Generation of Assessment Questions from Textbooks Enriched with Knowledge Models

Lucas Dresscher, Isaac Alpizar-Chacon[0000−0002−6931−9787], and Sergey Sosnovsky[0000−0001−8023−1770]

Utrecht University, Utrecht, The Netherlands
l.l.j.dresscher@students.uu.nl, i.alpizarchacon@uu.nl,s.a.sosnovsky@uu.nl

Abstract. Augmenting digital textbooks with assessment material improves their effectiveness as learning tools. It can be a laborious task requiring considerable amount of time and expertise. This paper presents an automated assessment generation tool that works as a component of the Intextbooks platform. Intextbooks extracts fine-grained knowledge models from PDF textbooks and converts them into semantically annotated learning resources. With the help of the developed assessment components, these textbooks become interactive educational tools capable to assess students' knowledge of relevant concepts. The results of an expert-based pilot evaluation show that generated questions are properly worded and have a good range in term of difficulty. From the point of assessment value, some generated questions types fall behind manually constructed assessment, while others obtain comparable results.

Keywords: Assessment generation · Interactive textbooks · Textbook models

1 Introduction

[1]Adding assessment to digital textbooks can greatly improve their effectiveness as learning tools from several perspectives. Being interactive learning activities, assessment questions allow students to break from mundane consumption of reading material, thus making learning more engaging [12]. They enable practice and training of knowledge acquired from textbooks, thus allowing students to work with the learning material on different levels of cognitive complexity [19]. And finally, they can provide solid evidence of students' knowledge which is a crucial step for transforming a textbook into an adaptive educational system (AES) [29]. Without such evidence, reliable modelling of students' knowledge becomes a much harder task and the AES has to do with less informative indicators of knowledge comprehension, such as annotations [18], browsing patterns [24] or reading time [13].

There are three principle approaches to add such assessment resources to a textbook: by carefully crafting them [10], by integrating textbooks with external

practice material [27] and by generating assessment directly from the textbook and/or models attached to it. In this paper, we propose a technology that follows the latter approach.

While the recently published studies on assessment generation do show promising developments (see [20] for a systematic overview), a number of aspects still prove to be a challenge. Some of them are related to certain questions types. For example, generation of effective distractors - the incorrect options - for multiple choice questions (MCQs) is a long-standing problem. Other issues are much more specific for the field of cognitive assessment and student modelling where questions are supposed to provide evidence of knowledge of an individual concept rather than estimate the level of mastery in the entire domain. In such a case, it is crucial that the assessment component can accurately define the scope of the questions - the key term/concept that should become the target of assessment. And as the next step, it should be able to formulate a question that is properly worded, grammatically correct, easy to understand, has a reasonable level of difficulty, and (most importantly) can be used to assess students' knowledge of the target concept.

To this end, we have developed an automated assessment generation tool that is used as a component within the Intextbooks platform [2]. Intextbooks extracts knowledge models from well-formatted PDF-based textbooks and transforms them into semantically-annotated educational resources. An important characteristic of these resources when used as input for assessment generation is that they become a source of both high-quality learning content and a semantic model annotating it. The Intextbooks platform can define which concept from the underlying model needs to be tested. As a response, the assessment component can utilise both the relevant parts of the textbooks as well as the semantic neighborhood of the target concept to generate a set of questions targeting the required concept.

The rest of this paper is structured as follows. Section 2 provides a brief overview of assessment generation research. Section 3 outlines most important details of the Intextbooks platform. Section 4 describes the proposed assessment generation component. Section 5 presents the results of an expert-based validation study. Finally, Section 6 concludes the paper with a discussion and a summary of potential directions for future work.

2 Related work

Automated question generation (AQG) is a well-researched area that has been studied for more than three decades, with a surge of activity over the past few years [20]. The main purpose of AQG systems is to aid in or to replace the manual construction of (assessment) questions by experts - a time consuming process with an often flawed outcome [28]. Many different systems have been described over the years that employ different generation methods and generate questions from varying sources. Text has proven to be the most popular form of input, rather than structured sources like ontologies [7, 20].

A system that uses text as input type often employs a rule-based generation method, an approach that uses rules to specify the conditions and transformations required to create a certain question [20]. It utilizes syntactic and semantic information of the text to do so, e.g. provided by annotations from a natural language processing tool. This information is then used to generate different types of questions, like true-false (yes-no) questions (TFQ) [11, 17], cloze (gap-fill) questions (CQ) [1, 8, 25] or multiple-choice questions (MCQ) [22, 23, 25]. A TFQ is a simple declarative sentence to which the answer is either true or false. A CQ consists of a sentence where one word or a sequence of words is replaced by a gap, to be filled in by the student. An MCQ is any question that contains multiple options from which the student needs to choose the correct answer.

Each question type introduces its own specific set of challenges. Gap selection for cloze questions and distractor generation for multiple-choice questions are the most notable ones. Gap selection is concerned with selecting the most appropriate word(s) in the sentence to be replaced by a gap. One approach for this is to use a set of features that evaluate and rank each candidate word based on its syntactic and semantic information [1, 25]. One of the biggest challenges for MCQs is the generation of good distractors [20] - the incorrect answers that accompany the correct answer (the key) as options. A lot of research has been done on generating appropriate distractors - concepts that should be semantically close to the key, but cannot serve as the right answer itself [16]. A dominant approach is to select distractor concept based on their similarity with the key concept [20], e.g. syntactical similarity [14] or contextual similarity [1].

3 Intextbooks

The Intextbooks (Intelligent textbooks) system [2] performs the complete transformation of PDF textbooks into online intelligent educational resources. After extracting a knowledge model from a PDF textbook, it converts it into an HTML/CSS representation with a fine-grained DOM (Document Object Model) enriched with semantic information extracted from the content and formatting of the textbook. Intextbooks consists of two main components. The offline component performs textbook modeling and conversion to HTML, while the online component supports students' interaction with the textbooks. For the current work, we are interested in the offline component.

As the first step, the semantic model of a textbook is extracted by a rule-based system. Its rule set captures common conventions and formatting guidelines for textbook formatting, structuring and organisation. Such elements and tables of contents and indices play the crucial role. However, more subtle aspects, such as formatting styles, repeated texts and commonly used labels, are employed as well. More information can be found in [4]. On the next stage, the domain terms extracted from the textbook index are linked to DBpedia[2]. As a result, the model is enriched with additional semantic information [3]. Finally,

[2] `http://dbpedia.org`

the knowledge model is serialized as an XML file using the Text Encoding Initiative (TEI)[3]; the additional semantic information from DBPEdia is added as RDFa annotations[4]. Altogether, three phases, seven main stages, 17 steps, and 54 unique rules have been defined to handle the extraction process (a detailed description of the complete workflow is provided in [5]).

The research presented in this paper mostly benefits of those steps of the Intextbooks workflow that deal with processing textbooks' indices. Figure 1 illustrates these steps. *Index identification* processes a variety of different index sections (multicolumn, flat, hierarchical) to identify individual index terms (main headings, subentries, locators, cross-references). Each index term has a set of associated page references, which are identified as well. Then, the *term recognition* step identifies the correct reading label and the corresponding sentences for each index term in its reference pages. The reading label is the right reading order for hierarchical index terms (e.g., 'gamma distribution' opposed to 'distribution gamma'). After that, several steps are used to complete *term linking* and *term enrichment* phases in order for index terms to become connected to their corresponding resources in DBpedia. As a result, the index terms are enriched and annotated with semantic information: abstract, categories, Wikipedia article, related terms, and domain specificity – the primary relationship of the index term to the domain of interest [6]. Finally, in the *TEI model construction* step, the structure, content, index terms, and semantic information are expressed using TEI and RDFa attributes.

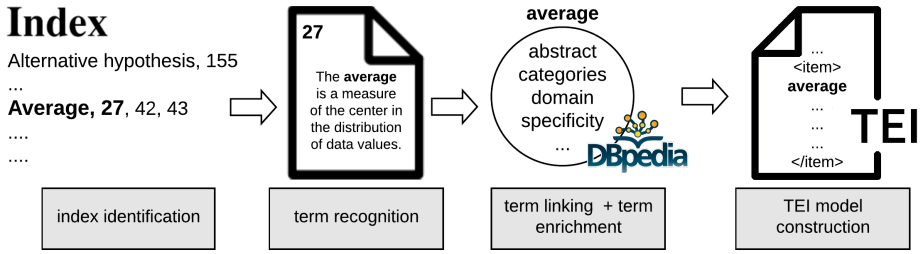

Fig. 1. Relevant steps to extract the index terms information.

In the resulting knowledge models, each content unit (page, subchapter, chapter) is annotated with its corresponding index terms. Additionally, each index term is associated with the exact sentences in which it appears in the reference pages and with additional semantic information.

[3] https://tei-c.org/
[4] http://rdfa.info/

4 Question generation system

Our AQG component broadly follows the pipeline regularly used by rule-based question generation systems [1, 20, 23, 25]. However, it uses a unique combination of both textual and semantic features as input, and therefore deviates from existing systems at a number of ways. An overview of our AQG component is shown in figure 2.

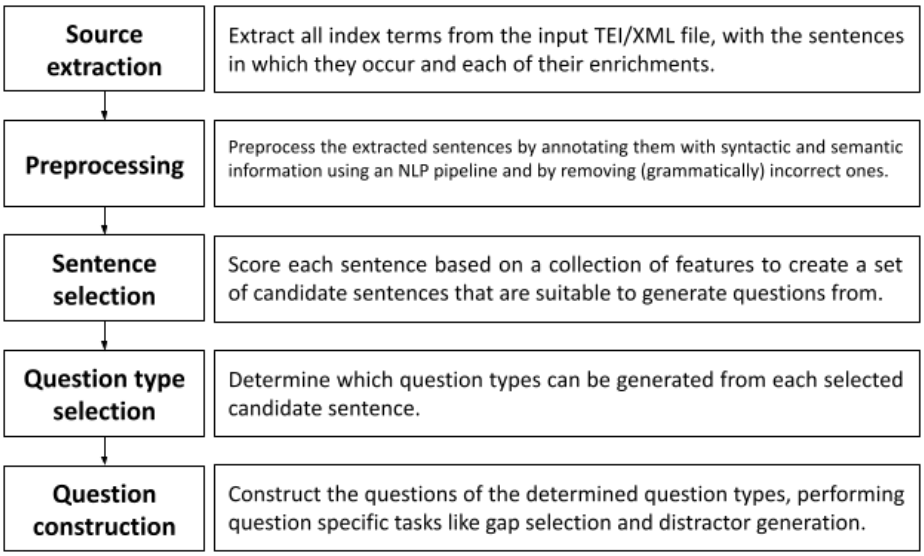

Fig. 2. An overview of the automatic question generation system.

First, the system extracts all sentences from the textbook that are related to the target domain concepts as defined in the TEI/XML(+RDFa) model. A range of Natural Language Processing (NLP) tools is applied to annotated sentences with syntactic and semantic information. This allows to filter out sentences that are grammatically incongruous. Each remaining sentence is then rated according to several criteria that utilize NLP annotations, together with additional information from the the model about the sentence's target concept. Finally, the best phrases are used to generate up to five different question types.

4.1 Source extraction

The AQG component uses the TEI/XML(+RDFa) model described in section 3 to extract all sentences from the textbook relevant to the target concept. The model specifies in which sections of a textbook the concepts are introduced (as defined in the index) and links them to all the sentences from these sections

that mention the concepts. In addition, the index terms' enrichments are extracted from the model. This includes related concepts, its DBPedia abstract and Wikipedia page and its *domain specificity*. The latter information is used to filter out concepts (and their corresponding sentences) that are unrelated to the domain (e.g. terms from other domains used as examples and usecases, such as *epidemic* in a statistics textbook). This step results in an initial set of sentences, corresponding to the target concepts from the main domain of the textbook.

4.2 Preprocessing

In the second step, standard preprocessing common to NLP tasks [20] is performed. We employ the Stanford CoreNLP[5] tool for this purpose, which offers a pipeline of NLP annotators: tokenization, sentence splitting, parts-of-speech (POS) tagging, named entity recognition (NER), lemmatization and dependency parsing.

Figure 3 displays an example phrase annotated by the Stanford CoreNLP pipeline. It is a sentence from the statistics textbook *OpenIntro Statistics* and has three target concepts: variance, standard deviation and random variable. It shows each word's part-of-speech (POS) - its function in the sentence - and the sentence's dependencies, i.e. its grammatical structure and the syntactic relations between the words.

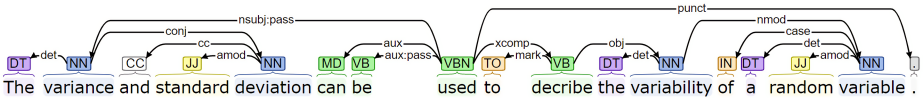

Fig. 3. An example sentence annotated by the Stanford CoreNLP pipeline, visualized by the CoreNLP live online demo (https://corenlp.run/).

Utilizing the above mentioned annotations, the system filters out several types of sentences from the initial list of phrases. First, sentences that are grammatically incorrect or of an unusable structure, like questions or imperative phrases, are removed. Then, sentences that contain verbal references to previously defined context are filter out as well. This involves phrases that start with a discourse connective (e.g. "so", "because") or a personal/possessive pronoun (e.g. "I", "theirs") and sentences that contain a demonstrative pronoun/adjective (e.g. "this", "those"). Sentences that refer to visual elements (e.g. a table, graph or formula), are also removed. Additionally, the component also excludes phrases that originally served as numerical examples, i.e. ones with a very high ratio of numbers. Overall, the preprocessing step transforms the initial set of input phrases into a set of grammatically congruous, standalone (not requiring additional context) sentences with NLP annotations.

[5] https://stanfordnlp.github.io/CoreNLP/

4.3 Sentence selection

The remaining sentences are rated according to a set of criteria, shown in table 1. Every criterion has a weight that indicates its relative importance. To compute the overall sentence score, the weighted sum of all features is taken, i.e. $s = \sum_{i=1}^{n} f_i * w_i$, where s denotes the overall sentence score, f a feature score and w its corresponding weight. Finally, the sentences are compared to a threshold score, producing a set of potential source phrases for question generation. The criteria, their weights and the threshold are selected based on existing research [1, 22, 23, 25] and our own calibration experiments.

Table 1. Feature set used for sentence selection.

Name	**Description**	**Weight**
Sentence header similarity	Considers the similarity between the sentence and the header of its chapter.	30%
Complexity	Considers the complexity, i.e. the number of clauses, of the sentence.	20%
Length	Considers the length, i.e. the number of words, of the sentence.	15%
Domain specificity	Considers the domain specificity of the subject term of the sentence.	15%
Superlatives	Considers the occurrence of a superlative in the sentence.	10%
Comparatives	Considers the occurrence of a comparative in the sentence.	10%

The *sentence header similarity* feature computes the textual similarity[6] between the sentence and the header of its chapter/section, highlighting central sentences of textbook sections. *Complexity* counts the number of clauses, i.e. a subject accompanied by a predicate, of the sentence with score being deducted exponentially for sentences with more than three clauses. It uses the sentence's parse tree to do so. Similarly, *length* considers the number of words of the sentence, with score being deducted exponentially for sentences with more than 25 or fewer than 10 words. Both features aim to select sentences that contain an optimal amount of context. *Domain specificity* utilizes the domain specificity of the terms present in a sentence. This metric is supplied by the TEI/XML(+RDFa). The *superlatives* and *comparatives* features detect informative sentences that contain either one or more superlatives or comparatives, using the sentence's POS tags.

4.4 Question type selection

The fourth step of the AQG component determines which question types can be generated from the selected set of remaining sentences. It looks at their structural and external properties. In systems that generate only a single question type,

[6] https://nlp.stanford.edu/IR-book/html/htmledition/dot-products-1.html

this step is typically incorporated in the sentence selection module as a small number of additional features (e.g., [1, 23]). Our system can generate up to five types of questions per source sentence: three types of true-false questions, cloze questions and multiple-choice questions. This step is also responsible for the final removal of sentences that cannot be used to generate at least one type of questions.

The *unmodified true-false question* (TFU) is a standard true-false question and only requires the phrase to be a declarative sentence. Such sentences follow a subject-verb-object (SVO) structure. The *negated true-false question* (TFN) is a modified version of the previous type, where the original phrase is negated. Such question type requires the source sentence to consist of a single independent clause to minimize the chance of generating a poorly-worded question [23]. The *substituted true-false question* (TFS) modifies the original phrase by replacing the target concept with a different concept. It requires the original concept to be substitutable, which means: its label can occur only once in the sentence and the rest of the sentence cannot provide cues about it. The choice of the substitute is also an interesting problem that generally follows the same rules as the selection of distractors for MCQs (see 4.5). Requirements to the source sentence for a *cloze question* (CQ) are similar to TFS: the target concept can occur only once, and the rest of the sentence should not hint towards it. We also do not generate CQs for concept labels that are longer than three words to avoid over-complicating the question [26]. Finally, the *MCQs* are implemented as a CQ for which the response format is multiple-choice instead of free response. Hence, it has the same requirements to the sources sentence and an additional condition that there are at least three generatable distractors for the sentence's target concept (see 4.5). As an example, the sentence shown in figure 3 meets all the above requirements and can be used to generate all five question types.

4.5 Question construction

In the final step, all questions are constructed from the definitive input set of source sentences, to be presented to the student. This requires performing question type specific tasks, like stem negation (TFN), term substitution (TFS), gap-selection (CQ and MCQ) and distractor generation (MCQ). Each subtask is discussed in the subsections below.

Stem negation and term substitution For a TFU, the source sentence is directly used as question stem to which the answer is *true*. To generate a more diverse set of true-false questions (and answers), the system also generates negated and the substituted TFQs. For a TFN, the original simple sentence's positive verb is modified to a negative verb and vice versa. It takes into account different verbal structures, by looking at the phrase's POS and dependencies annotations. For a TFS, the target concept is replaced by a related term. To not provide any cues to the student, the replacing term matches the original term's capitalization and the possibly preceding indefinite article, i.e. *a* or *an*, is

modified to match with the new term. The replacing term is selected using the same approach as for the distractor generation (see 4.5). As opposed to TFUs, the answer to both TFN and TFS questions is *false*. For example, the TFN of the sentence shown in figure 3 would be: *The variance and standard deviation can not be used to describe the variability of a random variable.* (Answer: *false*).

Gap selection Specific to the CQ type is gap selection, where the target term is replaced by a gap. Gap selection is based on three factors: the target concept's length (at most three words), its domain specificity (only core domain concepts are used) and its height in the syntactic tree of the sentence (a term higher in the tree is scored higher as it contains more context in its sub-trees to create an unambiguous question with a clearer aim [1]). For any term of three words or less, the average of the other two factors is taken as the overall score. The highest scoring target concept of a phrase is replaced by a gap and the correct answer to the CQ is the replaced term. The CQ resulting from the example sentence would be: *The variance and _____ _____ can be used to describe the variability of a random variable.* (Answer: *standard deviation*).

Distractor generation Our system utilizes a combination of syntactic and semantic information for the generation of distractors. Rather than using an external source to retrieve concepts that are semantically similar [23], our approach uses as candidate distractors concepts related to the target concept as defined in the TEI/XML(+RDFa). Table 2 shows an overview of the feature set used to score and select the most appropriate distractors. Similar to the sentence selection module, the weighted average is taken to determine the overall distractor score. Each distractor is ranked according to its score and is selected when it meets a given threshold, which can vary depending on the number of distractors required for the question type.

Table 2. Feature set used for distractor generation.

Feature	**Description**	**Weight**
Syntactic similarity	The syntactic similarity of the distractor and the target concept.	35%
Category similarity	The similarity between the distractor's and target concept's linked DBPedia categories.	20%
Relatedness	The number of ways in which the distractor is related to the target concept.	20%
Textual similarity	The textual similarity between the distractor's and the target concept's DBPedia abstracts.	15%
Domain specificity similarity	The similarity of the distractor's and the target concept's domain specificity.	10%

Example distractors for *standard deviation*, one of the target concepts of the sentence from figure 3, are *standard error*, *mean* and *sample statistic*. Finally,

note that for MCQs, the selected target concept is replaced by only a single gap. This is to avoid providing cues about the correct answer to the student.

5 Evaluation

5.1 Procedure

The developed AQG component has been evaluated in the domain of introductory statistics. We have used the Intextbooks platform to extract models from three university-level textbooks [9, 15, 21] and randomly selected ten core concepts that co-occurred in all three models. Five of these concepts were used to automatically generate questions of all five question types. The other five questions were created manually. The sentences for generated questions were selected by the AQG component from all three textbooks based on the highest scores. The sentences for manually created questions were selected by an expert who located corresponding pages according to the textbooks indices and chose the candidate sentences knowing how the resulting questions should look like.

The resulting set consisted of 25 generated and 25 crafted questions (ten per question type and five per concept) and was given to three domain experts to evaluated them based on several criteria: overall wording (i.e., if a question is both grammatical correct and naturally formulated), assessment value (i.e., if a question is capable to assess the target concept) and difficulty (i.e., how challenging the question is). The experts had to rate all 50 questions according to these 3 criteria on a 3-point scale (3 = max).

Such a setup has allowed us to focus on two main research questions:

- Is our approach potentially sound? In other words, can such a form of AQG potentially produce high-quality assessment questions of various difficulty?
- Is our approach already capable of producing high-quality assessment items of various difficulty?

If the experts rank manually crafted questions low, this means the approach needs a conceptual revamp, and these types of questions based on sentences selected from textbooks simply cannot produce good assessment items. If the experts rank generated questions low, but manually crafted questions high enough, this means our approach is potentially sound and its quality can be improved by fine-tuning the generation algorithm. If the experts rank generated questions high, this means we have already achieved good results.

5.2 Results

Fleiss' Kappa metric was computed for each metric to determine the inter-rater agreement. The results for wording and assessment value were 0.24, 0.27, which are reasonably low. The agreement for difficulty was -0.02. This was rather expected as difficulty of assessment items is a hard metric to estimate objectively. It is usually calibrated based on data produced by real test takers.

Table 3. An overview of the average ratings per metric for each question type separately.

Question Type	Question set	Wording	Assessment value	Difficulty
True-false (U)	Manually created	2.93	2.27	1.60
	Generated	2.67	1.80	1.53
	Difference (p-value)	-0.27 (0.19)	-0.47 (0.17)	-0.07 (0.83)
True-false (N)	Manually created	2.27	2.20	1.73
	Generated	2.07	1.93	1.47
	Difference (p-value)	-0.20 (0.67)	-0.27 (0.61)	-0.27 (0.33)
True-false (S)	Manually created	2.27	1.73	2.00
	Generated	2.40	1.47	1.53
	Difference (p-value)	+0.13 (0.75)	-0.27 (0.39)	-0.47 (0.19)
Cloze	Manually created	2.87	1.87	1.53
	Generated	3.00	1.93	1.80
	Difference (p-value)	+0.13 (0.90)	+0.07 (0.16)	+0.27 (0.65)
MC	Manually created	2.80	2.53	1.60
	Generated	2.87	1.87	1.47
	Difference (p-value)	+0.07 (0.42)	-0.67 (0.92)	-0.13 (0.37)

Table 3 shows a detailed overview of the results for each question type including the results of a Mann-Whitney U test comparing corresponding scores for generated and crafted questions. The average scores for **Wording** are quite high across all types and are comparable between the crafted and generated questions. The scores for **Assessment value** depend on question types. For manually crafted TFUs, TFNs and MCQs, assessment values are relatively high. Generated questions of these types have not reached the same results, even though the differences are not significant. For TFSs, neither crafted, nor generated questions reached high results on assessment value. Finally, for CQs, the results between generated and crafted questions are almost identical; however, not high enough. Difficulty scores are comparable for all assessment types and forms of questions and fluctuate between easy and medium questions.

Overall, such results indicate that the approach is sound enough for most question types. From the points of wording and difficulty, the approach has a reasonable quality. From the point of assessment value, it can be further improved. Also, perhaps, TFS type of questions should be dropped, as even manually crafted questions did not produce good results for it.

6 Discussion and future work

This paper has presented an approach towards automated generation of assessment questions from digital textbooks processed by the Intextbooks system [2]. This research shows the potential of textbooks enriched with linked data. The results from the expert-based validation of the approach show that the approach requires further work, yet it is potentially capable to generate good quality questions of various difficulty.

There are a number of concerns that need to be resolved before more reliable results can be obtained. Textbooks are different in nature from e.g. Wikipedia pages or dictionaries and the sentences selected from textbooks may require more textual transformations to be useful as a question than initially anticipated. Moreover, little to no domain-specific information is used in any of the components of the system, as the goal is to be able to generate (multiple types of) questions for any textbook of any domain. This might be too ambitious; rather than aiming for an open-domain system, it could be more feasible to design the system for a subset of domains, e.g. formal domains exclusively.

Furthermore, the quality of the system very much relies on the quality of two external components: the TEI/XML(+RDFa) input model and the NLP annotation tool. Errors in either of the two components, e.g. missing or incorrect input sentences or inaccurate annotations, propagate through the rest of the system and can have a severe impact on the quality of the output. An example of this is the Stanford CoreNLP coreference resolution, which we initially used to detect references from input phrases to their context sentence and to replace them by their referent. However, early experiments showed that it did not offer a satisfying solution. For future work, it would be interesting to see its performance when trained[7] on the specific domain of the input textbook.

Figure 4 shows how the generated assessment questions could be used within the Intextbooks system (see the top-right panel).

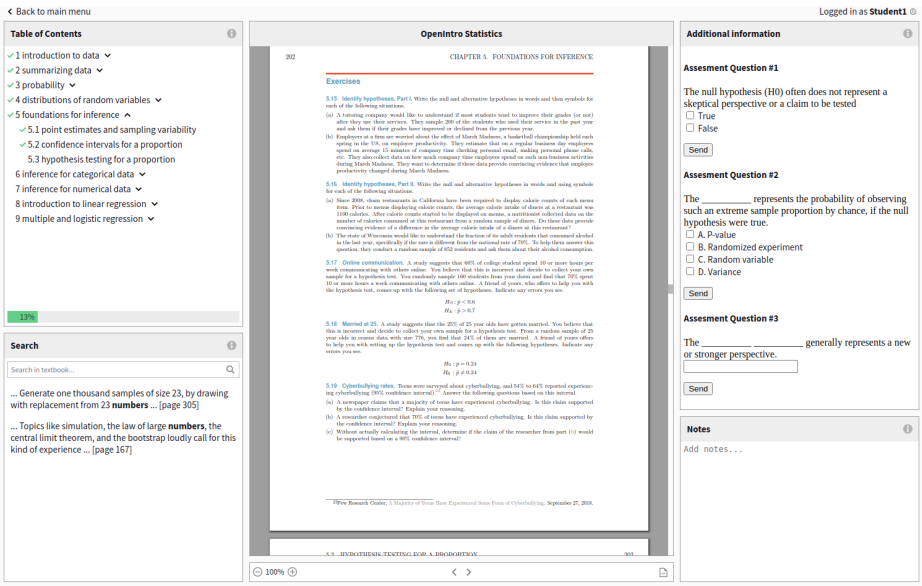

Fig. 4. Conceptual integration of assessment questions as additional content within Intextbooks

[7] https://stanfordnlp.github.io/CoreNLP/coref.html#training-new-models

References

1. Agarwal, M., Mannem, P.: Automatic gap-fill question generation from text books. In: BEA@ACL (2011)
2. Alpizar-Chacon, I., van der Hart, M., Wiersma, Z.S., Theunissen, L., Sosnovsky, S.: Transformation of pdf textbooks into intelligent educational resources. In: Proceedings of the Second Workshop on Intelligent Textbooks. vol. 2674, pp. 4–16. CEUR-WS (2020)
3. Alpizar-Chacon, I., Sosnovsky, S.: Expanding the web of knowledge: One textbook at a time. In: Proceedings of the 30th ACM Conference on Hypertext and Social Media. p. 9–18. HT '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3342220.3343671, https://doi.org/10.1145/3342220.3343671
4. Alpizar-Chacon, I., Sosnovsky, S.: Order out of chaos: Construction of knowledge models from pdf textbooks. In: Proceedings of the ACM Symposium on Document Engineering 2020. DocEng '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3395027.3419585, https://doi.org/10.1145/3395027.3419585
5. Alpizar-Chacon, I., Sosnovsky, S.: Knowledge models from pdf textbooks. New Review of Hypermedia and Multimedia pp. 1–49 (2021)
6. Alpizar-Chacon, I., Sosnovsky, S.: What's in an index: Extracting domain models from digital textbooks. In: Proceedings of the 32nd ACM Conference on Hypertext and Social Media (submitted). HT '21, Association for Computing Machinery, New York, NY, USA (2021)
7. Alsubait, T.: Ontology-based question generation. Ph.D. thesis, University of Manchester (2015)
8. Brown, J.C., Frishkoff, G.A., Eskenazi, M.: Automatic question generation for vocabulary assessment. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. p. 819–826. HLT '05, Association for Computational Linguistics, USA (2005). https://doi.org/10.3115/1220575.1220678, https://doi.org/10.3115/1220575.1220678
9. Diez, D.M., Barr, C.D., Mine, C.R.: OpenIntro statistics. openintro.org (2016)
10. Ericson, B.: An analysis of interactive feature use in two ebooks. In: Sosnovsky, S.A., Brusilovsky, P., Baraniuk, R.G., Agrawal, R., Lan, A.S. (eds.) Proceedings of the First Workshop on Intelligent Textbooks co-located with 20th International Conference on Artificial Intelligence in Education (AIED 2019), Chicago, IL, USA, June 25, 2019. CEUR Workshop Proceedings, vol. 2384, pp. 4–17. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2384/paper01.pdf
11. Flor, M., Riordan, B.: A semantic role-based approach to open-domain automatic question generation. In: Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications. pp. 254–263. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/W18-0530, https://www.aclweb.org/anthology/W18-0530
12. Hake, R.R.: Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. American journal of Physics **66**(1), 64–74 (1998)
13. Huang, Y., Yudelson, M., Han, S., He, D., Brusilovsky, P.: A framework for dynamic knowledge modeling in textbook-based learning. In: Proceedings of the 2016 conference on user modeling adaptation and personalization. pp. 141–150 (2016)

14. Jiang, S., Lee, J.: Distractor generation for Chinese fill-in-the-blank items. In: Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications. pp. 143–148. Association for Computational Linguistics, Copenhagen, Denmark (Sep 2017). https://doi.org/10.18653/v1/W17-5015, https://www.aclweb.org/anthology/W17-5015
15. Kaltenbach, H.M.: A concise guide to statistics. Springer (2012)
16. Karamanis, N., Ha, L.A., Mitkov, R.: Generating multiple-choice test items from medical text: A pilot study. In: Proceedings of the Fourth International Natural Language Generation Conference. pp. 111–113. Association for Computational Linguistics, Sydney, Australia (Jul 2006), https://www.aclweb.org/anthology/W06-1416
17. Killawala, A., Khokhlov, I., Reznik, L.: Computational intelligence framework for automatic quiz question generation. In: 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). pp. 1–8 (2018). https://doi.org/10.1109/FUZZ-IEEE.2018.8491624
18. Kim, D.Y., Winchell, A., Waters, A.E., Grimaldi, P.J., Baraniuk, R.G., Mozer, M.C.: Inferring student comprehension from highlighting patterns in digital textbooks: An exploration of an authentic learning platform (2020)
19. Krathwohl, D.R.: A revision of bloom's taxonomy: An overview. Theory into practice **41**(4), 212–218 (2002)
20. Kurdi, G., Leo, J., Parsia, B., Sattler, U., Al-Emari, S.: A systematic review of automatic question generation for educational purposes. International Journal of Artificial Intelligence in Education **30**(1), 121–204 (Mar 2020). https://doi.org/10.1007/s40593-019-00186-y, https://doi.org/10.1007/s40593-019-00186-y
21. Madsen, B.S.: Statistic fro non-statisticians. Springer (2018)
22. Majumder, M., Saha, S.K.: A system for generating multiple choice questions: With a novel approach for sentence selection. In: Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications. pp. 64–72. Association for Computational Linguistics, Beijing, China (Jul 2015). https://doi.org/10.18653/v1/W15-4410, https://www.aclweb.org/anthology/W15-4410
23. Mitkov, R., Ha, L., Karamanis, N.: A computer-aided environment for generating multiple-choice test items. Nat. Lang. Eng. **12**, 177–194 (2006)
24. Mouri, K., Suzuki, F., Shimada, A., Uosaki, N., Yin, C., Kaneko, K., Ogata, H.: Educational data mining for discovering hidden browsing patterns using non-negative matrix factorization. Interactive Learning Environments pp. 1–13 (2019)
25. Pino, J., Heilman, M., Eskenazi, M.: A selection strategy to improve cloze question quality (05 2011)
26. Smith, N.A., Heilman, M.: Automatic factual question generation from text (2011)
27. Sosnovsky, S., Hsiao, I.H., Brusilovsky, P.: Adaptation "in the wild": ontology-based personalization of open-corpus learning material. In: European Conference on Technology Enhanced Learning. pp. 425–431. Springer (2012)
28. Tarrant, M., Knierim, A., Hayes, S.K., Ware, J.: The frequency of item writing flaws in multiple-choice questions used in high stakes nursing assessments. Nurse Education Today **26**(8), 662 – 671 (2006). https://doi.org/https://doi.org/10.1016/j.nedt.2006.07.006, http://www.sciencedirect.com/science/article/pii/S0260691706001067, proceedings from the 1st Nurse Education International Conference

29. Weber, G., Brusilovsky, P.: Elm-art: An adaptive versatile system for web-based instruction. International Journal of Artificial Intelligence in Education (IJAIED) **12**, 351–384 (2001)