Análise de Sentimentos

Prof. Walmes Zeviani

walmes@ufpr.br

Laboratório de Estatística e Geoinformação
Departamento de Estatística
Universidade Federal do Paraná

Justificativa e objetivos

- Mineração de opinião e análise de sentimentos são as técnicas mais usadas de mineração de texto em rede social.
- Dar uma ideia geral e comentar as limitações.
- Apresentar os conjuntos léxicos de sentimentos.
- Fazer uma aplicação simples.

Análise de sentimento

Mineração da opinião

- Opiniões e pontos de vista.
  - E.g. classficação de posição política.
  - Podem haver várias classes.
  - Requer algoritmo de classificação e conjunto para seu treinamento.
  - Dados rotulados nem sempre são acessíveis.
- Análise de sentimento.
  - Determinar a polaridade (+1, 0, -1).
  - Pode ser resolvido com um dicionário de termos classificados.
  - Está associado com extração de informação.

Escopo da análise

- Nível de documento.
- Nível de sentença.
- Nível de palavra.
  - Análise de sentimento.
  - Soma algébrica das polaridades dos termos.
- Problemas com tratamento da negação:
  - Negação pré-verbal.
  - Negação pós-verbal.
  - Negação pré e pós-verbal.
- Problemas com variações imprevisiveis:
  - *Tony Stark? Amoooo!*.
  - *Estou com sdd de vc. Volta logo. ;-* .*

Bases de léxicos de sentimentos para português

- SentiLex-PT 02;
  - Contém 7.014 lemas e 82.347 formas flexionadas.
  - Abrange adjetivos, nomes, verbos e expressões.
  - Possui a polaridade e categoria gramatical de cada palavra.
- OpLexicon 3.0.
  - Contém 32.191 palavras polarizadas e classificadas gramaticalmente.
  - Inclui emoticons e hashtags.

Aplicando análise de sentimentos

Importação do dicionário

```
#-------------------------------------------------------------------
# Lendo o dicionário léxico de sentimentos.

sent <- read.table("oplexicon_v3.0/lexico_v3.0.txt",
                   header = FALSE,
                   sep = ",",
                   quote = "",
                   stringsAsFactors = FALSE)
names(sent) <- c("term", "class", "pol", "ann")
xtabs(~class + pol, data = sent)
```

```
##                pol
## class             -1     0     1
##   adj           12825  6226  5424
##   emot             45     9    12
##   vb             1297  2761  2831
##   vb adj           15     0    59
##   vb adv            4     0    18
##   vb det n prp     41     0    62
##   vb n prp         28     0    63
```

Importação dos textos

```
#-------------------------------------------------------------------------
# Lendo opiniões sobre Capitão América: Gerra Civil na Google Play.

library(XML)

# `pg` é a página tirada da Google Play com o RSelenium.
load("../data/Captain_America_Civil_War.RData")
doc <- htmlParse(pg)
path <- paste0("//div[@class = 'single-review']",
               "/div[@class = 'review-body with-review-wrapper']")

# Extrai os reviews.
reviews <- xpathSApply(doc,
                       path = path,
                       fun = xmlValue)

# Conserta problema de encoding. Deveria ser consertado na leitura.
reviews <- iconv(reviews, to = "iso-8859-1")

# Considerar apenas as primeiras opiniões.
rev <- reviews[1:12]
```

Criação do corpus

```
#-----------------------------------------------------------------------
# Criando o Corpus.

library(tm)

cps <- VCorpus(VectorSource(rev),
               readerControl = list(language = "pt"))
cps
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 12
```

```
replacePunctuation <- content_transformer(
    function(x) {gsub("[[:punct:]]"," ", x)})

# Fazendo as operações de limpeza.
cps <- tm_map(cps, FUN = content_transformer(tolower))
cps <- tm_map(cps, FUN = replacePunctuation)
cps <- tm_map(cps, FUN = removeNumbers)
cps <- tm_map(cps, FUN = stripWhitespace)
```

Criação da matriz de documentos e termos

```
# Criada com o vocabulário existente no corpus.
dtm_cor <- DocumentTermMatrix(cps)
dtm_cor
```

```
## <<DocumentTermMatrix (documents: 12, terms: 309)>>
## Non-/sparse entries: 484/3224
## Sparsity           : 87%
## Maximal term length: 17
## Weighting          : term frequency (tf)
```

```
# Criada com o vocabulário do dicionário.
dtm_dic <- DocumentTermMatrix(cps,
                              control = list(dictionary = sent$term))
dtm_dic
```

```
## <<DocumentTermMatrix (documents: 12, terms: 31648)>>
## Non-/sparse entries: 116/379660
## Sparsity           : 100%
## Maximal term length: 25
## Weighting          : term frequency (tf)
```

Cálculo da polaridade de cada avaliação

```r
# A intersecção entre os termos do corpus e do dicionário.
inter <- intersect(x = Terms(dtm_cor),
                   y = sent$term[sent$pol != 0])
# length(inter)

# Obter o vetor de polaridades associada aos termos na matriz.
lex <- merge(x = data.frame(term = inter,
                            stringsAsFactors = FALSE),
             y = sent,
             sort = FALSE)
# str(lex)

# Remover os termos na DTM que não tem polaridade atribuida.
m <- as.matrix(dtm_cor)
m <- m[, lex$term]

# Verifica dimensões e disposição.
all.equal(colnames(m), lex$term)
```

```
## [1] TRUE
```

```r
# Média aritmética das polaridades por termo em cada documento.
pol <- (m %*% cbind(lex$pol))/rowSums(m)
fra <- sapply(lapply(rev, strwrap, width = 60), "[[", 1)
knitr::kable(data.frame(Polaridade = pol, Fragmento = fra))
```

Polaridade	Fragmento
0.0000000	Mais um do mesmo de sempre da Marvel É divertido, porém
-0.0909091	Capitão América Depois dos eventos de Vingadores: Era de
-0.2000000	Depois dos eventos de Vingadores: Era de Ultron, 'Capitão
-0.2000000	À s Boa dia tia Boa dia tia Marly tá bem ele fica brecopada
0.5555556	Reinaldo junior Gente adorei o filme guerra civil. Só
0.1428571	Muito esperado! Já adicionei a minha coleção de filmes da
1.0000000	Galera da uma Passadinha no meu Canal GALERA EU TÓ
-0.2500000	Como sempre "diálogos que não levam a lugar nenhum" roteiro
1.0000000	Muito bom uito bom Eu mexendo muito bom naquela que apare
0.1428571	Sem dúvidas o melhor filme de "CAPITÃO AMÉRICA" Não é por
0.0000000	Razoável Reúne toda a liga, mas é um pco cansativo,
0.6000000	Essa filme é o Melhor Filme de 2016 O filme prometeu oque

Discussões

- É óbvio que a abordagem utilizada é muito simples.
- É óbvio que você não ficou satisfeito.
- Como lidar com o problema da negação?
- Como aproveitar um `amoooo` e expressões coloquiais?

Classificação

- Uma alternativa é treinar um algorítmo de classificação.
- Vai demandar textos rotulados com a polaridade.
- Vai precisar criar as características.
  - Pode usar a matriz de documentos e termos diretamente.
  - Utilizar métodos de *text to vectors* e *feature learning*.
- Vai precisar treinar para então se aplicado para novos documentos.