Doc2Vec Approach towards Constructed Short Answers Assessment in Tutorial Dialogue Context

Lasang J. Tamang
Department of Computer Science
University of Memphis
Memphis, TN, USA
ljtamang@memphis.edu

Abstract

Answer assessment has been one of the crucial aspect for success of intelligent tutoring system as the goal of every tutoring system is to be able to grade answers. We propose to use doc2vec representation of student answers and reference answer, compute cosine similarity score between them and use those score as features to build multinomial classifier in order to access student answer automatically. Our result shows that this approach can grade the student answer correctly by 43.22%

1 Introduction

Assessment of student answer is one of the most important process of any tutoring process; otherwise tutoring progression is not known. In typical classroom tutoring process, an instructor grades the student answer. However, with recent growth and advancement of various dialogue based intelligent tutoring system, such as (Rus et al., 2014; Lane and VanLehn, 2004; Graesser et al., 2004; Evens et al., 1997), where computer can naturally interact and tutor large number of student at once, the job of grading such large number of student is almost impossible or require large effort. In this scenario, the need of system that is able to grade student answer is critical requirement and has thus gained much research interest.

Constructed short answers are the response given by student to question during tutoring process in intelligent tutoring system. This answer are natural conversation and the assessment of Such constructed answer are very different and tougher than assessing multiple choice answer where the answer are fixed from available options. Their assessment requires proper understanding of their semantic and syntactic meaning. In our research, we use doc2vec representation ((Mikolov et al., 2013) to capture semantic meaning of student answer and the reference answers prepared by subject expert, and we then calculate cosine similarity to feed them to multinomial logistic classifier which can classify the student answers (i.e. correct, incorrect, contradictory, etc).

In the following section, we first describe background required to understand doc2vec representation (which is most essential to understand our rest of the work), then we present our approach in attempt to making automatic answer assessment, then we discuss about our experimental procedure and finding of the experiment in brief, and finally end with conclusion.

2 Background

The first step in our research is to convert both student answer and reference answer to vector representation. We achieve this using 1) a distributed memory model for paragraph vector (PV-DM) and 2) Paragraph Vector without word ordering: Distributed bag of words (PV-DBOW), a popular techniques for capturing semantic meaning of documents by (Mikolov et al., 2013). Thus, we briefly discuss about these two technique in the section below.

2.1 Paragraph Vector: A distributed memory model

In Paragraph Vector framework (see Figure 1), every word is mapped to a unique vector(by a column in matrix W) and also every paragraph is mapped to a unique vector(by a column in matrix D) . These word and paragraph vectors are averaged or concatenated to predict the next word in a context. Paragraph and word vector are trained using gradient descent. Once trained, the paragraph vector can be used as feature for input to conventional machine learning technique such as logistic regression in our case.

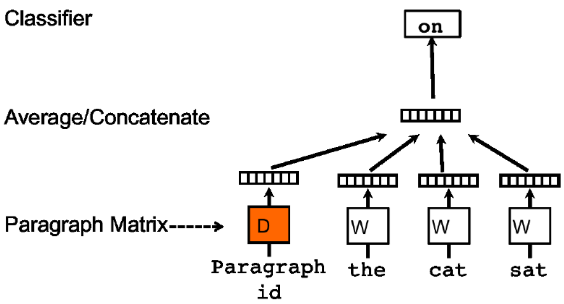


Figure 1: Distributed memory model version of paragraph vectors

2.2 Paragraph Vector without word ordering: Distributed bag of words

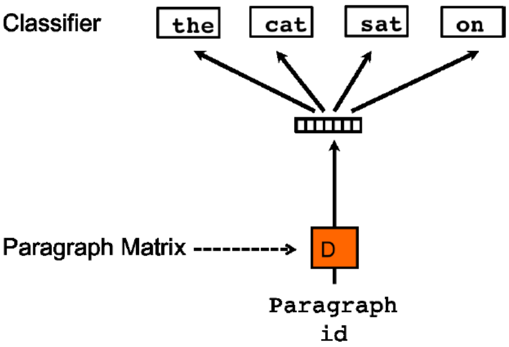


Figure 2: Distributed Bag of Words version of paragraph vectors

As shown in In this 2), this version of learning a paragraph vector trains the paragraph vector to predict the randomly sampled word (from the paragraph) in the output. It is similar to Skip-gram model in word vectors(Mikolov et al. (2013))

3 Approach

Our approach for assessing short answers in tutorial dialogue, first, generates doc2vec vector (Le and Mikolov, 2014) representation for student answer and all reference answers , then we compute cosine similarity score between them. Finally, we feed those scores as features input to multinomial logistic classifier to build the classifier to assess the quality of answer provided by the student(i.e. able to classify either the student answer is correct or incorrect or correct but incomplete, etc).

For each of the Student answer(A) and reference answers(R_1,R_2, ..., R_n), we generate both PV-DM and PV-DBOW vector representation. We also concatenate both PV-DM and PV-DBOW vector representation to form new concatenation vector. Thus, we have 3 vector representation, PV-DM, PV-DBOW and concatenation vector for each of student answers and reference answers. We then compute cosine similarity scores, cosSim(A, R_1), cosSim(A, R_2) , ..., cosSim(A, R_n) and form the following features:

- DM_max_cosSim: max of cosSim(A, R_1), cosSim(A, R_2) , ..., cosSim(A, R_n) with PV-DM vector representation .
- DM_mean_cosSim: mean of cosSim(A, R_1), cosSim(A, R_2) , ..., cosSim(A, R_n) with PV-DM vector representation .
- DBOW_max_cosSim: max of cosSim(A, R_1), cosSim(A, R_2) , ..., cosSim(A, R_n) with PV-DBOW vector representation .
- DBOW_mean_cosSim: mean of cosSim(A, R_1), cosSim(A, R_2) , ..., cosSim(A, R_n) with PV-DBOW vector representation .
- DM_DBOW_max_cosSim: max of cosSim(A, R_1), cosSim(A, R_2) , ..., cosSim(A, R_n) with concatenation vector representation .
- DM_DBOW_mean_cosSim: mean of cosSim(A, R_1), cosSim(A, R_2) , ..., cosSim(A, R_n) with concatenation vector representation .

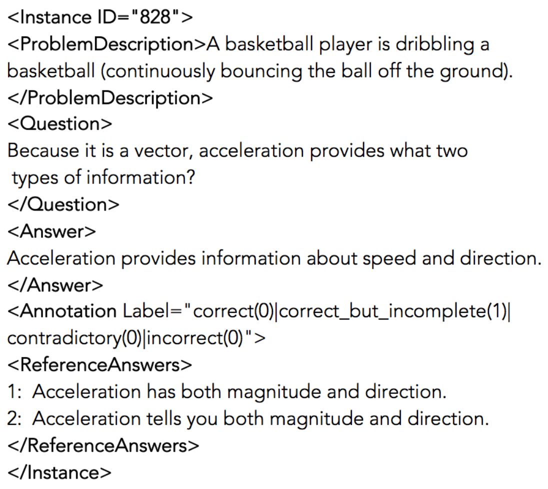
We build the different multinomial logistic classifier using different above features combination and compare their performance.

4 Experiment and Result

4.1 Dataset

To evaluate our approach, we use DT-Grade dataset (Banjade et al., 2016). The dataset contains 900 instances where each instance (see Figure 3) contained a) problem description (describes the scenario or context), b) tutor question, (c) student answer in its natural form (i.e., without any spelling and grammatical errors correction), (d) list of reference answers for the question (prepared by subject expert), e) annotation label of the student answer (annotated manually based on provided reference answers).

The annotation label can be either *correct* or *correct_but_incomplete* (answer is correct but



```
<Instance ID="828">
<ProblemDescription>A basketball player is dribbling a
basketball (continuously bouncing the ball off the ground).
</ProblemDescription>
<Question>
Because it is a vector, acceleration provides what two
 types of information?
</Question>
<Answer>
Acceleration provides information about speed and direction.
</Answer>
<Annotation Label="correct(0)|correct_but_incomplete(1)|
contradictory(0)|incorrect(0)">
<ReferenceAnswers>
1: Acceleration has both magnitude and direction.
2: Acceleration tells you both magnitude and direction.
</ReferenceAnswers>
</Instance>
```

Figure 3: Example of single instance of dataset

something is missing) or *incorrect* or *contradictory*(answer is very contrasting with reference answer). Label correct(1) means the instance is annotated as correct and all other annotation label must have 0 value; for example, the annotation of given instance (see Figure 3) is *correct_but_incomplete*.

4.2 Experimental procedure

Building model: We first learn both PV-DM and PV-DBOW doc2vec model using 7629 training paragraphs. The training paragraph consists of 900 problem description, 900, question, 900 answer and 4929 reference answer (an answer can have multiple reference answer). All paragraph available in 900 instances are used to build the model since it is unsupervised learning, i.e. the model takes the raw text input and makes no use of annotation information, and thus there is no need to hold separate test data, as it is unlabeled. Both of the models' parameters are set as dimensionality of feature vector size=300, initial learning rate $\alpha = 0.025$ and minimal learning $\alpha_{min} = 0.0001$. We choose window size of 8 and 5 for PM-DM and PM-DBOW respectively and train former with 1000 epochs and the later with 400. The choice of the parameter setting is largely influenced by the use of those setting by (Lau and Baldwin, 2016) in their work(similar in nature to our this experiment) as optimal setting, however, we choose smaller window size i.e. 5 since our answer paragraph are short and looking larger window size is not feasible intuitively.

Feature formation: We then obtain PV-DM and PV-DBOW vector representation, using respective model obtained above, as well concatenation of those two vector. We then compute their cosine similarity score for all student answer and reference answers and form score features: *DB_max_cosSim*, *DBOW_max_cosSim*, *DBOW_mean_cosSim*, *DM_DBOW_max_cosSim* and *DM_DBOW_mean_cosSim*.

Classifier: We build 7 multinomial logistic classifier all together; 6 of them has one distinct feature from our available features while the last classifier uses all features as input features. We build the classifier using WEKA logistic classier with default setting and use 10 fold cross validation.

Features used	Accuracy %
DM_max_cosSim	40.8889
DM_mean_cosSim	43.2222
DBOW_max_cosSim	40.8889
DBOW_mean_cosSim	40.8889
DM_DBOW_max_cosSim	40.8889
DM_DBOW_mean_cosSim	40.8889
All above 6 features	42.1111

Table 1: Accuracy using different feature combinations

4.3 Results

The result of the experiment using different combinations of features is reported in the Table 1. As we can see in the table, the highest level of accuracy is 43.22% obtained using feature *DM_mean_cosSim* alone. We obtain second highest level of accuracy as 42.11% using combination of all features, which is 1 percent lower than the obtained highest level of accuracy. Other features *DB_max_cosSim*, *DBOW_max_cosSim*, *DBOW_mean_cosSim*, *DM_DBOW_max_cosSim* and *DM_DBOW_mean_cosSim*, each of these features used alone, gives 40.88% accuracy.

5 Conclusion

We presented new and more general approach of answer assessment using doc2vec vector representation of answer and reference answers. We then evaluated our approach using DT-Grade dataset (Banjade et al., 2016) and obtained highest accuracy of 43.22 %. Our experiment shows that PV-DM for paragraph vector is alone more efficient than using concatenations of the PV-DM and PV-

DBOW or PV-DBOW. In the future, using of features that captures contradictory answers, answer similar to correct answer but just few words make them opposite in meaning, for example, the bus moves and the bus do not moves, will be interesting to explore and see their effect.

References

Rajendra Banjade, Nabin Maharjan, Nobal Bikram Niraula, Dipesh Gautam, Borhan Samei, and Vasile Rus. 2016. Evaluation dataset (dt-grade) and word weighting approach towards constructed short answers assessment in tutorial dialogue context. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 182–187.

Martha W Evens, Ru-Charn Chang, Yoon Hee Lee, Leem Seop Shim, Chong Woo Woo, Yuemei Zhang, Joel A Michael, and Allen A Rovick. 1997. Circsim-tutor: An intelligent tutoring system using natural language dialogue. In *Proceedings of the fifth conference on Applied natural language processing: Descriptions of system demonstrations and videos*, pages 13–14. Association for Computational Linguistics.

Arthur C Graesser, Shulan Lu, George Tanner Jackson, Heather Hite Mitchell, Mathew Ventura, Andrew Olney, and Max M Louwerse. 2004. Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 36(2):180–192.

H Chad Lane and Kurt VanLehn. 2004. A dialogue-based tutoring system for beginning programming. In *FLAIRS Conference*, pages 449–454.

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Vasile Rus, Dan Stefanescu, Nobal Niraula, and Arthur C Graesser. 2014. Deeptutor: towards macro-and micro-adaptive conversational intelligent tutoring at scale. In *Proceedings of the first ACM conference on Learning@ scale conference*, pages 209–210. ACM.