2
Trabalhos relacionados

Não foram encontrados trabalhos diretamente relacionados a auto-moderação de comentários. Contudo este trabalho possui uma semelhança com trabalhos que tratam de filtros anti-spam. Como muitos deles estão ligados a linguagem natural, utilizando técnicas de aprendizado de maquina, também serão estudados trabalhos sobre análise sentimental. Estes trabalhos servirão como guia para o desenvolvimento da dissertação e também ajudarão a construir um *base-line*.

2.1.Filtro anti-spam

Dentro os artigos pesquisados, (ANDROUTSOPOULOS, KOUTSIAS, *et al.*, 2000) e (ANDROUTSOPOULOS, KOUTSIAS, *et al.*, 2000) são muito citados. Nestes trabalhos, os autores mostram como utilizar Naive Bayes (LEWIS, 1998) para construção de um filtro anti-spam e qual o impacto que a seleção dos atributos possui no resultado final. Como métrica é utilizado uma acurácia com peso λ para mensagens legítimas que são classificadas incorretamente como spam.

Filter used ("NB" is Naive Bayesian)	λ	**no. of attr.**	**spam recall (%)**	**spam precision (%)**	**weighted accuracy (%)**	**TCR**
(a) NB bare	1	50	83.98	95.11	91.076	4.90
(b) NB with stop-list	1	50	84.19	96.76	91.167	4.95
(c) NB with lemmatizer	1	100	78.14	98.25	89.796	4.29
(d) NB with lemmatizer and stop-list	1	100	79.60	97.96	90.341	4.53
Keyword patterns (case insensitive)	1	–	53.01	95.15	78.253	2.01
Baseline (no filter)	1	–	0	∞	56.233	1
(e) NB bare	9	100	78.77	96.65	96.378	2.20
(f) NB with stop-list	9	150	74.83	97.34	96.508	2.28
(g) NB with lemmatizer	9	100	75.86	98.50	97.183	2.83
(h) NB with lemmatizer and stop-list	9	100	75.86	97.91	96.886	2.56
Keyword patterns (case insensitive)	9	–	53.01	95.15	94.324	1.40
Baseline (no filter)	9	–	0	∞	92.040	1
(i) NB bare	999	700	46.96	98.80	99.475	0.15
(j) NB with stop-list	999	700	47.17	98.76	99.475	0.15
(k) NB with lemmatizer	999	50	60.68	98.79	99.322	0.11
(l) NB with lemmatizer and stop-list	999	600	49.45	98.31	99.313	0.11
Keyword patterns (case insensitive)	999	–	53.01	95.15	97.862	0.04
Baseline (no filter)	999	–	0	∞	99.922	1

Tabela 1 - Resultados obtidos em (ANDROUTSOPOULOS, KOUTSIAS, *et al.*, 2000) e (ANDROUTSOPOULOS, KOUTSIAS, *et al.*, 2000).

A utilização de um peso para medir a acurácia é útil, pois os autores entendem que classificar como spam uma mensagem legítima é mais grave que classificar como mensagem legítima um spam. Este aspecto é muito importante para o trabalho de auto-moderação de comentários, porém o peso λ possui um efeito inverso. Neste caso é mais grave aprovar uma mensagem que deveria ser reprovada, pois tal ação pode ter um efeito muito negativo para a página que ela está sendo exibida. Se uma mensagem aprovada fosse reprovada, ainda poderíamos contar com a ajuda manual do moderador.

Conforme já publicado por (JOACHIMS, 1998), SVM tem-se mostrado eficiente em tarefas de classificação de texto em geral. (LAI, 2007) fez um estudo sobre a acurácia do Naive Bayes, k-NN e SVM para classificação de spam, utilizando dois corpora. Em ambos, o SVM mostrou-se mais eficiente que os outros algoritmos, chegando atingir acurácia de 94,59%, contra 81,59% do k-NN e 90,89% do Naive Bayes. Estes resultados são os melhores de cada algoritmo utilizando o corpus globo-twitter, com 30% do corpus para teste. (DRUCKER, WU e VAPNIK, 1999) também publicaram um estudo sobre o SVM e também concluíram que ele possui uma maior acurácia para a classificação de spam.

Para tentar melhorar ainda mais a acurácia, alguns trabalhos utilizam técnicas para combinar algoritmos. (CARRERAS e MÀRQUEZ, 2001) implementaram uma variante do AdaBoost e conseguiram precisão de 98,73% com recall de 97,09%, para λ=1. (SAKKIS, ANDROUTSOPOULOS, *et al.*, 2001) combinaram vários algoritmos com *stacked generalization* e conseguiram melhorar a acurácia de (ANDROUTSOPOULOS, KOUTSIAS, *et al.*, 2000), especialmente quando λ>1.

2.2.Sentiment Analysis

Analisando trabalhos sobre classificação de texto, encontramos um tema de muito interesse atualmente: *Sentiment Analysis*. Este consiste em uma classificação de texto em que se está interessado em saber se o mesmo tem um sentimento positivo ou negativo (alguns trabalhos consideram também neutro). Este tema ganhou muita evidência devido à proliferação das redes sociais. Ele permite, em larga escala, saber qual o sentimento dos usuários sobre um contexto definido, que pode ser um produto, pessoa, ou serviço.

O trabalho mais conhecido sobre sentiment analysis é (PANG, LEE e VAITHYANATHAN, 2002). Este trabalho parece ser o primeiro a utilizar técnicas de aprendizado de máquina para resolver o problema. Nele os autores mostram que contar palavras, selecionadas manualmente, que intuitivamente fornecem uma identificação sobre o sentimento (como brilhante, excelente ou terrível), não dão uma boa acurácia. A partir daí eles utilizam 3 algoritmos de aprendizado de máquina, Máxima Entropia (BERGER, PIETRA e PIETRA, 1996), Naïve Bayes (LEWIS, 1998) e SVM (JOACHIMS, 1998), combinando técnicas diferentes para a seleção dos atributos. Os autores mostram que o SVM utilizando como atributo somente a presença ou não das palavras e símbolos de pontuação tem a melhor acurácia. A análise também é feita com Part of Speech (POS), somente de adjetivos e adicionado as palavras sua posição no texto. Os autores ainda argumentam que sentiment analysis parece ser uma tarefa mais desafiadora do que a tradicional classificação de texto em tópicos.

Mais recentemente, (ALVIM, VILELA, *et al.*, 2010) propuseram o uso do SVM com Bigrams e POS Tagging conseguindo acurácia de 84,80% sob o mesmo corpus de (PANG, LEE e VAITHYANATHAN, 2002), com 3 folds, e 86,09% com 10 folds, tornando-se o estado da arte para este corpus. Utilizando a mesma técnica num corpus de notícias da Petrobrás (em português), (ALVIM, VILELA, *et al.*, 2010) conseguiram acurácia de 84,00%, desta vez com 5 folds; porém, para este corpus, a acurácia foi melhor quando se utilizou como atributo apenas a presença das palavras com unigrams, com acurácia de 85,94%.

Outro trabalho proposto sobre sentiment analysis é (TURNEY, 2002). Uma característica deste trabalho que difere bastante dos demais é que ele utiliza um método não-supervisionado, ou seja, não é necessário um corpus para fazer o aprendizado. Alternativamente, os autores utilizam um algoritmo baseado em três etapas, a saber:

1. Identificar os adjetivos e advérbios através de um algoritmo de POS Tag.
2. Calcular a orientação semântica de cada frase através de um algoritmo de PMI-IR (CHURCH e HANKS, 1990), porém fazendo acesso a um *search engine* para obter informações sobre cada palavra selecionada na primeira etapa e o quão mais aproximada de *excellent* ou *poor* ela estaria. A isto era atribuído um valor numérico.

3. Por fim, atribuir uma classe - recomendado ou não-recomendado - de acordo com a média da orientação semântica das frases calculadas no item acima.

Observe que este trabalho mostra uma heurística que não depende de um corpus para fazer o aprendizado (não-supervisionado), porém precisa de um acesso a um serviço Web de busca para avaliar se o adjetivo ou advérbio em questão define uma boa avaliação ou não, o que poderia tornar o algoritmo inviável. Além disto, a acurácia obtida não é melhor que o método proposto por (PANG, LEE e VAITHYANATHAN, 2002), sendo de apenas 65,83% para filmes, porém a comparação não é feita com o mesmo corpus.