Multi-document Cohesion Network Analysis: Automated Prediction of Inferencing across Multiple Documents

Bogdan Nicula[1], Cecile A. Perret[2], Mihai Dascalu[1],

and Danielle S. McNamara[2]

[1] University Politehnica of Bucharest

[2] Arizona State University

Multi-document Cohesion Network Analysis: Automated Prediction of Inferencing across Multiple Documents

Abstract

Open-ended comprehension questions are a common type of assessment used to evaluate how well students understand one of multiple documents. Our aim is to use natural language processing (NLP) to infer the level and type of inferencing within readers' answers to comprehension questions using linguistic and semantic features within their responses. Our taxonomy considers three types of responses to comprehension questions from students (N = 146) who read four documents: a) *textbase responses* (i.e., information required for the answer is present in a contiguous short sequence of text); b) *single-document inference responses* (i.e., requiring information from multiple text segments in a single document); and c) *multi-document inference responses* (i.e., information spanning multiple documents is required). The classification task was approached in two ways. First, we extracted features from students' answers to the comprehension questions using linguistic and semantic indices related to textual complexity and an extended Cohesion Network Analysis (CNA) graph to assess semantic links between the answers and the reference documents. Second, we compared different Recurrent Neural Networks (RNNs) architectures that rely on word embeddings to encode both answers and reference documents. Our best model based on RNN's predicts the answer type with an accuracy of 81%.

Keywords: Predicting question type, Natural Language Processing, Cohesion Network Analysis

Multi-document Cohesion Network Analysis: Automated Prediction of Inferencing across Multiple Documents

Bogdan Nicula
Computer Science Department
University Politehnica of Bucharest
Bucharest, Romania
bogdan.nicula@stud.acs.pub.ro

Cecile A. Perret
Department of Psychology
Arizona State University
Tempe, USA
cperret@asu.edu

Mihai Dascalu
Computer Science Department
University Politehnica of Bucharest
Bucharest, Romania
mihai.dascalu@upb.ro

Danielle S. McNamara
Department of Psychology
Arizona State University
Tempe, USA
dsmcnama@asu.edu

***Abstract*—Open-ended comprehension questions are a common type of assessment used to evaluate how well students understand one of multiple documents. Our aim is to use natural language processing (NLP) to infer the level and type of inferencing within readers' answers to comprehension questions using the linguistic and semantic features within their responses. Our taxonomy considers three types of responses to comprehension questions from students ($N = 146$) who read four documents: a) *textbase responses* (i.e., information required for the answer is present in a contiguous short sequence of text); b) *single-document inference responses* (i.e., requiring information from multiple text segments in a single document); and c) *multi-document inference responses* (i.e., information spanning multiple documents is required). The classification task was approached in two ways. First, we extracted features from students' answers to the comprehension questions using linguistic and semantic indices related to textual complexity and an extended Cohesion Network Analysis (CNA) graph to assess semantic links between the answers and the reference documents. Second, we compared different Recurrent Neural Networks (RNNs) architectures that rely on word embeddings to encode both answers and reference documents. Our best model based on RNNs predicts the answer type with an accuracy of 81%.**

Keywords—Predicting question type, Natural Language Processing, Cohesion Network Analysis.

I. INTRODUCTION

Reading assessments are frequently used to evaluate the coherence of a student's mental representation of a text. Coherence is defined as the degree to which the overall connections between disparate text ideas are cohesively linked with the reader's prior knowledge. In his seminal book on comprehension, Kintsch [1] outlined two separate but related levels of understanding that contribute to the reader's development of a coherent mental representation of the text information: the textbase level and the situation model. The textbase level relates to the basic meaning that is derived from ideas in the text, whereas the situation model integrates various aspects of the textbase with prior knowledge through spreading activation to create a dynamic and integrative process called inferencing. Through inferencing, a reader can link adjacent or distal information in a text with their prior knowledge. Although assessments can target surface level information in a text through text-based questions, the assessments typically must also include deeper level questions that require inferencing in order to truly evaluate the coherence of a reader's mental representation [2].

There are many ways to assess comprehension; these include multiple-choice, open-ended, recall, verification, and even essay questions. Open-ended questions are typically a better form of assessment since these questions force students to rely on their understanding and memory to generate a response as opposed to recognizing the appropriate answer (as is the case with verification and multiple-choice questions) [3]. Additionally, open-ended questions force readers to engage in active comprehension processes through inference generation because of limited retrieval cues [3, 4].

Typically, these reading assessments were conducted on individual texts and only few researchers have begun to assess comprehension across multiple texts [5-7]. Within a single text there exist cohesive devices (e.g. connectives) and anaphoric references to direct the reader to relevant information. These textual cues help facilitate inference generation within a text. Inference generation between texts cannot rely on such cues since each text is written independently of one another. As a result, the absence of these cues forces readers to rely even more on prior knowledge. Whereas inference questions for within text content (intra-textual questions) require generating connections between information that already contains cohesive devices, inter-textual questions require inferencing across separate documents. Therefore, due to their inherent complexity induced by the bridging of remote texts, these responses may be different from those generated for intra-textual inference questions.

Our research question is the following: to what extent can textual complexity indices, coupled with word embeddings and deep learning models, assess the level and type of inferencing evidenced within answers generated by students in response to open-ended questions? Our aim is to introduce a tool that provides feedback to students on whether their production matches the level of inference required by the given question. Doing so first requires a benchmarked set of questions that are

2375-0197/20/$31.00 ©2020 IEEE
DOI 10.1109/ICTAI50040.2020.00061

designed to tap into different levels of inferencing. As such, we leverage the corpus reported in Nicula, et al. [8], described below.

We conceive this problem as a classification task, such that we categorize participants' responses into three potential categories of increasing degrees of inferencing within the answer: textbase, single-document, or multi-document inference. As such, we assess the extent to which inferencing can be detected within readers' open-ended responses to comprehension questions.

Our objective was to compare the use of linguistic and semantic features within different types of machine learning models, and to also explore the use of deep learning (i.e., Recurrent Neural Networks). In Study 1, we used two types of handcrafted features, coupled with various machine learning models to predict the level and type of inferencing evidenced within student answers. In Study 2, we examine the accuracy of the classification model using word embeddings within a deep learning model (i.e., RNN). In contrast to Study 1, which relies on classical machine learning models applied to linguistic and semantic features, Study 2 analyzes classification accuracy using models that generate their own internal representations in order to make a prediction.

II. General Method

A. Corpus

The corpus reported by Nicula, et al. [8] consists of productions from 146 students who were asked to read four texts (referred to as text A, B, C, and D) on the same topic, green living (i.e. the feasibility of implementing sustainable living methods). In terms of length, text A is by far the most complex text, having a length of 720 words, while texts B, C, and D, are approximatively half the length –more precisely, 360, 334, and 369 words, respectively.

The 146 students were asked to answer 12 open-ended questions comprised of three questions per text. There were three different question types, each depending on the information that they targeted (see Table I). In total, there were 584 question answers per question type, resulting in 1752 responses in total. Out of these, 15 were blank and were skipped in further analyses.

TABLE I. List of questions.

	Question IDs	Number of examples
Text A	Q1, Q2, Q3	Intra-textual, Intra-textual, Inter-textual
Text B	Q4, Q5, Q6	Textbase, Intra-textual, Inter-textual
Text C	Q7, Q8, Q9	Textbase, Text-based, Inter-textual
Text D	Q10, Q11, Q12	Textbase, Intra-textual, Inter-textual

Textbase questions were designed to target information located in one sequence of text from one document (e.g., Q4 refers to one continuous sequence of text in text B) and they correspond to textbase answers. Intra-textual questions targeted information presented in multiple sections across a single document; the associated answer class is single-document inference. Finally, if the information was presented in multiple documents, then the question was considered inter-textual (e.g., Q3 refers to information mentioned across all texts but focused in text A) and the corresponding answers are labeled as multi-document inference answers.

Regardless of its type, each question has one of the four texts as its "reference text", indicating that the information targeted by that question can be located in that reference text. In the case of intra-textual questions, additional useful information can be found in other texts, but it is not necessary for answering the question. In contrast, even if a reference text is mentioned for inter-textual questions, information from multiple texts is necessary to generate a complete answer.

In terms of preprocessing, few changes were made to the texts before the two methods below were applied. Given that semantic models and word embeddings were used in both cases, no stemming or lemmatization techniques were applied to the text. Only a standard tokenization was applied to split the text into words. However, no preprocessing was performed to correct possible spelling errors made by the students. Misspelled words, for which there were no corresponding terms in our vocabularies, were ignored by the two methods.

II. Study 1: Cohesion Network Analysis and Textual Complexity Indices

We explored two directions of deriving relevant features for this study. First, we were interested in a global view, namely the links between the question answer and the reference documents. Thus, we considered features from the Cohesion Network Analysis (CNA) extended graph [8, 9] that establishes semantic links between the question answers and the source texts. We hypothesized that the semantic relations would differ between answer types because each of the corresponding questions was designed to tap into different relations. Cohesion Network Analysis is a technique inspired from Social Network Analysis [10], aimed at modelling text cohesion by generating a graph composed of links between different elements of the text(s), which can have different granularities (e.g. sentence, paragraph, entire text, etc.). These links represent the similarity between the two text elements, and can be computed using different lexical overlap measures or semantic models, such as Latent Semantic Analysis – LSA [10], Latent Dirichlet Allocation – LDA [11], GloVe [12] or word2vec [13]. In this study, the graph only contained links between the question answer and the source text (i.e., sentences or paragraphs from the text). This resulted in a set of 15 features per question answer (text features and paragraph and sentence aggregated features).

Second, we were interested in a local view centered on the question answer. We extracted linguistic features from each answer, reflective of specific writing characteristics. We relied on 700 textual complexity indices from the ReaderBench framework [14] covering surface, lexical, syntactic and semantic properties. The surface and lexical features are easily computable features such as: sentence or paragraph lengths (average values and standard deviations), commas per sentence or paragraph, number of unique words and others. Syntax-level indices are used for extracting both word-level features (e.g., the frequency for each part of speech) and sentence-level features, which are structural features derived from the parsing tree (e.g., the maximum depth of the parsing tree). Lastly, semantic features consider the CNA semantic links within a document, as well as specific word lists designed to capture certain semantic

valences (e.g., General Inquirer[1] – GI [15], or Lasswell [16] dictionaries). One example of such a list taken from the GI categories is the "Academ" list, which contains words related to academic, intellectual, or educational matters.

The resulting 715 features were filtered to eliminate multicollinearity. All pairs of two features with a Pearson correlation greater than .9 were analyzed using the Kruskal-Wallis H Test and the feature with the lowest χ^2 score was eliminated. After this step, 599 ReaderBench features and 7 CNA features remained, for a total of 606 features. These features were used as input for a set of machine learning classifiers from the SciKit Learn library [17] to predict the answer type: SVC linear (Support Vector Classifier with a linear kernel), SVC RBF (Support Vector Classifier with a Radial Basis Function kernel), Extra Trees classifier, and Multi-Layered Perceptron (MLP). The SVC models were chosen due to their popularity. We opted for both the RBF and linear kernels because the first one is better at classifying data that is linearly non-separable, while the second is better for linearly separable data. The Extra Trees model was chosen because it works well at separating a large number of features and it allows easy interpretation of feature importance. Out of the four selected classifiers, Extra Trees is also the only bagging and boosting algorithm, meaning that it is the only one that builds multiple weak classifiers and bundles their results together to create a more robust prediction. Thus, the model is more resilient to noise and able to work well in high dimensional feature spaces. The Multi-Layered Perceptron model was selected because it is simple, straightforward, and computationally fast.

III. Study 1: Results and Discussion

All experiments employ 5-fold cross-validation conducted 10 times to account for outliers. The reported metric is the average of the overall accuracy for the three given classes (i.e., number of correct predictions divided by the size of the test set), across all 10 runs. The five training and testing datasets for each run were compiled such that students' responses remained within the same set. This ensured that class balance was maintained, as each student answered an equal number of questions (4), from each of the three categories. Several experiments were conducted to assess the importance of the two types of features and their combination (see Table II). We further compared accuracy using the four types of machine learning models (i.e., Extra Trees, SVC RBF and linear, MLP).

A. Predicting answer type

The highest accuracy in predicting the correct answer type is bolded for each feature set in Table 2. Extra Trees produced good results for all three combinations, obtaining the highest accuracy when using the CNA features; in addition, results were close to the best ones in all other scenarios. The average accuracy of 60.80% was obtained with the Extra Trees model when using the CNA features, but this was the lowest accuracy among the three scenarios. When using the ReaderBench features, the accuracy was 67.83% with the same classifier, but an even better result was obtained using the MLP classifier. When using all features, accuracy was improved slightly more for all models except the SVC with RBF kernel. The best result was obtained by the MLP model, reaching an average of 68.47%.

We noticed that the SVC RBF model had a large drop in performance in the 2nd and 3rd scenario. This is probably due to the fact that it tries to find nonlinearly separable features and struggles with the high-dimensional space. The best results for the two scenarios with a large number of features (599 and 606) were obtained using SVC linear and MLP, respectively; however, both previous models and Extra Trees obtained similar results.

TABLE II. Classification accuracy and F1 using different feature sets

Feature set (# of features)	Model	Avg. F1-score	Avg. accuracy	Max accuracy
CNA (7)	Extra trees	**60.72%**	**60.80%**	**62.64%**
	SVC RBF	57.20%	57.14%	59.77%
	SVC linear	52.01%	52.53%	56.32%
	MLP	56.50%	56.86%	58.62%
ReaderBench textual complexity indices (599)	Extra trees	67.71%	67.83%	73.55%
	SVC RBF	45.25%	45.64%	49.30%
	SVC linear	**67.76%**	67.82%	73.00%
	MLP	67.58%	**67.95%**	**74.21%**
All features (606)	Extra trees	68.08%	68.15%	**72.12%**
	SVC RBF	45.33%	45.83%	49.71%
	SVC linear	**68.45%**	**68.47%**	70.97%
	MLP	67.58%	68.14%	71.26%

Given the large number of features obtained when combining the CNA features and the textual complexity indices, we also attempted to filter them by selecting only top-*k* features based on their importance, as reported by the Extra Trees models. We analyzed three cases with $k = 10$, $k = 50$, and $k = 100$ (see Table III). Although all four types of machine models were calculated, the best results by at least 4% in all cases was obtained by the Extra Trees model. It seems that this model performs better in these scenarios with fewer features. This could also be influenced by the hyperparameter addressing the number of estimators, i.e., how many weak classifiers the Extra Trees model uses. This parameter was kept fixed throughout the experiments. Given that the Extra Trees model obtained the best or second-best results in all three cases, we report only the results for those models.

The $k = 10$ scenario obtained a 1.5% better result than the scenario using only CNA features, underlining the importance of the textual complexity indices. We could notice a steady improvement in results as the number of features increased, the best result, out of the three, being the one for $k = 100$. This result improved upon the best nonfiltered results by 1%, reaching 71.06%, thus arguing for the necessity of this second filtering.

TABLE III. Extra Trees model classification F1 and accuracy as a function of the feature set size ($K = 10$, $K = 50$, $K = 100$) based on the feature importance

Model	Average F1 scores	Average accuracy	Max accuracy
Top 10	63.17%	63.16%	66.48%
Top 50	68.30%	68.20%	69.82%
Top 100	71.11%	71.06%	73.56%

[1] http://www.wjh.harvard.edu/~inquirer/homecat.htm

When analyzing the top features, we could also observe the importance of the CNA features (see Table IV). Despite representing only 1.1% of the total number of features, 4 out of 10 top features were CNA-based (see Table IV in which CNA features are marked with *). In the top 30, all 7 CNA features that remained after the multicollinearity filtering were present.

TABLE IV. TOP 10 FEATURES ACCORDING TO THE EXTRA TREES MODEL

ID	Feature name	Feature importance	$\chi^2(2)$	*p*
1	Links between QA and all texts (Stdev)*	0.0273	186.239	<.001
2	Links between QA and all paragraphs from the target text (Stdev)*	0.0188	71.871	<.001
3	Links between QA and all sentences from the target text (Stdev)*	0.0183	80.237	<.001
4	Average words per sentence related to negative affect category from GALC [18]	0.0173	182.211	<.001
5	Average number of nominal subject dependencies per sentence	0.0139	111.030	<.001
6	Average number of third person pronouns per paragraph	0.0136	124.036	<.001
7	Links between QA and all paragraphs from the target text (Max)*	0.0136	104.485	<.001
8	Average word length	0.0133	60.694	<.001
9	Average words per paragraph related to economy category from GI [15]	0.0130	106.728	<.001
10	Average number of sentences per paragraph	0.0124	30.387	<.001

One interesting finding is that the top three features consider the standard deviation of the semantic similarity between the question answer and all the texts, all the paragraphs in the reference text, and all the sentences in the reference text. These metrics quantify the variability of links between the question answer and different text elements of the same granularity. The top ReaderBench textual complexity features seem to be almost evenly distributed among surface-level features (e.g., average number of sentences per paragraph or average word length), syntax and morphology features (e.g., average number of third person pronouns, average number of nominal subject dependencies), and semantical features (e.g., average number of words appearing in General Inquirer or GALC word lists).

Another interesting finding is that the most important ReaderBench textual complexity feature, according to the Extra Trees model, covers the average valence related to the economy word list from the General Inquirer. We assume this is due to the topics of the texts, which are centered on green living; a topic with a strong economic emphasis. Some other features may seem more of an example of overfitting to the particularities of the text, such as average counts relative to negative affect. Nevertheless, the wide range of considered features allows the method to be adaptable to new problems. The machine learning models have to be retrained from scratch for every new dataset to learn what features are most important for the given task.

The confusion matrix for the best Extra Trees model is provided in Table V. The most difficult to predict class was the single-document inference one as it was the easiest to mistake with either simpler textbase answers from a cognitive point of view, or the slightly more complex multi-document inference answers. Nearly 18% of single-document inference answers were categorized as multi-document inference, and nearly 16% of multi-document inference answers were classified as single-document inference. This underlines the similarity of the two types of question answers, from the perspective of the types of features extracted using ReaderBench and CNA.

TABLE V. CONFUSION MATRIX WHEN RELYING ON THE TEXTUAL COMPLEXITY INDICES

	Predicted Textbase	Predicted Single-document inference	Predicted Multi-document inference
Actual Textbase	73.33%	13.33%	13.33%
Actual Single-document inference	15.00%	66.67%	18.33%
Actual Multi-document inference	13.33%	15.83%	70.83%

IV. STUDY 2: CLASSIFICATION USING RECURRENT NEURAL NETWORKS

As opposed to the Study 1 in which classical machine learning models were used on top of handcrafted features, in this study we analyzed models that generate their own internal representations in order to make a prediction. To this end, we implemented Recurrent Neural Networks using the Pytorch deep learning library [19]. Pretrained 300-dimensional GloVe [12] word embeddings were considered. These embeddings are a high dimensional representation of the meaning of each word and they allow the neural network to aggregate information from words to text level through multiple levels of processing. Every input for the deep learning model was translated from text into a set of indices, each index pointing to the corresponding GloVe embedding. The array of word embeddings is passed through a standard feature extractor module composed of two stacked bidirectional Long Short-Term Memory –LSTM [20] layers, a set of pooling operations, and a fully connected (FC) layer. Depending on the architecture type, after this step one or more fully connected layers are used to combine intermediary results and predict the answer type (see Fig. 1). The first architecture from Fig. 1 uses only the question answer as input for the feature extractor module.

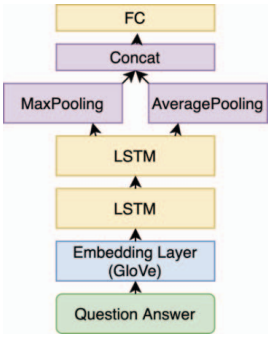

Fig. 1. Simple architecture using only the question answer as input

Afterwards, we experimented with a Siamese architecture [21] (see Fig. 2) in which we used the question answer and also a secondary input consisting of either the question, or the reference text. Both the question answer and the secondary input were processed separately by the model, using the same processing pipeline. This results in two sets of features which were concatenated and used as input for a fully connected layer (a single layer perceptron) that predicts the question type.

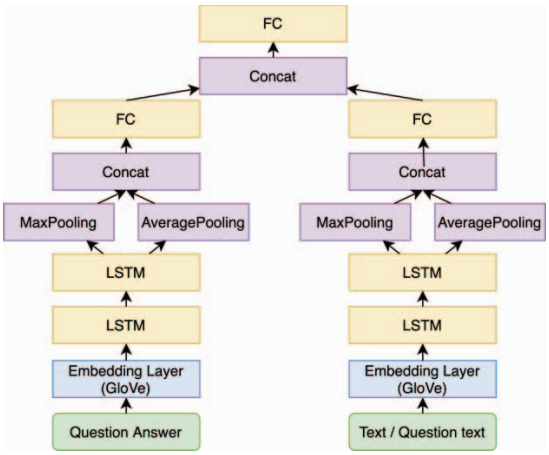

Fig. 2. Siamese architecture using as a first input the question answer and as a secondary input either the reference text, or the question.

V. Study 2: Results and Discussion

The neural network models introduced for this study takes into account both the student's response, as well as its potential links to the question text or reference documents. Three scenarios were tested, depending on the type of input that was used for making the prediction: a) prediction of the answer type based only on the question answers (QAs), b) prediction based on the question answer and the question text itself, and c) prediction based on the question answer and the reference text that was targeted by the question.

One of the scenarios was a failure. When trying to make the prediction based on the question answer and the question text, the model overfits. This happens because the model becomes biased due to the small number of questions (12 overall). We examined how the model fared when tested on answers to unseen questions, so we trained this model on answers from 9 of the 12 questions and test it on answers from the remaining 3. In this case the training accuracy remained at 100%, but the test accuracy dropped to 33% (i.e., the model learned the answer type from the question text without making any links to the answer; i.e., it was overfitting). Thus, Table VI considers only the two remaining relevant scenarios (i.e., only the question answers and the questions answers in combination with the reference text).

TABLE VI. Classification Accuracy Using the Different Deep Learning Models

Model	Average Test F1-score	Average Test Accuracy	Average Training Accuracy
Only QAs	78.47%	78.37%	79.65%
QA + text	**81.33%**	**81.20%**	**83.21%**

The best performance was obtained by the model using both question answers and the reference text. We notice a significant improvement in the prediction of the textbase and intra-textual examples (a 15-16% increase in accuracy), when analyzing the confusion matrix for the best performing model (81.2% accuracy) – see Table VII. The performance for the multi-document inference class, however, improved by only 3%, illustrating that this class of answers is more difficult to categorize.

TABLE VII. Confusion Matrix for the Deep Learning Models Using Both Question Answer and the Reference to the Read Texts

	Predicted Textbase	Predicted single-document inference	Predicted multi-document inference
Actual textbase	88.46%	4.81%	6.73%
Actual single-document inference	7.89%	82.46%	9.65%
Actual multi-document inference	12.32%	13.77%	73.91%

A second observation is that the textbase class remained the easiest to predict, but the margin between it and the other classes increased from 3% to a 6% margin. This may be explained by the decrease in the number of single-document inference samples classified as textbase and vice-versa (from 15% and 13% to 8% and 5%). This is probably due to the LSTM-based Recurrent Neural Network model which is better at extracting and synthesizing semantic information; thus, the model becomes more finetuned at separating between those two classes which are structurally and lexically more similar, than the third one. This assumption is also supported by the observation that the deep learning model only slightly improves the performance for classifying multi-document inference answers.

VI. Conclusions

Both CNA features and ReaderBench textual complexity indices play important roles in predicting the answer type. The CNA features that were preferred by the models were those underlining how spread out the information from the answer was across the source text. The two sets of features are focused on complimentary facets of the response. The top ReaderBench indices focused on surface elements (e.g., punctuation, word uniqueness), as well as semantic information (e.g., word valences from different dictionaries) from the question answers. The simple top-*k* feature selection method that we employed further improved the results. An improvement in terms of feature selection could be to prune features based on the type of information embedded in them.

The deep learning approach obtained considerably better results, despite the relatively small number of samples. The risk of overfitting with such small data was still visible in the scenario where we tried to use both the question answer and the question as inputs. This approach may have yielded better results if a wider variety of questions were available. The combination of question answer and reference text representation features yielded the best result, indicating that the deep learning model might also be looking at the connectivity between the question answer and the source text. However, the

question answer and the reference text are treated by our RNN as two large sequences of data, from which the network extracts features combined only at a later stage. The only link between different text elements that the model is able to comprehend is the order in which words appear. It cannot distinguish sentence or paragraph ends, for instance, and it considers the text as a chain of words, not a graph of knowledge. Thus, it seems unlikely that such a model can actually analyze the connections between ideas in multiple parts of the text in relation to the question answer.

This work could be improved in the future by increasing the dataset with more data of a similar kind, as well as by using a proxy task for training a more complex network that could then be used on the current dataset. A second line of action could be to integrate the CNA approach, which is better at analyzing the structure and cohesion of the texts, with a deep learning model, which seems better at analyzing semantics. This could be done by: a) training a deep learning model and using it to compute the links in the CNA graph, or b) considering a hierarchical deep learning approach, in which the network would analyze the text bottom up and aggregate the information at different granularity levels, in accordance with the structure of the text.

In conclusion, we conducted two studies to explore methods to classify question answers based on the type and level of inferencing targeted by the question in the context of multiple document comprehension. Such algorithms will contribute to a larger objective of developing the means to provide feedback to students on the quality and depth of their answers. Ultimately, the objective is to enhance students' ability to understand multiple documents at deep levels, generating connections both within and between documents. We provide comparisons of machine learning and deep learning models to assess natural language within student responses and ultimately provide initial steps toward providing automated feedback in the context of multiple document comprehension.

ACKNOWLEDGMENTS

This research was partially supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-III 54PCCDI / 2018, INTELLIT – "Prezervarea și valorificarea patrimoniului literar românesc folosind soluții digitale inteligente pentru extragerea și sistematizarea de cunoștințe", the Institute of Education Sciences (R305A190063, R305A180144 and R305A180261), and the Office of Naval Research (N00014-17-1-2300). The opinions expressed are those of the authors and do not represent views of the IES or ONR.

REFERENCES

[1] W. Kintsch, Comprehension: A paradigm for cognition. Cambridge, UK: Cambridge University Press, 1998.

[2] D. S. McNamara and W. Kintsch, "Learning from text: Effects of prior knowledge and text coherence," Discourse Processes, vol. 22, pp. 247–288, 1996.

[3] Y. Ozuru, S. Briner, C. A. Kurby, and D. S. McNamara, "Comparing text comprehension measured by multiple-choice and open-ended questions," Canadian Journal of Experimental Psychology, vol. 67, pp. 215–227, 2013.

[4] J. Magliano and K. Millis, "Assessing reading skill with a think-aloud procedure and latent semantic analysis," Cognition and Instruction, vol. 21, pp. 251–283, 2003.

[5] I. Bråten, L. E. Ferguson, Ø. Anmarkrud, and H. I. Strømsø, "Prediction of learning and comprehension when adolescents read multiple texts: The roles of word-level processing, strategic approach, and reading motivation," Reading and Writing, vol. 26, pp. 321–348, 2013.

[6] L. Le Bigot and J.-F. Rouet, "The impact of presentation format, task assignment, and prior knowledge on students' comprehension of multiple online documents," Journal of Literacy Research, vol. 39, pp. 445–470, 2007.

[7] I. Rukavina and M. Daneman, "Integration and its effect on acquiring knowledge about competing scientific theories for text," Journal of Educational Psychology, vol. 88, pp. 272–287, 1996.

[8] B. Nicula, C. A. Perret, M. Dascalu, and D. S. McNamara, "Predicting Multi-document Comprehension: Cohesion Network Analysis," in 20th Int. Conf. on Artificial Intelligence in Education (AIED 2019), Chicago, IL, 2019, pp. 358–369.

[9] M. Dascalu, D. S. McNamara, S. Trausan-Matu, and L. K. Allen, "Cohesion Network Analysis of CSCL Participation," Behavior Research Methods, vol. 50, pp. 604–619, 2018.

[10] J. Scott, Social Network Analysis. London, UK: Sage, 2017.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," Journal of Machine Learning Research, vol. 3, pp. 993–1022, 2003.

[12] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation," in The 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP 2014), Doha, Qatar, 2014.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representation in Vector Space," in Workshop at ICLR, Scottsdale, AZ, 2013.

[14] M. Dascalu, S. A. Crossley, D. S. McNamara, P. Dessus, and S. Trausan-Matu, "Please ReaderBench this Text: A Multi-Dimensional Textual Complexity Assessment Framework," in Tutoring and Intelligent Tutoring Systems, S. Craig, Ed., ed Hauppauge, NY, USA: Nova Science Publishers, Inc., 2018, pp. 251–271.

[15] P. Stone, D. C. Dunphy, M. S. Smith, D. M. Ogilvie, and associates, The General Inquirer: A Computer Approach to Content Analysis. Cambridge, MA: The MIT Press, 1966.

[16] H. D. Lasswell and J. Z. Namenwirth, The Lasswell Value Dictionary. New Haven: Yale University Press, 1969.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. J. J. o. m. l. r. Dubourg, "Scikit-learn: Machine learning in Python," vol. 12, pp. 2825–2830, 2011.

[18] K. R. Scherer, "What are emotions? And how can they be measured?," Social science information, vol. 44, pp. 695–729, 2005.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems, 2019, pp. 8024–8035.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, pp. 1735–1780, 1997.

[21] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," arXiv, vol. preprint arXiv:1404.2188, 2014.