



MP2 - Língua Natural

Elaborado por:

Afonso Gil Sobral Pena Soares - ist1110876

Tiago Miguel Gordo Barreiros - ist1110826

Modelos

Para alcançar as metas de *accuracy* requisitadas, tivemos que testar vários modelos, entre os quais, Soft Vector Machines e Neural Networks. Acabamos por escolher, como modelo final do nosso projeto, SVM com TF-IDF, pois foi com este modelo que alcançamos as melhores métricas e retirámos melhores conclusões.

No que toca a pré-processamento, testamos vários métodos como, *lower casing*, remover *stop-words* (biblioteca: `nltk.corpus`), tokenização (biblioteca: `nltk.tokenize`), *stemming* (PorterStemmer) e *lemmatization* (WordNetLemmatizer) (biblioteca: `nltk.stem`). No entanto, o modelo final apresenta apenas *lower casing*, isto, pois os restantes métodos não nos apresentaram melhorias.

Accuracy com pré-processamento completo representado na figura 1.

```
Accuracy for TRUTHFULPOSITIVE: 0.81
Accuracy for TRUTHFULNEGATIVE: 0.83
Accuracy for DECEPTIVEPOSITIVE: 0.84
Accuracy for DECEPTIVENEGATIVE: 0.80
Overall accuracy: 0.82
```

Figura 1: Imagem ilustrativa dos valores de *Accuracy* com pré-processamento completo.

Ambiente

Antes de apresentar os resultados finais do modelo, vamos falar do ambiente em que este foi testado. Foram nos dados dois *data sets*, no entanto, apenas o `train.txt` foi usado para criar os conjuntos de treino e teste, para posteriormente obtermos as métricas, dito isto, tivemos que dividir

os dados de treino em subconjuntos, fizemos K-Fold Cross Validation (StratifiedKFold) e dividimos o `train_set` em dez, usámos cada uma das partições como treino para obter resultados melhores do que um `train-test-split` normal.

Tomando em conta os parâmetros do modelo, estes foram decididos com a ajuda de Grid Search (GridSearchCV), que configurámos com a seguinte matriz de parâmetros:

```
param_grid = {
    'C': [0.1, 1, 10],
    'kernel': ['linear', 'rbf', 'poly'],
    'gamma': [0.1, 1, 10],
    'degree': [2, 3, 4],
    'coef0': [0.0, 1.0, 2.0]
}
```

Excerto de Código 1: Parâmetros do Grid Search.

Com isto obtivemos que os melhores parâmetros para o nosso modelo eram então `kernel='poly'`, `C=0.1`, `coef0=2.0`, `degree=3` e `gamma=1`. Como última nota do ambiente é importante mencionar que todo o texto, treino e teste, foi vetorizado a partir de um TF-IDF vectorizer (`TfidfVectorizer`).

Resultados

Estes foram avaliados a partir de *accuracy*, representado na tabela 1 e especificamente para cada *label* na matriz de confusão presente na figura 2.

Label	Accuracy
TRUTHFULPOSITIVE	82%
TRUTHFULNEGATIVE	83%
DECEPTIVEPOSITIVE	85%
DECEPTIVENEGATIVE	85%
OVERALL	84%

Tabela 1: Tabela de *Accuracy*.

As matrizes de confusão são usadas para resumir os resultados da classificação do modelo, mostra o número de instâncias da classe que foram classificadas corretamente (valores da diagonal principal) e incorretamente (valores fora da diagonal principal).



	TP	TN	DP	DN
TP	286	26	37	1
TN	24	293	0	35
DP	35	8	295	8
DN	2	43	9	298
39				

Figura 2: Matriz de Confusão de cada *label*.

Realizámos vários testes com algoritmos baseados em Naïve Bayes, Random Forest e Gradient Boosting, mas os resultados obtidos não se mostraram significantes, nem ajudaram a captar grandes conclusões.

Discussão

Explorámos a razão do pré-processamento baixar a *accuracy*, descobrimos que isto é verdade, no entanto, em primeira instância parece-nos afetar mais *reviews* DECEPTIVE que TRUTHFUL.

Ao criarmos um ficheiro com as *reviews* que foram classificadas incorretamente no formato REVIEW\NPREDICTED LABEL\NTRUE LABEL, e comparando os *outputs* com pré-processamento (incluindo remover *stop-words*, tokenização e lematização) e sem pré-processamento, descobrimos que existem 36 *reviews* mal classificadas sem pré-processamento enquanto existem 38 *reviews* mal classificadas com pré-processamento. A maioria das erradas na versão pré-processada aparentam ser DECEPTIVE, o que acreditamos que seja causado, pois, *reviews* enganadoras usam vocabulário específico que ao serem pré-processadas, o texto é perdido.

Temos de assumir que não é fácil classificar *reviews* como DECEPTIVENEGATIVE e DECEPTIVEPOSITIVE, pessoalmente tivemos algumas dificuldades em analisar os dados facultados, apesar de

notarmos algum padrão nas mesmas e algumas serem bastante óbvias, outras pareciam verdadeiras, o que denota a importância dos dados a serem explorados.

Trabalho Futuro

Todos os dados analisados ajudaram-nos a concluir que a sua quantidade e diversidade é importante, mas também a qualidade e o caminho percorrido, pois este último pode-nos distanciar do objetivo.

Acreditamos que existe um longo caminho a percorrer e este caminho passa por mais pesquisa, trabalho e investigação, que vise, por exemplo, otimizar o código de pré-processamento de dados e obter mais dados para treinamento, o que pode levar a resultados melhores. O tempo despendido com Redes Neurais foi curto e acredito que explorar esse meio seria interessante, começando por aumentar o número de camadas ocultas, pois estas podem compreender representações complexas de dados, com bastantes dependências textuais e aprender a hierarquia dessas características, de modo a captar melhor análise de sentimentos.

Contudo, a análise dos resultados permitiu-nos identificar áreas a melhorar e direções futuras para aprimorar o sistema, considerando que o foco futuro seja redirecionado para a análise sintática e do sentido das frases, como a deteção de sarcasmo e a compreensão do contexto, pois algumas *reviews* falsas exprimiam um excesso de sentimento e por vezes alguma descontextualização.