Towards Concept Map Based Free Student Answer Assessment

Nabin Maharjan and Vasile Rus
Department of Computer Science/Institute for Intelligent Systems
{nmharjan,vrus}@memphis.edu

Abstract

We propose a concept map based approach to assessing freely generated student responses. The proposed approach is based on a novel automated tuple extraction system, DT-OpenIE, for automatically extracting concept maps from student responses. The DT-OpenIE system is significantly better, for assessment purposes, in terms of concept map quality than state-of-the-art open information extraction (IE) systems such as Ollie or Stanford as evidenced by our experimental results. The concept map based approach can not only generate a holistic score assessing the accuracy of a student response but also enable diagnostic feedback.

Introduction

Assessing student responses has been approached primarily using a semantic textual similarity (STS) approach in which a student response is compared to an ideal, expert-generated response.

In general, STS solutions (Agirre et al. 2015; 2016; Maharjan et al. 2017) do not explain why the two texts are similar, related or unrelated. For example, consider a question asked by DeepTutor (Rus et al. 2013), an Intelligent Tutoring System (ITS) for Newtonian Physics, and the corresponding ideal answer or expectation shown in Table 1. A student response to the question is also shown in the table.

An STS approach would most likely assign a similarity score of 3 for the given student answer meaning that the student response is missing important information. However, it does not explain which information is missing. If such explanatory functionality existed that could explain that the student is missing information about *direction*, an ITS could use this diagnostic information to generate a follow-up question such as: *What other type of information is provided by acceleration?*

One approach to add an explanatory layer in STS systems is to align text chunks, e.g., phrases, in a given pair of texts and label them with semantic relation types and similarity scores as proposed in the pilot interpretable Semantic Textual Similarity task (iSTS; (Agirre et al. 2015)).

Another approach is to use a concept map approach such as the one proposed here to both assess and interpret the student answers. A concept map is a graphical representation of organized knowledge. Concepts are the labeled nodes and relationships between concepts are the directed labeled edges of the graph. It can be a hierarchical map (Novak and Musonda 1991) where the most general concepts are at the top. The map can be also associative where no hierarchy is assumed - the concept map is a semantic network of concepts and their interrelations (Deese 1966). Since the concept maps derived from student free responses in the domain of Newtonian Physics is typically associative, we use associative concept maps in our work.

Question: Because it is a vector, acceleration provides what two types of information?
Student Answer: Acceleration gives magnitude
Expectation: Acceleration provides magnitude and direction.

Table 1: A question and answer between student and DeepTutor with ideal expected answer, i.e., an expectation.

In our concept map approach, we first map ideal answers, i.e., expectations, to, say, Physics problems, into ideal concept maps consisting of one or more tuples. Similarly, student responses to the same problem are mapped into corresponding concept maps. Finally, by comparing the two, we can determine whether the student answer matches one or more of the tuples in the ideal concept map and which tuples are not matched. A tuple is a triplet consisting of a relation/labeled-edge and the corresponding concepts/nodes in a concept map.

Typically, the ideal concept maps are manually created by experts from ideal answers provided by domain experts. For example, the expectation in Table 1 can be represented by a concept map consisting of two tuples: *(acceleration, provides, magnitude)* and *(acceleration, provides, direction)*. On the other hand, the student concept map is automatically extracted by an open IE system from actual student responses. Ideally, a concept map with a single tuple, *(acceleration, gives, magnitude)*, is extracted from the student response in Table 1. The result of the comparison of the two concept maps is that one tuple is missing from the student answer. We thus infer that the student answer is partially correct. Furthermore, we can provide the feedback component

Problem: Two hockey players pass a puck between them on an ice rink. Assume that the ice is smooth so that there is no friction. What forces are acting on the puck while the puck is moving on the ice between the two players? Describe the motion of the puck. **Expectations**: 1. When an object moves with constant velocity, net force on the object is zero. 2. The forces acting on the puck while it is between the players are the force of gravity and the normal force from the ice. 3. Puck moves in a straight line with a constant speed.

Table 2: A task in the DeepTutor system and its expectations.

of an ITS with specific information regarding the missing tuple. The feedback component can in turn generate appropriate diagnostic feedback targeting the missing tuple, e.g, by triggering a hint in the form of a question to elicit the missing tuple.

The rationale for using concept maps for knowledge representation is grounded on a key assumption in most cognitive theories: "*the knowledge within a content domain is well structured and organized around central concepts*". Glaser and Bassok (Glaser and Bassok 1989) defined competence in a domain as "*the well-structured knowledge*". Therefore, as students acquire expertise in a domain, their knowledge becomes increasingly interconnected and resembles the subject-matter expert's representation of the domain (Glaser and Bassok 1989; Royer, Cisero, and Carlo 1993).

Using the concept map approach, we can break down an expectation, i.e., step in an ideal answer, into one or more tuples which essentially means that we end up with finer-grain learning components. That is, we can track students' knowledge at a finer grain level leading to more subtle differences among different knowledge states. It should be noted that a tuple may be considered a *normal* or *learning* tuple depending upon its pedagogical value. For illustration purposes, we will consider the Physics problem in Table 2 and its ideal response as a set of expectations. For example, *Expectation* 1 in Table 2 is represented by two *learning* tuples: *(an object, moves with, constant velocity) and (net force on the object, is, zero)*. *Expectation* 2 is represented by two normal tuples, *(forces, act on, the puck)* and *(puck, is between, the players)*, and one learning tuple, *(forces, are, the force of gravity and the normal force from the ice)*. Similarly, we represent expectation 3 by two *learning* tuples: *(the puck, moves in, a straight line)* and *(the puck, moves with, a constant speed)*.

In this paper, we focus primarily on automating and developing accurate solutions for the automated extraction of concept maps from student generated answers. There are several existing open information extraction (IE) tools that could be used including the state-of-the-art Ollie (Schmitz et al. 2012) and Stanford systems (Stanford-OpenIE; (Angeli, Premkumar, and Manning 2015)). However, these systems mostly focus on solving the Knowledge Base Problem (KBP) and as such tuples produced by these systems are not suited for the task of student answer assessment. We will discuss in detail later the issues with the open IE tools.

Related Work

Concept maps were first proposed by Novak (Novak and Musonda 1991) to represent children's knowledge of science when they faced difficulty. The goal was to identify learning specific changes in children by examining interview scripts. The concept maps were developed based on the learning psychology of Asubel(Ausubel 1963) whose fundamental idea was that people learn new concepts and propositions by asking questions and getting clarification about relationships between old concepts and new concepts and between old propositions and new propositions.

Concept maps have been used for many purposes such as checking student's knowledge on a topic (CMap Tools; (Cañas et al. 2004)) and collaborative learning on a topic/domain (Martinez Maldonado et al. 2012). Also, they have been used as instructional tools for meaningful learning (All, Huycke, and Fisher 2003; Wallace and Mintzes 1990; Novak, Bob Gowin, and Johansen 1983; Schmid and Telaro 1990). Some ITSs use concept maps as instructional tools to facilitate learning (Olney et al. 2012).

Concept maps have been used as an assessment tool as well. The assessment methods might vary in how they elicit information from the learners. For example, students might be asked to fill in a skeleton map (Anderson and Huang 1989), to construct a concept map (Roth and Roychoudhury 1993; Wu et al. 2012), or to write an essay (Lomask et al. 1992). Recently, assessment methods based on concept maps have been developed that provide prompt feedback to students. For example, Wu (Wu et al. 2012) evaluates student concept maps on-the-fly and provides real-time feedback by comparing the concept maps with the expert/teacher's concept map. Also, the COMPASS (Gouli et al. 2004) system provides individualized feedback based on diagnostic assessment of the learner's concept map against an ideal concept map.

Our method is more similar to the concept map based assessment approach of Lomask and colleagues (Lomask et al. 1992), with some differences. In their work, students wrote essays on two central topics in biology and then trained teachers derived concept maps from the essays. No hierarchical structure was assumed. Similarly, in our approach, we do not assume any hierarchical structure and the concepts maps are derived from student-generated responses during tutorial interactions for problem-solving with the state-of-the-art ITS DeepTutor. The target domain is conceptual Newtonian Physics. In our case, the concept map extraction is automated. Once the student concept maps are extracted, we compare them to the corresponding ideal concept maps to assess correctness.

As already noted, we use information extraction techniques to automatically extract concept maps. TextRunner (Yates et al. 2007) and ReVerb (Fader, Soderland, and Etzioni 2011) uses syntactic POS tag patterns for extracting entity-relation structures, i.e., tuples in the form of (conceptA, relation, conceptB). The CREATE system (Bhattarai and Rus 2013)) generates open-relation tuples by combining the ReVerb system approach and iterative pattern and tu-

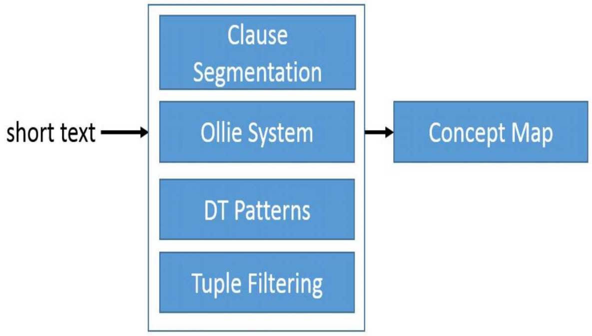

Figure 1: A DeepTutor Open Information Extraction (DT-OpenIE).

ple based extraction. Similarly, the OLLIE system (Schmitz et al. 2012) exploits learned dependency patterns to extract the tuples. The Stanford system(Stanford-OpenIE; (Angeli, Premkumar, and Manning 2015)) first generates shorter entailed clauses from given texts using a clause splitter model and a natural logic inference system and then applies a small set of patterns to extract the tuples. These systems are geared towards building knowledge bases through open tuple extraction with a focus on extracting factual tuples from professionally written texts. As such, these systems do not produce desirable tuples for student assessment tasks from. To address this drawback, we propose a novel open tuple extraction method, DT-OpenIE, which is more suited for the assessment task.

DeepTutor Open Information Extraction

As mentioned earlier, state-of-the-art IE systems such as Ollie (Schmitz et al. 2012) and Stanford-OpenIE (Angeli, Premkumar, and Manning 2015) are more suited for building knowledge bases by extracting factual tuples from professionally written texts such as newspaper articles.

The Stanford IE tool relies on its clause-splitter model and natural logic inference system to generate maximally shorter clauses which are simple enough for its few patterns to extract tuples. It means that shorter clauses not entailed from the original text are not generated which could lead to tuples not being retrieved. For example, given the text: *"T_1: If the acceleration of a system is zero, the net force is zero."*, no tuples are extracted. However, given the text: "*T_2: The acceleration of a system is zero. The net force is zero.*", many tuples are extracted including the desirable tuples *(acceleration of system, is, zero)* and *(net force, is, zero)*.

Another issue with the Stanford-OpenIE tool is that its natural logic inference system tends to over-produce tuples from texts. This is helpful for solving the KBP problem where the relations from the extracted tuples are mapped onto standard KBP relations based on co-occurrence statistics, which typically requires a large amount of tuples for better estimates. However, all maximally entailed shorter clauses might not be valid for tasks such as the student answer evaluation where the focus is on whether the student has mastered specific domain concepts or not. For example, the Stanford-OpenIE tool also generates (force,is,zero) from the text T_2 above, which is invalid because it is "*net force*" that is zero and not any individual force. Similarly, for the text "*the frictional force cancels normal force*", the desirable tuple output is *(frictional force, cancels, normal force)*; however, the Stanford-OpenIE tool also generates *(frictional force, cancels, force)*, *(force,cancels,normal force)* and *(force,cancels,force)* which are all misleading for assessment. Just to illustrate the proliferation of tuples, for Newton's first law: "*An object at rest will stay at rest and an object moving with constant velocity in a straight line will continue moving with constant velocity in a straight line as long as the net force acting on the object is zero*", the Stanford-OpenIE tool produces 38 different tuples.

The Ollie system does not suffer from the problem of over-generating the tuples. Its patterns are effective at extracting tuples that mostly cover the concepts in the given short text. However, the system might retrieve false tuples sometimes. For example, the tool retrieves the incorrect tuple *(the desk,increase its speed as, the net force)* and misses *(net force, is anymore, not zero)* when processing the following text: "*The desk increases its speed as the net force is not zero anymore*". Also, it successfully extracts the tuple *(the mover's push, equal, the oppose force of friction)* from this text "*The mover's push equals the opposing force of friction*" but fails to extract any tuple from the simpler text "*Mover's push equals friction*".

The newly proposed extraction method called DT-OpenIE, shown in Figure 1 builds the strengths of these open IE systems and avoids their weaknesses with respect to our target task. It consists of i) a Clause Segmentation Model, ii) the Ollie System iii) DT patterns and iv) Tuple Filtering. The tuple filtering removes any duplicate tuples and produces a final concept map from the given text. Next, we describe our clause segmentation model.

System	P	R	F
CM03	87.99	81.01	84.36
CMPR02	90.18	78.11	83.71
CM01	84.82	78.85	81.73
MP01 (Molina and Pla 2001)	70.85	70.51	70.68
DT-CS	81.21	74.25	77.57

Table 3: Results of different systems on CoNLL-2001 shared task test data. CM03 (Carreras and Marquez 2004), CMPR02 (Carreras et al. 2002), CM01 (Carreras and Màrquez 2001). P = Precision, R = recall, F= F-measure.

1. more force is being applied
2. since more force is being applied
3. the speed of the desk will increase
4. the speed of the desk will increase since more force is being applied

Table 4: An optimal clause split generated from text: *the speed of the desk will increase since more force is being applied.*

Extraction	Input Pattern
(velocity, increase,NONE)	NP + VP e.g. *velocity increases*.
(IMPERSONAL,impress,you)	To-clause e.g. He has ability *to impress you*.
(IMPERSONAL,is,no force)	NP_1 + VP + NP_2, $NP_1 \in$ EX tag e.g. *There is no force*.
(1st Law,says,COMPLEX)	Attribution relation e.g. *1st Law says* that the object moves with a constant velocity
(Push,equals,friction)	NP_1 + VP + NP_2, $NP_1 \notin$ EX tag e.g. *Push equals friction*

Table 5: A list of DT patterns for tuple extraction.

Clause Segmentation Model

A clause is a text segment containing a subject and a predicate. It constitutes a meaningful unit, which ideally is a proposition. Similar to Stanford-OpenIE, we extract shorter clauses from a given text and consider them candidates for tuple extraction. However, we do not use the entailment restriction or the natural logic inference system while extracting shorter clauses because of the issues discussed above.

We developed our model using the CoNLL-2001 shared task data for clause identification (Sang and Déjean 2001). We could not replicate Adaboost classifier used by Carreras (Carreras and Màrquez 2001) due to memory limitation. Instead, we used a liblinear classifier following their approach to build our model. First, we developed models to detect clause start and end boundaries and then a clause identification model that classifies whether a given clause candidate is a clause or not baed on a confidence score. We extract clause candidates $C_{(i,j)}$ such that $j > i, word_i \in S, word_j \in E$ from the text, where the $word_i \in S$ indicates the word at position i is tagged with a clause start label S while the $word_j \in E$ means that the word at j is tagged with a clause end label E.

We evaluated the clause candidates based on confidence scores to produce a clause split from the text. A clause split is a list of consistent clauses in which clauses are either nested or not overlapping. We produced a clause split from texts using both a greedy approach (Carreras and Màrquez 2001) and an optimal approach (Carreras et al. 2002).

We used all but *Sentence Pattern* features from Carreras' (Carreras and Màrquez 2001) as they were found to be not discriminating enough for our liblinear model. Besides these features, we used some additional context features in our models. We also used some post processing rules to correct the label predicted by clause start and end classifiers. We don't discuss them here because of space reason. Table 3 provides the performance of our clause segmentation model (DT-CS) using the optimal approach on the CoNLL-2001 shared task test data. The Adaboost algorithm with weak decision trees (CM01 and CMPR02) seemed more


```
<Tuples class="java.util.ArrayList">
  <Tuple requiresContextForCreation="0|1" weight="0.5" synsetId="6">
    <Text>(velocity,be,constant)</Text>
  </Tuple>
  <Tuple requiresContextForCreation="0|1" weight="0.5" synsetId="6">
    <Text>(acceleration,be,zero)</Text>
  </Tuple>
  <Tuple requiresContextForCreation="0|1" weight="0.5" synsetId="7">
    <Text>(sum of force,equal,zero)</Text>
  </Tuple>
</Tuples>
```


a.


```
<PredictedTuples>
  <Tuples class="java.util.ArrayList">
    <Tuple weight="0.0">
      <Text>(the sum of the force,therefore will equal,zero)</Text>
    </Tuple>
    <Tuple weight="0.0">
      <Text>(the sum of the force,will equal,zero)</Text>
    </Tuple>
    <Tuple weight="0.0">
      <Text>(velocity,be,constant)</Text>
    </Tuple>
    <Tuple weight="0.0">
      <Text>(the acceleration,be,zero)</Text>
    </Tuple>
  </Tuples>
</PredictedTuples>
```


b.

Figure 2: A comparison of a DT-OpenIE generated concept map (b) and an ideal concept map (a) for the ideal answer: "*When velocity is constant, the acceleration is zero; therefore the sum of the forces will equal zero*".

completely(1)	concept map is completely correct.
mostly(2)	at least half of concept map is correct.
slightly(3)	at least one extracted tuple is correct.
inaccurate(4)	none of the extractions are correct.

Table 6: An ordinal scale with four values for rating an extracted concept map of an ideal student answer along the metric *accuracy*.

predictive than our liblinear model. Our system results are comparable to the top performing systems (CM01, CMPR02 and CM03). More importantly, our system extracts clauses which are reasonably suited for the student answer assessment task. Table 4 shows the clause split output for an example text using the optimal approach.

DT Patterns

We passed the output of the clause segmentation model through the Ollie system to generate the tuples. However, as discussed above, there are certain sentence structures which are not captured by the Ollie system that might have pedagogical value. Therefore, we applied a set of patterns to extract tuples from such sentence forms as listed in the Table 5. We used the special keywords IMPERSONAL and NONE to indicate the absence of first and second arguments, respectively. We used the COMPLEX keyword to denote entities which are clauses.

Experiment and Results

Data

In order to evaluate the proposed approach, we used student answer data from logged interactions of 41 high school stu-

Measure	Stanford	Ollie	DT-OpenIE
Accuracy	2.71 (1.24)	2.19 (1.35)	1.89 (1.10)
Coverage	2.63 (1.28)	2.38 (1.27)	1.69 (1.12)
Pedagogy	2.52 (1.40)	2.41 (1.44)	1.70 (1.24)

Table 7: Mean ratings for concept maps of ideal student answers generated by different open information extraction methods. The standard deviations are provided in bracket alongside means.

dents with the DeepTutor ITS. During the summer of 2014, high-school students participated in an experiment on which they were given 9 different Physics problems to solve. The experiment produced 370 tutorial interactions in total (one student performed a task twice).

Creation of ideal concept maps: Two subject-matter-experts (SMEs) manually created ideal concept maps for all 9 tasks used in the experiment. The SMEs were provided with a reference guide for creating the concept map. Each task was provided as an XML document which allowed the annotators to annotate tuples with relevant attributes and also make comments, as necessary. For example, Figure 2 a) shows a human generated concept map for one of the steps in the ideal solution of a problem. It should be noted that the tuples covering identical concepts are assigned identical *synsetId* values. Similarly, the tuples are also weighted for their pedagogical value. After creating three concept maps, the SMEs met and revised their maps to resolve any discrepancies. A refined annotation guide was created which was then followed for the whole data.

Quality Evaluation

To assess the quality of the extracted tuples, we automatically extracted concept maps for 133 ideal student answers from the nine tasks using Stanford OpenIE, Ollie and DT-OpenIE extraction method. Then, we asked the two SMEs to rate the generated concept maps against the ideal/gold standard concept maps. Figure 2 shows a comparison of an automatially generated concept map for an ideal student answer against its gold standard concept map.

The annotators rated the automatically generated concept maps along three dimensions, i) *accuracy*, ii) *coverage* and iii) *pedagogy* following the approach adopted by Olney and colleagues (Olney, Cade, and Williams 2011). In other words, the annotators rated the degree of correctness, completeness, and pedagogical value of the extracted tuples in the concept maps while comparing against the gold standard concept maps. We provided the annotators with an annotation guideline for the annotation. The annotators met after annotating two tasks for revising annotations to resolve discrepancies if any. The guidelines were updated accordingly and used for annotating the whole data.

We used an ordinal scale of 4 values for the ratings. The Table 6 describes our ordinal scale for the *accuracy* measure where 1 denotes the highest level of accuracy.We used Cronbach's α to measure inter-rater reliability because of the ordinal ratings. The Cronbach's α were 0.991, 0.993 and 0.997 for accuracy, coverage, and pedagogy, respectively, which indicated a highly significant inter-annotator agreement.

Results and Analyses

Table 7 shows the mean and standard deviations of the ratings for each of the quality measures. The mean ratings for the concept maps generated by Stanford-OpenIE were 2.71, 2.63 and 2.52 for accuracy, coverage, and pedagogy, respectively. The Ollie-generated concept maps were slightly better with relatively lower mean ratings of 2.19, 2.38 and 2.41 for accuracy, coverage, and pedagogy, respectively. Our DT-OpenIE concept maps were the best in terms of their mean quality ratings with the scores of 1.89, 1.69 and 1.70, respectively. We performed paired t-test significance analysis between the different extraction methods for each of the quality measures. We found that our DT-OpenIE ratings were significantly better. Compared against the Stanford-OpenIE, the significance were all $p < 0.001$ for accuracy, coverage and pedagogy, respectively. Similarly, the significance were $p = 0.004$, $p < 0.001$ and $p < 0.001$ when comparing mean accuracy, coverage, and pedagogical values of DT-OpenIE extractions against Ollies'.

We also found that accuracy, coverage, and pedagogy are significantly corelated with each other (0.85 for accuracy vs coverage, 0.8 for accuracy vs pedagogy, and 0.92 for coverage vs pedagogy).

The results are promising - the concept maps generated by our method, on an average, fall between complete and mostly accurate for all of the three quality scales. One case where the system fails to generate tuples is a list-type student response. For example, a list-like representation for the Expectation 2 of the problem in Table 2 might be "*Force of gravity and normal force*". A simple resolution might be extracting such text as *(force of gravity and normal force, NONE, NONE)*. In another approach, the tool might extract the tuple *(force of gravity and normal force, **ACT ON**, **PUCK**)* by inferring the missing relation and second argument (capitalized) from the dialogue context, which in this case, is a question by the DeepTutor: "*What forces are acting on the puck?*"

Conclusion

We presented a novel automated concept map extraction method and system, called DT-OpenIE. The experiments indicate that the generated tuples are significantly better in quality than those extracted by the state-of-the-arts open information extraction tools such as Stanford-OpenIE and Ollie systems. Our future work will focus on better tracking student's knowledge states by using the proposed concept map approach. We also plan to exploit concept maps for dynamically providing diagnostic feedback in an automated tutoring environment and study its impact on tutoring effectiveness, i.e., on the ability of the tutoring system to induce learning gains for the learners.

Acknowledgments. This work was partially supported by The University of Memphis, the National Science Foundation (awards CISE-IIS-1822816 and CISE-ACI-1443068), and a contract from the Advanced Distributed Learning Initiative of the United States Department of Defense.

References

Agirre, E.; Banea, C.; Cardie, C.; Cer, D. M.; Diab, M. T.; Gonzalez-Agirre, A.; Guo, W.; Lopez-Gazpio, I.; Maritxalar, M.; Mihalcea, R.; et al. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *SemEval@ NAACL-HLT*, 252–263.

Agirre, E.; Banea, C.; Cer, D. M.; Diab, M. T.; Gonzalez-Agirre, A.; Mihalcea, R.; Rigau, G.; and Wiebe, J. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval@ NAACL-HLT*, 497–511.

All, A. C.; Huycke, L. I.; and Fisher, M. J. 2003. Instructional tools for nursing education: Concept maps. *Nursing Education Perspectives* 24(6):311–317.

Anderson, T. H., and Huang, S.-c. C. 1989. On using concept maps to assess the comprehension effects of reading expository text. *Center for the Study of Reading Technical Report; no. 483*.

Angeli, G.; Premkumar, M. J. J.; and Manning, C. D. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, 344–354.

Ausubel, D. P. 1963. The psychology of meaningful verbal learning.

Bhattarai, A., and Rus, V. 2013. Towards a structured representation of generic concepts and relations in large text corpora. In *RANLP*, 65–73.

Cañas, A. J.; Hill, G.; Carff, R.; Suri, N.; Lott, J.; Gómez, G.; Eskridge, T. C.; Arroyo, M.; and Carvajal, R. 2004. Cmaptools: A knowledge modeling and sharing environment.

Carreras, X., and Màrquez, L. 2001. Boosting trees for clause splitting. In *Proceedings of the 2001 workshop on Computational Natural Language Learning-Volume 7*, 26. Association for Computational Linguistics.

Carreras, X., and Marquez, L. 2004. Phrase recognition by filtering and ranking with perceptrons. *Recent advances in natural language processing III: selected papers from RANLP 2003* 260:205.

Carreras, X.; Màrquez, L.; Punyakanok, V.; and Roth, D. 2002. Learning and inference for clause identification. In *European Conference on Machine Learning*, 35–47. Springer.

Deese, J. 1966. *The structure of associations in language and thought*. Johns Hopkins University Press.

Fader, A.; Soderland, S.; and Etzioni, O. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 1535–1545. Association for Computational Linguistics.

Glaser, R., and Bassok, M. 1989. Learning theory and the study of instruction. *Annual review of psychology* 40(1):631–666.

Gouli, E.; Gogoulou, A.; Papanikolaou, K.; and Grigoriadou, M. 2004. Compass: an adaptive web-based concept map assessment tool.

Lomask, M.; Baron, J.; Greig, J.; and Harrison, C. 1992. Connmap: Connecticuts use of concept mapping to assess the structure of students knowledge of science. In *annual meeting of the National Association of Research in Science Teaching, Cambridge, MA*, 21–25.

Maharjan, N.; Banjade, R.; Gautam, D.; Tamang, L. J.; and Rus, V. 2017. Dt_team at semeval-2017 task 1: Semantic similarity using alignments, sentence-level embeddings and gaussian mixture model output. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 120–124.

Martinez Maldonado, R.; Kay, J.; Yacef, K.; and Schwendimann, B. 2012. An interactive teachers dashboard for monitoring groups in a multi-tabletop learning environment. In *Intelligent Tutoring Systems*, 482–492. Springer.

Molina, A., and Pla, F. 2001. Clause detection using hmm. In *Proceedings of the 2001 workshop on Computational Natural Language Learning-Volume 7*, 25. Association for Computational Linguistics.

Novak, J. D., and Musonda, D. 1991. A twelve-year longitudinal study of science concept learning. *American Educational Research Journal* 28(1):117–153.

Novak, J. D.; Bob Gowin, D.; and Johansen, G. T. 1983. The use of concept mapping and knowledge vee mapping with junior high school science students. *Science education* 67(5):625–645.

Olney, A.; D'Mello, S. K.; Person, N. K.; Cade, W. L.; Hays, P.; Williams, C.; Lehman, B.; and Graesser, A. C. 2012. Guru: A computer tutor that models expert human tutors. In *ITS*, 256–261. Springer.

Olney, A. M.; Cade, W. L.; and Williams, C. 2011. Generating concept map exercises from textbooks. In *Proceedings of the 6th workshop on innovative use of NLP for building educational applications*, 111–119. Association for Computational Linguistics.

Roth, W.-M., and Roychoudhury, A. 1993. The concept map as a tool for the collaborative construction of knowledge: A microanalysis of high school physics students. *Journal of research in science teaching* 30(5):503–534.

Royer, J. M.; Cisero, C. A.; and Carlo, M. S. 1993. Techniques and procedures for assessing cognitive skills. *Review of Educational Research* 63(2):201–243.

Rus, V.; DMello, S.; Hu, X.; and Graesser, A. 2013. Recent advances in conversational intelligent tutoring systems. *AI magazine* 34(3):42–54.

Sang, E. F., and Déjean, H. 2001. Introduction to the conll-2001 shared task: Clause identification. *arXiv preprint cs/0107016*.

Schmid, R. F., and Telaro, G. 1990. Concept mapping as an instructional strategy for high school biology. *The Journal of Educational Research* 84(2):78–85.

Schmitz, M.; Bart, R.; Soderland, S.; Etzioni, O.; et al. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 523–534. Association for Computational Linguistics.

Wallace, J. D., and Mintzes, J. J. 1990. The concept map as a research tool: Exploring conceptual change in biology. *Journal of research in science teaching* 27(10):1033–1052.

Wu, P. H.; Hwang, G.-J.; Milrad, M.; Ke, H.-R.; and Huang, Y.-M. 2012. An innovative concept map approach for improving students' learning performance with an instant feedback mechanism. *British Journal of Educational Technology* 43(2):217–232.

Yates, A.; Cafarella, M.; Banko, M.; Etzioni, O.; Broadhead, M.; and Soderland, S. 2007. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 25–26. Association for Computational Linguistics.