Using Sentence Compression to Develop Visual Analytics for Student Responses to Short Answer Questions

Aneesha Bakharia
Queensland University of Technology
Queensland
Australia
aneesha.bakharia@gmail.com

Shane Dawson
University of South Australia
South Australia
Australia
shaned07@gmail.com

ABSTRACT
In this paper, we report on early research to visualize and summarize student responses to short answer questions. Recently published, graph-based multi-sentence compression algorithms have been successfully applied to summarize opinions – a domain area with many similarities to short answer responses. Initial investigations reveal that visual analytics for short answer questions can be derived from the output of graph-based multi-sentence compression algorithms. A proposed open source short answer analytics tool is also briefly discussed, along with an evaluation plan. The proposed analytics tool will allow lecturers and tutors to have a high level overview of how students have responded to questions, identify knowledge gaps and provide feedback to students.

Categories and Subject Descriptors
H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms
Algorithms, Measurement, Design, Human Factors.

Keywords
Learning Analytics, Visual Analytics, Sentence Compression, Natural Language Processing, Summarization, Graph Layout.

1. INTRODUCTION
Short answer questions are a useful form of summative and formative assessment as they allow students to explain concepts in their own words without providing prompts to students. Grading and providing feedback for short answer questions, depending upon the number of student responses is a tedious task. Multiple choice questions which are easily automatically graded are predominantly used when student numbers are large such as MOOCs. Within flipped classroom scenarios, students are given pre-lecture readings and required to answer short answer questions, with the lecturer analyzing the student answers and addressing knowledge gaps during the lecture. The need for visual analytics to help lecturers and tutors gain an overview of student responses is therefore becoming increasingly important.

A rapid turnaround between students submitting their responses and the lecturer analyzing the response is required. This research is not focused on automatically grading short answer questions, rather the focus in on providing insight into how students have answered questions and allowing lecturers to easily determine the appropriate feedback and support that students require using visual analytics.

Lecturers and tutors require a way to analyse and visualize student responses so that they can:

- understand how students have responded to a question
- review the vocabulary being used
- identify knowledge gaps
- provide feedback to groups of students with similar knowledge gaps

The visual analytics tool that is proposed in this paper will apply sentence compression to summarize student responses to short answer questions. Graph-based sentence compression algorithms have recently been developed and applied to summarize multiple related sentences [3] and opinions within textual reviews [4]. The fact that student responses are made up of short sentences with common phrases being used amoung students, makes the summarization problem an ideal candidate for sentence compression because there is high similarity and redundancy between student responses. Graph-based approaches have also been applied to automatically grade student responses [4].

2. MULTI-SENTENCE COMPRESSION
The first algorithm that has been investigated is the multi-sentence compression algorithm by Filippova [3]. The Filippova algorithm summarizes similar or related sentences and outputs a single short sentence that summarizes the most salient theme conveyed in the cluster of sentences. The algorithm constructs a word graph and uses an approach based upon the shortest paths between words in the graph to produce a summary sentence. The algorithm is easy to implement because sentences must only be tokenized and part of speech tagged. The Filippova algorithm is the first sentence compression algorithm that does not require "hand-crafted rules, nor a language model to generate reasonably grammatical output".

All of the words contained in the sentences form the nodes in the word graph. A word graph is a directed graph where an edge from word A to word B represents an adjacency relation. It also contains start and end nodes (i.e., punctuation). Part of speech information is used to prevent verbs and nouns from being merged in the word graph which would result in the summarization sentence having ungrammatical sequences. Edges within the word graph are used to connect words that are adjacent in a sentence with the edge weight incremented by 1 each time a word occurs after another in a sequence.

Filippova [3] says that good sentence compression goes through all the nodes which represent important concepts but does not pass the same node several times. This is achieved by inverting the edge weights and finding the *K* shortest paths from the start to the end node in the word graph that don't include a verb. The path

through the graph with the minimum total weight is selected as the summary sentence. Additional graph scoring and ranking metrics are used to take into consideration strong links between words and determine salient words.

3. VISUAL ANALYTICS BASED ON MULTI-SENTENCE COMPRESSION

Graph-based multi-sentence compression produces *K* candidate summary sentences, with the sentence with the minimum shortest path score being selected as the summary. Within the context of applying the algorithm to develop visual analytics for short answer questions, we propose to use all *K* candidates because difference common pathways are captured and these may have branches that identify different concepts or vocabulary being used by students.

The following 3 approaches are being considered as summarization and visualization tools for short answer questions:

- Approach 1: Display the K candidate sentences that are derived from the Filippova [3] algorithm. This is only a textual display of the sentences.
- Approach 2: Construct a graph from the *K* candidate sentences and use a graph layout algorithm to display the graph is a visual manner [5]. The advantage over the textual display of the sentence is that loops of words and branches between words would be more easily identifiable.
- Approach 3: Display the full word graph and highlight the *K* candidate paths (sentences) on the graph display. This visualization would allow the lecturer/tutor to see the range of words used. Approach 3 is not presented in this paper but will be evaluated in future work.

4. INITIAL INVESTIGATION

An initial investigation on using the Filippova [3] algorithm to produce a summary and visualization of student responses to short answer questions has been conducted using the open dataset provided by Mohler and Mihalcea [6]. The dataset consists of three assignments of seven short answer questions each given to an introductory computer science class at the University of North Texas. Each assignment includes the question, the teachers answer, and the student responses (usually a few short sentences).

An open source implementation of the Filippova [3] multi-sentence compression algorithm, from the Takahe library (https://github.com/boudinfl/takahe) was used. Tokenization and part of speech tagging was done using NLTK [1]. Numerous spelling errors were noted, but were not fixed for the initial investigations. The minimum number of words in the derived the compressions was set to 6. The number of sentence candidate generated for the 2 examples shown in this paper was 10. The visualization for each of the examples was created using the Yifan Hu [5] Layout in Gephi.

The top 10 summary sentences (Approach 1) and the graph visualization of the word graph constructed from the summary sentences (Approach 2) is included for 2 questions from the Mohler and Mihalcea [6] dataset in Section 4.1 and 4.2.

Initial results show that the summary candidate sentences provide a good overview of the common concepts used by students. The graph visualization of the word graph also shows multiple branches and loops.

4.1 Example 1

The summary candidate sentences in the first example shows that most students have included the 2 key similarities between iteration and recursion related to a termination condition and that both can execute infinitely. Students however have not used the word "repetition" but refer to programming code syntax (i.e., control statement).

Question: What are the similarities between iteration and recursion?
Teachers Answer: They both involve repetition; they both have termination tests; they can both occur infinitely.

Table 1. Top 10 candidate summary sentences for Example 1

Score	Candidate Summary Sentence
0.016	both are based on control statement.
0.015	both are based on a control statement.
0.023	they are based on control statement.
0.021	they are based on a control statement.
0.021	both are based on control statement, termination test.
0.02	both are based on control statement both can infinitely.
0.02	both are based on a control statement , termination test.
0.017	both are based on control statement , both involve a termination test.
0.019	both are based on a control statement , both can infinitely.
0.023	based on control statement , both can occur infinitely.

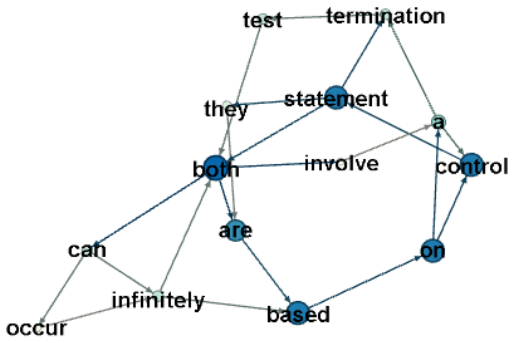

Figure 1. Graph visualization of the top 10 summary sentence candidates in Example 1.

4.2 Example 2

In the second example, very few students include "abstraction" in their answer or concepts that would be associated with "abstraction" such as "encapsulation". Most students mention reusability and maintenance/debugging but it is actually "abstraction" that leads to easier maintenance/debugging of object

oriented programming code. The proposed visualizations would therefore allow the lecturer/tutor to identify the concepts that the students have missed or explained incorrectly and guide the lecturer in providing feedback.

Question: What are the main advantages associated with object-oriented programming?
Teachers Answer: Abstraction and reusability.

Table 2. Top 10 candidate summary sentences for Example 2

Score	Candidate Summary Sentence
0.025	existing classes can be reused program.
0.025	existing classes can be reused program maintenance.
0.023	existing classes can be reused program maintenance and verification are easier.
0.042	objects can be reused program maintenance.
0.04	existing classes can be reused and program.
0.038	existing classes can be reused and program maintenance.
0.05	the classes can be reused program .
0.034	objects can be reused program maintenance and verification are easier.
0.047	the classes can be reused program maintenance.
0.059	objects can be reused and program.

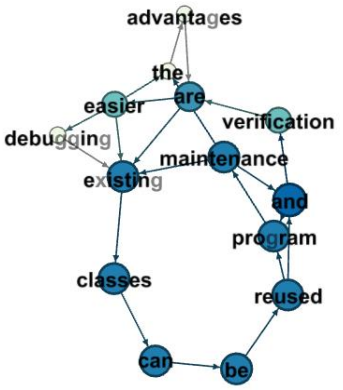

Figure 2. Graph visualization of the top 10 summary sentence candidates in Example 2.

5. TOOL DESIGN AND FUNCTIONALITY

We intend to create an open source tool that incorporates the sentence compression algorithm and the proposed visualizations described in Section 3 that will be made available on Github. The tool will allow lecturers to view student responses that match word graph loops and branches. This will help lecturers to determine context by viewing exemplar student responses. The tool will also allow lecturers to attach feedback to nodes and paths in the word graph as a means of providing specific and targeted feedback to students. Integration with quiz tool export formats from popular Learning Management Systems is also planned.

6. PROPOSED EVALUATION

A between subjects comparative study is being planned. The study will be comprised of two groups. Group A will be required to read all student responses and identify student knowledge gaps. Group B will use the visualizations produced from the output of sentence compression to identify knowledge gaps in the student responses. Identified knowledge gaps from Group A and Group B will then be compared.

Participants in Group B will be shown all 3 approaches described in Section 3 and asked to rate each approach based on principles of visual analytics.

7. CONCLUSION

In this paper, ideas on using graph-based multi-sentence compression as the basis for the visual analysis of student responses to short answer questions were explored. The multi-sentence compression algorithm was introduced and ideas for potential visualizations were discussed. Example visualizations were then presented along with plans to embed visualizations with in an open source tool that is able to integrate with quiz responses from Learning Management Systems. Preliminary results indicate that visualizations derived from the output of sentence compression are able to allow lecturers to identify knowledge gaps and provide feedback to groups of students with similar knowledge gaps. In the future an evaluation of the visualizations will be conducted. The keyphrase extraction algorithm for reranking summary sentences [2] and the Opinosis algorithm [4] will also be evaluated in addition to the Filippova algorithm [3].

8. REFERENCES

[1] Bird, S., Edward L., & Ewan K. 2009. Natural Language Processing with Python. O'Reilly Media Inc.

[2] Boudin, F., & Morin, E. 2013. Keyphrase Extraction for N-best Reranking in Multi-Sentence Compression. In Proccedings of the NAACL HLT 2013 conference.

[3] Filippova, K. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In Proceedings of the 23rd International Conference on Computational Linguistics (pp. 322-330). Association for Computational Linguistics.

[4] Ganesan, K., Zhai, C., & Han, J. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In Proceedings of the 23rd International Conference on Computational Linguistics (pp. 340-348). Association for Computational Linguistics. Chicago.

[5] Hu, Y. F. 2005. Efficient and high quality force-directed graph drawing. The Mathematica Journal, 10 (37-71).

[6] Mohler, M., & Mihalcea, R. 2009. Text-to-text Semantic Similarity for Automatic Short Answer Grading, in Proceedings of the European Chapter of the Association for Computational Linguistics (EACL 2009), Athens, Greece.