Implementação baseada no artigo “A Rule-Based Approach to Implicit Emotion Detection in Text”

Vinicius N. Silva[1]

[1]Centro de Matemática, Computação e Cognição

Universidade Federal do ABC (UFABC) - Santo André, São Paulo - Brasil

vinicius.narciso@aluno.ufabc.edu.br

Abstract. *This paper aims to present an implementation proposed in the paper “A Rule-Based Approach to Implicit Emotion Detection in Text”, authored by Orizu Udochukwu and Yulan He, from Aston University, UK. Besides that, will be presented comparisons between results obtained by my implementation and the results of the paper, describing step-by-step what was done.*

Resumo. *Este artigo visa apresentar uma implementação proposta no artigo “A Rule-Based Approach to Implicit Emotion Detection in Text”, de autoria de Orizu Udochukwu e Yulan He, da Universidade de Aston, no Reino Unido. Além disso, serão apresentadas comparações entre os resultados obtidos pela minha implementação e os resultados do artigo, descrevendo passo a passo o que foi feito.*

1. Introdução

A análise de sentimentos é uma subárea muito popular e relativamente recente em Processamento de Linguagem Natural. Os primeiros artigos datam do final dos anos 90 e cada vez mais temos um aumento de artigos publicados sobre o tema. O desenvolvimento desta área permite que empresas (principalmente) consigam processar digitalmente as avaliações que contém pareceres sobre elas e utilizar a informação da maneira que lhe for mais conveniente. Essa é apenas uma das várias aplicações que este tema pode auxiliar no processamento da informação.

Entretanto, entender o que uma pessoa (ser biológico, cheio de subjetividade) quer transmitir com seu comentário ou avaliação não é tarefa das mais simples para um computador. Este artigo busca atacar uma das grandes dificuldades dessa área, que é justamente diminuir a subjetividades de textos, procurando emoções implícitas em textos.

2. Modelagem

O problema foi modelado com base no artigo-base utilizado para este trabalho. Toda a implementação foi feita na linguagem Python, utilizando-se de sua biblioteca NLTK para facilitar algumas operações de análise objetiva (operações de separação de sentenças, por exemplo) e subjetiva (correlacionar gramaticalmente os verbetes).

Como proposto no artigo, primeiramente foi mapeada a tabela de regras:

Table 1. Rules for emotion detection.

Input Variables				*Output*
Direction	**Tense**	**Overall polarity**	**Event polarity**	**Emotion**
Self	Future	Positive	Positive	Hope
Self	Future	Negative	Negative	Fear
Self	Present	Positive	Positive	Joy
Self	Present	Negative	Negative	Distress
Self	Past	Positive	Positive	Satisfaction
Self	Past	Negative	Negative	Fears-confirmed
Self	Past	Positive	Negative	Relief
Self	Past	Negative	Positive	Disappointment
Other	All	Positive	Positive	Happy-for
Other	All	Negative	Positive	Resentment
Other	All	Positive	Negative	Gloating
Other	All	Negative	Negative	Sorry-for

(a) Event-based.

Input Variables		*Output*
Direction	**Polarity**	**Emotion**
Self	Positive	Pride
Self	Negative	Shame
Other	Positive	Admiration
Other	Negative	Reproach

(b) Action-based.

Input Variables		*Output*
Event	**Action**	**Emotion**
Joy	Pride	Gratification
Distress	Shame	Remorse
Joy	Admiration	Gratitude
Distress	Reproach	Anger

(c) Compound emotions.

Após o pré-processamento do texto, o desafio é calcular cada uma das variáveis envolvidas nesta tabela. Note que esta implementação foi focada apenas na parte (a) da tabela, dado que é o maior conjunto de variáveis a se calcular e cobrir uma grande quantidade de dados.

3. Implementação e Desafios

A implementação do algoritmo proposto foi feita em Python, utilizando a biblioteca NLTK, a qual disponibiliza diversas funções para auxiliar o pré-processamento do texto.

No geral, a implementação consistiu nos seguintes passos:

- Obter as sentenças do arquivo;
- Tokenizar as sentenças;
- Tokenizar as palavras;
- POS tagging de cada token;
- Encontrar a estrutura gramatical das frases;
- Analisar o tempo verbal;
- Calcular a polaridade geral da frase;
- Calcular a polaridade da relação entre verbo e predicado.

Após todo esse processo, eu tinha todas as ferramentas disponíveis para: 1) Calcular os valores das variáveis; e 2) Estimar o sentimento implícito da frase.

A dificuldade se concentrou na parte de calcular a polaridade da relação entre o verbo e o predicado. A primeira dificuldade foi filtrar apenas a parte em que há a relação. Por exemplo, imagine a frase “Eu falo com você mais tarde”. Para esta frase, temos a parte “falo com você mais tarde” pertencente ao predicado. Porém nem todos os termos estão relacionados com o verbo. Como ponto de melhoria do meu código, sugiro um melhor filtro

para extração dessas relações, tentando padronizar esta extração. Por hora, foi pego apenas a parte relacionada com o verbo, o que na maior parte dos casos, já resolve o problema, visto que o complemento atuaria apenas como um multiplicador, isto é, não mudaria a polaridade, apenas a intensificaria. Para a implementação dessa parte em específico, foi utilizado um desambiguador, de modo que melhore a assertividade da polaridade.

3.1. Complexidade Computacional

Quanto a complexidade computacional, podemos analisar da seguinte forma:

- Obter as sentenças do arquivo; (O(sent), onde sent é o número de sentenças do arquivo)
- Tokenizar as sentenças; (O(sent), onde sent é o número de sentenças do arquivo)
- Tokenizar as palavras; (O(words), onde words é o número de palavras contidas no arquivo)
- POS tagging de cada token; (O(words), onde words é o número de palavras contidas no arquivo)
- Encontrar a estrutura gramatical das frases; (O(words), onde words é o número de palavras contidas no arquivo)
- Analisar o tempo verbal; (O(1), pois, como nesse passo, já foi feito o POS tagging, basta pegar o verbo e analisar)
- Calcular a polaridade geral da frase; (O(words), onde words é o número de palavras contidas no arquivo)
- Calcular a polaridade da relação entre verbo e predicado. (O(words), embora seja apenas uma parte da sentença, é necessário analisar quais são as palavras contidas nessa relação)

Sendo rigoroso, a complexidade é: max(O(sent), O(words), O(1)). Porém, visto que words é pelo menos igual a sent e 1, e que, na maior parte das vezes, words é maior que sent, podemos estimar que a complexidade seria O(words). Vale salientar que words pode ser escrito em função do número de registros e sentenças, visto que, quanto mais registros, mais sentenças e, consequentemente, mais palavras. Polinomialmente, podemos estimar que o grau de words é 3, caso words = reg * sent * palavras. Portanto, uma boa estimativa para este algoritmo é $O(n^3)$.

4. Comparação de resultados

A técnica foi aplicada na base de dados do ISEAR, uma das bases também utilizadas no artigo, e servirá de base de comparação entre a implementação dos autores com a minha. Vale salientar que algumas partes do artigo não foram implementadas, por questão de que a complexidade, versus o tempo disponível para minha implementação não tinha um custo-benefício alto.

4.1. Resultados do artigo

Primeiramente, é necessário dizer que o artigo utilizou 3 bases de dados para analisar quão melhor é a técnica aplicada para melhorar a classificação de sentimento em frases. A tabela abaixo mostra os resultados obtidos em cada uma das bases:

Table 2. Statistics of the datasets. "Total" denotes the original number of sentences in each emotion category while "Implicit" denote the number of sentence which do no contain any emotion words according to WordNet-Affect.

Emotion	Total	Implicit
Joy	1095	537
Fear	1095	366
Anger	1096	483
Sadness	1096	488
Disgust	1096	484
Shame	1096	581
Guilt	1093	482
Total	7667	3421

(a) **ISEAR**

Emotion	Total	Implicit
Joy	362	317
Fear	160	130
Anger	66	60
Sadness	202	182
Disgust	26	24
Surprise	184	160
Total	1000	873

(b) **SemEval**

Emotion	Total	Implicit
Happy	406	103
Fearful	121	33
Angry-Disgusted	174	84
Sad	247	90
Surprised	92	50
Total	1040	360

(c) **Alm's**

Cada base tem sua peculiaridades, por este motivo, para cada uma foi avaliada um conjunto de sentimentos diferentes.

Para avaliação dos resultados, foi utilizada a pontuação F-measure:

Table 3. Performance comparison of F-measure results on the three datasets. Bold face values denote the best results obtained in each dataset.

Emotion	ISEAR			SemEval			Alm's		
	Lexicon	NB	Rule	Lexicon	NB	Rule	Lexicon	NB	Rule
Joy/Happy	33.4	61.2	**69.6**	39.7	**71.7**	59.9	58.8	63.5	**81.8**
Fear/Fearful	0	**47.6**	18.3	0	**52.2**	31.8	0	**26.7**	14.0
Anger/Angry-Disgusted	23.0	47.1	**61.3**	55.8	16.2	**61.3**	48.9	58.6	**86.6**
Sadness/Sad	25.6	55.4	**68.0**	47.8	56.0	**71.5**	61.0	56.0	**79.6**
Disgust	25.6	**51.0**	39.2	38.5	34.5	**61.7**	-	-	-
Average	21.5	**52.5**	51.3	36.4	**58.2**	57.3	42.2	56.0	**65.5**
Average (− `Fear`)	27.0	53.7	**59.5**	45.5	44.6	**63.6**	56.12	65.8	**82.7**

Para essa avaliação, foram montados dois modelos: O primeiro consiste num modelo de matching léxico que utiliza um Lexicon de emoções NRC. O segundo foi a implementação de treino supervisionado de um classificador Naive Bayes no Weka (um software de machine learning).

Dessa análise, podemos perceber duas coisas: no geral, matching léxico performa muito mal e que o baseline em Naive Bayes tem um desempenho bem próximo da proposta do artigo em bases com um grande número de registros.

Estatisticamente, por termos uma pequena quantidade de bases testadas, teremos resultados esmagadores a favor da proposta do artigo, mas é interessante observar que, para algumas bases, o desempenho do aprendizado supervisionado é equiparável ao algoritmo proposto.

4.2. Meus resultados

Antes de apresentar os resultados, é necessário salientar alguns pontos:

- O primeiro ponto, como dito anteriormente, apenas uma parte do algoritmo foi implementada. Por este motivo, não é esperado de que os resultados sejam

iguais aos obtidos pelo artigo. Como será mostrado, para algumas emoções, os resultados são muito próximos.

- O segundo ponto, foram feitos alguns arredondamentos: como as variáveis de polaridade utilizam o valor neutro, caso alguma polaridade seja diferente de neutro, ela predominaria sobre o neutro, trazendo mais resultados para a polaridade Joy/Sadness, o que explica a grande quantidade de emoções avaliadas para estas categorias. O outro arredondamento é em questão de sentenças: analisei apenas a primeira sentença de cada resposta. Muito mais por uma questão de simplicidade, visto que, embora existam 7667 registros, foram computadas 11649 sentenças (foram ignoradas 3982 sentenças). Por questão de tempo e complexidade da solução, foram ignoradas essas sentenças.
- O último ponto trata de casos que são totalmente neutros. Para esses casos, é impossível de levar o resultado para algum pólo. Por este motivo, foi criada uma nova categoria, chamada de Impossible To Determinate (ITD).

Após essas observações, os resultados obtidos foram:

JOY	2457
SADNESS	1799
FEAR	1450
ANGER	431
DISGUST	316
ITD	1213

A primeira diferença a se notar é a grande quantidade de registros que foram classificados, comparando com os resultados do artigo (3421 do artigo contra 6453 da minha implementação). Uma boa parte dessa maior avaliação se deve aos arredondamentos.

5. Conclusões

Após toda a implementação do código, é possível concluir que esse método baseado em regras pode obter bons resultados, porém há algumas ressalvas, tanto a sua aplicatividade quanto a sua implementação.

O método é muito bem aplicável em algumas situações, em outras ele fica bem semelhante à uma aplicação de Naive Bayes. Ou seja, mesclando as duas propostas, talvez seja o melhor dos mundos, pois há situações que o NB performa muito melhor, e quando mesclada com os casos em que a proposta leva a melhor, podemos ter uma melhora significativa de precisão.

Quanto à implementação, a tabela consiste no maior problema. Para aumentar a precisão do sentimento, seria necessário aumentar a relação das variáveis ou até mesmo a quantidade delas, criando por consequência mais relações. Visto que o maior custo e complexidade do algoritmo se dá em calcular os valores das variáveis e a interações entre elas, quanto maior a precisão necessária, maior a complexidade e vice-versa.

Talvez mais alguns refinamentos no método o tornem comercialmente viável, porém, por hora, abre um leque de possibilidades de pesquisas na academia.

References

Udochukwu, O. and He, Y. (2015) “A Rule-Based Approach to Implicit Emotion Detection in Text”, https://publications.aston.ac.uk/id/eprint/27397/1/Implicit_emotion_detection_in_text.pdf, June.

bogdani (2016), “Getting Started with Sentiment Analysis“, https://nlpforhackers.io/sentiment-analysis-intro/, November