Notas:

O algoritmo produz várias saídas na pasta files, dentre elas um arquivo de probabilidade para os termos de cada classe e um arquivo contendo o resultado da classificação dos testes (classification.text), onde, ao final, está o erro global da classificação.

* O erro ainda está alto quando usadas todas as classes (~0,53). É preciso verificar se é possível reduzir esse erro ou se ele é esperado dada a proximidade de algumas classes.
* Antes de remover as 100 palavras mais freqüentes, a classificação errava muito mais (~0,83).
* O erro com as classes inclusas (rec.autos, talk.politics.mideast e talk.politics.guns), no momento, está em torno de 0,22.

Próximo passo: implementar algoritmo de seleção de tuplas para treino e para teste, conforme especificação do trabalho.

10/09/11

* Seguindo dicas da professora, o vocabulário está sendo primariamente construído sobre todas as palavras do texto concatenado de cada classe, previamente limpo.
* Utilizando um algoritmo para verificar se determinada linha é cabeçalho, o erro do algoritmo diminuiu sensivelmente. Verificações posteriores mostraram que havia uma presença recorrente da palavra “article” nos textos concatenados. A adição da verificação do termo “in article” a esse algoritmo melhorou um pouco a saída da classificação.

É interessante notar que toda melhoria que puder ser feita nos textos concatenados das classes tem grande chance de produzir melhora no resultado da classificação, já que o vocabulário já será criado com termos mais “importantes”.

* Em um dos testes, deixei de limpar o vocabulário, retirando as palavras que aparecem menos de 3 vezes e as 100 palavras que mais aparecem. O erro em questão passou de 25% para 78%!