Língua Natural 2018/2019

Grupo 43
78093– Gonçalo Santos
79197– Maria Oliveira

Relatório do 2º Miniprojeto

I. Introdução

Neste projeto pretende-se classificar questões de cinema, atribuindo-lhes uma classe. Uma vez que nos são dadas questões conhecidas já classificadas, construiu-se um classificador supervisionado.

II. Proposta de solução

De modo a se obter melhores resultados, foi realizado o seguinte pré processamento dos datasets de treino e de teste, pela seguinte ordem :

1. Substituição dos nomes de filmes e atores por uma keyword correspondente;
2. Conversão do texto para letras minúsculas;
3. Substituições, por exemplo "what's" passa a "what is";
4. Remoção espaços no início e no fim de cada frase;
5. Remoção de pontuação;
6. Separação das frases em tokens, usando o NLTK;
7. Remover stopwords, usando o corpus do NLTK;
8. Lematização das palavras, usando também o NLTK;

Substituição por keywords

De modo a reduzir a dimensionalidade do problema e, consequentemente, tornando os dados menos esparsos substitui-se os nomes de filmes e atores por keywords. Adicionalmente, também se reduz a probabilidade do modelo criado sofrer de *overfitting* para determinados filmes ou atores. Apenas podemos aplicar esta técnica uma vez que a especificação destas entidades são irrelevantes para a classificação das frases, e.g. na frase " What actors entered Titanic?" o filme não influencia a classificação da frase como "*actor_name*", apenas interessa que *Titanic* se está a referir a um filme.

Apenas foram utilizámos os recursos *list_movies* e *list_people* uma vez que apenas estes apresentaram melhorias significativas nos resultados, sendo relevantes o suficiente para os dados em questão.

Conversão para letras minúsculas

Todos os dados foram convertidos para letras minúsculas com o objectivo de eliminar a diferenciação entre palavras iguais e.g. "*What"* e "*what".* Isto contribui para um dataset menos esparso.

Expansão de expressões

Tal como no ponto anterior, de modo a evitar o tratamento de expressões idênticas como diferentes. Neste caso apenas se realizou a expansão de "*what's"* para *"what is"* uma vez que era a única que aparecia nos dados utilizados. Apesar disso a língua inglesa tem outras instâncias desta situação.

Remoção de espaços, pontuação e stopwords

Removeram-se os espaços brancos e a pontuação uma vez que estes são irrelevantes para a classificação dos dados em questão. Para além disso removeram-se as palavras mais comuns da língua inglesa (*"stopwords"*) uma vez que estas podem contribuir para classificações erradas de dados. Para isto utilizou-se o *corpus* de *stopwords* do NLTK.

Lematização de palavras

Com o objetivo de agrupar palavras semelhantes utilizou-se o lematizador do *WordNet,* parte do pacote NLTK, e.g. *"actors"* lematizado para *"actor".*

Todos estes passos tiveram como principais objetivo limpar e remover palavras que possam ter e feitos negativos na classificação e a condensação da dimensão do problema. Finalizando este processo resta apenas o treino de um classificador utilizados os dados processados.

De modo a treinar os classificadores utilizando *SklearnClassifiers* recorreu-se à extração de *features* para cada entrada dos dados.

Adotou-se a solução usando Logistic Regression, que é um método de classificação probabilístico, e a partir de um conjunto de observações, permite a predição de classes que uma frase pode ter. Usou-se a biblioteca scikit-learn.

III. Resultados Experimentais e Discussão

Foram testadas várias modelos de classificador, usando a biblioteca scikit-learn:

- K-Nearest Neighbors
- Decision Tree
- Random Forest
- Logistic Regression
- SGD Classifier
- Naive Bayes
- SVM Linear

Obtiveram-se os seguintes valores para a *accuracy*:

```
K Nearest Neighbors accuracy : 85.71428571428571
Decision Tree accuracy : 88.09523809523809
Random Forest accuracy : 83.33333333333334
Logistic Regression accuracy : 90.47619047619048
SGD Classifier accuracy : 85.71428571428571
Naive Bayes accuracy : 83.33333333333334
SVM Linear accuracy : 88.09523809523809
```

Optámos pelo modelo de Logistic Regression uma vez que foi o que apresentou melhor accuracy.

Verifica-se que o processo aplicado produz resultados consistentes com a classificação correta. Apesar disso, uma vez que apenas temos em conta a *accuracy* e o *dataset* de treino não é balanceado não é garantido que este modelo classifique com igual *accuracy* para as diferentes classes possíveis.

Para além disso, uma vez que os recursos utilizados não são 100% relevantes para esta classificação e não englobam todas as entradas possíveis, e.g. o filme “Joan of Arc” não se encontra na lista de filmes dos recursos, é possível melhorar o resultado final realizando uma filtração mais extensiva dos recursos utilizados e/ou a utilização de recursos mais completos e melhor adaptados à classificação relativa aos dados.

IV. Conclusão e trabalho futuro

Através do método descrito anteriormente conseguiu-se um classificador com cerca de 90% de *accuracy* para classificar questões relativas ao cinema. Esta *accuracy* podia ser melhorada utilizando técnicas de pré processamento mais sofisticadas e recursos de maior qualidade.

O método de substituir palavras apenas teve resultados com algum sucesso porque o espaço do discurso utilizado é algo limitado, raramente havendo sobreposição entre o nome de uma pessoa e o título de um filme. Para classificações mais gerais, e até para melhores resultados nesta classificação, deve-se utilizar técnicas de NER (*Named Entity Recognition*) que consigam com sucesso identificar e diferenciar entre várias entidades.

Para além do métodos de classificação referidos, existem outras modelos que utilizam diferentes medidas de semelhança, e.g. Jaccard, que acabámos por não explorar.

V. Bibliografia

Jurafsky, D. and Martin, J. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2nd ed. Upper Saddle River, N.J.: Prentice Hall.

Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O'Reilly Media Inc.

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Logistic regression. In: *Wikipedia.* Disponível em: *https://en.wikipedia.org/wiki/Logistic_regression.* Acesso em: *05/Nov/2018.*