2016 IEEE 16th International Conference on Advanced Learning Technologies

A Software Tool to Measure the Alignment of Assessment Instrument with a Set of Learning Objectives of a Course

Rekha Ramesh
Department of educational Technology,
Indian Institute of Technology Bombay,
Mumbai, India
e-mail: rekha.ramesh@iitb.ac.in

Sasikumar M.
CDAC Bombay,
Mumbai, India
e-mail: the.little.sasi@gmail.com

Sridhar Iyer
Department of CSE,
Indian Institute of Technology Bombay,
Mumbai, India
e-mail: sri@iitb.ac.in

***Abstract*— In this paper, we present a software tool to measure alignment of Assessment instrument (AI) with a set of learning objectives (LOs) of a course. Alignment of syllabus, LO and AI is a major parameter determining the quality of an AI. The tool helps to considerably reduce the time and effort needed by teachers to ensure this alignment. It takes syllabus, a set of LOs and domain ontology as input. An ontology based knowledge representation mechanism is designed to integrate the contents of syllabus, LOs and AI. Alignment is measured in terms of both concepts covered and cognitive level used. The Data Structures course of second year engineering curriculum is chosen as the domain. The accuracy of this tool is tested by comparing the system generated alignment measure with the expert teachers and a confusion matrix is generated. We got an average accuracy of 90% for concepts alignment and 95% agreement in cognitive level alignment.**

Keywords- Quality of Assessment Instrument; Alignment of AI with learning objectives; Data structures; Domain Ontology;

I. INTRODUCTION

Alignment with the LOs of a course is a primary concern while designing any AI for summative or formative evaluation in a course [1]. LOs are clear statements that describe the competences that students should possess upon completion of a course. One of the key objectives in design of an AI is ensuring that the instrument actually measures these competencies. LOs are usually designed to span the entire syllabus of the course. So if the instrument is aligned with the LO fairly, it is assumed that it covers the syllabus fairly [2][3].

Today, teachers have to spend a lot of time and effort for ensuring the alignment manually. We did a study looking at past 5 years AIs of various courses of Mumbai University CS curriculum and found that average alignment is poor. Most of them contained biased distribution for cognitive level and questions and were catering only to lower order thinking skills (Recall, Understand and Apply). Similarly, the content coverage was also not fair. This mismatch of learning objective and assessment objective leads to non-alignment of the intended outcome of assessment [3]. This can be improved by having an automated mechanism that measures this alignment and provide constructive feedback to teachers before the instrument is given to students for solving. The IQuE, discussed in this paper is such a system. Even though various researchers have stressed the importance and benefits of aligning the AI to course LOs, to the best of our knowledge, none of them has reported attempts to formulate and automate this task. This motivated us to build such a system.

Alignment of LOs can be measured in terms of cognitive level alignment and content/concept alignment [3]. It is not easy to extract cognitive level reliably from these as teachers can frame the LOs and questions in different ways. For example, keywords can be misleading unless the system has contextual and domain specific knowledge. Similarly, concepts covered need to be identified. It is also not a straight forward process as the concepts are not explicitly present in the LO/Assessment statement.

The concepts and the cognitive level information are captured from LOs and mapped onto the nodes of the domain ontology which is then called as LO annotated ontology (LAO). Further, similar information from set of questions in an AI is mapped to LAO to get Instrument annotated ontology (IAO). The differences in these 2 dimensions from LO and AI reflect the unfairness in the LO coverage of AI. The IQuE provides a visual and numeric representation of this alignment.

IQuE includes the process of extracting concepts and cognitive level from an LO using NLP techniques and the complex process of mapping these to the nodes of the ontology [6][13]. Color coding is introduced to reflect the result of the mapping. The accuracy of this framework was tested by comparing the system generated results to the manually generated results by the experienced teachers.

Section 2 gives the description of the overall system. Section 3 explains the construction of domain ontology. Sections 4 and 5 provide details of design of LAO and IAO. Result of testing is discussed in section 6.

II. THE IQUE SYSTEM

IQuE takes a syllabus, a set of LOs and domain ontology as input and when given an AI to it, calculates its “Goodness measure”. The overall process is shown in figure 1. The

2161-377X/16 $31.00 © 2016 IEEE
DOI 10.1109/ICALT.2016.10

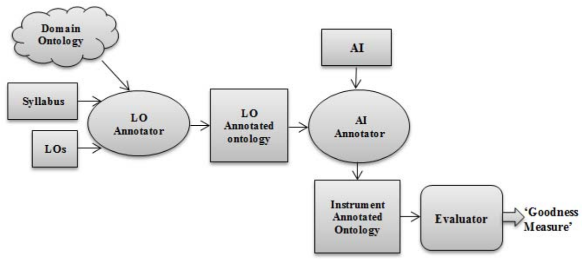

Figure 1. The IQuE System

domain ontology contains all the concepts related to a particular domain and relationship among them. The structure of domain ontology is discussed in detail in section 3. Syllabus is considered as set of keywords which have a corresponding match to nodes of ontology [4]. These keywords are extracted by LO annotator and mapped to the corrsponding nodes of the ontology. This defines the subset of the domain covered in the course. It then maps the content and cognitive level information from LOs into this ontology. This is called LAO. The AI annotator extracts the content and cognitive level information contained in questions in AI and maps it into LAO to form instrument annotated ontology (IAO).

In initial ontology all the nodes are colored as white. When the syllabus is mapped to it, the matching nodes will be colored as black and when LOs are loaded, the matching nodes will be partially colored as red. When questions are loaded the matching nodes will be partially colored as blue. Different cognitive levels will be indicated by varying shades of red or blue. The shade/intensity of the color is dependent on the cognitive level of LO or question involving those concepts.

The evaluator uses this information to perform many tasks such as: (i) find and print the statistics on the number of concepts within/outside the syllabus with items addressing them, the number of concepts within/outside the syllabus with no items addressing them, concepts that are within the scope/ outside the scope of some LO, etc. (ii) Find the syllabus fairness of AI and (iii) measure the alignment of AI with a set of LOs of a course.

III. CONSTRUCTION OF DOMAIN ONTOLOGY

An ontology can capture hierarchical structure among various concepts in a domain and also dependencies and relationships among them in a machine parsable way [10]. The nodes in the ontology represent the concepts from the domain and the links determine the relationship among them [6]. For example, for the domain of Data Structures, it will contain concepts relating to data structures including various known data structures, their representation and applications and operations on them [7][8][9]. Such an ontology structure is used as the base representation mechanism in IQuE [10]. The concepts in the domain are finalized by compiling the contents of various standard textbooks and also the data structures course contents of many different Universities. Fig. 7 shows the domain ontology. Every node in the ontology represents a concept/topic from the domain. All the major topics form the level 1 nodes in the ontology. The major topics can be further narrowed down to subtopics that form the subclasses in the ontology. The relations formed the links in ontology. The links are used to traverse the ontology to locate the neighborhood nodes which are relevant in the ontology. The type of links decides what nodes are to be included for mapping. Some are general links like

- 'hasSubClass'--- indicates one concept is a subclass of another concepts

Some links are domain specific. For data structures, we are assuming following domain specific links.

- "hasRepresentation"--- indicate that every data structure has some type of representation .
- "hasOperation"--- indicate that every data structure has some operation defined on them such as insert, delete, search, etc.
- "hasApplication"--- indicate that every data structure has some applications in real world
- "isA"--- One concept is a kind of another concept. E.g. Every Data Structure *isA* ADT.
- "includes" --- One concept has many other parts included in it. In other words, to understand and implement one concept you need to understand other parts. So, if an explicit concept has '*includes*" link then the connected nodes are considered as implicit links.
- The links have inverses. In the statement "Heap sort uses binary tree" The ***uses*** is inverse of ***hasApplication***.

These set of links may have to be revised when the domain changes.

IV. DESIGN OF IQUE

The design of IQuE involves the design of three major processing components: LO annotator, AI annotator and Evaluator.

A. LO Annotator

The LO annotator takes syllabus and LO as input and generates LAO. The keywords from syllabus are mapped to nodes of the ontology. The key challenge here is how to automatically extract relevant information (concepts and cognitive level) from the LO text and map to the nodes of the ontology. Every LO contains 2 attributes. A set of topics/concepts (c_1, c_2, , … c_l) from the syllabus addressed by that LO and the cognitive level defined by Bloom's Taxonomy [2][3]. Extracting these requires some amount of NLP techniques. Initially, the LO statements are preprocessed using simple NLP techniques such as tokenization and Lemmatization [13]. The words or tokens are matched to the nodes/concepts of domain ontology. But the matching process is not direct. Following subsections explains the process of parsing the LOs to identify relevant concepts and cognitive level, mapping to nodes of ontology and generating the LAO.

1) Extracting concepts from LOs:

There are direct and indirect concepts in an LO. Direct concepts are explicit wordings in LO text but they may be present in different forms. Indirect concepts are to be identified by traversal algorithm.

Step1. Identifying explicit concepts: Explicit concepts can be multi-worded or differently worded. Multi-worded concepts are identified using N-grams algorithm. Differently worded concepts are identified by annotating node in an ontology with a set of synonyms. Synonyms form possible alternative names that the examiner may use in place of the node names in the ontology. For example, in

LO1: Students should be able to demonstrate and implement different methods of traversal for binary trees.

Binary tree is a multi-worded and can be found using 2-grams. The concept ***methods of traversing*** is a synonym to the node ***traversal operation*** in the ontology. We assume that such synonyms are available in the ontology; and new ones may be added as needed for improving performance.

Step2. Identifying implicit or hidden concepts: Consider *LO2: Students should be able to implement various sorting algorithms.*

Here the parser locates the ***sorting algorithms*** as an explicit concept that is at a higher level in the ontology which encompasses all the sorting techniques mentioned in the syllabus and which forms the nodes in the sub tree below it. In an AI one may find references to specific algorithms like selection sort and we need to consider that as matching the above LO. For this to happen, we need to color these nodes when processing this LO. Such nodes can be reached by traversing the ***hasSubclass*** relation from the explicitly found nodes. After analyzing many such LOs, we found that there are some words typically associated with these LOs such as **various, different, any, all, plural form** of a concept etc., indicating that the associated concepts act like slot variables. Then annotator can find all the valid concepts that can be substituted for these slot variables. In this case, 'various' is a slot indicator and 'sorting algorithm' is a slot variable.

Sometimes, the implicit concepts are connected to explicit concepts by domain specific links. These link names are extracted from text LOs by matching the token/ words in LO with the link names and its synonyms stored in dictionary. For example,

LO3: Students should be able to explain operations on stack.

In this case, ***stack*** and ***operation*** are the only concepts that are explicitly identified from LO. Here implicit concepts **push and pop operations** are identified using ***hasOperation*** link. If the concept from LO do not have the identified link connected to it, then each of the super classes can be traversed to see whether they have the link. If they have, then all the nodes connected to that link are considered for mapping by traversing that link.

Sometimes explicitly identified concepts form isolated nodes when mapped to domain ontology. If they are connected by only one intermediate node, then our system considers coloring that also.

2) Extracting cognitive level from LOs

Revised Bloom's taxonomy forms the basis for cognitive level identification of an LO. Every level of Bloom's taxonomy namely, Recall, Understand, Apply, Analyze, Evaluate and Create is associated with an elaborate set of keywords that are stored in a dictionary. Some domain specific action verbs are also added in the dictionary. In case of data structures, these includes: 'write a program', 'provide a stepwise execution', etc. The tokens are matched to the keywords in the dictionary and accordingly its cognitive level is identified. If two tokens match with the keywords of two different Bloom's level, then the higher level one is chosen as cognitive level of the complete LO.

For example, in LO1, given the keyword '***Demonstrate***' is at Understand level and '***Implement***' is at Apply level. So the cognitive level of the LO1 is identified as Apply. Once the cognitive level of LO is identified, all the concepts involved with this LO are annotated/color coded with this cognitive level.

3) Generating LAO

All the relevant concepts and the cognitive level identified by LO annotator from each of the LOs are mapped to the nodes of the domain ontology and color coded to generate the LAO. The red colored nodes in Fig 7 indicate the concepts covered by LOs.

B. AI Annotator

The steps followed by AI annotator for finding the concepts and cognitive level from a question are same as that of LO Annotator in LOs. The questions that can be generated by teachers have very high variability and they can be very creative. So extracting concepts from questions in AI can be much more challenging than in LOs.

The concepts and the cognitive levels extracted from questions in AI are mapped to the nodes of the LAO and color coded to generate IAO. The blue colored nodes shown in Fig. 7 indicate concepts covered by questions in the AI selected.

C. Evaluator

Based on information associated with each concept in IAO, the evaluator computes the following statistics: number of concepts (i) within or out of syllabus (ii) within or outside syllabus and LO coverage of them (iii) within or outside syllabus and question coverage on them (iv) covered by LOs at each cognitive level (v) covered by questions at each cognitive level. From these statistics the evaluator can evaluate the quality of AI. The difference in concepts covered and the mismatch of cognitive levels of LOs and questions amounts to misalignment among them. These can be aggregated to measure the overall alignment between a set

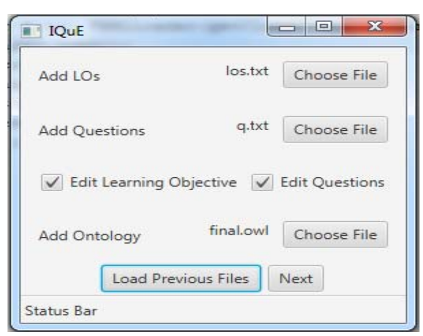

Figure 2. Interface for file upload

of LOs and AI of a course.

V. IMPLEMENTATION

The system is implemented using Java programming language. The ontology is created using protégé application. The Protégé OWL file is parsed by the OwlParser class of Java [6]. IQuE takes three files containing: LOs, questions, and an OWL file for domain ontology as input through the interface for file upload as shown in Fig. 2.

All the components of IQuE discussed in sections 4 are implemented using the Ontology Mapper Algorithm.

```
Algorithm Ontology Mapper
    stmt = read (LQ)
    LQ.stmt = preprocess (stmt)
    tagged_ stmt = POS_tagger (LQ.stmt)
    c_level_LO = find_cog-level (tagged_ stmt, action_verbs)
    LQ.relation = find_relation ( LQ.stmt)
    onto_concepts = extract_concepts (OWLfile)
    LQ.explicit = find_explicit_concepts (onto_concepts, LQ_stmt)
    LQ.implicit = onto_traversal (LQ.explicit, onto_concepts, LQ.slot)
    c_level_concepts = assign_cog_level (LQ.implicit, c_level_LO)
    color _code = assign_color (c_level_concepts)
   color_onto_nodes ( LQ.implicit, color_code, OWLfile)
```

Figure 3. Ontology Mapper Algorithm

The ontology traversal algorithm (highlighted) is used to find implicit concepts to be mapped to nodes of the ontology.

```
Algorithm Onto_Traversal
    LQ.slot = find_slot (LQ.stmt)
    for each slot_varible
          traverse hasSubClass link and include all connected nodes into LQ.implicit
    for each relation in LQ.relation
          traverse the link and include all connected nodes into LQ.implicit
   if the LQ.implicit do not have the identified link connected to it, then
   {  traverse each of the super classes to see whether they have the link.
      If they have the link then
      { traverse that link to include all the nodes connected into LQ.implicit
        remove the redundant concepts from LQ.implicit
       }
    }
    If concepts in LQ.implicit are isolated then
    { find the path between them
       if path length = 1 then
       { include intermediate nodes into LQ.implicit }
    }
```

Figure 4. Ontology Traversal Algorithm

The ontology mapper algorithm is implemented and run on the uploaded files. Fig. 5 shows the output instance for an LO and its corresponding cognitive level, slot, slot variable, relation and all explicit and implicit concepts extracted by

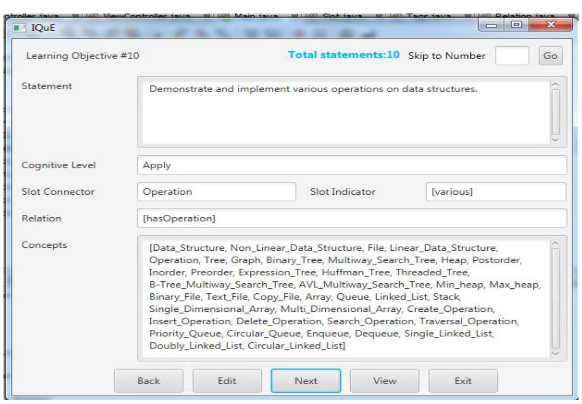

Figure 5. Output of Ontology mapper routine

system. The same is applicable for a question also. The IAO output can be viewed in graphical form using the 'view' command as shown in Fig. 6. The left panel of the output window shows the statistics calculated by the evaluator as discussed in section (IV) C.

VI. TESTING OF IQUE

The testing was primarily done to check whether the annotator is annotating correctly by giving the right color and right shade of color to the nodes. We submitted 10 LOs, 66 AI questions and ontology for the data structures domain to IQuE system. The same set of LOs and questions and domain ontology were also given to expert teachers who were manually told to create LAO. The teachers who have teaching experience of more than 5 years and have thorough domain knowledge were considered. The teacher generated output was compared with the system generated LAO in terms of both concepts and cognitive levels. A confusion matrix was generated which classifies total number of concepts from all LOs and questions into 4 classes: number of concepts in which (a) both the teacher and system colored the nodes (b) both the teacher and system did not color (c) only the system has agreed to color but teacher did not color and (d) only teacher has agreed to color but system did not color. The generated confusion matrix is shown in Table 1. There are total 88 concepts/nodes in the domain ontology.

TABLE 1. CONFUSION MATRIX

	Both agree (a)	**Both disagree (b)**	**Only system agree (c)**	**Only teacher agree (d)**	**% match (a+b)/ (a+b+c+d)**
T1	70	8	6	4	88.6%
T2	76	5	3	4	92.04 %
T3	75	5	3	5	90.9%
T4	69	5	6	8	84.09%
T5	70	5	5	8	85.22%

The agreement in terms of cognitive level of LOs is shown in Graph of Fig. 7.

From table 1 it can be seen that the teachers' notion of concepts covered by an LO or question and system extracted concepts is matching with an average accuracy of 90%. It can also be seen that there is not much variation among the

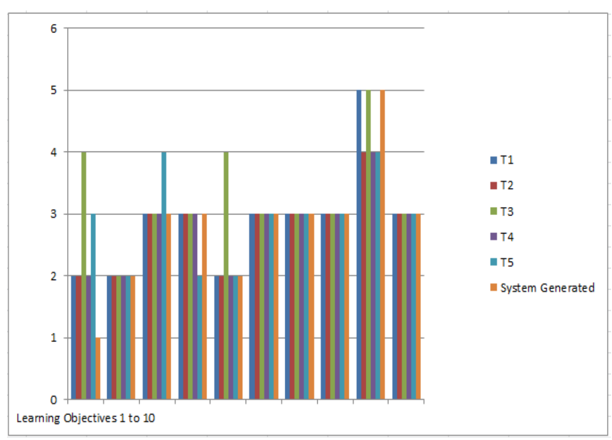

Figure 6. Teachers generated cognitive level Vs. system generated

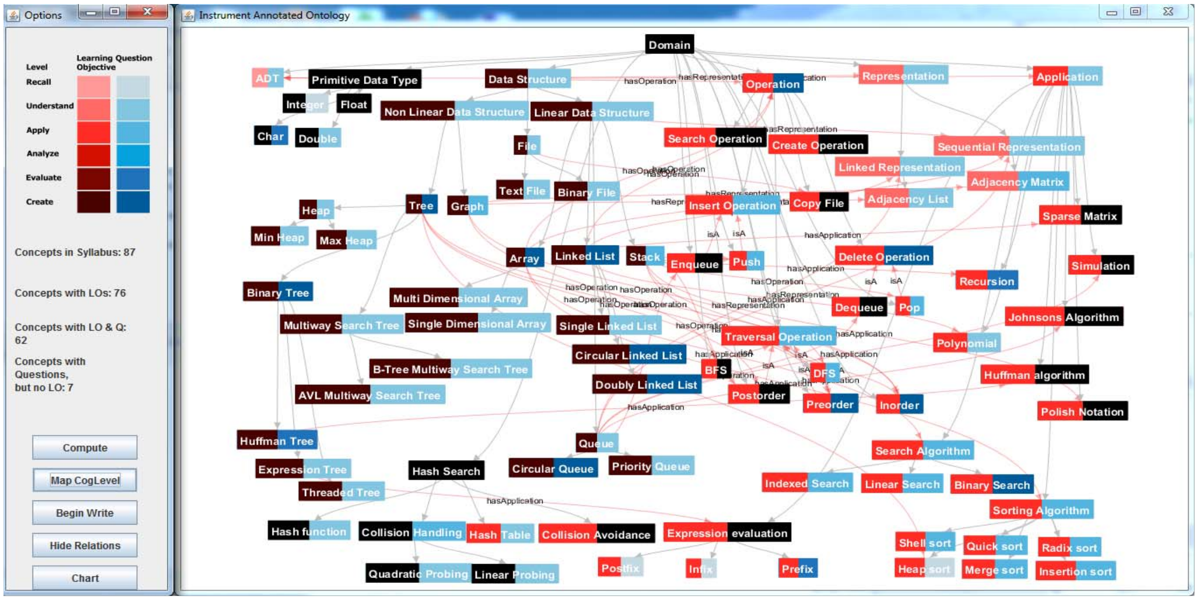

Figure 7. Instrument Annotated Ontology

teachers. Each LO and question was further analyzed to determine which ones have contributed to False Positives (c) and False Negatives (d) and what the characteristics of such LOs are. Sometimes the problem is because of the framing the LOs in a particular way. There can be many ways to handle this such as using more sophisticated NLP techniques, using templates to find semantically close concepts from a question or manually annotating the question. Currently, this is done manually by modifying such questions before giving it to system for parsing. So, we have added a question transformation process where the teacher can edit and reframe the question to submit again.

The teachers were in agreement with system for cognitive level also in most of the cases. The slight mismatch in cognitive level may be because of the inherent ambiguity at adjacent levels in Bloom's taxonomy.

VII. CONCLUSION

We have proposed an approach to capture the alignment of LO and AI in a course using ontology based representation. IQuE is an implementation of this approach. IQuE also provides a visual representation of alignment. The accuracy of this tool is tested by comparing the system generated alignment with the manually generated by expert teachers and a confusion matrix is generated. We got an average accuracy of 90% for concepts and 95% agreement in cognitive level.

IQuE can be used to train teachers in generating well aligned instruments and also for assessing the alignment before an AI is used.

Future plan includes more rigorously testing IQuE and extending the system to other domains. We also plan to do the usability testing of IQuE.

REFERENCES

[1] A. J. Nitko, *Educational assessment of students*. Des Moines, IA 50336-1071, Prentice-Hall, 2001

[2] J. Biggs, "Aligning teaching and assessing to course objectives," Teaching and Learning in Higher Education: New Trends and Innovations, vol. 2, pp.13-17, 2003.

[3] D. R. Krathwohl, " A revision of Bloom's taxonomy: An overview. Theory into practice," Vol. 4 41, pp. 212-218, 2002.

[4] Rekha, R., et al., "Ontology driven framework for assessing the syllabus fairness of a question paper., " IEEE International Conference on Technology Enhanced Education (ICTEE), IEEE, 2012.

[5] B. Simon, and J. Taylor, "What is the value of course-specific learning goals? Journal of College Science Teaching," pp. 53-57, Nov/Dec: 2009.

[6] Natalya F. Noy and Deborah L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford University, 2001.

[7] S. Kriglstein and R. Motschnig-Pitrik, "The Curriculum as an Ontology-A Human Centered Visualization Approach," In World Conference on Educational Media and Technology, Vol. 2011, No. 1, pp. 1343-1352, June 2011.

[8] Y. Chi, "Developing curriculum sequencing for managing multiple texts in e-learning system, " In *Proceedings of International Conference on Engineering Education*, 2010.

[9] H. Chung and J. Kim, "An Ontological Approach for Semantic Modeling of Curriculum and Syllabus in Higher Education," International Journal of Information and Education Technology, Vol. 6, No. 5, May 2016

[10] R. Mizoguchi and J. Bourdeau, "Using Ontological Engineering to Overcome AI-ED Problems: Contribution, Impact and Perspectives, " *International Journal of Artificial Intelligence in Education*, 1-16, 2015.

[11] M. Ronchetti and J. Sant, J., "Curriculum Management and Review: an ontology-based solution", 2007.

[12] M. Sahami, S. Roach, E. Cuadros-Vargas and D. Reed, "Computer science curriculum 2013: reviewing the strawman report from the ACM/IEEE-CS task force, " In *Proceedings of the 43rd ACM technical symposium on Computer Science Education,* pp. 3-4, ACM, February 2012.

[13] C. D. Manning, et al., "The Stanford CoreNLP Natural Language Processing Toolkit." *ACL (System Demonstrations)*, 2014.