An Unsupervised Approach to Automated Selection of Good Essays

Arijit De
TCS Innovation Labs, Mumbai
Tata Consultancy Services
Mumbai, India
arijit6.d @tcs.com

Sunilkumar Kopparapu
TCS Innovation Labs, Mumbai
Tata Consultancy Services
Mumbai, India
sunilkumar.kopparapu@tcs.com

***Abstract*— Evaluating essays automatically has been an area of active research for some time. In this paper, we propose an unsupervised technique to select good essays from a corpus of essays based on the statistical content of essays that make up the corpus. The approach is unsupervised as it uses intrinsic properties of the corpus itself to select good essays. Our approach is a 'bag of words' approach, not requiring any computationally intensive deep parsing. We evaluate our approach on a data set of essays submitted to a competition internally within our organization.**

Keywords-Natural Language Processing, Information Retrieval, Kullback–Leibler divergence.

I. INTRODUCTION

Essays have become an integral component of standardized tests such as the GRE, GMAT and TOEFL. Faced with the enormity of evaluating hundreds and thousands of essays, standardized testing agencies such as ETS, have turned to scientists working in the area of Natural Language Processing (NLP) to develop algorithms that can evaluate essays with minimal to no human effort. Challenges to automated grading/scoring of essays are enormous. To put it in simple words, every author of an essay has a different writing style, varying command over language and vocabulary, different choice of words. Writing styles at a more intrinsic level are hard to quantify as good of bad. That makes it practically impossible to identify one particular writing style as better than another. Even when human experts evaluate essays, their grading or selection often vary considerably. On top of that, computer algorithms can not read digitalized text and recognize writing styles, grammatical errors etc.

In this paper, we take a first step towards unsupervised automated ranking of essays by proposing an algorithm for selecting good essays from a specified corpus of essays. Our paper is organized as follows. In section II we delve into some related work in this area. In section III we define the problem of selecting ‘good’ essays from a corpus and then describe our proposed approach to essay selection. In section IV we describe our experiments and results and conclude our discussion in section V.

II. PREVIOUS WORK

Automated rating of essays has been a topic of much research amongst scientists working in the area of Natural Language Processing. Attali et.al. [1] proposes a system e-rater which has been used experimentally by ETS in the past decade or so. The e-rater engine assigns scores to essays based on grammatical correctness, quality of writing, usage and choice of words, organization of material content, and writing style. NLP is used to identify and extract linguistic features from stored digitalized essays. While the engine's score prediction has correlated well with expert ratings, the main disadvantage with e-rater is that it utilizes a complex model built from a vast training set of essays which have already been rated.

Chen et.al. [3] creates a system for unsupervised ranking of essays but essentially uses a voting algorithm to agglomerate rating from various human reviewers. Kakkonen’s [4] PLSA based automated scoring system compares pre rated essays with text book passages, then the to-be-rated essays are compared with the same text book passages to create a similarity function that can be used for evaluations. While this does not involve computationally complex model building, it still needs both human input as well as a set of text book passages. This also makes the system, topic specific. Larkey [6] does not essentially rate essays but categorizes them using k-nearest neighborhood and Bayesian classifiers.

III. PROPOSED APPROACH

Most systems for automated rating of essays are supervised and/or require some kind of textual references. Very few of them employ linguistic features and deep parse natural language text. However, in most real world situations pre rated text passages are few or not available at all. Under such conditions it becomes difficult to build complex models required by systems such as e-rater [1] or Kakkonen’s [6]. Even in the age of advanced multi-processor technology, deep natural language analysis and parsing still remains a computationally intensive task. This makes deep parsing on a large scale very expensive and time consuming.

This clearly calls for a system for unsupervised essay rating. The primary motivation for our work was to take a first step at creating a system which could be able to select good essays from a collection without any external stimulus or computationally intensive deep linguistic parsing.

Our approach to selecting good essays from a corpus is based on "bag of words" and statistical in nature and employs the Kulback-Liebler [5] distance as a metric for computing distance between documents. KL divergence is often used in text mining application to compute document similarities. Bigi [2] uses KL divergence for text categorization and Tsagkias [7] shows how the KL-distance for two documents can be computed.

A. Problem Definition

Let us define the problem of selecting good essays from a corpus of essays. Let us say we have a corpus of n essays, $C = \{E_i, 1 \le i \le n\}$. Here E_i is the i^{th} essay in the corpus. Let us say we want the m *best* essays from this corpus. The problem is to select the m essays as a set $G = \{E_i\}$ and $|G| = m$.

B. Document Segmentation and Feature Selection

The first step in the proposed approach calls for segmenting the digitized essay into paragraphs and then further segmenting paragraphs into sentences. A set of features (F) are extracted from sentences and paragraphs. Let the feature set $F = \{F_p, F_s\}$. Here features associated with a paragraph (F_p) and features associated with the sentence (F_s). The features extracted from the essay are listed as follows:

- F_p: These are features associated with a paragraph. These include (1) Number of sentences per paragraph (2) Number of words per paragraph, (3) Number of nouns per paragraph, (4) Number of verbs per paragraph and (5) Number of adjectives per paragraph)
- F_s: These are features extracted from sentences. These include: (1) Number of words per sentence, (2) Number of nouns per sentence, (3) Number of verbs per sentence, (4) Number of adjectives per sentence.

C. Creating Probability Density Functions for Each Feature

For each feature in F_p and F_s we create a probability density function (PDF), Φ. Algorithm 1 shows how Φ can be calculated for a feature in F_p. Similarly the PDF of features in F_s can also be calculated. For each essay $E_i \varepsilon C$ and a feature $f \varepsilon F_p$ or F_s we can obtain a probability density function $\Phi(E_i, f)$.

D. Computing Kullback–Leibler divergence

In our methodology we use the Kullback-Liebler (KL) divergence between two probability density functions as a metric to compute the distances between two documents. Let E_i and E_j be two documents with PDF functions as Φ_1 and Φ_2 respectively. Equation (1) shows the KL divergence of the two distributions (E_i, E_j). Note that the KL divergence of document pair (E_j, E_i) is different as shown in equation (2).

$$D_{KL}^{ij} = \sum \varphi_i \log\left(\frac{\varphi_i}{\varphi_j}\right) \quad (1)$$

$$D_{KL}^{ji} = \sum \varphi_j \log\left(\frac{\varphi_j}{\varphi_i}\right) \quad (2)$$

From Equation (1) and (2) it is clear that the commutative property does not hold for KL Divergence. Tsagkias [7] shows how to obtain the KL distance from the KL Divergence. Equation (3) & (4) shows the KL Distance between two documents (E_i, E_j).

$$d_{ij} = D_{KL}^{ij} + D_{KL}^{ji} \quad (3)$$

$$d_{ij} = \sum (\varphi_i - \varphi_j) \bullet \log\left(\frac{\varphi_i}{\varphi_j}\right) \quad (4)$$

Algorithm 1: Constructing probability density function Φ for a feature in F_p.

Input:
An essay E_i
A feature $f \varepsilon F_p$

Output:
Φ: probability density function array

Algorithm:
Identify all paragraphs P in E_i
W: frequency counts array
R: raw values list
TOT: total count of f in all P
for i = 1 to P do
    k ← count(f) in P
    Add k to R
    Increment (W[k])
    TOT ← TOT +k
end for
for each $k \varepsilon R$
    Φ[k] ← W[k]/TOT
End

E. Creating a Corpus Document

We merge all documents in the corpus to create a corpus essay E_C. Thus this corpus document is a union of all documents within the corpus as show in equation (5)

$$E_C = \bigcup_{i=1}^{n} E_i \quad (5)$$

We can compute a PDF $\Phi(E_C, f)$ for all features $f \varepsilon F$. We then compute the KL distance $D_{KL}(i, f)$ between each document $E_i \varepsilon C$ with the corpus document E_C for each feature $f \varepsilon F$. If one particular feature is to be used to select m best essays, then there is no aggregation required as mentioned in the next section. However aggregation is necessary when multiple features need be utilized. We use Yager's [9, 10] Fuzzy Ordered Weighted Average (OWA) operator for aggregation of KL Distances for the whole set of features in F. Actual aggregation is discussed in detail in section G.

F. Selecting Essays based on a Feature

Essays can be selected based on one specific feature or on all features as mentioned above. In the first case, for each essay E_i in the corpus C we have a KL distance $D_{KL}(i, f)$. Here the

distance is for the i^{th} document and feature f. We sort these values in descending order of distance from the corpus essay E_c and select the best m essays. Thus the essay with the least distance from E_c is ranked at the top and so on.

G. Aggergration of KL Distances

For each essay $E_i \varepsilon C$ the KL distance $D_{KL}(i , f)$ for short a feature $f \varepsilon F$. So for each essay $E_i \varepsilon C$ we have a set of KL distances $\{D_{KL}(i , f_1), D_{KL}(i , f_2), D_{KL}(i , f_3),\ldots.. , D_{KL}(i , f_p)\}$ where $p = | F |$ or $|F_p + F_s|$. Let $D_{KL}(i , f_j)$ be represented as d_k for all $f_k \varepsilon F$. So the set of distances for E_i can be written as $D_i = \{d_1, d_2, , ,d_p\}$ with p as defined above. The problem of aggregation is a problem of Multi-Criteria Decision Making (MCDM). Yager's [9, 10] MCDM OWA operator is a good aggregation operator in MCDM problems.

To explain the OWA operator let us take a multi-criteria decision making approach. Assume $A_1(x), A_2(x)\ldots.. A_r(x)$ is r criteria of concern in a multi-criteria problem. Let x be some proposed alternative. Then $A_j(x)$, where $A_j(x) \varepsilon [0,1]$ indicates the degree to which x satisfies the jth criteria. Yager [9] comes up with a decision function FOWA by means of which we can combine these criteria and evaluate the degree to which the alternative x satisfies the criteria. Let $a_1= A_1(x)$, $a_2= A_2(x)$, and $a_n=A_n(x)$.The OWA decision function is as shown in equation (6). Here b_j is the j^{th} greatest a_i. Here w_j is the ordered weight vector attached to the j^{th} criteria and such that the sum of the ordered weights is always unity as shown in equation (7).

$$F_{OWA}(a_1, a_2, a_3, \ldots, a_r) = \sum_{j=1}^{r} w_j b_j \quad (6)$$

$$\sum_{j=1}^{r} w_j = 1 \quad (7)$$

The ordered weight vector $W = [w_1, w_2, \ldots.. w_n]$ associated with the OWA operator is key to determining the "orness" of the aggregation. When $W = [1, 0, \ldots.. 0]$ the value of the largest a_i is the value of F_{OWA}. In this case we get maximum orness. On the other hand when $W = [0, 0, \ldots.. 1]$ the value of the smallest a_i is the value of F_{OWA}. In this case we get minimum orness/maximum andness. Yager proceeds to compute OWA weights using linguistic quantifiers. The weight associated with the i^{th} criterion (positional value associated with a search engine) is as shown in equation (8). Here, Q is a RIM (Regular Increasing Monotone) quantifier of the form $Q(x) = x^{\alpha}$. Yager computes the orness associated with the quantifier, orness(Q) as shown in equation (9). It should be noted that when α varies from 0.5 to 2 orness of the aggregation increases. When $\alpha = 1$ orness of aggregation is balanced and it is in effect the same as averaging the values.

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right) \quad (8)$$

$$orness\ (Q) = \frac{1}{1+\alpha} \quad (9)$$

For each essay E_i in corpus C we have a set of distances D_i which represent the distances between the document and the corpus document E_c. Using the KL distance values in the set D_i as arguments to the function F_{OWA} and using a RIM quantifier of the form x^{α} we obtain ordered weights using equation (8). We vary the value of α from 0.5 (high orness) to 2.0 (low orness). The value of function F_{OWA} so computed, is the composite distance dc_i for the E_i .

H. Selecting Essays

To select essays and obtain the m good essays we sort the essays in ascending order and select the m top essays. Essays that are ranked by the least composite distance from the corpus document E_c are ranked at the top.

IV. Experiments

A. Data Sets Used

We tested our essay selection methodology a dataset consisting of 19 essays written on a technical topic 'Next Generation Mobile Phones' that were part of a technical essay writing competition within our organization. These essays had been evaluated by a set of experts and the best 10 documents had been selected. The objective of our experiments was to see if we could re-create the best 10 documents by using our methodology.

B. Metric

Precision and recall are two widely used metrics for evaluating the correctness of an Information Retrieval algorithm. Baeza-Yates [10] defines Precision as the number of relevant documents retrieved by an IR algorithm to the total number of relevant documents available. In our case relevant documents are best 10 documents. In our case the Precision metric is a measure of system accuracy. It is the ratio of number of essays common between the human judged essays and the automated methodology created by us, and the total number of essays in the human judged set.

C. Results

In our experiments, we first work with individual features to observe the accuracy of essay selection based on that particular feature. The results for experiments are shown in Table 1 below. Clearly for features, nouns per sentence and adjectives per sentence, the precision is the highest at 0.7. This essentially means that using the feature nouns per sentence and Adjectives per sentence we are able to correctly identify the best ten documents with an accuracy of 70%. Surprisingly sentences per paragraph feature did poorly with a lower precision at 0.3. This means that the accuracy was lower at 30%. This essentially means that in ranking essays, number of sentences per paragraph is a less relevant feature as opposed to nouns per sentence or adjectives per sentence. Overall sentence based features F_s did better than paragraph based features F_p as the average precision of the F_s group was 0.65 and the average precision of F_p group was 0.5. This means that sentence based features are more significant in measuring the quality of a document.

Table II shows the effect of aggregating KL Distances and then selecting essays using the OWA operator. Since the OWA

operator is a Fuzzy operator we vary the orness of aggregation. Average precision is high under high orness and low orness conditions. There is a drop-off in performance as we hover around close to averaging condition. Selecting through aggregation of KL distances does as well as or better than all but two features, nouns per sentence and adjectives per sentence.

TABLE I. RESULTS FOR DATA SET 1

Feature	Precision
noun per sentence	0.7
verb per sentence	0.6
adjectives per sentence	0.7
word per sentence	0.6
nouns per paragraph	0.6
verb per paragraph	0.5
adjectives per paragraph	0.6
word per paragraph	0.5
sentences per paragraph	0.3

TABLE II. RESULTS FROM AGGREGATION

Alpha	Orness	Average Precision
0.25	0.80	0.60
0.50	0.67	0.60
1.00	0.50	0.50
1.50	0.40	0.50
2.00	0.33	0.50
2.50	0.29	0.60
3.00	0.25	0.60

V. CONCLUSION

In this paper we take a first step at automated ranking of essays by proposing a method for selecting good essays from a corpus by using intrinsic properties of the corpus to compute the distance between each essay and the overall corpus. We use nine parts-of-speech based features on which the distance between each essay is compared with the overall corpus. We also aggregate distances from various features to create a composite distance. This composite distance can also be used to select good documents. We test our method against a corpus of documents from which ten best documents had already been determined by consensus of a set of judges. Experimental results show that for a few features we are able to select the ten best essays about 60-70% of the times. As future work we plan to extend our strategy for selecting good essays to a complete automated unsupervised ranking of documents.

REFERENCES

[1] Y. Attali and J Burstein, "Automatic essay scoring system", The Journal of Technology, Learning and Assessment, vol. 4(3), pp 53-84, February 2006.

[2] B. Bigi,"Using Kullback-Leibler distance for text categorization" in Proc. of the 25th European Conference on Information Retrieval, Pizza, Italy, April, 2003, vol. 2633, pp. 305–319.

[3] Y.Y. Chen, C.L. Liu, T.H. Chang and C. H. Lee, "An unsupervised automated essay scoring system", IEEE Intelligent Systems, vol. 25, pp 61–67, 2010.

[4] T. Kakkonen, N. Myller, J. Timonen and E. Sutinen, "Automatic essay grading with probabilistic latent semantic analysis" in Proceedings of the second workshop on Building Educational Applications Using NLP, Association for Computational Linguistics, pp. 29–36, Morristown, NJ, USA (2005).

[5] S. Kullback and R.A Leibler,"On Information and Sufficiency", Annals of Mathematical Statistics vol. 22(1), pp 79–86, 1951.

[6] L.S. Larkey, "Automatic essay grading using text categorization techniques" in Proceedings of the 21st annual International ACM SIGIR conference on Research and Development in Information Retrieval,SIGIR '98, pp. 90–95. New York, NY,USA (1998), ACM Press.

[7] E. Tsagkias, "KL-divergence of two documents", URL: http://staff.science.uva.nl/~tsagias/?p=185.

[8] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. New York: ACM Press, Addison-Wesley,1999.

[9] R. R. Yager, "On ordered weighted averaging aggregation operators in multi-criteria decision making", *Fuzzy Sets and Systems*, vol. 10, 2, pp. 243-260, July 1983.

[10] R. R. Yager, "Quantifier guided Aggregating using OWA operators", *International Journal of Intelligent Systems*, vol. 11, 1, pp. 49-73, March 1996.

[11] R. R. Yager, and V. Kreinovich, "On how to merge sorted lists coming from different web search tools". *Soft Computing Research Journal*, vol. 3, 1, pp. 83-88, March 1999.