EVALUATION OF SUBJECTIVE ANSWERS USING GLSA ENHANCED WITH CONTEXTUAL SYNONYMY

Parag A. Guruji[1], Mrunal M. Pagnis[2], Sayali M. Pawar[3] and Prakash J. Kulkarni[4]

Department of Computer Science and Engineering,
Walchand College of Engineering, Sangli, India 416415

ABSTRACT

Evaluation of subjective answers submitted in an exam is an essential but one of the most resource consuming educational activity. This paper details experiments conducted under our project to build a software that evaluates the subjective answers of informative nature in a given knowledge domain. The paper first summarizes the techniques such as Generalized Latent Semantic Analysis (GLSA) and Cosine Similarity that provide basis for the proposed model. The further sections point out the areas of improvement in the previous work and describe our approach towards the solutions of the same. We then discuss the implementation details of the project followed by the findings that show the improvements achieved. Our approach focuses on comprehending the various forms of expressing same entity and thereby capturing the subjectivity of text into objective parameters. The model is tested by evaluating answers submitted by 61 students of Third Year B. Tech. CSE class of Walchand College of Engineering Sangli in a test on Database Engineering.

KEYWORDS

N-gram, Synonym-Set, natural language processing, GLSA, SVD, Wordnet

1.INTRODUCTION

The progress in computer network technologies and increased inclination towards making human interactions „paperless", online test systems are being adopted widely. Automation of the assessment activity comes to a great help in saving resources (time, efforts & cost) and in increasing the consistency of the assessment by mitigating potential human biases of intangible nature. Realization of Computer Aided Assessment (CAA) for objective types of answers is a well addressed deterministic problem which involves „matching" of submitted answers to the corresponding predefined expected answers. Realization of the CAA in case of descriptive answers becomes comparatively more challenging by virtue of the subjectivity of the answers. It involves „mapping" of underlying semantics of submitted answers to that of the expected answers. Pattern recognition, artificial cognition, and natural language processing are required in such process of CAA.

Significant work has been done in this field. Project Essay Grader (PEG) was developed by Ellis Page in 1966. PEG ignored the semantic aspect of essays and relied on the surface structures, for which it had been criticized [1, 2]. Thomas K. Landauer, Boulder & Peter W. Foltz developed

DOI : 10.5121/ijnlc.2015.4105

Latent Semantics Analysis (LSA) [1, 2] – the technique widely used in automated assessment. Group lead by Jill Burstein designed E-rater that deals with automatedessay scoring of GMAT [3]. Automated Text Maker (ATM) is another computer aided assessment system and can be adopted to address different disciplines [4]. Md. Monjurul Islam and A. S. M. Latiful Hoque came up with Automated Essay Scoring Using Generalized Latent Semantic Analysis in 2010 [5] and also applied GLSA for scoring essays in Bangla Language in 2013 [6]. M. S. Devi and Himani Mittal applied the LSA Technique to evaluate technical answers in 2013 [7] wherein they have used single word as atomic unit for computations and suggested in future work that use of synonymy would enhance the results.

In this paper, automation of assessment of technical answers written by undergraduate students is further enhanced by modifying GLSA based techniques. Section II outlines the fundamental model and describes fundamental techniques used. In Section III, we discuss the enhancements made to the basic model. It includes the use of N-grams as atomic computation unit, capturing context based synonymy, the concept of Synonym-Sets and change in document representation and the N-gram formation over Synonym-Sets which enables representation of many by one.

2. THE FUNDAMENTAL MODEL

The existing model of the system for evaluating subjective answers that uses Generalized Latent Semantic Analysis (GLSA) is discussed in this section. We have referred to it as the fundamental model.

2.1. GLSA over LSA:

Normally LSA represents documents and their word content in a large two-dimensional matrix semantic space. Using a matrix algebra technique known as Singular Value Decomposition (SVD), new relationships between words and documents are uncovered, and existing relationship are modified to more accurately represent their true significance.[1] Each word represents a row in the matrix, while each column represents the sentences, paragraphs and other subdivisions of the context in which the word occurs. The traditional word by document matrix creation of LSA does not consider word sequence in a document [2]. Here the formation of word by document matrix the word pair “concurrent transactions” makes the same result of “transactions concurrent”. Thus, LSA fails to capture the sematic effect of collocations in the document.

GLSA addresses this issue by considering n-gram as atomic unit of the document instead of individual word. An n-gram is a contiguous sequence of n words from a given sequence of text. Thus now, “concurrent transactions” is not recognized same as “transactions concurrent”.

2.2. System Architecture Overview:

The system works in two phases – training set generation and testing of newly submitted answers. Pre-processing and frequency computation operations to be done on an individual document are common in both the phases.

2.3. Pre-processing:

The pre-processing is comprised of stopwords removal, word-stemming and lingual error handling in the fundamental model. We have decided to eliminate the step of word-stemming because we found that dealing with contextual synonymy (discussed in section III) with the stemmed words as inputs introduces too much of noise and thus inaccuracies.

The pre-processing thus now consists of two steps - stopwords removal and lingual errors handling.

2.4. Phase 1: Training set generation:

The input for this phase is set of pre-graded answers as per earlier model. We include the question-domain in the input set for the purpose discussed in section III. Output of this phase is the training-set.

After pre-processing every pre-graded answer – a training document, n-grams created from all the training documents and their respective frequencies for corresponding documents are computed to form the n-gram-by-document vectors. Each such vector represents one training document. The n-gram-by-document matrix is formed by putting together all such vectors as columns of the matrix. Every row of the matrix represents an n-gram and every column represents a training-document. Every cell of this matrix holds the frequency value of the n-gram corresponding to its row-index in the training document that corresponds to its column index. The n-gram-by-document matrix then undergoes Singular Value Decomposition (SVD) to generate the singular value matrices. According to SVD a matrix At×n is decomposed as follows:

$$A_{t\times n} = U_{t\times n} \times S_{n\times n} \times V_{t\times n}^{T} \quad \ldots(1)$$

Here, A is n-gram by documents matrix, U is an orthogonal matrix, S is diagonal matrix and VT is the transpose of an orthogonal matrix V. After applying dimensionality reduction on matrices U, S and V we get reduced singular value matrices that are used to generate training vectors. The dimensionality reduction eliminates one or more least significant singular values in order to reduce the noise involved in the sematic space developed in the form of SVD matrices. Training vector corresponding to every pre-graded answer is then formed as per the following equation:

$$d_j^{"} = d_j^{T} \times U_{t\times k} \times S_{k\times k}^{-1} \quad \ldots(2)$$

Here, d_j^T is the transpose of document vector d_j, $U_{t\times k}$ is truncated left orthogonal matrix and $S_{k\times k}$ is truncated singular matrix from truncated SVD matrices. The training vectors d_j' along with human grades of pre-graded answers makes the training set.

Phase 2: Evaluation of submitted answers:

The input for this phase is set of submitted answers for corresponding question which will be evaluated by the system. For the purpose of testing the system, these answers are also evaluated

by the human graders. Output of this phase is the machine-generated grades for the submitted answers/(s).

Submitted answer first undergoes pre-processing. Frequency values for all the n-grams (generated in training phase) for this document are computed to form the n-gram-by-document vector q of the submitted answer. This vector is used to generate the query vector for submitted answer as follows:

$$q" = q \times U_{t\times k} \times S_{k\times k}^{-1} \quad \ldots(3)$$

Here, q is query matrix $U_{t\times k}$ is truncated left orthogonal matrix and $S_{k\times k}$ is truncated singular matrix generated from SVD in training phase. Similarity between query vector q' and training set vectors d_j' is calculated by Cosine similarity [1]

$$Similarity(q', dj') = \frac{\sum_{j=1}^{t} w_{qj} \mathbf{x} d_{ij}}{\sqrt{\sum_{j=1}^{t}(d_{ij})^2 \mathbf{x} \sum_{j=1}^{t}(w_{ij})^2}} \quad \ldots(4)$$

Here, w_{qj} is the j^{th} weight of query vector q' and d_{ij} is the i^{th} weight of training essay set vectors $d_j"$. The highest correlation value from the Cosine of query vector and the training essay vector is used for grading the submitted essay.

3. THE ENHANCEMENTS

In the n-gram by document matrix, each row represents an n-gram and each column represents a document. Every cell contains a value that represents the frequency of n-gram associated with the row index of that cell in the document associated with the column index of the same cell. We use the Term Frequency/Inverse Document Frequency (TFIDF) notation to compute this frequency value.

Now, to capture the significance of a given n-gram, the system depends on the frequency value associated with it. The subjectivity of the natural language text has an inherent feature which is generally known as "saying same thing in different words". For instance, while talking about networks, somebody may refer to a term, say, „connected nodes'; whereas, in some other text, same entity might have been referred to as „joint peers" or „connected peers" or „joint nodes". Now, all these four expressions represent same entity, given the same context – networks. But, in the existing model that uses GLSA, all of them are treated as independent bigrams and thus, their frequencies are computed separately. This introduces inaccuracy in the computed significance of the given entity if the same entity is expressed in different forms at its different occurrences in the text. We propose a solution to this problem with the use of Synonym-Set.

3.1. Concept of Synonym-Set & Document redefinition:

For a given n-gram, say, $N = (W_1\ W_2\ \ldots\ W_i\ \ldots\ W_n)_c$ which represents certain entity, say E in a given context C, if $W_{11}, W_{12}, \ldots, W_{1S1}$ are synonyms representing W1 then, we define Synonym-Set S_{ic} as

$$S_{ic} = (W_i, \{W_{i1}, W_{i2}, \ldots, W_{iSi}\})$$

The document d is defined as the sequence of words $(W_1, W_2, \ldots, W_j, \ldots W_n)$. We redefine d as

$$d = (S_{1c}, S_{2c}, \ldots S_{jc}, \ldots, S_{nc})$$

Every Synonym-Set Sic is associated with a unique integer say I_{ic}. Therefore the document d is represented as

$$d = (I_{1c}, I_{2c}, \ldots I_{jc}, \ldots, I_{nc})$$

Now, n-grams are formed over this redefined representation of the document. One such n-gram now represents a set of all such permutations of expressing the entity represented by that n-gram which are valid in the given context. In the example stated previously,

C = ‘computer networks’,
S_{1c} = (I_{1c}, {‘connected’, ‘joint’}),
S_{2c} = (I_{2c}, {‘node’, ‘peer’}),
N_{old} = { ‘connected node’, ‘connected peer’, ‘joint node’, ‘joint peer’}
N_{new} = (I_{1c} I_{2c})

Thus, a single new n-gram now represents all P permutations of forms in which E can be expressed in the subjective text in context C, where P is given by:

$$P = S_1 \times S_2 \times \ldots \times S_i \times \ldots \times S_n$$

Now, occurrence of any of the P forms of expressing E will be identified by the system as single entity represented by N_{new}. Hence, while computing the significance of any n-gram (and in turn the entity which it represents) using TFIDF, the weightage of the entity is now converged and put together under single unique entity unlike earlier models where it was getting scattered among P ¬ways of expressing E. This increases the precision of the system and deals with an inherent feature of subjectivity – ‘saying same thing in different words’.

3.2. Contextual Synonymy:

In the solution discussed above, there lies an assumption that a synonym-set will consist of words having same semantics in a given context, i.e. every word in a given synonym-set shares a relation of contextual synonymy with the key-word of that synonym-set.

The number of false positive n-grams representing different ways of expressing same entity is thus inversely proportional to the accuracy of formation of the contextual synonym-sets. For instance, word ‘node’ is related also to the word ‘knot’ by the relation of synonymy. But, in given

context of ‘computer networks’, ‘knot’ does not mean ‘node’. Hence, ‘joint knots’ cannot be a valid form of representing the entity ‘connected nodes’. So, the implicit challenge in making the concept of synonym-set and document redefinition functional is to device a process that accurately forms the contextual synonym-set for every word in the pre-processed training text.

Our method to prepare such synonym-sets uses the database provided by the Wordnet. Along with inter-relations between words such as synonymy, hyponym, etc., the Wordnet also provides definitions of every word. We observed that the strings of context-specific definitions in Wordnet show certain pattern wherein the context or the domain of the definition is specified at the beginning of the definition in parentheses.

We first fetch all the synonyms of the word under consideration. Then, to filter out only contextual synonyms, we make use of the aforesaid pattern in the Wordnet definitions and apply string matching techniques along with the context of the question to which the subjective answers are written to identify the contexts or the domains of the words which are initially fetched as synonyms of the key-word.

The presence of domain-specific definition in Wordnet for a given context also serves as a qualifying criterion for any word to be identified as a terminological, and hence significant, word in the context of the question. Since, a subjective text of informative nature which talks about certain domain holds the major part of its semantics in the form of terminological entities, the process discussed above proves to be effective in dealing with informative answers pertaining to an identified domain.

3.3. Feedback-driven incremental advancement:

We attempt to incorporate an aspect of human behavior into our model by using a feedback system. The objective of this feedback system is to make the system avoid same mistakes twice. The system can make mistake in grading an answer which has content semantically or syntactically ‘unknown’ to the system, i.e., it contains large number of new entities which were not part of the training set at all. Such cases are identified as outliers and the grades assigned to them are forwarded for human moderation. Identifying potential outliers is important for the System’s reliability. To identify outliers from a given set of submitted answers which is normally distributed in terms of probability of presence of outliers, we use the characteristic of an outlier by virtue of which, even the highest amongst the cosine correlations of the query vector of outlier with the training vectors has comparatively low value than the highest of cosine correlations of an average case test answer (graded with acceptable reliability) with the training vectors. Using this property, we identify an outlier as the answer in case of which, the highest cosine correlation of the query vector with the training vectors is below the average of such highest cosine correlations of query vectors of all answers in the given set.

We feedback every such outlier along with its human-moderated grade to the system for its inclusion in the training set. This leads to addition of new n-grams and re-computation of the SVD matrices which in turn modifies the training set.

Figure1 shows the overview of our system model.

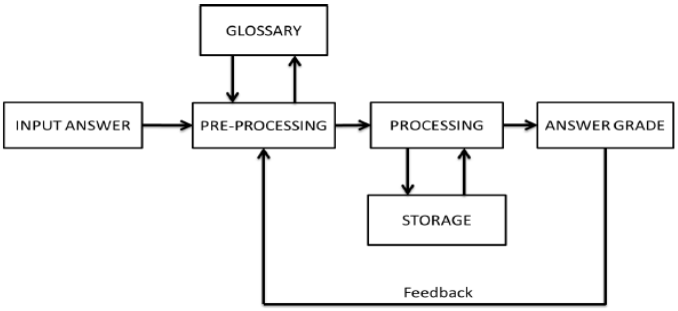

Figure 1. System overview

4. EXPERIMENT AND RESULTS

We have experimented on our model by conducting an online test on Database Engineering course for a class of 61 students in our departmental lab. We collected the human grades for each answer from the concerned faculty. Further, 30 answers were used for training the system and rest for testing. Following (Fig. 2) is the graph comparing the machine generated grades and corresponding average human grades for each test answer. Three human graders have graded the answer and we have used the average of their scores. These results are obtained for value of n=1,2,3.

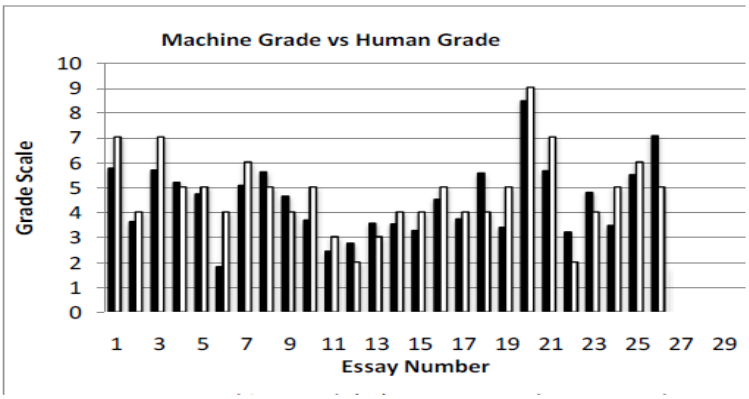

Figure 2 Graph: Machine grade Vs Human grade

Table 1. Variation in Training Time, Testing Times and Standard Deviation with various combinations of n-grams

N	T1	T2	n-grams count	Standard Deviation
2	4.76	15	1095	1.6269
3	10.14	19	1503	1.5565
2,3	15.29	25	2598	1.4891
1,2,3	20.09	33	3476	0.8697

T_1 : Training time in minutes with 30 answers
T_2 : Testing time in minutes with 30 answers

Table 1 shows the comparison of various values of n and the time taken for training 30 answers, testing answers, number of n-grams and the standard deviation.

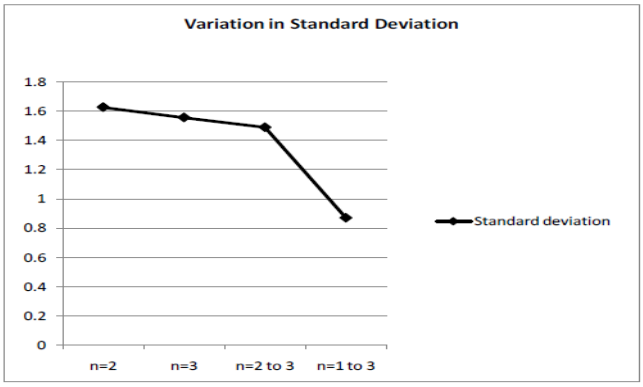

Figure 3 Graph: Variation in Standard Deviation

Graph in Figure 3 shows that when we increase the value of n in n-gram we get improved results. The standard deviation approached zero as we increased n. However, we observed that if we try to improve the results with 4-grams then it has a reverse effect. This is because average number of words in a sentence (for third year undergraduate students) is about 6-7 after stop word removal. So when we have 4-grams then we are actually trying to find phrases as long as four words. This introduces noise in the result. This is the major reason for limiting the value of n to 3

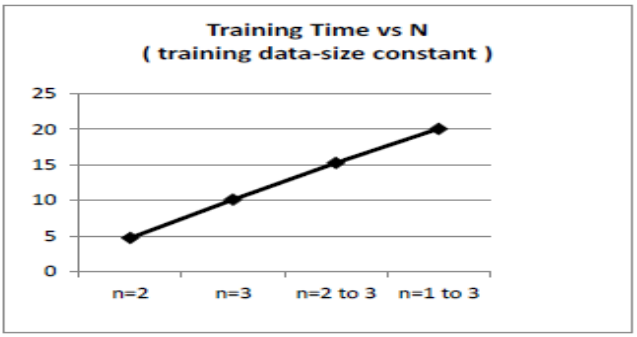

Figure 4 Graph: Variation in Training Time with N

Graph in Figure 4 shows that the training time increases with number of n-grams, which is dependent on the number of training documents. Thus, we can say that the number of training documents determines the time taken for training.

As a part of reliability measure we compute the error with respect to human grades. When error goes beyond a threshold we add the outliers to SVD and recomputed the matrix to improve the results.

5. CONCLUSIONS

We have developed a system that focuses on the concept of synonyms while it grades an answer. This enhancement has given significant results in comparisons with the recently proposed models that use latent semantic analysis without incorporating synonyms. Section 2 is the basic model that currently is considered successful for grading answers. Section 3 introduces a new aspect due to which the model can perform better. Implementation and results have proved that the enhancement costs negligible resources in terms of time and space. The improvement in performance definitely outweighs the negligible costs. Section 4 shows the experimental results taken on collected data. The feedback driven model improves the results every time we train the system. This regulates error periodically and also improves the results after each time we train.

ACKNOWLEDGEMENTS

The authors would like to thank firstly the Department of Computer Science and Engineering, Walchand College of Engineering, Sangli for facilitating the experimentation by providing required infrastructure. The authors also thank in particular the esteemed faculty members of Dept. of CSE, WCE, Sangli Mrs. Hetal Gandhi and Dr. Smriti Bhandari, for their valuable assistance in human expert grading of test answers used in training and testing of the system. Lastly the authors thank all the students of 2013-14‟s Third Year B.Tech. CSE class of WCE, Sangli who participated in the experiments as test takers. Without their sincere voluntary participation, the experiments could not have proceeded.

REFERENCES

[1] Md. Monjurul Islam and A. S. M. Latiful Hoque, "Automated Essay Scoring Using Generalized Latent Semantic Analysis," in Proceedings of 13th International Conference on Computer and Information Technology (ICCIT 2010), 23-25 December, Dhaka, Bangladesh

[2] M. Syamala Devi and Himani Mittal, "Subjective Evaluation using LSA Technique," in International Journal of Computers and Distributed Systems Vol. No.3, Issue I, April-May 2013 ISSN: 2278-5183 (www.ijcdsonline.com)

[3] S. Valenti, F. Neri, and A. Cucchiarelli, "An overview of current research on automated essay grading," in Journal of Information Technology Education, vol. 2, 2003, pp. 319-330

[4] Md. Monjurul Islam and A. S. M. Latiful Hoque, "Automated Bangla Essay Scoring System: ABESS," in 978-1-4799-0400-6/13/$31.00 ©2013 IEEE

[5] "Java API for WordNet Searching (JAWS)" [Online] Available: http://lyle.smu.edu/~tspell/jaws/index.html

[6] Mark Watson, "Practical Artificial Intelligence Programming with Java", Ed. 3, 11/11/2008 [Online] Available: http://archive.org/stream/PracticalArtificialIntelligenceProgrammingWithJava/JavaAI3rd_djvu.txt

[7] JAMA: A Java Matrix Package [Online]. Available: http://math.nist.gov/javanumerics/jama/

Authors

Parag A. Guruji
Data Scientist at Zlemma Analytics Inc, Pune, India.
Former student of Walchand College of Engineering, Sangli, India.
Received Bachelor of Technology Degree in Computer Science and Engineering in May 2014.
Team Member of the project discussed in this paper.

Mrunal M. Pagnis
Graduate Student at Indiana University at Bloomington, Indiana, USA.
Former student of Walchand College of Engineering, Sangli, India.
Received Bachelor of Technology Degree in Computer Science and Engineering in May 2014.
Team Member of the project discussed in this paper.

Sayali M. Pawar
Graduate Engineer Trainee at Tata Consultancy Services, Pune, India.
Former student of Walchand College of Engineering, Sangli, India.
Received Bachelor of Technology Degree in Computer Science and Engineering in May 2014.
Team Member of the project discussed in this paper.

Dr. Prakash J. Kulkarni
Professor in Computer Science and Engineering and Deputy Director at Walchand College of Engineering, Sangli, India.
Faculty Mentor of the project discussed in this paper.