ANÁLISE DE SENTIMENTOS EM MENSAGENS DE E-MAILS SOBRE RESOLUÇÃO DE INCIDENTES DE TI

Júlio Campos[1]

Fernanda Araujo Baião[2]

João Carlos de Almeida Rodrigues Gonçalves[3]

Flávia Maria Santoro[4]

Resumo

Este artigo apresenta uma análise dos sentimentos expressos pelos clientes de uma empresa de TI em mensagens de e-mails trocadas com a equipe técnica durante a resolução de incidentes. Para isso, foi utilizada a técnica de análise de sentimentos a fim de investigar duas situações distintas no processo de resolução de incidentes: o sentimento dos clientes cujos incidentes foram classificados como bem-sucedidos, ou seja, resolvidos antes de um prazo estabelecido por um Acordo de Nível de Serviço (SLA), e o sentimento dos clientes cujos incidentes foram classificados como mal-sucedidos, isto é, resolvidos depois do prazo definido por esse indicador de desempenho. A extração de sentimentos foi realizada com a utilização de uma técnica não-supervisionada, a qual não prevê treinamento de um modelo de aprendizado de máquina. Os sentimentos presentes nas mensagens de e-mails foram classificados pela API do Google Cloud Natural Language com a utilização do pacote “googleLanguageR”.

Palavras-chave: Análise de Sentimentos, Mineração de Opinião, Mineração de Texto, R

Abstract

This paper presents an analysis of the feelings expressed by the clients of an IT company in e-mails exchanged with the technical team during the resolution of incidents. For this, the technique of sentiment analysis was used in order to investigate two different situations in the incident resolution process: the sentiment of the clients whose incidents were classified as successful, that is, solved before a deadline established by a Service Level Agreement (SLA), and the sentiment of the clients whose incidents were classified as unsuccessful, that is, resolved after the deadline defined by this performance indicator. The extraction of sentiments was performed with the use of an unsupervised technique, which does not provide training for a model of machine learning. The sentiments in the email messages were graded by the Google Cloud Natural Language API using the "googleLanguageR" package.

Keywords: Sentiment Analysis, Opinion Mining, Text Mining, R

[1] Universidade Federal do Estado do Rio de Janeiro (UNIRIO), julio.campos@uniriotec.br

[2] Universidade Federal do Estado do Rio de Janeiro (UNIRIO), fernanda.baiao@uniriotec.br

[3] Universidade Federal do Estado do Rio de Janeiro (UNIRIO), joao.goncalves@uniriotec.br

[4] Universidade Federal do Estado do Rio de Janeiro (UNIRIO), flavia.santoro@uniriotec.br

Introdução

Segundo Liu (2012, p.7), a análise de sentimentos - também chamada de mineração de opinião - “é o campo de estudo que analisa opiniões, sentimentos, avaliações, atitudes e emoções para entidades como produtos, serviços, organizações, indivíduos, problemas, eventos, tópicos e seus atributos”. O autor ainda afirma que, desde os anos 2000, a análise de sentimentos se tornou uma das áreas de pesquisa mais ativas em processamento de linguagem natural, tendo sido definida pioneiramente no trabalho de Nasukawa e Yi (2003); já o termo mineração de opinião apareceu pela primeira vez no trabalho de Dave, Lawrence e Pennock (2003). Os dois termos representam o mesmo campo de estudos e “ambos focam principalmente nas opiniões que expressam ou implicam sentimentos positivos ou negativos” (LIU, 2012, p.7).

O principal objetivo da análise de sentimentos é “definir técnicas automáticas capazes de extrair informações subjetivas de textos em linguagem natural, como opiniões e sentimentos, a fim de criar conhecimento estruturado que possa ser utilizado por um sistema de apoio ou tomador de decisão” (BENEVENUTO, RIBEIRO e ARAÚJO, 2015).

Liu (2012, p.10) aponta os principais problemas de pesquisa em análise de sentimentos classificados em três níveis de granularidade: nível de documento, nível de frase e nível de entidade e aspecto.

O primeiro nível busca classificar se a opinião de um documento expressa um sentimento positivo ou negativo. Este nível de análise assume que cada documento expressa opiniões sobre uma única entidade como, por exemplo, um produto único.

O segundo nível busca determinar se uma frase de um documento expressa uma opinião positiva, negativa ou neutra. Segundo Wiebe, Bruce e O'Hara (1999), o nível de frase está intimamente relacionado com a classificação de subjetividade, que distingue as frases em objetivas ou subjetivas. Enquanto as frases objetivas são factuais, ou seja, se atém aos fatos, as frases subjetivas expressam pontos de vista e opiniões pessoais.

O terceiro nível já foi chamado por Hu e Liu (2004) como nível de característica (mineração e resumo de opinião baseada em características). Em vez de olhar para as construções linguísticas como documentos, parágrafos ou frases, por exemplo, o nível de entidade e aspecto olha apenas para a opinião em si. Baseia-se na ideia de que uma opinião consiste em um sentimento (positivo ou negativo) e um alvo (de opinião). Por exemplo, embora a frase "apesar de o serviço não ser tão bom, eu ainda adoro esse

restaurante" possua claramente um tom positivo, não se pode dizer que ela seja inteiramente positiva.

Benevenuto, Ribeiro e Araújo (2015) citam duas principais técnicas utilizadas para extrair sentimentos em textos: a supervisionada e a não-supervisionada. Enquanto a primeira exige uma etapa de treinamento de um modelo com amostras previamente classificadas, a segunda não realiza treinamento de modelos de aprendizado de máquina e faz uso de um dicionário de termos. Na técnica não-supervisionada, cada termo está associado a um sentimento, que possui um significado qualitativo ou quantitativo, ou seja, um valor numérico que varia em uma escala de -1 a 1, onde -1 é o valor sentimental mais negativo e 1 o mais positivo.

Objetivo

Nesse contexto, o objetivo deste trabalho é aplicar uma técnica não-supervisionada de análise de sentimentos em um log de dados de uma empresa que presta serviços de infraestrutura de tecnologia de informação e comunicação (TIC) para cerca de uma centena de clientes. Para isso, foi realizada uma análise quantitativa a partir de histogramas ou distribuições de frequências, que permitiram extrair informações dos dados classificados pela técnica de acordo com as ocorrências dos diferentes resultados observados.

Um dos principais processos de negócios da empresa é o de resolução de incidentes de TIC relacionados aos ativos do cliente, por exemplo interrupções do servidor de e-mail ou problemas de conexão de rede. Um incidente é um episódio inesperado e não planejado que, se não for resolvido corretamente, pode causar perda, danos ou mesmo algum tipo de acidente. Quando um cliente relata um incidente, é aberto um ticket no sistema de atendimento da empresa para acompanhar toda a resolução do incidente e registrar toda e qualquer comunicação (através de e-mails) entre o cliente e a equipe técnica da empresa.

Dessa forma, pretende-se analisar os sentimentos dos clientes em relação a falhas apresentadas pelos serviços da empresa após o término do processo de resolução de incidentes em duas situações distintas: o sentimento dos clientes cujos tickets foram classificados como bem-sucedidos, ou seja, resolvidos antes do prazo estipulado por um Acordo de Nível de Serviço (SLA), e o sentimento dos clientes cujos tickets foram classificados como malsucedidos, isto é, resolvidos depois do prazo estipulado por esse indicador de desempenho. Com isso, espera-se verificar se a classificação desses serviços baseada no SLA (bem-sucedido *versus* malsucedido) corresponde ou não aos sentimentos

expressos pelos clientes em comunicações com a empresa. Busca-se descobrir se o sentimento predominante dos clientes que tiveram incidentes resolvidos dentro do prazo estabelecido é necessariamente positivo, o que pode auxiliar a empresa a perceber a necessidade de revisão de seus indicadores de desempenho, ou se os clientes estão satisfeitos com os serviços prestados pela empresa mesmo que seus problemas tenham sido resolvidos dentro de prazos estabelecidos.

Material e Método

Coleta de dados

Uma consulta SQL foi realizada no banco de dados da empresa a fim de coletar uma amostra para estudo. A amostra coletada (arquivo JSON) possui 233 tickets. Cada ticket contém registros de e-mails enviados por clientes sobre tipos de serviços prestados pela empresa categorizados como o rótulo “Relatar Falha”. Os dados são referentes ao ano de 2015. As variáveis presentes na base de dados são:

- **article_id**: identificador de uma mensagem (e-mail) enviada por um cliente;
- **ticket_id**: identificador de abertura de um ticket;
- **ticket_service_type**: tipo de serviço prestado pela empresa, filtrados com o rótulo “Relatar Falha”;
- **ticket_priority_id**: prioridade de atendimento de um ticket;
- **ticket_solution_time**: tempo de solução de um ticket;
- **SLAMissed** (booleano):
  - Y - indica que o tempo de solução de um ticket foi maior que o SLA (Acordo de Nível de Serviço), ou seja, ticket malsucedido;
  - N - indica que o tempo de solução de um ticket foi menor que o SLA, ou seja, ticket bem-sucedido;
- **article_a_body**: o corpo da mensagem de e-mail

As variáveis “article_id”, “ticket_id”, “ticket_priority_id”, “ticket_solution_time” são quantitativas discretas. Todas as demais variáveis são qualitativas nominais.

Limpeza de dados

Após a coleta de dados, foram aplicadas técnicas de mineração de texto na variável "article_a_body", em linguagem R. O objetivo da limpeza foi eliminar das mensagens de e-mails dados irrelevantes para a análise de sentimentos, como quebras de linhas, espaços em branco, números de telefones, endereços de e-mails, urls de sites e assinaturas de e-mails.

Também foi removido todo o histórico de respostas presentes nos corpos das mensagens, pois a sua presença afetaria a pontuação realizada durante a classificação dos sentimentos, o que levaria a uma análise enviesada da base de dados. Ao final da limpeza, o arquivo JSON que antes possuía 168 kB passou a ter 114 kB, uma redução aproximada de 32%.

Google Cloud Natural Language API

Após a limpeza de texto na variável "article_a_body", foi aplicada uma técnica não-supervisionada de análise de sentimentos com a utilização da última versão[5] da Google Cloud Natural Language API. Lançada em novembro de 2016, a Google Cloud Natural Language API disponibiliza poderosos modelos de aprendizado de máquina em uma REST API. Possibilita a extração de informações sobre pessoas, lugares e eventos, bem como entender os sentimentos expressos em um documento de texto.

A definição de análise de sentimentos pela Google Cloud Natural Language API[6] se refere "a atitude de modo geral, positiva ou negativa, expressa no texto". Os sentimentos são representados por dois valores numéricos: score e magnitude. O score dos sentimentos varia entre -1.0 (negativo) e 1.0 (positivo), e corresponde à tendência emocional geral do texto. A magnitude indica a intensidade geral da emoção (positiva e negativa) no texto fornecido e varia entre 0.0 e +infinito.

A escala de pontuação (score range) utilizada pela Google Cloud Natural Language API é mostrada na Figura 1. Um texto que obtém um score entre -1.0 e -0.25 é classificado como negativo; entre -0.25 e 0.25 é classificado como neutro e entre 0.25 e 1.0 é classificado como positivo.

[5] Versão: 2017-08-04

[6] https://cloud.google.com/natural-language/docs/basics?hl=pt-br

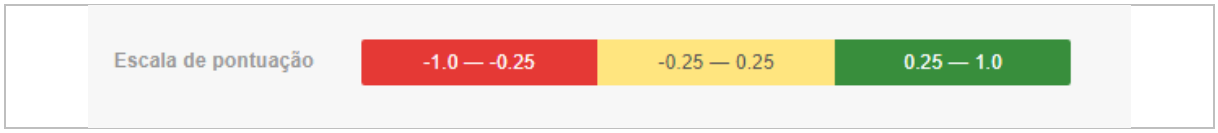

Figura 1 – Escala de pontuação da Google Cloud Natural Language API
Fonte: Google Cloud Natural Language API/Reprodução

Ao contrário do score, a magnitude não é normalizada. Cada expressão de emoção no texto (positiva e negativa) contribui para a magnitude do texto. Por isso, blocos de texto mais longos podem ter magnitudes maiores.

Segundo a documentação da Google Cloud Natural Language API, "um documento com uma pontuação (score) neutra de aproximadamente 0.0 pode indicar baixa emoção, ou indicar emoções mistas, com valores altamente positivos e também negativos que se anulam". Assim, o valor da magnitude ajuda a "eliminar a ambiguidade nesses casos, já que documentos verdadeiramente neutros têm um valor baixo de magnitude, enquanto documentos mistos têm valores de magnitude maiores".

Para acessar e utilizar os recursos disponíveis pela Google Cloud Natural Language API, foi utilizado o pacote "googleLanguageR"[7], publicado por Edmondson (2017). Os histogramas foram criados com o software Tableau.

Resultados e Discussão

A Figura 2 mostra o score dos sentimentos dos tickets bem-sucedidos (SLAMissed = N; as três barras à esquerda da figura) e malsucedidos (SLAMissed = Y; as três barras à direita da figura), obtido com a Google Cloud Natural Language API.

Nos dois casos, observa-se que a quantidade de sentimentos negativos é maior em relação aos sentimentos positivos e neutros e a quantidade de mensagens com sentimentos positivos é a menor entre todas. Para os tickets bem-sucedidos, poderia se esperar uma maior frequência de sentimentos positivos, o que não ocorreu.

Na Figura 2, observa-se ainda que a quantidade de mensagens neutras é maior nos tickets malsucedidos, em relação aos bem-sucedidos. Para verificar se as mensagens classificadas como neutras são realmente neutras, comparou-se o valor do score com o da magnitude, como mostra a Figura 3.

[7] https://cran.r-project.org/package=googleLanguageR

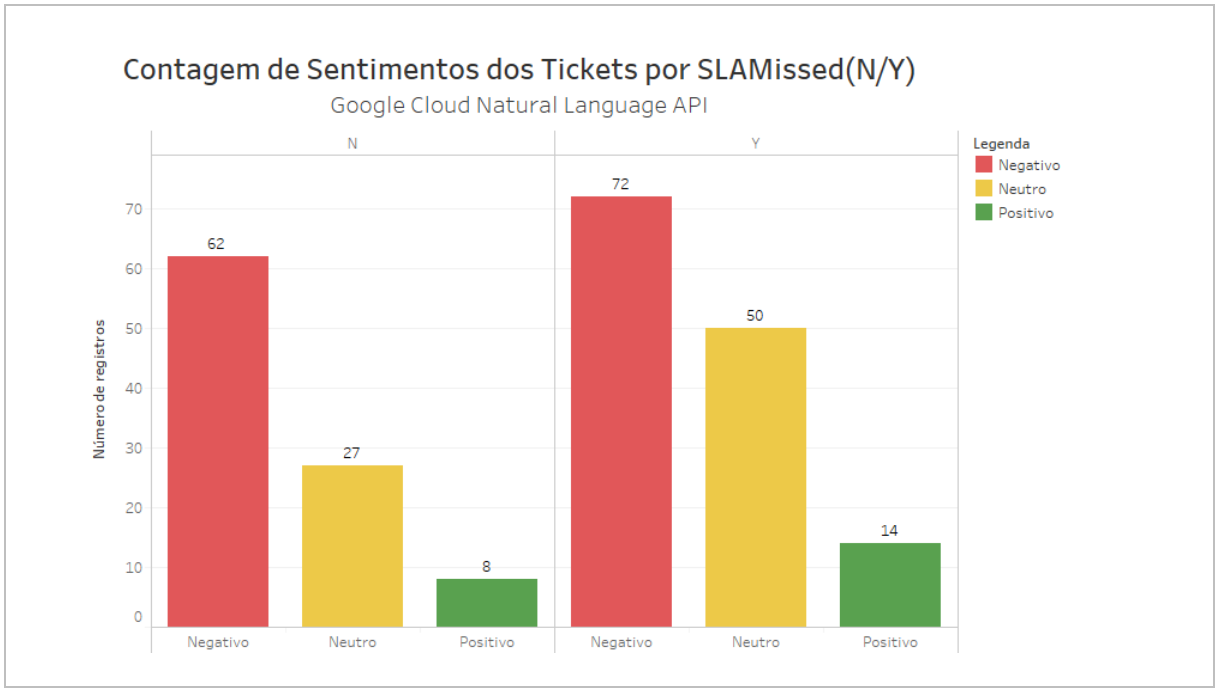

Figura 2 – Contagem de sentimentos dos tickets bem- e malsucedidos
Fonte: Autor

Na Figura 3, o eixo horizontal superior representa os valores da magnitude (que variam entre 0.0 a + infinito), enquanto o eixo horizontal inferior representa os valores de score. Nota-se que a maioria das mensagens neutras, em tickets bem e malsucedidos, se encontram situadas em intervalos de valores baixos de magnitude (0.0 a 0.2), um indicativo de que a maioria das mensagens classificadas como neutras são realmente neutras e não mistas (com valores altamente positivos que anulam negativos). Na Figura 3, observa-se ainda a existência de cinco mensagens neutras com valores altos de magnitude (acima de 0.2) em tickets bem e malsucedidos.

Também foram analisados os sentimentos das mensagens em relação aos tipos de serviços prestados pela empresa, como mostra a Figura 4. Nota-se que tanto nos tickets bem-sucedidos quanto nos malsucedidos, a categoria de tipo de serviço “hardware”, que inclui computador, estabilizador, impressora, internet, modem, monitor, print server, roteador, switch e teclado, é a mais frequente e possui mais mensagens negativas.

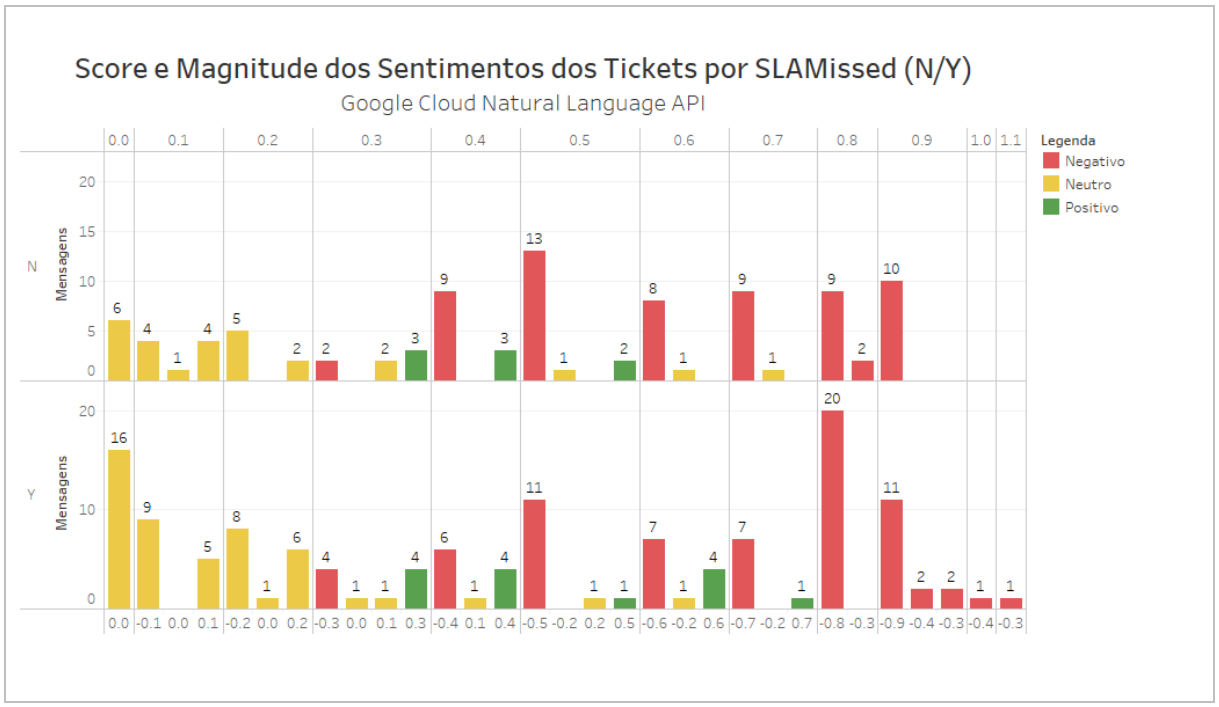

Figura 3 – Score e magnitude dos sentimentos dos tickets bem e malsucedidos
Fonte: Autor

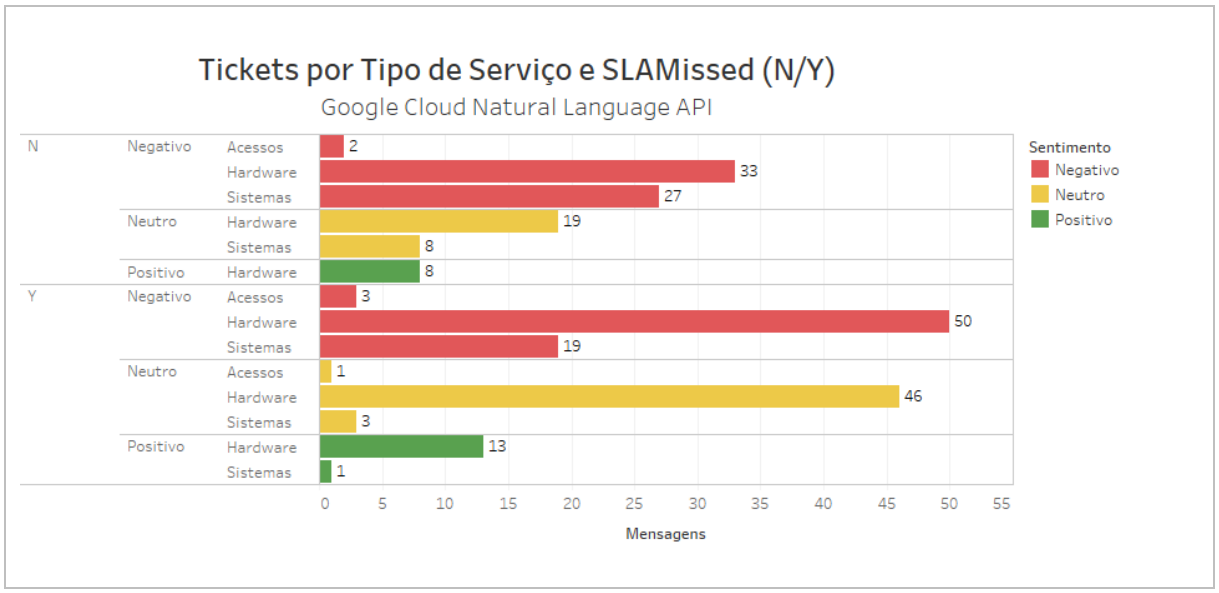

Figura 4 – Sentimentos dos tickets bem e malsucedidos por tipo de serviço
Fonte: Autor

Em relação ao tipo de prioridade dos tickets, verifica-se que a maioria das mensagens possui prioridade alta. Observa-se ainda que, tanto em tickets bem quanto malsucedidos, os tickets com prioridade alta são mais negativos, como mostra a Figura 5.

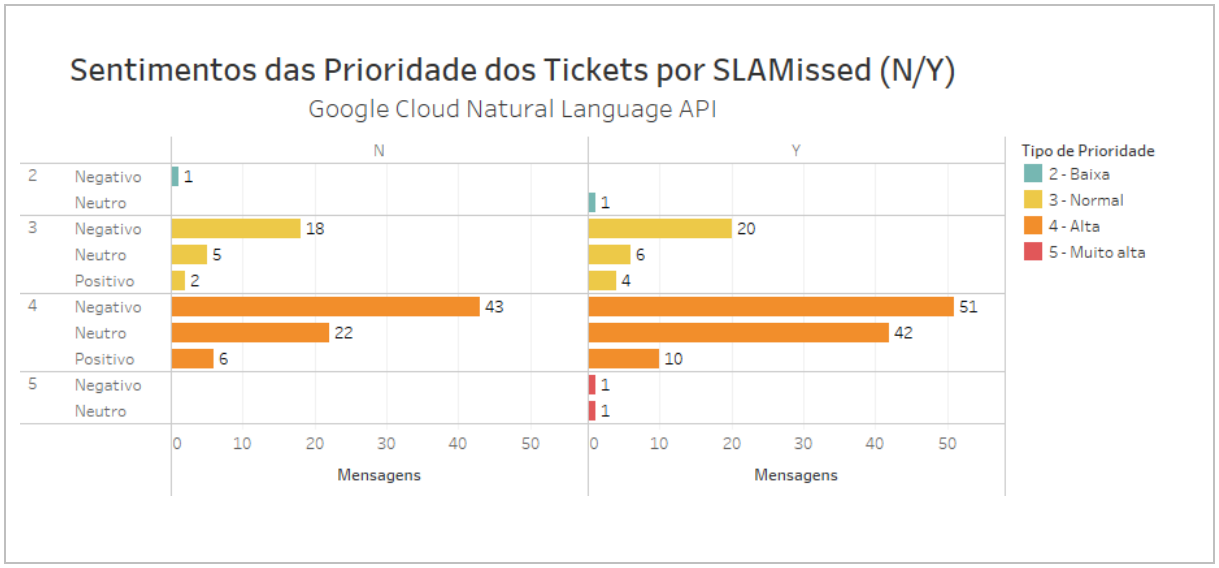

Figura 5 – Sentimentos dos tickets bem e malsucedidos por prioridade do ticket
Fonte: Autor

Alguns exemplos de classificação realizados pela Google Cloud Natural Language API, para tickets bem e malsucedidos, são mostrados nas Figuras 6 e 7, respectivamente. Nota-se que, apesar de considerar o contexto (análise semântica da mensagem), nem sempre as classificações de sentimentos das mensagens são realizadas de forma correta pela API.

A Figura 6 mostra um exemplo de ticket bem-sucedido e classificado como positivo, de forma incorreta, pela API: “Boa noite, srs. venho através desta informa (sic) que a impressora de etiqueta não esta (sic) em funcionamento necessito urgentimente (sic) de um técnico para o reparo ainda hoje”. Pelo contexto da mensagem, verifica-se que há um descontentamento do cliente em relação a um problema, que demanda urgência. Assim, infere-se que o conteúdo emocional predominante dessa mensagem não pode ser positivo.

Na Figura 7, observa-se mais um exemplo de classificação incorreta. A mensagem “a impressora não esta (sic) se comunicando com o pdv” foi classificada como neutra, mas deveria ser negativa, uma vez que pelo contexto da mensagem é possível perceber a existência de uma falha técnica. Um outro exemplo de incorreção é verificado na mensagem “informa que está sem internet na loja, novamente ressalta que está sendo recorrente”, classificada como fortemente positiva, quando na realidade não é. Nesse caso, nota-se a ocorrência de um incidente que já se repetiu outras vezes e, dessa forma, fica implícita na mensagem a insatisfação do cliente em relação ao problema.

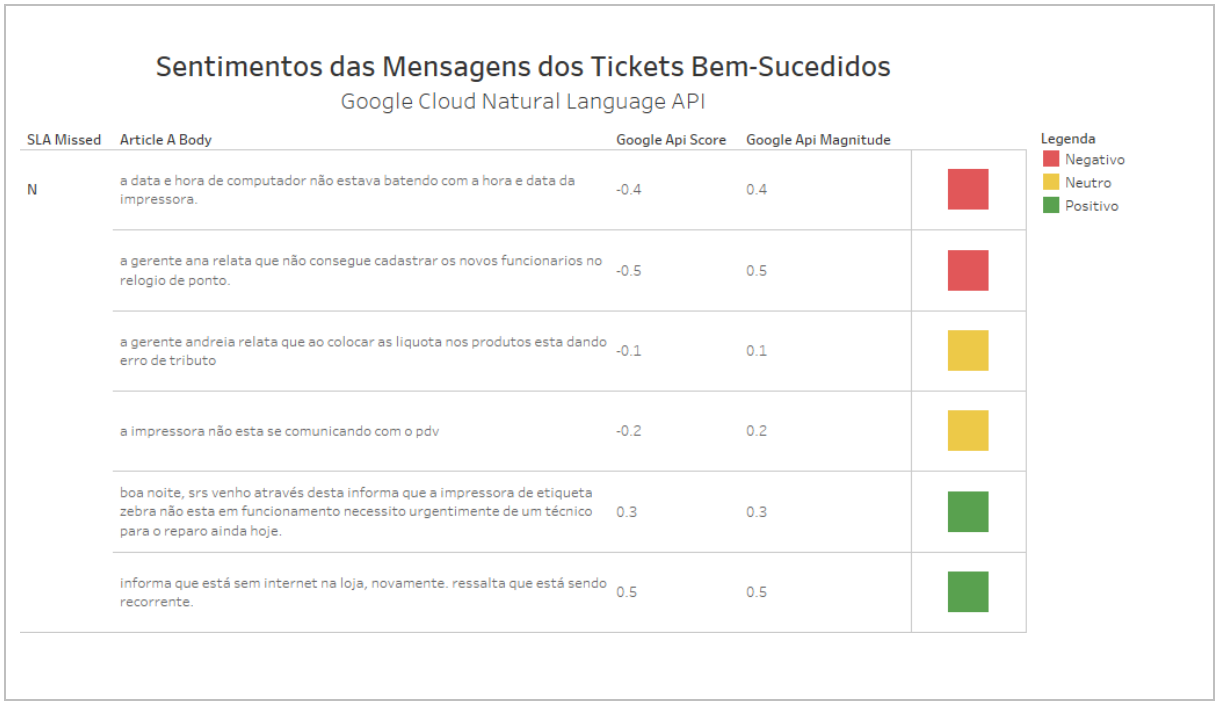

Figura 6 – Sentimentos dos tickets bem-sucedidos classificados pela API do Google
Fonte: Autor

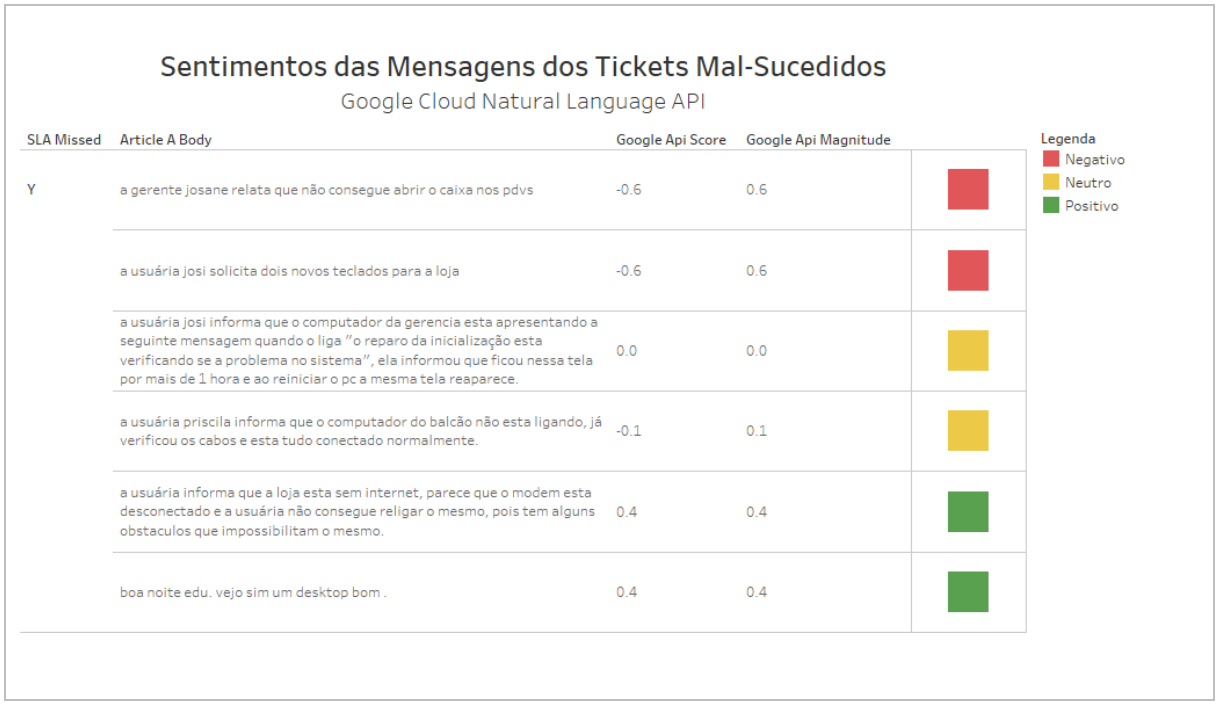

Figura 7 – Sentimentos dos tickets malsucedidos classificados pela API do Google
Fonte: Autor

A fim de verificar o nível de precisão da classificação de sentimentos feita pela Google Cloud Natural Language API, realizou-se uma classificação manual com 21 tickets,

escolhidos de forma aleatória. A classificação manual levou em consideração o contexto da mensagem (análise semântica), pois a API do Google é sensível ao contexto.

Ao final da classificação manual, verificou-se que a Google Cloud Natural Language API acertou corretamente apenas 11 das 21 mensagens, o equivalente a 52,3% do total. Apesar de acertar mais da metade das mensagens escolhidas, o resultado está abaixo do que poderia ser considerado como satisfatório.

Conclusão

De acordo com os resultados do Google Cloud Natural Language API, o sentimento preponderante nas mensagens analisadas é o negativo. Não foram encontradas informações detalhadas na documentação do Google Cloud Natural Language API sobre o funcionamento de seus métodos de classificação de sentimentos a fim de avaliar melhor o resultado obtido. Os resultados obtidos durante a classificação utilizaram a escala de pontuação padrão da API e estão condicionados aos recursos disponíveis na versão utilizada.

Portanto, baseado na análise realizada e assumindo que o nível de percepção dos clientes sobre os serviços pode ser inferido pelos sentimentos expressos nas mensagens, pode-se dizer que, de um modo geral, os clientes demonstram mais sentimentos negativos do que positivos em relação aos serviços prestados, mesmo em situações em que incidentes foram resolvidos dentro de um prazo estabelecido.

Porém, deve-se considerar que a base de dados não possui mensagens sobre o feedback dos serviços prestados após o encerramento dos tickets, o que pode explicar a baixa presença de sentimentos positivos e a alta presença de sentimentos negativos e neutros encontrados pela Google Cloud Natural Language API.

Outro ponto importante a ser considerado é que, nas mensagens analisadas, nem sempre os sentimentos são expressos pelo ponto de vista dos clientes. Existem muitas mensagens nas quais são os funcionários que relatam os problemas dos clientes em relação aos serviços prestados pela empresa, o que pode ter influenciado os resultados obtidos durante a classificação. Como trabalhos futuros, pretende-se realizar uma nova análise na base de dados com a utilização de uma técnica supervisionada para comparação de resultados.

Referências

BENEVENUTO, Fabrício; RIBEIRO, Filipe; ARAÚJO, Matheus. **Métodos para Análise de Sentimentos em Mídias Sociais**. In: WebMedia2015 (minicurso). Disponível em: <http://homepages.dcc.ufmg.br/~fabricio/download/webmedia-short-course.pdf>. Acesso em março de 2018.

DAVE, Kushal; LAWRENCE, Steve; PENNOCK, David M. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: **Proceedings of the 12th international conference on World Wide Web**. ACM, 2003. p. 519-528.

HU, Minqing; LIU, Bing. Mining and summarizing customer reviews. In: **Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining**. ACM, 2004. p. 168-177.

LIU, Bing. Sentiment analysis and opinion mining. **Synthesis lectures on human language technologies**, v. 5, n. 1, p. 1-167, 2012.

EDMONDSON, Mark. **googleLanguageR: Call Google's Natural Language API, Cloud Translation API and Cloud Speech API from R**. R package version 0.1.0. Disponível em: <http://code.markedmondson.me/googleLanguageR/>.

NASUKAWA, Tetsuya; YI, Jeonghee. Sentiment analysis: Capturing favorability using natural language processing. In:**Proceedings of the 2nd international conference on Knowledge capture**. ACM, 2003. p. 70-77.

R CORE TEAM. **R: A language and environment for statistical computing**. R Foundation for Statistical Computing, Vienna, Austria, 2017. Disponível em: <https://www.R-project.org/>.

SILGE, Julia; ROBINSON, David. **Text Mining with R: A tidy approach**. " O'Reilly Media, Inc.", 2017.

WIEBE, Janyce M.; BRUCE, Rebecca F.; O'HARA, Thomas P. Development and use of a gold-standard data set for subjectivity classifications. In: **Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics**. Association for Computational Linguistics, 1999. p. 246-253.

Anexo

```
# Script em R
# Autor: Júlio Campos
# E-mail: julio.campos@uniriotec.br

# ETAPA 1 - Limpeza de texto

# 1 - Instalar bibliotecas

        install.packages("RJSONIO")
        install.packages("stringr")

# 2 - Carregar bibliotecas

        library(RJSONIO)
        library(stringr)

# 3 - Carregar o log de dados em formato JSON

        log <- RJSONIO::fromJSON("original.json", encoding = "UTF-8")
        head(log)

# 4 - Selecionar todas as mensagens de e-mails do log: 'article_a_body' é o décimo primeiro
nome de cada instância do arquivo JSON

        limpeza <- sapply(log, function(x) x[[11]])
        head(limpeza)

# 5 - Arrumar espaços em branco (remover excessos)

        arrumarEspacos <- stringr::str_replace(gsub("\\s+", " ", str_trim(limpeza)), "B",
"b")
        limpeza <- arrumarEspacos

# 6 - Transformar todos os caracteres em minúsculos

        minusculos <- gsub(pattern = '([[:upper:]])', perl = TRUE, replacement = '\\L\\1',
limpeza)
        limpeza <- minusculos

# 7 - Remover assinaturas, textos e imagens de e-mails

        # a) remover todos os textos após e incluindo 'att'

        remover <- sub('att.*', '', limpeza)
        limpeza <- remover

        # b) remover todos os textos após e incluindo 'equipe de operação de serviços'

        remover <- sub('equipe de operação de serviços.*', '', limpeza)
        limpeza <- remover

        # c) remover todo o texto após e incluindo 'abs' (abraços)

        remover <- sub('abs.*', '', limpeza)
        limpeza <- remover

        # d) remover imagens dentro de assinaturas. Exemplo:
[cid:image003.png@01cfd728.d6519c10]

        remover <- sub('\\[cid:image.*', '', limpeza)
```

```
        limpeza <- remover

        # e) remover números de telefones e celulares com padrões de dígitos como xx-xxxx-
xxxx, xx-xxxxxxxxx, xx-xxxxxxxx, xxxx-xxxx, xxxxxxxx

        remover <- gsub('[0-9]{2}-[0-9]{4}-[0-9]{4} | [0-9]{2}-[0-9]{9} | [0-9]{2}-[0-9]{8}
| [0-9]{4}-[0-9]{4} | [0-9]{8}', '', limpeza)
        limpeza <- remover

# 9 - Exibir resultado da limpeza

        limpeza

# 10 - Exportar para arquivo de texto (.txt)

        df <- data.frame(limpeza)
        write.table(df, "textos.txt", quote = FALSE, row.names = FALSE, col.names = FALSE,
fileEncoding = "UTF-8", sep="\t")


# ETAPA 2 - Análise de Sentimentos

# googleLanguageR
# url: https://cran.r-project.org/web/packages/googleLanguageR/index.html

# 11 - Instalar bibliotecas

        install.packages("devtools")
        devtools::install_github("ropensci/googleLanguageR")
        install.packages("xlsx")

# 12 - Carregar bibliotecas e realizar autenticação com a chave json

        library(devtools)
        library(googleLanguageR)
        library(xlsx)

# 13 - Criar um projeto e realizar autenticação com a chave privada da conta de serviço do
Google Cloud Natural Language API

        # obter a chave em:
https://console.cloud.google.com/apis/api/language.googleapis.com/overview

        gl_auth("key.json")

# 14 - Ler o arquivo de texto que contém as mensagens de e-mails

        textos <- readLines("textos.txt", encoding="UTF-8")
        textos

# 15 - Executar função de acesso a Google Cloud Natural Language API para realizar a
análise de sentimentos nos textos

        nlp_result <- gl_nlp(textos, nlp_type = c("analyzeSentiment"), type =
c("PLAIN_TEXT"), language = c("pt"), encodingType = c("UTF8"))
        nlp_result

# 16 - Armazenar os resultados da análise de sentimentos em listas

        texto <- list()
        sentimento <- list()
        score <- list()
        magnitude <- list()
```

```
        length <- length(nlp_result$sentences)

        for(i in 1 : length) {

                texto[[i]] <- textos[i]
                score[[i]] <- nlp_result$documentSentiment$score[i]
                magnitude[[i]] <- nlp_result$documentSentiment$magnitude[i]

                score_value <- nlp_result$documentSentiment$score[i]

                if (score_value < -0.2) {
                        sentimento[[i]] <- "Negativo"
                } else if ((score_value >= -0.2) && (score_value <= 0.2)) {
                        sentimento[[i]] <- "Neutro"
                } else {
                        sentimento[[i]] <- "Positivo"
                }
        }

        head(texto)
        head(sentimento)
        head(score)
        head(magnitude)

# 17 - Converter as listas em vetores

        texto <- unlist(texto, recursive = TRUE, use.names = TRUE)
        sentimento <- unlist(sentimento, recursive = TRUE, use.names = TRUE)
        score <- unlist(score, recursive = TRUE, use.names = TRUE)
        magnitude <- unlist(magnitude, recursive = TRUE, use.names = TRUE)

# 18 - Criar data frame para agrupar todos os vetores em um único objeto

        df <- data.frame(texto, sentimento, score, magnitude)
        head(df)

# 19 - Salvar data frame em uma planilha no Excel

        write.xlsx2(df, file = "analise_de_sentimentos.xlsx", row.names = FALSE,
fileEncoding = "UTF-8")
```