# Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination

Êmile Lopes, Carol Bras

OLGA KOLCHYNA, THÁRSIS T. P. SOUZA, PHILIP C. TRELEAVEN AND TOMASO ASTE

DEPARTMENT OF COMPUTER SCIENCE, UCL, GOWER STREET, LONDON, UK

SYSTEMIC RISK CENTRE, LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCES, LONDON, UK

# Introdução

"…an area of research that investigates people's opinions towards different matters: products, events, organisations" (Bing, 2012)

Abordagens:

- Baseada em dicionário
- Aprendizado de máquina

# Pré-processamento de dados

- Part-of-Speech Tagging (POS)
- Stemming and lemmatisation
- Stop-words removal
- Tratamento de negações
- But-clauses
- Tokenisation into N-grams

# Lexicon-Based Approach

**patty spivot**
@flasht

Follow

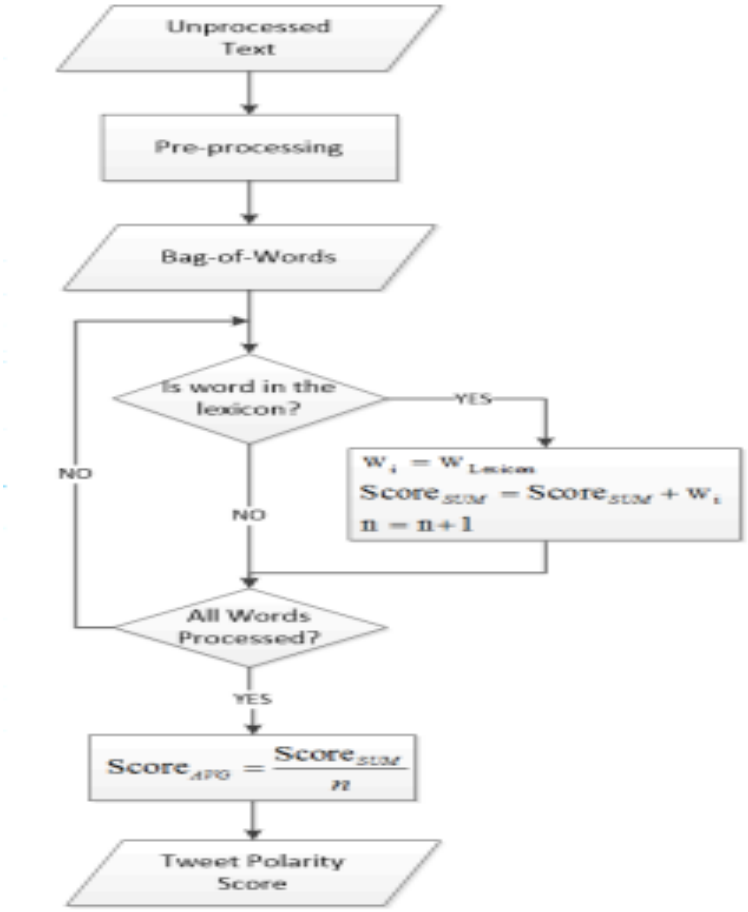
esse tweet eu gostaria de dedicar p galera q
legenda série,
gente fina
manja dos idioma
coloca recadinho no final
faz s/ nada em troca
amo

$$Score_{AVG} = \frac{1}{m}\sum_{i=1}^{m} W_i.$$

***Pré-processamento -> BOW -> Compara -> Calcula Sentimento -> -1 / 0 / +1***

# Lexicon-Based Approach

- Manualmente
- Base de dados treinada (Aparição)
- Bootstraping Techniques (AND)



# A Machine Learning Based Approach

- Pré-processamento
- Feature Generation
  - Number of words with positive/negative sentiment;
  - Number of negations;
  - Length of a message;
  - Number of exclamation marks;
  - Number of different parts-of-speech in a text (for example, number of nouns, adjectives, verbs);
  - Number of comparative and superlative adjectives
- Feature Selection

# Learning an Algorithm

- Search procedure. A process that generates a subset of features for evaluation. A procedure can start with no variables and add them one by one (forward selection) or with all variables and remove one at each step (backward selection), or features can be selected randomly (random selection).
- Evaluation procedure- Estatístico
- Stopping criterion. The process of feature selection can be stopped based on a: i) search procedure, if a predefined number of features was selected or predefined number of iterations was performed; ii) evaluation procedure, if the change of feature space does not produce a better subset or if optimal subset was found according to the value of evaluation function.
- **Classificadores:** Decision Trees, Naive Bayes, SVM

# Model Evaluation

- Accuracy: correto em meio ao numero de predições
- Error Rate: errado em meio ao numero de predições

$$F\text{-}Score = \frac{2 * Precision * Recall}{Precision + Recall}.$$

Table 3: Example of ArkTweetNLP (Gimpel et al., 2011) tagger in practice.

**Sentence:**
**ikr smh he asked fir yo last name so he can add u on fb lolololol**

| word | tag | confidence |
|---|---|---|
| ikr | ! | 0.8143 |
| smh | G | 0.9406 |
| he | O | 0.9963 |
| asked | V | 0.9979 |
| fir | P | 0.5545 |
| yo | D | 0.6272 |
| last | A | 0.9871 |
| name | N | 0.9998 |
| so | P | 0.9838 |
| he | O | 0.9981 |
| can | V | 0.9997 |
| add | V | 0.9997 |
| u | O | 0.9978 |
| on | P | 0.9426 |
| fb | ^ | 0.9453 |
| lolololol | ! | 0.9664 |

"ikr" means "I know, right?", tagged as an interjection.
"so" is being used as a subordinating conjunction, which our coarse tagset denotes P.
"fb" means "Facebook", a very common proper noun (^).
"yo" is being used as equivalent to "your"; our coarse tagset has posessive pronouns as D.
"fir" is a misspelling or spelling variant of the preposition for.
Perhaps the only debatable errors in this example are for ikr and smh ("shake my head"): should they be G for miscellaneous acronym, or ! for interjection?

**Negative Handling** -
Polarizar da palavra negativa até o pontuação

N(common noun), V(verb), A(adjective), R(adverb), !(interjection), E(emoticon), G(abbreviations, foreign words, possessive endings).

$$positiveSentScore = \frac{\#Positive\ sentences}{(\#Positive\ sentences + \#Negative\ sentences)}.$$

$$positiveSentScore = \frac{\#Positive\ sentences}{(\#Positive\ sentences + \#Negative\ sentences)}.$$

[3]

For example, we calculated that the word "*pleasant*" appeared 122 times in the positive sentences and 44 times in the negative sentences. According to the formula, the positive sentiment score of the word "*pleasant*" is

$$positiveSentScore = \frac{122}{(122 + 44)} = 0.73.$$

Similarly, the negative score for the word "*pleasant*" can be calculated by dividing the number of occurrences in negative sentences by the total number of mentions

$$negativeSentScore = \frac{\#Negative\ sentences}{(\#Positive\ sentences + \#Negative\ sentences)},$$

[4]

$$negativeSentScore = \frac{44}{(122 + 44)} = 0.27.$$

**+ 0.6**
**0 [0.4; 0.6]**
**- 0.4**

| | GOOD | BAD | LIKE |
|---|---|---|---|
| Positive Score | 0.675 | 0.213 | 0.457 |
| Negative Score | 0.325 | 0.787 | 0.543 |

$$PolarityScore = 2 * positiveSentScore - 1.$$

# Emoticons

| Emoticon | Score | Emoticon | Score | Abbreviation | Score | Abbreviation | Score |
|---|---|---|---|---|---|---|---|
| l-) | 1 | [-( | -1 | lol | 1 | dbeyr | -1 |
| :-} | 1 | T_T | -1 | ilum | 1 | iwiam | -1 |
| x-d | 1 | :-(( | -1 | iyqkewl | 1 | nfs | -1 |
| ;;-) | 1 | :-[ | -1 | iwalu | 1 | h8ttu | -1 |
| =] | 1 | ((( | -1 | koc | 1 | gtfo | -1 |

Table 6: Combinations of lexicons tested

| | Lexicons combinations |
|---|---|
| 1. | OL |
| 2. | OL + EMO |
| 3. | OL + EMO + AUTO |

Table 7: Results of K-Means clustering for different lexicon combinations.

| **Accuracy** | **OL** | **OL + EMO** | **OL + EMO + AUTO** |
|---|---|---|---|
| $Score_{AVG}$ | 57.07% | 60.12% | 51.33% |
| $Score_{Log10}$ | 58.43% | 61.74% | 52.38% |

$$Score_{Log10} = \begin{cases} sign(Score_{AVG}) Log_{10}(|\ 10 Score_{AVG}\ |), & \text{if } |\ Score_{AVG}\ | > 0.1, \\ 0, & \text{otherwise} \end{cases} \quad [6]$$

# Abordagem de Aprendizado de Máquina(aplicação)

- ✓ Pré-processamento + Filtragem + substituição de *tokens*
- ✓ Extração das características

  - N-grams
  - Lexicon Score
  - Número de palavras alongadas
  - Emoticons
  - Último *token*
  - POS
  - Pontuação
  - Número de emoticons
  - Número de *tokens* negativos
  - Número de *tokens* positivos

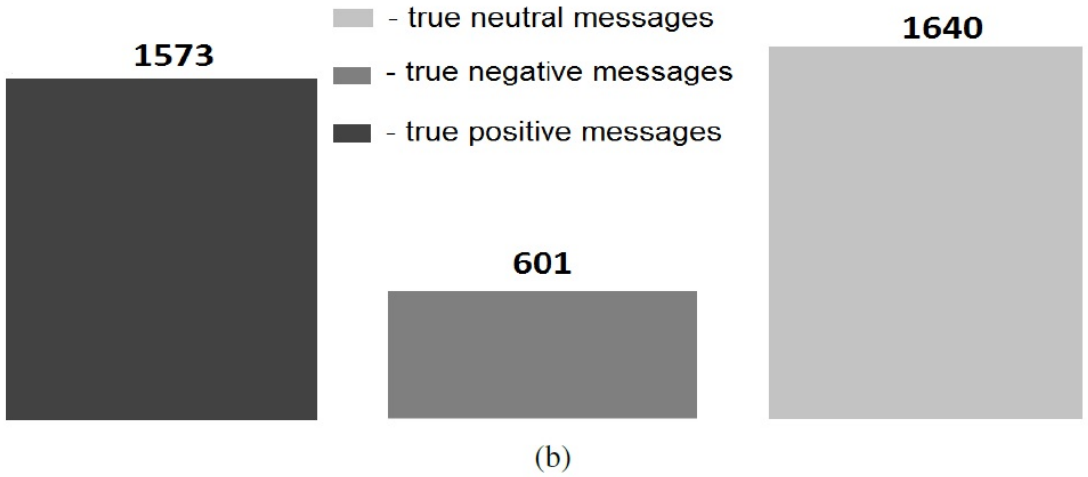
# Abordagem de Aprendizado de Máquina(aplicação) *cont.*

- ✓Seleção das características

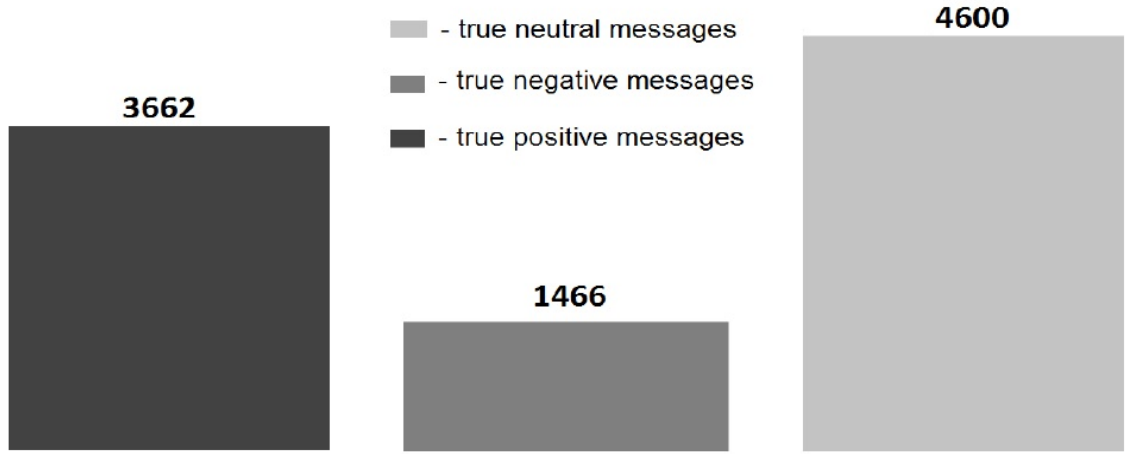
| **TOP FEATURES** | 11. great | 22. fun | 33. hope |
|---|---|---|---|
| 1. LexiconScore | 12. posV | 23. lastTokenScore | 34. thanks |
| 2. maxScore | 13. happy | 24. i love | 35. luck |
| 3. posR | 14. love | 25. don | 36. best |
| 4. minScore | 15. excited | 26. don't | 37. i don't |
| 5. negTokens | 16. can't | 27. amazing | 38. looking forward |
| 6. good | 17. i | 28. fuck | 39. sorry |
| 7. posE | 18. not | 29 love you | 40. didn't |
| 8. posN | 19. posA | 30. can | 41. hate |
| 9. posU | 20. posElongWords | 31. awesome | 42. ... |

# Abordagem de Aprendizado de Máquina(aplicação) *cont.*

- ✓ Treinamento + validação




(b)




(a)

# Resultados

| Classifier | Naive Bayes | Decision Trees | SVM | Cost Sensitive SVM |
|---|---|---|---|---|
| **F-SCORE** | 0.64 | 0.62 | 0.66 | 0.73 |

| TEAM NAME | F-SCORE |
|---|---|
| NRC-Canada | 0.6902 |
| GUMLTLT | 0.6527 |
| TEREGRAM | 0.6486 |
| AVAYA | |
| BOUNCE | 0.6353 |
| KLUE | 0.6306 |
| AMI and ERIC | 0.6255 |
| FBM | 0.6117 |
| SAIL | |
| AVAYA | 0.6084 |
| SAIL | 0.6014 |
| UT-DB | 0.5987 |
| FBK-irst | 0.5976 |

# Conclusão

No artigo foram apresentadas as duas principais abordagens usadas para análise de sentimentos: abordagem baseada em dicionário e o método de aprendizado de máquina. Na abordagem baseada em dicionário foram comparados três dicionários([OL],[OL + EMO], [OL + EMO + AUTO]). Os resultados desta comparação mostram o quanto é essencial a incorporação de expressões como emoticons, gírias e abreviações no dicionário, e que dicionarios muito grandes afetam negativamente a performance do algorítmo. No método de aprendizado de máquina foi proposto o uso do dicionário, gerado pela classificação baseada em dicionário, como principal característica no treinamento dos classificadores. Além disso, foi demostrado que quando a base de dados esta desbalanceada o uso de classificadores *cost-sensitive* aumentam a acurácia da predição das classes: no caso do conjunto de dados tirados do Twitter, uma SVM *cost-sensitive* teve performance 7% maior que uma SVM normal.