PUC

Utilizando machine learning para classificação das narrativas em um jogo de futebol

Silvano Nogueira Buback

Departamento de Informática

INF 2030

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900

RIO DE JANEIRO - BRASIL

Sumário

Introdução

A principal característica da Web 2.0 é a possibilidade de qualquer usuário de internet ser capaz de produzir conteúdo. Blogs, comentários em notícias, opiniões em redes sociais como Orkut, Facebook ou Twitter são utilizados pelos seus usuários para expressar sua própria opinião a respeito de diversos assuntos. Porém a participação em massa passou a gerar uma quantidade excessiva de conteúdo e os meios no qual este conteúdo é gerado normalmente não oferecem um mecanismo eficiente de classificação sobre o aspecto computacional, ao contrário do conteúdo gerado por editores profissionais e cadastrados e classificados em sistemas especializados para o assunto.

Para a participação durante uma partida de futebol o mesmo problema se repete. Cada vez mais é comum as pessoas acompanharem o jogo através do computador. *Streaming* de vídeo (1) e técnicas como *Comet* (2) permitem o torcedor o acompanhamento da partida de futebol com vídeo e narração em tempo real. Porém para a experiência social estas tecnologias não auxiliam tanto, pois normalmente os internautas fazem comentários fora do instante do lance. É comum haver mais comentários através do Twitter durante o intervalo e após o jogo. Com a participação dos usuários sem uma ordem cronológica entre eles, temos uma participação confusa, com comentários dispersos na seqüência do texto. Seria interessante que pelo menos os principais eventos de uma partida pudessem ser classificados automaticamente para permitir que os usuários possam ler as opiniões sobre os lances que acharem mais polêmicos, incentivando a participação.

Objetivo

Para conseguir estimular mais a participação dos usuários, o objetivo deste trabalho é de classificar os comentários de usuários de acordo com os seguintes eventos de uma partida: **gol**, **pênalti**, **substituição** e **cartão**. Tudo que não se encaixar nos critérios acima será classificado como uma **narração** (comentário) sobre a partida. Pênalti, apesar se ser uma falta, possui classificação especial, pois normalmente é um lance muito polêmico, por isto o tratamento especial.

Corpus/Desenvolvimento

Para o desenvolvimento deste trabalho utilizamos o classificador Naive Bayes (3). Este classificador, apesar de simples de implementar, produz bons resultados na classificação de textos (3) (4).

Como o foco do trabalho é a classificação, sendo o classificador apenas uma ferramenta, foi utilizado uma implementação do Naive Bayes chamada de NLTK, escrita em linguagem Python. O NLTK é uma biblioteca com diversos algoritmos para manipulação de linguagem natural. Além disto, ela também possui diversos corpora, a maior parte relacionada a textos de língua inglesa.

O corpus utilizado neste trabalho consiste da narração de um editor esportivo durante as partidas de futebol do site GloboEsporte.com. Tanto jogos nacionais, quanto internacionais estão incluídos e, em todos, a narração é em português. Abaixo os números do corpus.

Evento	Total	Treino	Teste
Gols	6.516	4.888	1.628
Cartões	12.339	9.254	3.085
Pênalti	272	205	67
Substituição	12.750	9.562	3.188
Narração	127.296	95.473	31.823
Total de sentenças	159.173	119.381	39.792
Total de jogos			**4.640**

Tabela 1 - Descrição quantitativa do corpus utilizado

Uma solução pedestre para o problema seria tentar procurar por palavras chaves para fazer a classificação do texto, por exemplo, onde houvesse a palavra "gol" a sentença seria classificada como **gol**, a palavra "cartão" classificaria como **cartão**, e para cada categoria uma ou mais palavras poderiam ser utilizadas. O problema desta abordagem é que ela não funciona tão bem para a massa de dados utilizada. Veja algumas sentenças extraídas do corpus.

"Marquinhos - O meia faz falta dura no meio-campo" (cartão)

"Djair recebe na entrada da área e chuta. A bola passa por cima do gol de Roberto" (narração)

Se usássemos features predefinidas acima certamente as sentenças acima teriam classificação errada. Como reportado por (5) para o problema de *sentiment analysis*. Assim, vamos deixar o classificador definir os peso de cada palavra para uma classe.

Assim, para realizar a classificação do texto, foi utilizado como feature a existência de cada palavra na sentença, sem levar em consideração maiúsculas/minúsculas e os caracteres acentuados, ou seja, "Pênalti" e "penalti" são consideradas a mesma feature.

Na segunda etapa, utilizamos as palavras mais informativas para tentar melhorar a precisão do classificador.

Resultados

Com o algoritmo implementado, executamos o classificador e conseguimos uma acurácia de 68%. Analisando as 10 features identificadas como mais informativas, temos:

Palavra	Classificação
Convertido	Pênalti
Goooooooool	Gol
Falta	Cartão
Em	Cartão
Gooool	Gol
Gooooooooooooooooool	Gol
Goleiro	Gol
Substitui	Substituição
Dura	Cartão
Gooooooooooool	Gol

Tabela 2 - Palavras mais informativas (melhores features)

Na tentativa de melhorar a precisão do classificador, utilizamos uma técnica muito comum em *sentiment analysis* que implica em utilizar somente as features mais relevantes [6]. Fazendo isto, retiramos o peso das palavras que não mudam a classificação do texto, retirando o ruído.

Para fazer isto, é necessário utilizar o resultado acima, selecionando-se as N palavras mais informativas. Com esta lista de palavras, as sentenças são novamente classificadas, porém na nova classificação somente serão features a presença de palavras que estiverem nesta lista das mais informativas. As demais são ignoradas. Para determinar qual o número de palavras mais informativas utilizar, é necessário executar a classificação utilizando diversos valores de N e comparando a acurácia obtida. Abaixo tabela e gráfico com os resultados.

Número de palavras	Acurácia
5	76%
30	89%
50	89%
80	83%
100	81%
200	75%
300	75%
400	74%
500	74%
800	75%
900	75%
1000	75%
3000	76%
5000	76%
10000	76%
Todas	68%

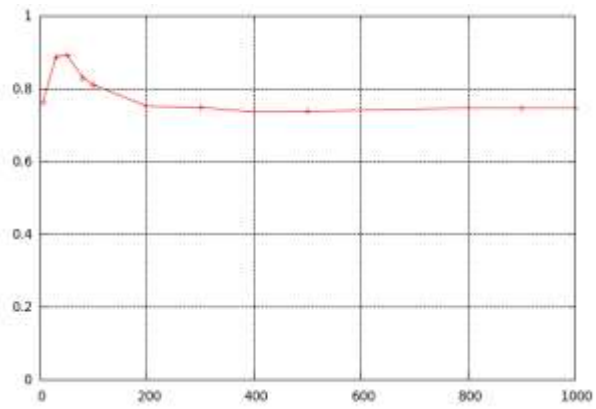

Figura 1 - Gráfico número palavras X acurácia

Portanto utilizando o classificador para utilizar como feature somente a presença das 30 a 50 palavras mais informativas temos uma acurácia de 89%.

Conclusão e trabalhos futuros

Utilizando somente as 50 melhores palavras para a classificação temos um resultado satisfatório, próximo de 90%, muito bom considerando a maioria dos resultados dos trabalhos de classificação de texto (5). Embora uma comparação direta não possa ser feita com outros trabalhos, pois cada domínio e corpus podem possuir particularidades, a inexistência de trabalhos semelhantes a este domínio nos força a fazer esta comparação para permitir alguma orientação.

Uma melhoria no classificador seria a possibilidade de utilizar bigrams ou até trigrams e verificar os resultados obtidos. Para alguns domínios de classificação de texto eles permitem uma melhora na

acurácia. Outra melhoria seria a utilização de algoritmos mais sofisticados de *machine learning* como o SVM.

Todos os dados e o código fonte do trabalho estão disponíveis em: http://github.com/snbuback/trab-INF2030.

Referências

[1] FOUNDATION, Wikimedia (Org.). **Streaming**. Disponível em <http://pt.wikipedia.org/wiki/Streaming>. Acesso em: 21/agosto/2010.

[2] FOUNDATION, Wikimedia (Org.). **Comet (Programming).** Disponível em <http://en.wikipedia.org/wiki/Comet_%28programming%29>. Acesso em: 21/agosto/2010.

[3] S. Chakrabarti, **Mining the Web: Discovering Knowledge from Hyper-text Data**, Morgan Kaufmann, 2002. 3.5, 4, 7.

[4] A. McCallum and K. Nigam, **A comparison of event models for Naïve bayes text classification**, In Proc. of the AAAI-98 Workshop on learning for text categorization, pp. 41-48. 2.2, 4

[5] PANG, Bo; LEE, Lillian; VAITHYANATHAN, Shivakumar. **Thumbs up? Sentiment Classification using Machine Learning Techniques.** New York: Emnlp, 2002.

[6] StreamHacker.com. **Text Classification for Sentiment Analysis – Eliminate Low Information Features.** Disponível em: http://streamhacker.com/2010/06/16/text-classification-sentiment-analysis-eliminate-low-information-features/. Acesso em: 30/agosto/2010.