Is your Statement Purposeless? Predicting Computer Science Graduation Admission Acceptance based on Statement Of Purpose

Diptesh Kanojia[†,♣,⋆], **Nikhil Wani**[‡,†], **Pushpak Bhattacharyya**[†]
[†]Center for Indian Language Technology, IIT Bombay, India
[♣]IITB-Monash Research Academy, India
[⋆]Monash University, Australia
[†]{diptesh, pb}@cse.iitb.ac.in
[‡]nick.nikhilwani@gmail.com

Abstract

We present a quantitative, data-driven machine learning approach to mitigate the problem of unpredictability of Computer Science Graduate School Admissions. In this paper, we discuss the possibility of a system which may help prospective applicants evaluate their Statement of Purpose (SOP) based on our system output. We, then, identify feature sets which can be used to train a predictive model. We train a model over fifty manually verified SOPs for which it uses an SVM classifier and achieves the highest accuracy of 92% with 10-fold cross validation. We also perform experiments to establish that Word Embedding based features and Document Similarity based features outperform other identified feature combinations. We plan to deploy our application as a web service and release it as a FOSS service.

1 Introduction

Computer Science (CS) graduate admissions process often involves holistic evaluation of prospective applicant based on multiple subjective and quantitative parameters (Ward, 2006). Amongst these parameters the applicant's Statement of Purpose (SOP) serves as a document to convince its readers' *i.e.* the faculty on the selection committee - that one has recorded solid achievements which reflect promise for success in graduate study and hence submission of such a good quality SOP becomes of paramount importance.

Furthermore, Graduate admissions to most Elite universities in the United States of America (USA) only open twice every year - Fall and Spring semesters.

Terminology: We use the terms essay and SOP interchangeably further during our discussion of the work.

2 Motivation

Applicants spend a great deal of time writing SOPs for the admissions process. A well written SOP is a must for an applicant to ensure their admission in any university, and more so for elite universities. Their thoughts and ideas should be organized in their statement. University guidelines[1,2], Alumni blogs[3], and Admission consultancy blogs[4] recommend spending ample time on each SOP and tailoring it to perfection. They also recommend stylometry for writing an essay i.e. word limit, active voice, coherence, and continuity. Various NLP applications like Essay grading (Larkey, 1998), Text Summarization (Gupta and Lehal, 2010) and Sentiment Analysis (Joshi et al., 2015) utilize these features. Hence, we believe that an application that evaluates their statement is crucial. The key question that this paper attempts to answer is:

' *Can information gained from an SOP be used to predict the outcome of a candidate application for graduate school admissions?* '

3 Related Work

Ward (2006) discuss a qualitative model for Graduate Admissions to Computer Science programs but do not use any Machine Learning or Deep Learning based techniques for estimating a likelihood. According to them, other factors which affect the decision of the committee reviewing the applications include Graduate Record Examinations (GRE) score, Undergraduate Grade

[1] http://grad.berkeley.edu/admissions/apply/statement-purpose/
[2] http://admission.stanford.edu/apply/freshman/essays.html
[3] http://alumnus.caltech.edu/~natalia/studyinus/guide/statement/q&a.htm
[4] http://www.happyschools.com/strengthen-your-graduate-school-application/

Proc. of ICON-2017, Kolkata, India. December 2017 ©*2017 NLPAI*, pages 141–145

Point Average (GPA), Letters of Recommendation (LORs), Financial preparation of a candidate, Alignment with institute needs keeping in mind the diversity goals of the university, and lastly the Undergraduate Major of the candidate. They require the user to rate the application parameters and provide ratings as an input to their system. As an output, they provide an estimate of acceptance based on their model[5].

On the other hand, we employ the existing state-of-the-art techniques, identify features and use some of them to predict the acceptance of a candidate. We acknowledge that we do not model all parameters described above.

Another similar study (Raghunathan, 2010) tries to subjectively discuss the admissions process and details the factors which participate in the decision making process of an admission committee. They break the components of a graduate school admissions process and state that SOP is one of the trickiest components of an overall application. They also note that too long an SOP would deter the chances of selection of the candidate. In light of these studies, we focus on creating a model which is able to grade an SOP based on ML techniques.

Text Similarity and related measures (Choi et al., 2010; Adomavicius and Tuzhilin, 2005; Gomaa and Fahmy, 2013) have been extensively studied and used for various NLP applications *viz.* Information Retrieval (Salton et al., 1983), Sense Disambiguation (Resnik and others, 1999). To the best of our knowledge, there is no reported study which evaluates SOPs based on the features identified by us, or use ML and DL based techniques of this kind, at the time of submission. Most of the articles list various parameters which are considered by an admissions committee and a Statement of Purpose (SOP) is a common factor among all.

4 Experiment Design and Setup

In this section, we provide details about our experiment setup and features used for the classification task.

4.1 Dataset

We create our dataset by collecting essays from i) Acquaintances ii) Publicly disclosed SOPs from personal websites, and iii) Admission consultancy blogs. For calculating the similarity measures, we concatenate the essays of the successful applicants, and create a corpus which is used for comparison with both training and testing data.

We collect a total of 50 manually verified SOPs from Elite Universities (low acceptance rate <= 15%) and rejected essays equally split into two sets. We plan to release the dataset publicly under the CC-BY-SA-4.0[6] license.

4.2 Methodology

We use conventional Machine Learning (ML) algorithms (Hall et al., 2009) like Support Vector Machines (SVM) (Vapnik, 2013), Logistic Regression (LR) (Walker and Duncan, 1967), and Random Forest Decision Trees (RFDT) (Ho, 1998) for the task and provide a comparison in Section 5.

We use deep learning approaches and deploy a simple Feed Forward Neural Network to classify the SOPs. We split our data in two folds where the first half is used for training, and the second half is then split into tuning and testing datasets. We also use Multilayer Perceptron, another simple Feed Forward Neural Network (NN) and perform a standard 10-fold cross validation on our dataset. We do acknowledge the modest size of our dataset, but we provide rigorous experimentation including an ablation test to verify that our performance on all classes of our data are unbiased.

4.3 Experiment Design

We cluster the set of features in the following groups - **a) Textual Features** - Feature values based on text contained within the document, **b) Word Embedding based Features** - Features based on average of vector values provided by pre-trained model on Google News Corpora, **c) Similarity Score based and Error based features** - Features based on Document Similarity, and other features based on errors in the document. The last set of features have been identified by us, and are our contribution to the work. We, then, use the algorithms mentioned above to calculate precision, recall and F-Score on each feature set.

We also perform an Ablation test to see which feature set combination is performing the best.

[5] http://www.cs.utep.edu/nigel/estimator/

[6] https://creativecommons.org/licenses/by-sa/4.0/

Classifier	P_{acc}	P_{rej}	$\mathbf{P_{avg}}$	R_{acc}	R_{rej}	$\mathbf{R_{avg}}$	F_{acc}	F_{rej}	$\mathbf{F_{avg}}$
RFDT	0.86	0.79	0.83	0.76	0.88	0.82	0.81	0.83	0.82
LR	0.69	0.83	0.76	0.88	0.60	0.74	0.77	0.70	0.74
SVM	0.89	0.96	**0.92**	0.96	0.88	**0.92**	0.92	0.92	**0.92**
Neural Network Based									
Multilayer Perceptron (Train-Test Split)	-	-	0.82	-	-	0.82	-	-	0.82
Feed Forward NN (FFNN) (Train-Tune-Test Split)	-	-	0.36	-	-	0.60	-	-	0.45

Table 1: Performance of our model on 10-fold cross validation

4.4 System Architecture

Our architecture, shown in figure 1, provides the necessary details about the working of our system. The system takes as input the essay of a prospective applicant, calculates feature values for Similarity Score and Error based features along with Word Embedding based features and predicts an **accept** or **reject** based on the classification model being used.

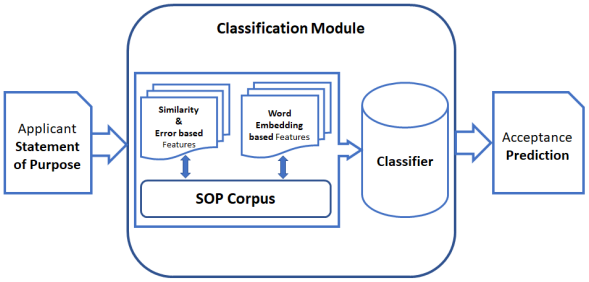

Figure 1: System Architecture

4.5 Features Used

We use the following textual features for evaluating the SOPs. These features have been identified via surveying linguistic properties of a text which may affect the organization and quality of an essay.

4.5.1 Word Embeddings based Features

1. **Average Word Vector Scores** - Average of word vectors of each word in the statement calculated using pre-trained Google News word vectors (Mikolov et al., 2013).

4.5.2 Textual Features

1. **PoS Ratios** - Ratio of nouns, adjectives, adverbs, and verbs to the entire text, obtained using NLTK[7] (Loper and Bird, 2002).

[7] http://www.nltk.org/

	Individual Feature Sets (N-fold)			
Features	2-F	5-F	10-F	50% Split
T [14]	54	46	44	40
WE [300]	48	78	40	44
SE [3]	48	56	56	49
	Combination of Feature Sets			
T + WE [314]	56	62	62	52
T + SE [17]	48	50	38	30
SE + WE [303]	**90**	**92**	**92**	**92**
T + WE + SE [318]	52	50	53	43

Table 2: Ablation test on feature sets using Multi-fold Cross Validation

2. **Discourse Connectors** - It is the number of discourse connectors in the essay computed using a list of discourse connectors[8].

3. **Count of Named Entities** - Number of named entities in the essay. We tried using this as a feature but this drastically lowered the F-scores, and had to be avoided in the final set of reported experiments.

4. **Readability** - The Flesch Reading Ease Score (FRES) of the text (Flesch, 1948).

5. **Length features** - Number of words in the sentence, number of words in the paragraph, and average word length.

6. **Coreference Distance** - Sum of token distance between co-referring mentions.

7. **Degree of Polysemy** - Average number of WordNet (Fellbaum, 2010) senses per word.

4.5.3 Document Similarity Score and Error based Features

1. **Cosine Similarity** - Cosine Similarity Score of an SOP with the corpus of accepted essays dataset, where we ensure that the SOP being compared is not a part of the accepted essay corpus.

[8] http://www.cfilt.iitb.ac.in/cognitive-nlp/

2. **Similarity-based features using GloVe** - The similarity between every pair of content words in adjacent sentences. The similarity is computed as the cosine similarity between their word vectors from the pre-trained GloVe word embeddings (Pennington et al., 2014). We calculate the mean and maximum similarity values.

3. **Spell Check Errors** - We use PyEnchant[9] to embed a spell checker and count the number of errors in each document. The count is then used as another feature for training classifier.

4. **Out of Vocabulary Words** - We use the pre-trained Google news word embeddings and find out word vectors for every token in the document. The tokens which do not return any vector are either rare words or in all probability out of vocabulary words. We use the count of such tokens as another feature set.

5 Results

We perform the experiments detailed in section 4.3 and report our results on 10-fold cross validation. Among the experiments we perform, we achieve the highest F-score of 92% using the SVM classifier with an RBF Kernel. The results are shown in table 1 and discussed in Section 6.

Table 1 clearly indicates that SVM outperforms Random Forest Decision Trees (RFDT) with a margin of 9%, Logistic Regression (LR) with a margin of 18%, Neural Network based Multilayer Perceptron with a margin of 10%, and another Feed Forward Neural Network (FFNN) with a margin of 47%. We further discuss the impact and justifications of these results in Section 6.

We also perform a multi-fold ablation test, using SVM Classifier, on the feature sets identified in section 4.3. The results for the ablation test are shown in Table 2. The table clearly identifies that Similarity Scores and Error based features along with Word Embedding based features give us the best results.

6 Discussion

In order to identify the features that contribute to the modeled non-linearity of SVM and our best reported accuracy of 92%, we conduct a comprehensive ablation test. Feature sets mentioned in Section 4.3 were considered. A total of 317 features were ablated based on their sets via multi-fold stratified cross validation experiments and additionally in an experiment with 50% split of the dataset as shown in the Table 2.

It was found that the 14 identified Textual (T) features do not contribute significantly to our model. We extrapolate that these features may have worked better in another context such as Sentiment Analysis (Mishra et al., 2017), or Essay Grading (Valenti et al., 2003), but not for the task of SOP Classification. Our task primarily aims at labeling an SOP with an accept or reject, however, we observe that Textual features do not differentiate well between coherent and incoherent essays. We also observe that Word Embedding (WE) features of 300 dimensions contribute significantly towards the accuracy of our final model. While they do not contribute notably when used to perform classification independently, combining them with Similarity Score and Error Based (SE) feature set form our best reported model i.e. SE + WE.

7 Conclusion and Future Work

In this paper we demonstrate the applicability of a data driven approach to mitigate the unpredictability of Computer Science graduate admissions process. We build a corpus of fifty manually verified SOPs from Accepted applicants to Elite Universities (low acceptance rate <= 15%) rejected SOPs. We show that a combination of Cosine Similarity, Error based features and Word Embedding based features outperform any of the textual features based combinations, for this task. Based on the ablation tests conducted, we model an SVM classifier that predicts with significantly high accuracy.

In future, we plan to integrate Parts-of-speech (POS) based similarity measures and Recurrent Neural Networks (RNN) (Cho et al., 2014) which have been shown to work well with textual data. Integration of other traditional metrics of a candidates application performance measure such as GRE, Test of English as a Foreign Language (TOEFL) / International English Language Testing System (IELTS) score and GPA will further robustly extend this model. We also plan to translate this novel research to an open source web application which would allow prospective applicants to evaluate their SOPs with our system.

[9] http://pythonhosted.org/pyenchant/

References

Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 17(6):734–749.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Seung-Seok Choi, Sung-Hyuk Cha, and Charles C Tappert. 2010. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48.

Christiane Fellbaum. 2010. Wordnet. *Theory and applications of ontology: computer applications*, pages 231–243.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Wael H Gomaa and Aly A Fahmy. 2013. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13).

Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Tin Kam Ho. 1998. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *ACL (2)*, pages 757–762.

Leah S. Larkey. 1998. 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, pages 90–95, New York, NY, USA. ACM.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics.

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.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2017. Leveraging cognitive features for sentiment analysis. *arXiv preprint arXiv:1701.05581*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Karthik Raghunathan. 2010. Demystifying the american graduate admissions process. *StudyMode. com*.

Philip Resnik et al. 1999. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res.(JAIR)*, 11:95–130.

Gerard Salton, Edward A Fox, and Harry Wu. 1983. Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036.

Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education: Research*, 2(1):319–330.

Vladimir Vapnik. 2013. *The nature of statistical learning theory*. Springer science & business media.

Strother H Walker and David B Duncan. 1967. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179.

Nigel Ward. 2006. Towards a model of computer science graduate admissions decisions. *JACIII*, 10(3):372–383.