Aplicando classificação não-supervisionada para detecção de cola em provas escolares

Elmano R. Cavalcanti[1], José S. Jackson[2]

[1] Universidade Federal de Campina Grande (UFCG), 58.109-970 - Campina Grande - PB - Brazil

[2] Faculdades Integradas de Patos (FIP), 58.700-250 - Patos - PB - Brazil

Abstract Neste artigo, foi desenvolvida uma solução para a detecção de cola em provas escolares utilizando-se de classificação não supervisionada, do modelo de vetor de espaços e do cálculo da similaridade por cosseno. O resultado permitiu detectar cola de diversos tamanhos em um conjunto de 30 provas. A acurácia do modelo de detecção foi de 72,73% e o índice Kappa apresentou um valor substancial: 0,63.

Keywords: classificação, similaridade, mineração de texto

1 Introdução

A cola de alunos em provas escolares é uma prática indesejável porém, amplamente disseminada. Um dos fatores que dificulta a detecção de cola é grande quantidade de dados que são analisados manualmente pelo professor. Embora não exista uma definição concreta de cola, o senso comum de que quando duas provas apresentam um grau de semelhança razoável, supõe-se que tenha ocorrido a cola. Com o advento da informática é de esperar-se que num futuro próximo as correções das provas sejam feitas por *softwares* capazes de detectar diversos tipos e tamanhos de cola, desde as maiores às mais sutis. A seguir, é descrita uma solução para detecção de cola em provas escolares.

2 O Processo de Mineração de Texto

No escopo deste trabalho, os dados foram obtidos diretamente de fontes eletrônicas ou criados manualmente em arquivos de textos comuns. Dessa forma, não foi necessário realizar as atividades iniciais de seleção, limpeza e amostragem dos dados. Ao todo, foram utilizadas 24 provas reais, cada uma contendo quatro questões subjetivas. Adicionalmente, foram criadas seis provas fictícias simulando alunos que colaram de alguma das provas reais.

Foi necessário definir um dicionário para cada questão, ou seja, um conjunto de palavras que faz parte do contexto da questão. Seguindo as etapas de mineração de texto apresentadas em [1], temos as seguintes fases:

1. *CodeMapper*: Nesta etapa, foi realizada a remoção dos acentos das palavras. Todas as etapas a seguir foram feitas utilizando-se a ferramenta RapidMiner [4] com o plugin de mineração texto [3];

2. *Tokenizer*: Separação do texto em palavras (i.e., *tokens*);

cola	grande	razoável	pequena	nenhuma	total
grande	10	2	0	0	12
razoável	1	6	3	0	10
pequena	0	0	4	1	5
nenhuma	0	1	1	4	6
total	11	9	8	5	33

Table 1: Matriz de Confusão

3. *WordFilter*: Em seguida, foi feita uma filtragem da lista de palavras da etapa anterior. Utilizou-se a lista de *stopwords* disponível no projeto Snowball [2]. Foram feitos alguns acréscimos de palavras, de acordo com o domínio das questões da prova;

4. *Stemmer/Reducer*: Antes de iniciar a garimpagem dos dados foi necessário mapear todos os sinônimos ou palavras que possuem o mesmo radical (e.g., processar e processado) para uma única palavra-base. Para esta etapa foi utilizado o algoritmo de *stemming* do Snowball.

5. *Criação do vetor*: O modo usado para criação do vetor foi o *term frequency-inverse document frequency* (TFIDF), que é uma medida estatística utilizada para avaliar a importância que uma palavra tem dentro de um documento. O tamanho do vetor ficou em aproximadamente 500 colunas. Todos os vetores foram normalizados para o tamanho unitário Euclidiano.

Utilizou-se classificação não-supervisionada através do cálculo de similaridade da função cosseno, a qual retorna um valor real $sim(i, j)$ no intervalo [0,1], indicando o nível de semelhança entre duas provas i e j. Foram consideradas quatro categorias de cola: grande ($sim(i, j) > 0,70$), razoável ($0,40 < sim(i, j) < 0,70$), pequena ($0,25 < sim(i, j) < 0,40$), nenhuma ($sim(i, j) < 0,25$). Em seguida, foi possível construir a Matriz de Confusão (Tabela 1), que representou uma acurácia de 72,73% e índice Kappa de 0,63, o que é considerado um valor substancial [5] para o modelo de inferência construído.

References

[1] Bilenko, M., Mooney, R. J.: Adaptive Duplicate Detection Using Learnable String Similarity Measures. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003), Washington DC, August 2003 pp.39–48

[2] http://snowball.tartarus.org/

[3] Wurst, M.: The Word Vector Tool and the RapidMiner Text Plugin. Dortmund, Germany (2007). Disponível em: ¡http://wvtool.sf.net/¿

[4] RapidMiner 4.0 - User Guide, Operator Reference and Developer Tutorial. Dortmund, Germany (2007). Disponível em: ¡http://www.rapidminer.com/¿

[5] Landis, J. R. and Koch, G. G. The measurement of observer agreement for categorical data In Biometrics, vol. 33, 1977, pp. 159–174