Análise de Sentimento de Tweets
Text Mining 2019-2020

Tiago Almeida, 69362
MEI-SIGCA1
ISCTE-IUL
tfpaa@iscte-iu.pt

Patricia Santos, 64732
MEI-SIGCA1
ISCTE-IUL
psssa@iscte-iul.pt

***Abstract*— No nosso artigo vamos investigar a Análise de Sentimentos através de *tweets*. Numa primeira fase recorremos ao processamento e tratamento de dados, tais como, tratamento de *retweets*, *hashtags*, *urls*, pontuação, *targets*, abreviaturas, *emoticons*, maiúsculas para minúsculas, stop words, palavras alongadas, números, dinheiro e tempo. Depois do tratamento de dados, aplicamos o léxico de sentimentos com e sem negação, com e sem *Stemming e, com e sem Lemmatization* para classificar a polaridade dos *tweets*. Na última fase da análise, aplicamos três algoritmos de aprendizagem automática de forma a saber qual deles se aplicariam melhor ao nosso problema.**

Keywords—twitter, sentiment, sentiment analysis, tweets, text mining

I. INTRODUÇÃO

Hoje em dia o Twitter é uma rede social que permite aos utilizadores exporem todo o tipo de pensamentos e, consequentemente sentimentos, em relação a vários tópicos. A análise de sentimentos em *tweets* é cada vez mais frequente para ajudar a detetar casos de depressão, solidão ou outras situações que coloquem o bem-estar de uma pessoa em causa, através do conteúdo de um *tweet*, como por exemplo, *hashtags*, *url's*, *targets* a outros utilizadores, palavras com caracteres repetidos e emoticons.

Ao longo deste artigo vamos avaliar diferentes abordagens para a análise de sentimento tendo em conta a informação que podemos obter através dos *tweets*.

II. REVISÃO DE LITERATURA

Neste capítulo apresentamos uma revisão de literatura onde analisamos artigos realizados anteriormente e que abordam este tema comparando as abordagens de cada um deles.

Na fase de pré-processamento, uma abordagem comum em todos os trabalhos encontrados durante a nossa investigação, foi o tratamento de *emoticons*. Nos artigos [1] e [2] os autores optaram por remover os *emoticons*, por outro lado, nos artigos [3]–[5] os autores decidiram criar um dicionário, relacionando o *emoticon* com o sentimento associado ao mesmo.

A técnica dos dicionários foi também aplicada para abreviaturas nos artigos [3]–[5], enquanto que nos artigos [1] e [2] os autores não fizeram o tratamento de abreviaturas .

Outro tratamento tido em conta foi em relação a *URLs*, *targets*, negações e *hashtags*. Nos artigos [1], [3], [4] a solução apresentada para os *URLs* e *targets* foi a substituição destas expressões pela palavra *URL*, *USERNAME* ou por ||U|| e ||T||, contudo, para os artigos [2] e [5] a abordagem escolhida foi a remoção dos mesmos.

Relativamente ao tratamento das *hashtags*, no artigo [5] o autor decidiu remove-las, enquanto que nos restantes artigos não foi mencionado tal tratamento. Já em relação ao tratamento da negação, os artigos [2]–[5] são os únicos que relatam tal tratamento.

A remoção de caracteres repetidos foi mais um tratamento tido em conta para os artigos [3]–[5], onde se modificaram palavras com mais de 2 caracteres repetidos para 2 caracteres, 3 caracteres e 1 caracter, respetivamente.

Em relação às *features* mencionadas nos artigos, temos *unigrams*, *bigrams*, a junção de *unigrams* com *bigrams*, *Part-of-Speech* (POS) *tags* e por fim, a junção de POS com u*nigrams*.

Os autores do artigo [1] após determinarem a polaridade dos tweets, tendo em conta as palavras positivas e negativas que se encontravam nos mesmos, aplicaram os classificadores *Naive Bayes*, *Max Entropy* com *Stanford Classifier* e *Support Vector Machine* (SVM) com *kernel* linear.

Os autores concluíram que para o objetivo proposto, as POS *tags* não são úteis e também que o uso de *unigrams* e *bigrams* separadamente não é uma boa escolha, no entanto, a combinação destes 2, apresentou bons resultados acabando por ser a melhor escolha.

Por fim, concluíram também que os classificadores *Naive Bayes* e *Max Entropy* melhoram quando combinados com *unigrams* e *bigrams*.

No artigo [3] os autores optaram por definir *unigrams* como *baseline*, dois tipos de modelos, Feature Based Model e Tree Kernel Based Model por fim, como classificador o SVM com 5-fold cross-validation para testar os resultados obtidos. Como conclusão verificaram que o Feature Based Model e o Tree Kernel Based Model superam o *unigram baseline* e, as features mais importantes são aquelas que combinam a polaridade inicial das palavras com as suas POS *tags*.

Continuando a análise dos artigos lidos, no artigo [4] os métodos baseiam-se num classificador AdaBoost e combinações *unigram* e *bigrams* com ajuda de *features* baseadas na remoção das *stop words*, tratamento da negação, léxico e por fim POS *tags*. Após a análise concluíram que as POS *tags* não são úteis e que o léxico é uma mais-valia **l**

O penúltimo artigo analisado, [5] , após o pré-processamento teve-se em conta as features de POS tags, léxico, tratamento das negações, contagem da frequência que um termo e/ou palavra tem, *unigrams*, *bigrams* e *trigrams*. Para complementar usaram Naive Bayes, SVM e Max Entropy como classificadores, Corpus-Based com abordagem Latent Semantic Analysis, WordNet como dicionário de sinónimos e antónimos.

Com esta abordagem, os auttores concluíram que os melhores classificadores são Naive Bayes e SVM, o léxico

pode ser importante e, o modelo *bigrams* é melhor comparativamente ao *unigrams* e *trigrams*.

Por fim, no último artigo [2] utilizam a ferramenta, TreeTagger, como auxiliar de anotações de tweets com POS tags, contabilizaram o número de palavras positivas e negativas e recorreram aos modelos de *unigrams, bigrams, trigrams.* Os autores concluíram que as POS *tags* são um forte indicador e o modelo *bigrams* é considerado um melhor modelo em relação ao *unigrams* e *trigrams.*

III. TRATAMENTO DOS DADOS E *BASELINE*

O conjunto de dados escolhido para análise foi o "Tweets_EN_sentiment.json". Cada dado deste conjunto representa um *tweet* e dependendo do sentimento associado encontra-se etiquetado como positivo ("pos") ou negativo ("neg"). Para obtermos uma análise mais justa fizemos um balanceamento dos dados, uma vez que existem muito mais tweets positivos do que negativos, o que provocaria resultados desajustados à realidade, para colmatar este desfasamento procedemos a uma redução dos dados de maneira a encontrar um equilíbrio entre o número de *tweets* com sentimento positivo e negativo. Após o balanceamento obtivemos um total de 17088 *tweets* sendo que 80% são para treino dos modelos (6795 positivos e 6875 negativos) e os restantes 20% (1749 positivos e 1669 negativos) para testar esses modelos.

A. *Baseline*

A *Baseline* foi criada recorrendo ao *TextBlob* antes da preparação e tratamento dos dados. Com isto obtivemos os seguintes resultados das previsões dos modelos:

- *True positives* : 995
- *True Negatives* : 560
- *False Positives* : 490
- *False Negatives* : 1373

Observando estes valores concluímos que foi possível prever corretamente o sentimento de 1555 *tweets* e que 1863 tiveram uma previsão errada. Estes resultados foram alcançados através da polaridade do TextBlob, sendo que quando a polaridade apresentava valores maiores que 0, era classificado como sentimento positivo e quando a polaridade era menor ou igual a 0 era classificado como negativo.

Depois de obtermos esta informação das métricas de performance, *Accuracy*, *Precision*, *Recall* e *F-Measure*, em que para cada uma delas usámos fórmulas apresentadas nas imagens abaixo:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \times 100$$

$$Precision = \frac{TP}{TP + FP} \times 100$$

$$Recall = \frac{TP}{TP + FN} \times 100$$

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100$$

Após a definição das fórmulas e cálculos das métricas os valores obtidos foram:

- *Accuracy:* 45.49%
- *Precision:* 67%
- *Recall*: 42.02%
- *F-Measure:* 51.65%

B. *Preparação e tratamento dos dados*

Nesta fase de preparação e tratamento dos dados, vamos enumerar os procedimentos tidos em conta para o nosso conjunto de dados bem como a alteração tida em conta.

1. *Retweets*
   Para o tratamento de tweets repetidos decidimos remover aqueles que no seu texto contivessem a palavra "RT", visto ser desnecessário o processamento de tweets com o mesmo conteúdo.

2. *Hashtags*
   Conversão do caracter de *hashtag* "#" para a palavra *hashtags*, com esta alteração ficamos com a palavra *hashtag* e a palavra que já existia anteriormente.

3. *URL's*
   Muitos dos *tweet*s não têm só textos mas também *url's* para músicas, vídeos, entre outros conteúdos de multimédia. No nosso entender, os *url's* não alteram o sentimento associado ao *tweet* por isso, decidimos remover as referências *url's*.

4. Pontuação
   A pontuação está presente em praticamente quase todos os tweets e de forma a melhorar o tratamento, decidimos remover a pontuação presente.

5. *Targets*
   Mais uma vez e como aconteceu nos *url's,* decidimos remover as menções a outros utilizadores, por não acrescentar nenhum valor ao sentimento do *tweet.*

6. Conversão Maiúsculas para Minúsculas
   Para evitar que a mesma palavra, escrita em maiúsculas ou minúsculas, seja interpretada de maneira diferente, por exemplo "Olá" e "olá", decidimos converter todos os caracteres para minúsculas.

7. Números, Dinheiro e Tempo
   Todas as referências a números, dinheiro e tempo/horas foram convertidas para a palavra "*number*", "*money*" e "*time*", respetivamente.

8. Caracteres repetidos numa palavra
   Existem *tweets* que transmitem um sentimento mais intenso por ter a repetição de um ou mais caracteres numa palavra, na tentativa de demonstrar entusiasmo ou outro sentimento de euforia. Para estas situações retirámos os caracteres repetidos e duplicamos a palavra em causa continuando a intensificar o sentimento do *tweet*, por exemplo, "yeeeessss" passa a ser representado por "yes yes".

9. Abreviaturas
   Uma situação bastante presente nos *tweets* é o recurso a abreviaturas para certas palavras, como por exemplo, "fb" para a palavra "facebook". Para nos ajudar a interpretar e associar um significado a estas abreviaturas, decidiu-se criar um dicionário composto por 123 abreviaturas e respetiva interpretação e, quando encontradas num *tweet* efetuou-se a conversão para o seu significado.

10. *Emoticons*
    Uma vez mais recorremos a um dicionário com 71 emoticons, que contem o emoticon e o significado associado ao mesmo. Com isto, E, tal como nas abreviaturas convertemos o emoticon no seu significado com a ajuda do dicionário.

11. Stop words
    Segundo o artigo [5] as *stop words* afetam negativamente as previsões, por isso, decidimos remover dos nossos *tweets* palavras que sejam consideradas *stop words* como por exemplo, *"the", "an", "for", "but", "yet", "towards"* entre outras.

12. Negação
    Por fim, quanto ao último procedimento para o nosso tratamento de dados, todas as palavras que se encontrem depois de uma palavra de negação, i.e., depois de "no", "not" ou palavras terminadas "n't", vão ter um prefixo "_NEG" até se encontrar um caracter de pontuação. Para este procedimento é necessário manter a pontuação presente nos tweets, portanto não se aplica o tratamento da pontuação neste caso.

IV. LÉXICO

Nesta etapa, construímos um classificador de sentimentos com a ajuda do léxico NCR *Word-Emotion Association Lexicon* (EmoLex). Este léxico é um ficheiro CSV constituído por várias colunas, onde as mais relevantes são a "English", "Positive" e "Negative". A informação destas colunas complementa-se uma vez que a primeira, "English", diz respeito às palavras que se encontram em inglês e, tanto a segunda como a terceira colunas, "Positive" e "Negative", se referem à classificação da palavra, positiva ou negativa, respetivamente.

Quando a coluna "Positive" assumia o valor 1 e a coluna "Negative" assumia o valor 0, ou se ambas as colunas assumiam o valor 1, a palavra era considerada como uma palavra positiva. No caso em que a coluna "Positive" assumia o valor 0 e a coluna "Negative" assumia o valor 1, ou se em ambas assumiam o valor 0, a palavra era classificada como palavra negativa.

Após a implementação do tratamento de dados no léxico, como foi anteriormente explicado, concluímos que nas 14182 palavras disponíveis no léxico, 8708 eram palavras neutras, 2231 palavras positivas e, por último, 3243 palavras negativas. Visto que o número de palavras neutras é elevado (como palavra neutra assumimos que ambas as colunas "Positive" e "Negative" assumem o mesmo valor), decidimos considerar como palavra positiva quando ambas as colunas assumem o valor 1 e palavra negativa quando ambas as colunas assumem o valor 0.

Antes de aplicar o léxico, foi aplicada uma técnica para saber o número de ocorrências de cada palavra num tweet. Este dicionário tem como informação a palavra encontrada no tweet e respetivamente o seu número de ocorrências nesse tweet.

Para perceber qual é a melhor abordagem para o léxico de sentimentos, fizemos as seguintes experiências:

1. Léxico sem tratamento da negação: Para classificar cada tweet, verificámos a polaridade de cada palavra presente no texto, através do léxico e fizemos a contagem dessa mesma polaridade, se o número de palavras positivas fosse maior do que o número de palavras negativas, o tweet era classificado como positivo e caso contrário era classificado como negativo.
2. Léxico com *WordNet Lemmatization*: Nesta abordagem é aplicado o *Lemmatization* ás palavras correspondentes e de seguida aplica-se o processo anterior.
3. Léxico com Stemming: É aplicado Stemming à palavra respetiva e de seguida é aplicado o processo referido no ponto 1.
4. Léxico com o tratamento da negação: Todas as palavras que se encontrem depois de uma palavra de negação, i.e., depois de "no", "not" ou palavras terminadas "n't", vão ter um prefixo "_NEG" até se encontrar um caracter de pontuação. Para este procedimento é necessário manter a pontuação presente nos tweets, portanto não se aplica o tratamento da pontuação. Posteriormente é apagado o prefixo "_NEG" e é pesquisado no léxico se esta palavra existe, caso exista é classificada como negativa.
5. Léxico com o tratamento da negação e *WordNet Lemmatization*: Aplicada a abordagem do ponto 4 e antes da pesquisa no léxico é aplicada a abordagem do ponto 2, caso exista no léxico é classificada como palavra negativa.
6. Léxico com o tratamento da negação e Stemming: Aplicada a abordagem do ponto 4 e antes da pesquisa no léxico é aplicada a abordagem do ponto 3, caso exista no léxico é classificada como palavra negativa.

Os resultados obtidos das diferentes abordagens explicadas acima foram:

	Accuracy	***Precision***	***Recall***	***F-Measure***
Léxico sem tratamento negação	36.4 %	65.1 %	30.6 %	41.6 %
Léxico com WordNet	38.3 %	65.1 %	32.5 %	43.3 %
Léxico com Stemming	33.9 %	65.8 %	27.4 %	38.6 %
Léxico com tratamento negação	51.0 %	51.6 %	97.2 %	67.4 %
Léxico com tratamento negação e WordNet	51.1 %	51.6 %	97.3 %	67.4 %
Léxico com tratamento negação e Stemming	51.0 %	51.6 %	97.2 %	67.4 %

Com estes valores conseguimos verificar que a utilização do léxico isoladamente apresentou piores resultados comparando com o *Baseline* definido no início desta análise contudo, a utilização das outras técnicas com o léxico mostram resultados melhores sendo que o que obteve um melhor resultado foi a utilização do Léxico com tratamento de negação e *WordNet Lemmatization*.

V. APRENDIZAGEM AUTOMÁTICA

Na aprendizagem automática desenvolvemos funções auxiliares de suporte aos algoritmos de Aprendizagem automática, com tratamento geral. Os métodos aplicados foram *WordNet Lemmatization*, *Stemming*, POS-*tagging* e tratamento da negação e, os algoritmos utilizados foram *Naive Bayes*, *Logistic Regression* e SVM.

Combinámos os métodos descritos anteriormente com estes algoritmos na perspetiva de alcançar o melhor resultado possível. Estes foram os nossos melhores resultados:

	Accuracy	***Precision***	***Recall***	***F-Measure***
Naïve Bayes	54,3 %	60,5 %	30,2 %	40,3 %
Logistic Regression	55,9 %	54,9 %	76,5 %	63,9 %
SVM	55,5 %	54,5 %	77,6 %	64,0 %

	Previsões corretas	***Previsões erradas***
Naïve Bayes	1848	1555
Logistic Regression	**1903**	**1500**
SVM	1889	1514

VI. CONCLUSÃO

Concluímos que a análise de sentimentos é um processo muito importante na atualidade e tem vindo a ser vastamente explorado ao longo dos últimos anos. Têm sido aplicados vários mecanismos e abordagens, com o objetivo de alcançar melhores resultados, tal como foi descrito no trabalho relacionado dos artigos abordados. Ficou bastante claro que o tratamento dos dados é um fator determinante para se conseguir melhores resultados de classificação. No entanto o léxico sem tratamento da negação ou com *Lemmatization* ou com *Stemming*, demonstraram piores resultados face ao Baseline. Já com o tratamento da negação em ambos, demonstrou melhorar bastante os resultados.

No final da nossa investigação concluímos que Logistic Regression e SVM mostram ser os melhores algoritmos para analisar sentimentos de *tweets.Quanto* às técnicas aplicadas Stemming demonstrou ser a mais eficiente enquanto que POS-tagging piorou os resultados obtidos na Baseline. No nosso caso o tratamento da negação não teve um impacto relevante visto que os resultados não apresentaram melhoras.

VII. REFERÊNCIAS

[1] A. Go, R. Bhayani, e L. Huang, «Twitter Sentiment Classification using Distant Supervision», p. 6, Jan. 2009.

[2] V. N. Patodkar e S. I.R, «Twitter as a Corpus for Sentiment Analysis and Opinion Mining», *IJARCCE*, vol. 5, n. 12, pp. 320–322, Dez. 2016.

[3] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, e R. Passonneau, «Sentiment analysis of twitter data», em *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, 2011, pp. 30–38.

[4] E. Kouloumpis, T. Wilson, e J. Moore, «Twitter Sentiment Analysis: The Good the Bad and the OMG!», p. 4, Jul. 2011.

[5] V. A. e S. S. Sonawane, «Sentiment Analysis of Twitter Data: A Survey of Techniques», *Int. J. Comput. Appl.*, vol. 139, n. 11, pp. 5–15, Abr. 2016.