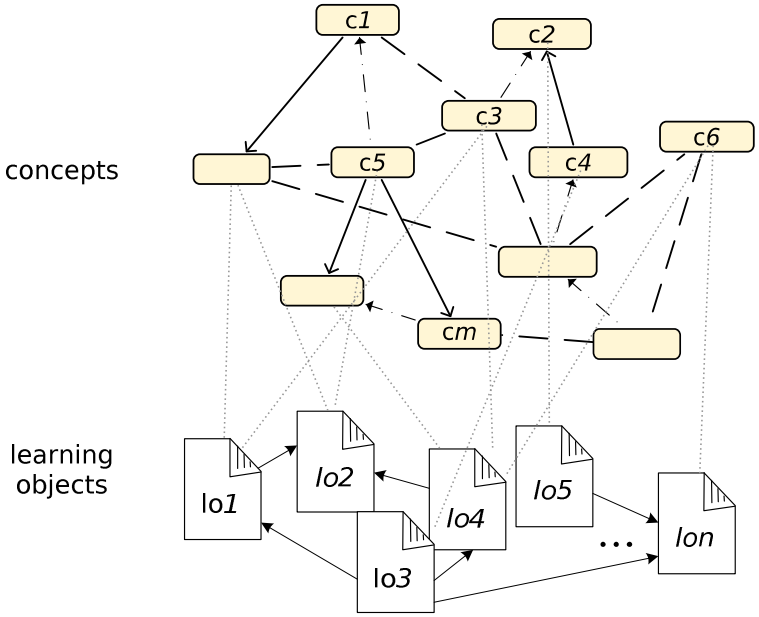

Figure 1. E-course domain model.

The rest of the paper is structured as follows. In section 2 we discuss related work. Section 3 introduces the proposed method, while sections 4 and 5 provide its description in more detail. Section 6 provides evaluation details and the results of performed experiments. In section 7 we sum up our contribution and discuss future work.

2 Related work

The work related to concept relationship discovery in the area of adaptive e-learning is presented in [6]. Concept similarities are computed based on the concept domain attributes comparison. However, the meaning of *concept* in the terminology of Cristea is different from one described by Brusilovsky [1]. Although it contains domain attributes, it also holds textual representation. This should be considered an intentional description from the ontological point of view, but then the reusability of such concepts is arguable. We are not aware of any further evidence of (semi-)automatic concept relationship generation in the adaptive e-learning field.

Finding relations between concepts is a subtask of ontology learning field [10]. The relations being created typically have taxonomic character (*is-a*). Considering text mining, related approaches mainly utilize natural language processing (NLP) techniques. Relations are induced based on linguistic analysis relying on preceding text annotation [2], incorporating formal concept analysis (FCA) [4] or using existing resources such as Wikipedia or WordNet [5]. The drawback of relationship discovery as an ontology

learning task (in relation to e-course authoring) is its dependency on precise linguistic analysis. Most of the approaches rely on lexical or syntactical annotations, the presence of powerful POS taggers, existing domain ontologies, huge corpuses or other external semantic resources (e.g. WordNet). As mentioned above, this knowledge is often not available during an e-course authoring. The solution for teachers should involve unsupervised approaches to unburden them from additional work. This need we address in the method we propose.

The task of structuring the concept space is also present in the area of the topic maps. In this field, the elementary units – topics – we can view analogical to concepts. Authors in [7] generate relations between topics by analyzing the HTML structure of Wikipedia documents. The results indicate that learning objects representation structure should be considered when structuring the domain space. Categorization methods are used in [9] where similar topics are discovered by latent semantic indexing (LSI) and K-means clustering. Unsupervised methods serve as guidance in topic ontology building. Similar approach is missing in the area of adaptive e-learning.

Our method is based on statistical unsupervised text processing and graph analysis related to actual knowledge about the domain. We explore generated or existing concept associations in order to reveal hidden semantic relationships. We were motivated by good results of graph analysis employment achieved in the area of Web search. Our method does not depend on external semantic resources allowing to be used in various domain environments. However, additional semantic connections need not to be excluded.

3 Automatic concept relationship discovery

The concept relations discovery is one of several steps in the adaptive e-course authoring process. Prior to this step we assume a teacher has already created (eventually reused) learning objects, put them into reasonable structure (e.g., hierarchical), identified domain concepts and assigned them to learning objects. Concept extraction can also be done semi-automatically as we show later.

Our goal is to utilize the actual knowledge and discover relationships between concepts. As we represent a domain using a graph model, we conduct a graph analysis. We employ two alternative graph algorithms suitable to this task. Before the algorithms are applied we perform learning objects preprocessing. Because we are interested in e-learning domain, we consider several specifics when dealing with the knowledge discovery.

Learning objects are present mostly in a textual form. Thus we employ text mining techniques. We suppose the number of learning objects is known. This enables us to compute inverse document frequency when building term vector-based learning objects representations. Moreover, the learning objects we process are related to one domain area. This fact reduces the concept ambiguity problem unlike when processing heterogeneous sources. Learning objects are often mutually interconnected. We usually know the hierarchical relationships between learning objects in the course or references to other course parts may exist.

Due to the specifics we are able to compose relatively accurate vector representation during the preprocessing. For example, we can employ bag-of-words model with tf-idf weights. Based on the weights we determine the degree of each concept's relatedness to all learning objects.

After the preprocessing step we apply concept relationship discovery method itself. Using selected graph analysis algorithm we (1) compute concept similarity scores and finally (2) create relationships between each concept and his top-*k* most similar neighbors. For graph analysis we employ two alternatives: spreading activation and PageRank-based approach. Both approaches are described in more detail in the following sections.

In the second step, for each concept the most similar neighbors according to computed similarity are chosen. The top-*k* set we define as a set where the most similar neighbors sorted by score accumulate *k*% of all neighbor similarity values. For example, top-20 neighbors accumulate 20% of sum of all neighbors' similarities. We typically set coefficient *k* from <10; 20>. Between each concept and its top-*k* neighbors we create relationships. The relations' weights are normalized with regard to the whole domain model.

4 Spreading activation

The principle of the spreading activation approach is to consider the domain model to be a contextual network. Contextual network is network where several types of nodes exist. We recognize two node types: learning object nodes and concept nodes. In contextual networks, the spreading activation method is often used for similarity search [3]. The queried node is activated with energy E that spreads to neighbor nodes via incident edges. Final energy distribution in the graph determines the similarity of nodes. We use this principle to compute the degree of the concepts' similarity.

Basic steps of the algorithm can be described as follows:

```
For each concept c_i:
   1. activate concept with initial energy E_0,
   2. spread activation to entire graph,
   3. determine the degree of similarity to all concepts.
```

In step 2 the energy spreads from an activated node to all neighbor nodes proportionally according to outgoing relationships weight. Weights derived from learning object vector representation determine concept relatedness to different learning objects (refer to example depicted in Figure 2).

The crucial step is step 3. After activation spreading finishes, each concept in the network is activated with energy according to its relatedness to concept c_i. The degree of similarity between concept c_i and its neighbor c_j is computed as follows:

$$sim_{i,j}^{sa} = \frac{E_j}{\sum_k E_k} \log(sd_{i,j}) \qquad (1)$$

where $sim^{sa}_{i,j}$ is spreading activation similarity between concepts c_i and c_j, E_j is the energy of concept c_j and $sd_{i,j}$ is the shortest distance between the concepts c_i and c_j in the contextual network graph. The purpose of the formula is to normalize the power-law distributed activation energy values.

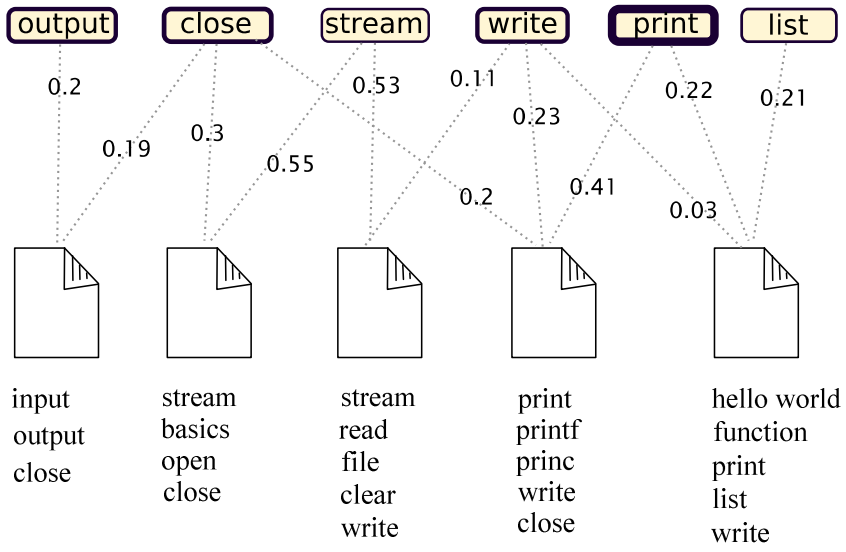

Figure 2. Example of a contextual network with two types of nodes: concepts and learning objects with the simplified vector representation. After activation spreading finishes, each concept in the network is activated with energy depending on the initially activated concept "print" (accumulated activation is visualized by the concepts' border width).

5 PageRank-based Analysis

The graph representation of the domain model forms the basis for the second variant of concept-to-concept similarity computation. This approach builds on the algorithms for estimating relative importance in networks [12]. Such algorithms are used to compute the quantitative measure of node similarity with respect to a given node (or set of nodes). We employ PageRank with Priors in particular, successfully used in categorization systems [8]. We modify Diedrich's and Balke's core idea to allow for the inclusion of weights between learning objects and concepts. The approach principle lies in the propagation of actual domain model topology characteristics into the explicit links between concepts.

The algorithm consists of the following steps:

1. for each concept c_i select concepts connected via exactly one learning object,
2. for concept c_i and selected neighbors compute relation weight $ow_{ij,}$
3. build a temporary domain graph where concepts are nodes and $ow_{i,j}$ weights are assigned to edges,
4. for each concept c_i select the most related co-occurring neighbors,
5. for each concept c_i compute the PageRank scores biasing on the selected neighbors.

For $ow_{i,j}$ computation in step 2 we use the following equation:

$$ow_{i,j} = \sum_{lo_k \in LO} \sqrt{w_{i,k} w_{j,k}} \quad (2)$$

where $ow_{i,j}$ is the relation weight between concepts c_i and c_j, and $w_{i,k}$ is the weight assigned to the association between the concept c_i and learning object lo_k. The idea is that weights are considered to be probabilities that the concept is related to learning object. The resulting probability of concepts' similarity is the mean of individual probabilities.

After building a graph in step 3, the most related co-occurring neighbors are determined in step 4. The most related neighbors we consider neighbors that accumulate top-k % of the sum of all neighbors' similarity scores to a given concept.

In step 5 we use the PageRank analysis algorithm (concepts and temporal links in between are analogical to the Web) to adjust the graph and compute the prestige of nodes. PageRank scores represent similarity $sim^{prp}_{i,j}$ towards a biased node set with regard to relations within the whole graph. The sorted set of all graph concepts is subject to top-*k* selection step in order to obtain only relevant relationships.

6 Evaluation in the programming learning domain

The proposed method we evaluated in the domain of programming learning. For an experiment we used Functional programming course being lectured at the Slovak University of Technology in Bratislava. The Functional programming course is a half-term course consisting of 70 learning objects on the functional programming paradigm and programming techniques in the Lisp language. The learning material is hierarchically organized into chapters and sections according to a textbook used in the course. Learning objects are represented using the DocBook markup language enabling easy processing.

The method results we evaluated against manually constructed functional programming concept map. The course lecturer together with randomly chosen sample of 2007/08 course students were involved. Manual creation of concept map comprised the assignment of weighted values to concept relationships. As assigning continuous values from interval <0; 1> is non-trivial task, possible weight values were limited to set {0, 0.5, 1} implying:

- 0 – concepts are not related to each other (no relation),
- 0.5 – concepts are partially (maybe) related to each other (weak relation),
- 1 – concepts are highly (certainly) related to each other (strong relation).

There were 366 relationships created, 216 were weak relations while 150 were strong relations.

In first step of experiment we preprocessed learning objects and composed their bag-of-words term vector representation. The frequency of the terms presented as domain keywords in the textbook index was boosted. We extracted concepts as most frequent

terms and their normalized tf-idf values we used as weights for concept-to-learning object associations. Hereby we executed all necessary steps prior to concept relationship discovery.

After preprocessing we separately applied both proposed method alternatives and obtained concept relationships. The relationships were compared to manually constructed reference concept map using well-known *precision* and *recall* measures and their harmonized mean, the *F-measure*. In order to gain more accurate evaluation, we extended the original recall measure to involve the manually constructed domain model relationship types:

$$R^* = \frac{|\, retrieved \cap (correctA \cup correctB) \,|}{|\, correctA \cup (correctB \cap retrieved) \,|} \tag{3}$$

where R^* is the extended recall measure, *retrieved* is the set of all relationships retrieved by the method, *correctA* is the set of manually created "strong" relationships with weight 1.0 and *correctB* is the set of manually created "weak" relationships with weight 0.5.

The purpose of equation (3) is to take into consideration the fact that "weak" relationships need not necessarily be the part of a domain model. Table 1 sums up the results of the performed experiments.

Table 1. Experimental results. The F* contains the harmonized mean using R* recall.

Variant	P	R	**F**	R*	**F***
Spreading Activation	0.544	0.443	**0.488**	0.784	**0.566**
PageRank-based Analysis	0.501	0.569	**0.532**	0.741	**0.652**

The results show that performance of PageRank-based approach is better than spreading activation. This result is evidence that PageRank-based analysis enables more precise processing of the underlying graph representation.

The resulted F/F* measure we interpret as "completeness" of generated concept map. However, generated relationships not contained in the manually constructed concept space were all considered incorrect that should not reflect the reality. Though manual relationship creators made their best effort to match real-world relations, relationships retrieved automatically need not to be irrelevant. They might represent bindings, which were not explicitly realized even by the most concerned authors. As the proposed method goal is to serve as assistant to a teacher, the amount and accuracy of recommended relationships can be considered helpful.

7 Conclusions

In this paper we presented a method of automatic concept relationship discovery for an adaptive e-course. The method is applied as a step in the process of an adaptive e-course authoring and its goal is to help teacher and contribute to overall authoring automation. We proposed and evaluated two variants of the concept score similarity computation

representing two approaches to domain model graph processing. We found that our PageRank-based variant achieves better results and identifies more correct relationships against manually constructed concept map.

The main contribution of this paper is a novel approach to adaptive e-course authoring automation. No similar approaches relying on domain model processing and yielding similar results (F*-metrics 65.2%) are applied in the field of adaptive e-learning. The method we propose is independent "component" of a course authoring process. It has clearly defined interface and does not depend on preprocessing techniques. Prior to relationship discovery the concept extraction and weight assignment techniques may vary: various IR methods can be employed or manual annotations can be used. Both approaches may be eventually combined, which seems reasonable especially when considering social and collaborative aspects of e-course authoring. The method addresses the specifics of e-learning domain and does not rely on external resource presence like similar approaches do. Moreover, it is not dependent on the language of an e-course.

As the results of evaluation seem promising, we currently work on more complex evaluation in a real-world environment to obtain even more objective feedback on discovered relationships accuracy and relevancy during functional programming learning.

The further advantage of our method is that although the variants are targeted at the e-learning domain, they are not limited to it – the presented computations are also applicable to different environments. A similar situation with acute "metadata" need is on the Web. Concept maps constructed over the Web pages should in first step serve as backbone for development of richer semantic descriptions. Involvement of social annotations or folksonomies shifts our method applicability even further.

Acknowledgements

This work was partially supported by the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 3/5187/07.

References

[1] Brusilovsky, P. Developing adaptive educational hypermedia systems: From design models to authoring tools. In T. Murray, S. Blessing and S. Ainsworth (eds.): *Authoring Tools for Advanced Technology Learning Environment*. Dordrecht: Kluwer Academic Publishers, pp. 377–409.

[2] Buitelaar, P., Olejnik, D., and Sintek, M. A protégé plug-in for ontology extraction from text based on linguistic analysis. In *Proc. of the 1st European Semantic Web Symposium* (ESWS), 2004.

[3] Ceglowsky, M., Coburn, A., Cuadrado, J. *Semantic Search of Unstructured Data using Contextual Network Graphs*. 2003.

[4] Cimiano, P., Hotho, A., Staab, S. Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. In *JAIR - Journal of AI Research*, vol. 24, pp. 305–339, 2005.

[5] Cimiano, P., et al. Learning Taxonomic Relations from Heterogeneous Evidence. In *Proc. of ECAI Workshop on Ontology Learning and Population*, 2004.

[6] Cristea, A. I., de Mooij, A. Designer Adaptation in Adaptive Hypermedia. In *Proc. of Int. Conf. on Information Technology: Computers and Communications* ITCC'03. Las Vegas, 2003. IEEE Computer Society.

[7] Dicheva D., Dichev C. Helping Courseware Authors to Build Ontologies: the Case of TM4L. In *13th Int. Conf. on Artificial Intelligence in Education*, AI-ED 2007, July 9-13, 2007, LA, California, pp. 77–84.

[8] Diedrich, J., Balke, W-T. The Semantic GrowBag Algorithm: Automatically Deriving Categorization Systems. In *Proc. of the 11th European Conf. on Research and Advanced Technology for Digital Libraries*, ECDL 2007, Budapest, Hungary, 2007, pp 1-13.

[9] Fortuna, B., Grobelnik, M., Mladenic, D. Semi-automatic Construction of Topic Ontology. In *Semantics, Web and Mining, Joint Int. Workshop*, EWMF 2005 and KDO 2005, Porto, Portugal, October 3-7, 2005.

[10] Maedche, A., Staab, S. Ontology Learning for the Semantic Web, In *IEEE Intelligent Systems*, Vol. 16, No. 2, pp. 72–79, 2001.

[11] Šimún, M., Andrejko, A., Bieliková, M. Maintenance of Learner's Characteristics by Spreading a Change. In Kendall, M., Samways, B. (eds.). *IFIP Int. Federation for Information Processing*, Vol. 281, Learning to Live in the Knowledge Society, Boston: Springer, 2008. pp. 223–226.

[12] White, S., Smith, P. Algorithms for estimating relative importance in networks. In *Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data mining*. ACM Press, 2003, pp. 266–275.