

Língua Natural

Grupo 7 - Relatório do Mini-Projeto 2

83567 - Tiago Gonçalves

83576 - Vítor Nunes

Introdução

O projecto consiste na classificação, atribuição de categorias, de questões sobre cinema. É fornecido a um classificador um set de questões com categorias previamente conhecidas e depois é possível colocar novas questões e o mesmo atribuir a categoria da mesma.

Proposta de Solução

A primeira abordagem consistiu em utilizar a biblioteca nltk ¹ para realizar:

- *Parsing*: ler as questões conhecidas do ficheiro e criar uma estrutura de dados compatível com o classificador.
- *Tokenizing*: separar as palavras da frase e, posteriormente, utilizar os lemas.
- *Stemming*: ignorar palavras que aparecem em todas as frases e não são relevantes. Articuladores e pronomes, designados de *stop words*.
- Naive Bayes: os classificadores utilizados foram baseados em Naive Bayes

A métrica utilizada, para atribuir pesos a determinados lemas, corresponde a atribuir um valor booleano (TRUE ou FALSE) consoante o novo lema já foi observado anteriormente ou não.

Numa segunda abordagem, após realizada uma pesquisa e análise de um tutorial [8], decidiu-se utilizar:

- *Parsing e Tokenizing*: COUNTVECTORIZER, que constroi uma matriz com contagens dos tokens, isto é, a frequência com que aparecem nas frases de treino.

¹Disponível em <http://www.nltk.org/>

Resultados experimentais

A primeira abordagem revelou os seguintes dados:

Classificador	Accuracy
Multinomial Naive Bayes	$\approx 9.5238\%$
Bernoli Naive Bayes	$\approx 2.3810\%$
Complement Naive Bayes	$\approx 9.5238\%$

Tabela 1: *Accuracy* usando Naive Bayes sobre o ficheiro NovasQuestoes.txt

Os resultados não foram bons, isto porque os algoritmos Naive Bayes utilizam a definição de probabilidade condicionada para determinar a probabilidade de dois lemas aparecerem seguidos.

Porém o que se pretende é classificar frases em categorias específicas portanto foi necessário seguir outra abordagem.

A segunda abordagem, usando a biblioteca SKLEARN revelou os seguintes dados:

Classificador	Kernel	Accuracy
SGD ²	<i>hinge</i>	$\approx 92,8571\%$
SGD	<i>log</i>	$\approx 80,9524\%$
SGD	<i>modified_huber</i>	$\approx 61,9048\%$
SGD	<i>squared_hinge</i>	$\approx 73,8095\%$
SGD	<i>perceptron</i>	$\approx 73,8095\%$
SGD	<i>epsilon_insensitive – l2</i>	$\approx 95,2381\%$
SGD	<i>epsilon_insensitive – l1</i>	$\approx 78,5714\%$
SGD	<i>epsilon_insensitive – l2 – invscaling</i>	$\approx 88,0952\%$
SGD	<i>epsilon_insensitive – l2 – constant</i>	$\approx 78,5714\%$
KNeighbors	<i>K = 3</i>	$\approx 83,3333\%$
SVC	<i>linear</i>	$\approx 00,0000\%$
DecisionTree		$\approx 90,4762\%$
RandomForest		$\approx 78,5714\%$

Tabela 2: Comparação de vários classificadores usando o ficheiro de teste NovasQuestoes.txt

O SGD (*epsilon_insensitive*) apresentou o melhor resultado, porém após realizar mais testes, mudando o training set e o *testing set*, obtemos melhor resultados usando o SGD (*hinge*). Observámos também, que o nível de falha reside em categorias que sejam bastante próximas. Por exemplo, detetámos alguns erros em distinguir a categoria *budget* da categoria *revenue*.

Conclusão e trabalho futuro

Por forma a resolver erros em categorias próximas, uma possível solução seria criar uma lista de palavras mais comuns por categoria por forma a aperfeiçoar o critério. Conclui-se então, que não existe nenhum classificador pré-determinado, isto é, o classificador deve ser obtido à custa de testes com casos reais.

Referências

- [1] Text Classification using Algorithms - chatbot
- [2] dk., Soul of the Machine: How Chatbots Work
- [3] PythonProgramming.net, Text Classification with NLTK
- [4] nltk.org Probability in NLTK
- [5] Syed Sadat Nazrul, Multinomial Naive Bayes Classifier for Text Analysis
- [6] Olli Huang, Applying Multinomial Naive Bayes to NLP Problems: A Practical Explanation
- [7] scikit-learn.org, 1.9. Naive Bayes
- [8] scikit-learn.org, Working With Text Data