UNIVERSIDADE FEDERAL DO CEARÁ
INSTITUTO UNIVERSIDADE VIRTUAL
CURSO DE SISTEMAS E MÍDIAS DIGITAIS

MÁRIO SILVA RIBEIRO

APLICAÇÃO DE TÉCNICAS DE ANÁLISE DE SENTIMENTO PARA CLASSIFICAÇÃO DE POSTAGENS EM AMBIENTE VIRTUAL DE APRENDIZAGEM

FORTALEZA
2018

MÁRIO SILVA RIBEIRO

APLICAÇÃO DE TÉCNICAS DE ANÁLISE DE SENTIMENTO PARA CLASSIFICAÇÃO DE POSTAGENS EM AMBIENTE VIRTUAL DE APRENDIZAGEM

Trabalho de Conclusão de Curso em formato de monografia a ser apresentado à banca de professores como requisito parcial para a obtenção do grau de Bacharel em Sistemas e Mídias Digitais.

Orientador: Prof. Dr. Emanuel Ferreira Coutinho.

FORTALEZA

2018

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

R37a Ribeiro, Mário Silva.
Aplicação de técnicas de análise de sentimento para classificação de postagens em ambiente virtual de aprendizagem / Mário Silva Ribeiro. – 2018.
64 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Instituto UFC Virtual, Curso de Sistemas e Mídias Digitais, Fortaleza, 2018.
Orientação: Prof. Dr. Emanuel Ferreira Coutinho.

1. Análise de sentimento. 2. Aprendizagem de máquina. 3. Multinomial Naive Bayes. 4. Ambiente Virtual de Aprendizagem. 5. SOLAR. I. Título.

CDD 302.23

MÁRIO SILVA RIBEIRO

APLICAÇÃO DE TÉCNICAS DE ANÁLISE DE SENTIMENTO PARA CLASSIFICAÇÃO DE POSTAGENS EM AMBIENTE VIRTUAL DE APRENDIZAGEM

Trabalho de Conclusão de Curso em formato de monografia a ser apresentado à banca de professores como requisito parcial para a obtenção do grau de Bacharel em Sistemas e Mídias Digitais.

Orientador: Prof. Dr. Emanuel Ferreira Coutinho.

Aprovada em: ___/___/_____.

BANCA EXAMINADORA

_____
Prof. Dr. Emanuel Ferreira Coutinho (Orientador)
Universidade Federal do Ceará (UFC)

_____
Prof. Dr. Ernesto Trajano de Lima Neto
Universidade Federal do Ceará (UFC)

_____
Prof. Dr. Leonardo Oliveira Moreira
Universidade Federal do Ceará (UFC)

À minha família.

AGRADECIMENTOS

Primeiramente acima de tudo e todos, Deus, que me guiou por todos os momentos da vida, até mesmo naqueles em que minha fé foi abalada.

À minha família, que sempre acreditou em mim e em minha capacidade mesmo quando eu repetidas vezes negava ter qualquer coisa do tipo; nunca me deixando desistir quando achei que alguma situação não tinha solução; e também por me suportarem e me alegrarem durante esses 23 complicados anos.

Aos meus amigos que, ainda que não muitos, foram parte crucial nessa jornada. Em especial à André Albuquerque, João Paulo Sabino, Matheus Campelo, Matheus Oliveira e William Victor, que sempre estiveram me apoiando de forma direta ou indireta, seja rindo do meu humor absolutamente ridículo ou me empurrando sempre pra fazer o melhor que eu pudesse nas disciplinas. Também aos mais recentes mas que ainda assim me são tão especiais quanto: Alexandre Machado, Ana Carolina e David Cabral.

Ao meu professor e orientador Emanuel, que além de ser um ótimo professor, me aceitou como orientando quando eu não possuía a mínima ideia de que tema ou até mesmo área ia querer escrever sobre.

Aos professores da banca, Leonardo Moreira e Ernesto Trajano, que, além de serem ótimos professores com quem tive o prazer de fazer não menos que 3 disciplinas cada, disponibilizaram seu tempo para analisar este trabalho e me ajudarem nesse grande passo final.

Um segundo agradecimento ao professor Ernesto, pois sem ele eu provavelmente nunca teria por conta própria arrumado um orientador tão compatível com meu estilo de ser e trabalhar.

À Universidade Federal do Ceará, por me proporcionar uma segunda chance em um curso superior, que dessa vez escolhi de forma correta.

E por fim, a qualquer um que ler esse trabalho, pois foi escrito com a intenção de ajudar quaisquer pessoas que tenham interesse sobre algum desses assunto aqui discutidos, que infelizmente não são alvo de tanta atenção quanto deveriam.

“Se uma estrela tirânica cobrir os céus, eu serei a tempestade da rebelião que cobre o chão”
(FATE/GRAND ORDER, 2018)

RESUMO

Em uma era de tecnologias aplicadas para auxiliar o trabalho humano, é possível relegar a tarefa de descoberta de concordância ou não com um assunto via a análise de sentimentos. Como os professores/tutores em um Ambiente Virtual de Aprendizagem têm que lidar com grandes quantidades de mídia textual, propõe-se a criação de uma ferramenta que possa auxiliá-los nesta tarefa. O trabalho aqui descrito tem por objetivo demonstrar, via implementação de tal ferramenta, como a utilização de técnicas de Análise de Sentimento podem ser úteis em Ambientes Virtuais de Aprendizagem, de forma que um professor e/ou tutor possa descobrir a tendência geral de uma turma em relação ao tema que se tem interesse sem a necessidade de ler todos os textos, que podem ser de grande tamanho e em grande quantidade. A metodologia aplicada envolve quatro passos distintos. Num primeiro momento, é feita a coleta dos dados de *login* de professores ou alunos que possuam contas ativas na plataforma SOLAR. De posse dos dados de acesso ao sistema de ao menos dez estudantes, ou um estudante com pelo menos dez fóruns, ou um professor com disciplinas ativas no semestre atual, os textos de fóruns são colhidos e armazenados em um banco de dados por uma parte da aplicação. Com os textos obtidos, é feito então o uso da segunda parte da aplicação, onde a Análise de Sentimento é executada e classifica os textos em positivo, negativo ou neutro. Então, em posse dos resultados das classificações, uma parcela de amostras de textos são passadas a uma ou mais pessoas que não possuem conhecimento das classificações feitas pelo algoritmo, de forma a se fazer uma referência cruzada e atestar a eficácia do sistema. Os resultados obtidos mostram que o sistema tem a capacidade geral de acertar até aproximadamente 79% das classificações, sendo que algumas chegam até em 93%, o que caracteriza o sistema como sendo confiável o suficiente para exercer o seu papel. Como conclusão, pode-se apontar que a utilização de um método de Análise de Sentimento que é considerado simples pode apresentar resultados extremamente satisfatórios, dependendo somente do contexto que é utilizado; que ainda não existem dicionários em português brasileiro que sejam bons o suficiente para esse tipo de aplicação, mas que mesmo um relativamente simples ainda apresenta resultados satisfatórios; e que a aplicação desenvolvida supre todos os objetivos a que se propôs.

Palavras-chave: Análise de sentimento. Aprendizagem de máquina. Multinomial Naive Bayes. Ambiente Virtual de Aprendizagem. SOLAR.

ABSTRACT

In this age of technologies aimed at helping on human labor, it is possible to relegate the activity of discovering concordancy with a subject via sentiment analysis. As the professors on a Virtual Learning Environment have to work with a lot of text, we propose the conception of a tool which can help them on this matter. This work has the objective of demonstrating, via the implementation of such tool, how the utilization of Sentiment Analysis may be useful in Virtual Learning Environments, as a way for a professor/tutor to discover the general tendency of a group about some subject that they are interested in without the need of reading all the texts – those may be of great length and in great quantity. The methodology applied here has four distinct steps. First, we collect the login data of professors/students that have active accounts on SOLAR. Then, with the data of no less than ten students, or one student with ate least ten forums, or one professor with active course subjects, the forum texts are collected and written to a database by the first module of this application. The second module is then executed, which does the Sentiment Analysis per se and classifies the texts as positive, negative or neutral. Then we get a third person, unrelated to the experiment, and give them a percentage of the classified texts so we can make a cross-reference to test the application accuracy. The results show that the application has the overall capability of getting 79% of the classifications right, and individually some classifications can go as high as 93% - which characterizes the application as being trustworthy enough to do the job it's supposed to. As a conclusion, we can attest that this classification method – which is considered very simple – can show extremely satisfactory results, depending only on the context it is applied to; that, as of the time of writing, there are no brazilian portuguese dictionaries that are good enough for this type of application, but even a relatively simple one can still show satisfactory results; and that the developed application is enough to cover all the objectives it is built to do.

Keywords: Sentiment Analysis. Machine Learning. Multinomial Naive Bayes. Virtual Learning Environment. SOLAR.

LISTA DE ILUSTRAÇÕES

LISTA DE GRÁFICOS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AI	*Artificial Intelligence*
API	*Application Programming Interface*
AS	Análise de Sentimento
AVA	Ambiente Virtual de Aprendizagem
CLI	*Command-Line Interface*
CSV	*Comma-Separated Values*
HTTP	*Hyper-Text Transfer Protocol*
IA	Inteligência Artificial
IUVI	Instituto Universidade Virtual
JISC	*Joint Information Systems Committee*
JSON	*JavaScript Object Notation*
KDD	*Knowledge Discovery in Databases*
MNB	*Multinomial Naive Bayes*
NB	*Naive Bayes*
NLP	*Natural Language Processing*
SA	*Sentiment Analysis*
SVM	*Support Vector Machine*
UFC	Universidade Federal do Ceará
WEKA	*Wakaito Environment for Knowledge Analysis*

SUMÁRIO

1 INTRODUÇÃO

Seguindo a explosão da internet no começo da década de 1990, muitas novas ferramentas e produtos foram desenvolvidos para explorar completamente seus benefícios. Desde o meio da década de 1990 a comunidade de software educacional vem produzindo os chamados Ambientes Virtuais de Aprendizagem (AVA, no plural AVAs), que aparecem com a proposta de suportar as atividades de ensino e aprendizagem através da internet (O'LEARY; RAMSDEN, 2002).

Segundo o *Joint Information Systems Commitee* (JISC), Ambientes Virtuais de Aprendizagem (AVAs) são "os componentes em que os alunos e tutores participam em interações *online* de vários tipos, incluindo o aprendizado *online*" (O'LEARY; RAMSDEN, 2002, p. 2).

As principais funcionalidades disponíveis em um AVA podem ser resumidas nas seguintes: a existência de um canal de comunicação entre estudantes e tutores, entregas de trabalhos compartilhados, estrutura de navegação clara e concisa, e, por fim, alguma espécie de espaço virtual que seja capaz de mimicar as funcionalidades de um espaço físico (O'LEARY; RAMSDEN, 2002).

Mineração de Dados (*Data Mining*) é o processo de descoberta de padrões em conglomerados de dados. É um processo automático ou semiautomático que se executa de forma a obter os anteriormente citados padrões de um volume geralmente bem grande de dados, de forma a se obter alguma vantagem, seja ela qual for (WITTEN et al., 2016).

Mineração de Dados tipicamente lida com dados que já foram coletados para algum objetivo além da análise de dados propriamente dita. Isso significa que os objetivos do exercício da Mineração de Dados não tem relação com a estratégia de coleta de dados. Esta é uma das formas em que a Mineração de Dados difere de boa parte da estatística, em que os dados são na maior parte das vezes coletados usando estratégias eficientes para responder questões específicas. Por esta razão, a Mineração de Dados é geralmente considerada como uma análise de dados secundária (HAND et al., 2001).

Atualmente, são muitas as áreas onde Mineração de Dados é utilizada para resolver, auxiliar ou mitigar seus problemas. Exemplos de tarefas que a Mineração de Dados pode auxiliar as demais áreas são de predição, agrupamento e classificação.

Uma de suas subáreas é a Análise de Sentimento. A Análise de Sentimento, às vezes chamada de *opinion mining*, é um campo de estudo que analisa as opiniões,

sentimentos, valores, apreciação, atitudes e emoções das pessoas, direcionadas às entidades como produtos, serviços, organizações, indivíduos, problemas, eventos, tópicos e os seus atributos, embora estes conceitos não sejam necessariamente equivalentes. O significado de opinião é ainda muito amplo. Análise de Sentimentos, ou *opinion mining,* foca principalmente em opiniões que expressam ou implicam em sentimentos positivos ou negativos (LIU, 2012).

Um bom exemplo da utilização da Análise de Sentimentos pôde ser visto na copa das confederações de 2013, onde foi feita uma grande Análise de Sentimento no Twitter, baseada nos *tweets* que continham como principal assunto os jogos da seleção brasileira e os *tweets* foram marcados como neutros, positivos ou negativos. Os dados resultantes foram então disponibilizados exclusivamente ao treinador da seleção brasileira de futebol através do aplicativo Ei!, mediante *login* e senha (IBM, 2013).

Num AVA, a interação entre aluno e professor/tutor e de aluno com outros alunos é principalmente feita através da escrita de textos, sejam conversas em *chats* virtuais ou fóruns virtuais. Um fórum é um espaço virtual onde criam-se postagens de texto, geralmente de acesso público (ou seja, é possível ser visualizado por qualquer pessoa ou aplicação que possua os direitos), em que qualquer um participante pode criar uma resposta à postagem inicial. Já um *chat* é uma conversa, que pode ou não ficar armazenada, em que somente os participantes da mesma possuem acesso ao conteúdo.

Por limitações do próprio meio de comunicação, no caso texto, é possível que hajam dificuldades na hora de avaliar um *post* por parte do tutor, seja por ele não ter compreendido a mensagem que se desejava ser passada ou simplesmente por não conseguir decidir se um texto, baseado em seu conteúdo, mostra que há uma concordância ou não no ponto de vista apresentado. Também é possível que o tutor apenas deseje dar um *feedback* apenas para alunos que demonstrem a tendência de ir por um certo caminho de pensamento, seja este o desejado ou não. Por fim, é ainda possível que um tutor apenas queira saber qual seria o momento ideal de intervir em uma discussão, de forma a incentivar, instigar ou redirecionar o tema que está sendo discutido pelos alunos. Por possuir essas características, a aplicação da Análise de Sentimento em AVAs se torna atrativa, de forma a se analisar os dados obtidos.

Neste contexto, por exemplo, seria possível aplicar técnicas da Análise de Sentimento de forma a auxiliar o tutor na avaliação da tendência de uma determinada turma em relação a um assunto exposto, baseada somente em se a maior parte dos *posts* no fórum são considerados neutros, positivos ou negativos.

Este trabalho tem como objetivo geral a aplicação de técnica de Análise de Sentimento em textos de fóruns de um AVA de forma a auxiliar um tutor em avaliações, e como objetivos específicos a utilização de Mineração de Dados e Análise de Sentimentos no contexto de problemas acadêmicos; estudo da biblioteca de componentes do AVA SOLAR; a criação de uma ferramenta capaz de fazer uma Análise de Sentimento básica em *posts* de fórum de um AVA; e por fim o estudo da aplicação da mesma no ambiente do AVA SOLAR.

A monografia aqui apresentada é dividida em oito capítulos e alguns deles possuem subcapítulos. Os capítulos são: a **introdução**, onde se expõem as principais razões pelas quais o assunto é interessante de ser estudado; o **referencial teórico**, onde toda a base teórica construída por outros autores é demonstrada, de forma a basear este trabalho em fatos concretos; **o ambiente SOLAR**, onde são demonstradas algumas das características deste AVA e acontece a exposição à API do mesmo; os **trabalhos relacionados**, onde são expostas as semelhanças e diferenças de trabalhos que tocam em assuntos similares ao desta monografia; a **metodologia**, onde se discorre sobre todo o processo científico necessário para a criação da ferramenta e execução da Análise de Sentimentos nos dados obtidos; **projeto e execução**, onde se entra em detalhes mais técnicos sobre a criação da ferramenta; os **resultados**, onde os resultados obtidos na execução da aplicação são demonstrados e discutidos em grandes detalhes; e por fim as **conclusões e considerações finais**, onde as dificuldades envolvidas na criação da aplicação, os resultados (de forma mais subjetiva) e projetos futuros são expostos e discutidos.

2 REFERENCIAL TEÓRICO

Este capítulo se divide em várias seções, de forma a manter o conteúdo organizado e indexável, e segue uma ordem estruturada de forma que se percorra o teoria por trás do tema do trabalho de maneira lógica, indo do assunto mais abstrato ao conteúdo mais concreto.

2.1 Mineração de Dados

Data mining, ou Mineração de Dados, é definido como o processo de descoberta de padrões em dados. O processo precisa ser automático ou, mais comumente, semiautomático. Os padrões descobertos precisam ser significativos o suficiente de forma a

prover alguma vantagem, por exemplo econômica. Os dados sempre estão invariavelmente presentes em quantias substanciais (WITTEN et al., 2016). É um processo de análise de (geralmente grandes) conjuntos de dados observacionais para descobrir relações não imaginadas e para sumarizar os dados em novos modos que são, ao mesmo tempo, compreensíveis e úteis para o dono dos dados (HAND et al., 2001). Se somente conjuntos pequenos de dados fossem envolvidos, meramente discutiríamos análise de dados exploratória clássica como já é praticada por estatísticos. Quando lidamos com grandes grupos de dados, novos problemas aparecem (HAND et al., 2001). As relações e os sumários derivados de um exercício de Mineração de Dados são geralmente chamados de modelos ou de padrões. Exemplos incluem equações lineares, regras, agrupamentos, grafos, estruturas em árvore e padrões recorrentes em um período de tempo (HAND et al., 2001). As definições anteriores se referem somente a dados observacionais, e não a dados experimentais. Mineração de Dados tipicamente lida com dados que já foram coletados para algum objetivo além da análise de dados. Isso significa que os objetivos do exercício de Mineração de Dados não tem relação coma estratégia de coleta de dados. Esta é uma forma em que a Mineração de Dados difere de boa parte da estatística, em que os dados são em boa parte das vezes coletados usando estratégias eficientes para responder questões específicas. Por esta razão, a Mineração de Dados é geralmente considerado como uma análise de dados secundária (HAND et al., 2001).

Mineração de Dados é geralmente inserida no contexto mais amplo de *knowlegde discovery in databases* (KDD). O processo de KDD envolve vários estágios: selecionando os dados-alvo, preprocessando os dados, transformando eles caso necessário, executando a Mineração de Dados para extrair padrões e relações, e, por fim, interpretando as estruturas descobertas (HAND et al., 2001).

O processo de procurar relações dentro de um conjunto de dados – procurar um sumário acurado, conveniente e útil representativo de alguns aspectos dos dados – envolve um número de passos:

a) determinando a natureza e estrutura da representação a ser utilizada;
b) decidindo como quantificar e comparar o quão bem diferentes representações se encaixam nos dados (isto é, escolhendo uma função de "pontuação");
c) escolhendo um processo algorítmico para otimizar a função de pontuação;
d) decidir quais princípios de gerenciamento de dados são requeridos para a implementação dos algoritmos eficientemente (HAND et al., 2001).

Mineração de Dados é um exercício multidisciplinar. Estatística, tecnologia de banco de dados, aprendizado de máquina, reconhecimento de padrões, inteligência artificial e visualização de dados – todas possuem seus papéis. E tão difícil quanto é definir fronteiras claras entre essas disciplinas, também é difícil de definir fronteiras entre cada uma delas em Mineração de Dados. Nestas fronteiras, a estatística, o banco de dados ou aprendizado de máquina de uma pessoa é o problema de Mineração de Dados de outra pessoa (HAND et al., 2001).

2.2 Análise de Sentimento

Análise de Sentimento, também chamada de *opinion mining*, é um campo de estudo que analisa as opiniões, sentimentos, valores, apreciação, atitudes e emoções das pessoas, direcionadas a entidades como produtos, serviços, organizações, indivíduos, problemas, eventos, tópicos e os seus atributos (LIU, 2012). Neste caso específico, opinião é o termo utilizado para denotar a expressão de sentimentos, valores, apreciação, atitudes e emoções, ainda que esses conceitos não sejam equivalentes, já que significado de opinião é ainda muito amplo. Análise de Sentimentos, ou *opinion mining,* foca principalmente em opiniões que expressam ou implicam sentimentos positivos ou negativos (LIU, 2012).

Embora a linguística e o processamento de linguagem natural (NLP – do inglês *Natural Language Processing*) possuem uma longa história, pouca pesquisa havia sido realizada sobre as opiniões e sentimentos das pessoas antes do ano 2000 (LIU, 2012). Desde então, o campo se transformou em uma área bastante ativa para pesquisas.

Existem 3 principais razões para o campo ser atrativo. Primeiro, existem várias aplicações práticas em basicamente todos os domínios, especialmente em aplicações comerciais. Segundo, oferece diversos problemas desafiadores de pesquisa, que nunca tinham sido estudados antes. Por último, pela primeira vez na história humana, possuímos um grande volume de dados contendo opiniões nas mídias sociais da *Web* (LIU, 2012).

Segundo Liu, encontrar, monitorar e destilar informações contidas em opiniões é uma tarefa interessante. Todo *site* tipicamente possui um grande volume de texto opinativo que nem sempre é facilmente decifrado em longos *posts* de fóruns e *blogs*. Uma pessoa mais leiga terá dificuldade em identificar e sumarizar as opiniões nestes *posts*, tornando a análise de sentimento automatizada necessária (LIU, 2012).

Dentro da Análise de Sentimento, é possível identificar várias subtarefas, todas relacionadas à classificação de um determinado documento com uma opinião expressada:

a) determinar a subjetividade de um documento, onde se decide se um texto é de natureza factual (descreve uma situação ou evento, sem expressar uma opinião positiva ou negativa sobre o mesmo) ou expressa uma opinião sobre o assunto à que se refere;
b) determinar a orientação (polaridade) de um documento, onde se decide se o documento que possui a opinião é positivo ou negativo em relação ao assunto discorrido;
c) determinar a força da orientação de um documento, onde se define, por exemplo, se uma opinião positiva expressada pelo texto sobre o seu assunto é fracamente positiva, mediamente positiva ou fortemente positiva (ESULI; SEBASTIANI, 2006).

De forma a auxiliar estas atividades, trabalhos recentes têm como objetivo identificar a orientação de termos subjetivos contidos em texto, como, por exemplo, determinando se um termo que carrega conteúdo opinativo tem uma conotação positiva ou negativa (ESULI; SEBASTIANI, 2006).

2.3 Aprendizagem de Máquina

O aprendizado é um fenômeno multifacetado. Processos de aprendizado incluem a aquisição de conhecimento declarativo novo, o desenvolvimento de habilidades motoras e cognitivas através da instrução ou prática, a organização de novo conhecimento em representações gerais e efetivas, e a descoberta de novos fatos e teorias através de observação e experimentação (MICHALSKI et al., 2013).

Desde o início da era dos computadores, pesquisadores vêm procurando implantar estas capacidades em computadores. Resolver esse problema vem sendo, e continua a ser, um dos maiores desafios de longo prazo da inteligência artificial (IA). O estudo e a modelagem computacional de processos de aprendizagem em suas múltiplas manifestações constituem o tema de estudo do Aprendizagem de Máquina (*machine learning*) (MICHALSKI et al., 2013).

No presente momento, o campo de aprendizado de máquina é organizado em volta de três focos primários de pesquisa:

a) **estudos orientados a tarefas:** o desenvolvimento e a análise de sistemas de aprendizado para melhorar a performance em certos tipos de tarefas (também conhecido como *engineering approach*);
b) **simulação cognitiva:** a investigação e a simulação computadorizada dos processos de aprendizagem humanos;
c) **análise teórica:** a exploração teórica do espaço de possíveis métodos e algoritmos de aprendizado independentes de aplicação (MICHALSKI et al., 2013).

No momento, instruir um computador ou um robô controlado por computador a realizar uma tarefa requer que alguém defina um algoritmo correto e completo para aquela tarefa, e então trabalhosamente implementar o algoritmo em um computador. Estas atividades geralmente envolvem um esforço tedioso e que demanda bastante tempo de pessoal especialmente treinado (MICHALSKI et al., 2013).

Pesquisas de aprendizado de máquina procuram abrir possibilidades de instruir computadores por novos meios, e promete facilitar o trabalho que a programação manual dos computadores futuros. A rápida expansão de aplicações e a disponibilidade de computadores hoje em dia torna essa possibilidade ainda mais atrativa e desejável (MICHALSKI et al., 2013).

Ao tentar uma atividade de aquisição de conhecimento baseada em atividades, deve-se ser ciente de que os sistemas computacionais resultantes têm que interagir com humanos, e por consequência deve ser capaz de imitar habilidades humanas. O argumento tradicional de que uma abordagem da engenharia não precisa refletir a performance humana ou biológica não é realmente aplicável ao aprendizado de máquina. Máquinas que aprendem devem interagir com as pessoas que fazem uso dela, e consequentemente os conceitos e habilidades que eles adquirem devem ser entendíveis por humanos (MICHALSKI et al., 2013).

O método Naive Bayes, algoritmo de aprendizado de máquina supervisionado[1], é comumente utilizado em situações deste tipo (YU; HATZIVASSILOGLOU, 2003) por ser bem estudado, rápido, de fácil implementação e eficácia relativa (RENNIE et al., 2003).

Quatro diferentes tipos básicos de aprendizagem aparecem comumente em

---

[1] aprendizagem supervisionada é aquela em que a saída do processamento deve ter relação e semelhança com o tipo de dado inserido previamente.

aplicações de Mineração de Dados. No **aprendizado por classificação**, o esquema de aprendizagem é apresentado como um conjunto de exemplos classificados, de onde se é esperado que ele (o esquema de aprendizagem) aprenda uma forma de classificar exemplos não vistos. No **aprendizado por associação**, procura-se qualquer associação, não somente as que predizem um valor particular para uma classe. No **agrupamento**, grupos de exemplos que devem ficar juntos são procurados. Na **predição numérica**, a resultado predito não é uma classe discreta, mas sim uma quantidade numérica. Independente do tipo de aprendizagem envolvida, chamamos a coisa a ser aprendida de *conceito* e a saída produzida por um esquema de aprendizado de *descrição do conceito* (WITTEN et al., 2016).

O aprendizado por classificação é às vezes chamado de *supervisionado* porque, de certa forma, o esquema opera sob supervisão, sendo provido com as respostas de cada um dos exemplos de treinamento. Esta saída é chamada de classe do exemplo. O sucesso do aprendizado por classificação pode ser julgado ao se testar a descrição do conceito que foi aprendida em um conjunto independente de dados de teste, para os quais a classificação verdadeira é conhecida mas não disponibilizada para a máquina (WITTEN et al., 2016).

Regras para o aprendizado por associação diferem das regras de classificação de duas maneiras: elas podem "predizer" qualquer atributo, não somente a classe, e elas podem predizer mais de um valor de atributo por vez. Por causa disso, existem muito mais regras de associação que de classificação (WITTEN et al., 2016).

Quando não há uma classe específica, o agrupamento é utilizado para agrupar itens que estão na mesma classe, obedecendo certos critérios ou atributos previamente definidos. O sucesso do agrupamento é geralmente medido subjetivamente em termos de quão útil os resultados parecem para um usuário humano. Pode até mesmo ser seguida de um aprendizado por classificação em que as regras são aprendidas se forma a resultar em uma descrição inteligível de como novas instâncias devem ser colocadas nos agrupamentos (WITTEN et al., 2016).

A predição numérica é uma variação do aprendizado por classificação, em que o resultado é um valor numérico em vez de uma categoria. Com a problemas de predição numérica, assim como qualquer outra situação de aprendizado de máquina, o valor predito para novas instâncias é geralmente menos interessante que a estrutura da descrição que é aprendida, expressada em termos de quais são os atributos importantes e como eles se relatam com a saída numérica (WITTEN et al., 2016).

2.4 Multinomial Naive Bayes

Multinomial Naive Bayes, ou *Multinomial NB* é um método de aprendizagem probabilística. A probabilidade de um documento *d* ser de uma classe *c* é computado como

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

onde $P(t_k|c)$ é a probabilidade condicional de de um termo t_k ocorrer em um documento de classe *c*. A medida $P(t_k|c)$ pode ser interpretada como a quantidade de evidência de que t_k contribui para que *c* seja a classe correta. *P(c)* é a probabilidade anterior de que um documento ocorra na classe *c*. Os termos são então classificados em suas devidas classes baseados em sua probabilidade. $\langle t_1, t_2, \ldots, t_{n_d} \rangle$ são os símbolos em *d* que são parte do vocabulário que é usado para a classificação e n_d é o número desses símbolos em *d*. Por exemplo, $\langle t_1, t_2, \ldots, t_{n_d} \rangle$ para um documento de uma única frase *Beijing and Taipei join the WTO* podem ser $\langle \text{Beijing}, \text{Taipei}, \text{join}, \text{WTO} \rangle$, com $n_d = 4$, se tratarmos os termos *and* e *the* como as *stopwords*[2] (MANNING et al., 2009, p. 258).

Em classificação de texto, o objetivo é sempre encontrar a melhor classe para o documento. A melhor classe em uma classificação *Naive Bayes* é a mais provável ou *maximum a posteriori* (*MAP*) classe c_{map} (MANNING et al., 2009, p. 258):

$$c_{map} = \arg\max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg\max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c).$$

A complexidade de computar os parâmetros no NB é $\Theta(|\mathbb{C}||V|)$ pois o *conjunto* de parâmetros consistem de $|\mathbb{C}||V|$ condições probabilísticas e $|\mathbb{C}|$ anteriores. O pré-processamento necessário para computar os parâmetros (extração de vocabulário, contagem de termos, etc.) pode ser feita em um *pass* nos dados de treinamento. A complexidade desse componente é, então, $\Theta(|\mathbb{D}|L_{ave})$, onde $|\mathbb{D}|$ é o número de documentos e L_{ave} é o tamanho médio de um documento. Em geral, temos $|\mathbb{C}||V| < |\mathbb{D}|L_{ave}$, então tanto a complexidade de treinar quanto a de testar são lineares no tempo que levam para analisar os dados. Por ter que

---

[2] *stopwords* são palavras que, mesmo que removidas, não alteram o sentido de uma frase.

observar os dados ao menos uma vez, *Naive Bayes* pode ser considerado como tendo uma relação de tempo e complexidade ideal. Essa eficiência é uma das razões que tornam o *Naive Bayes* um método de classificação de texto popular (MANNING et al., 2009, p. 261-262).

Classificadores *Naive Bayes* não fazem estimativas de classe muito boas, mas quando classificam, geralmente o fazem muito bem. Ainda que não seja o método com maior grau de acerto para textos, o *Naive Bayes* possui muitas virtudes que o fazem um competidor bastante forte para a classificação de textos. Ele é excelente quando existem muitas características igualmente importantes que contribuem em conjunto para a decisão da classificação. É também bastante robusto em relação a características de ruídos e desvio de conceito (desvio de conceito é definido como a mudança gradual ao longo do tempo dos conceitos que permeiam a classe. Como exemplo: "o presidente é ruim" pode, com a eleição de um novo presidente, afetar bastante a classificação da opinião pública em métodos mais avançados) (MANNING et al., 2009, p. 269).

A maior força do Naive Bayes é sua eficiência: treino e classificação podem ser concluídos em apenas uma passagem em cima dos dados fornecidos. Porque combina eficiência com boa acurácia, é geralmente utilizado como uma linha-base em pesquisas de classificação de textos (MANNING et al., 2009, p. 270).

É geralmente o método escolhido se:

a) extrair o máximo de confiabilidade de acerto da classificação de texto não vale o esforço envolvido com relação à outros métodos;
b) uma base de dados grande para o treinamento é previamente existente e se tem mais a ganhar em treinar com muitos dados do que se utilizar de um método classificador melhor em uma base de dados de treino menor;
c) sua robustez contra desvio de conceito pode ser explorada (MANNING et al., 2009, p. 270).

2.5 Ambiente Virtual de Aprendizagem

Seguindo a explosão da internet no começo da década de 1990, muitas novas ferramentas e produtos foram desenvolvidos para explorar completamente seus benefícios. Desde o meio da década de 1990 a comunidade de software educacional vem produzindo os chamados ambientes virtuais de aprendizado (AVA, no plural AVAs), que aparecem com a proposta de suportar as atividades de ensino e aprendizado através da internet. O J*oint*

Information Systems Commitee (JISC) diz que AVAs são “os componentes em que os alunos e tutores participam em interações *online* de vários tipos, incluindo o aprendizado *online*” (O’LEARY; RAMSDEN, 2002, p. 2).

AVAs permitem aos professores a criação de recursos de forma rápida e sem a necessidade de desenvolvimento de conhecimentos técnicos (O’LEARY; RAMSDEN, 2002). Tipicamente *web-based*, AVAs provêm uma série de ferramentas integradas na internet, permitindo fácil upload de materiais e oferecem um design consistente que pode ser customizado pelo usuário (O’LEARY; RAMSDEN, 2002).

As ferramentas que compõem um AVA geralmente incluem:

a) comunicação entre tutores e estudantes;
b) autoavaliação e avaliação somativa;
c) entrega de recursos e materiais para o aprendizado;
d) áreas de trabalho compartilhado;
e) suporte para os estudantes;
f) gerenciamento e acompanhamento de estudantes;
g) ferramentas para o estudante;
h) aparência consistente e customizável;
i) estrutura de navegação (O’LEARY; RAMSDEN, 2002).

Um AVA pode ser identificado por possuir as seguintes características:

a) um lugar com espaços designados para informação;
b) um espaço social: interações educacionais ocorrem no ambiente, tornando espaços em lugares;
c) o espaço virtual é representado de forma explícita;
d) estudantes não são somente ativos mas também devem co-construir o espaço virtual;
e) AVAs não são destinados somente à educação à distância (EAD);
f) AVAs integram tecnologias heterogêneas e múltiplas formas de pedagogia;
g) a maioria dos AVAs tem relação com espaços físicos (DILLENBOURG et al., 2002).

Os AVAs são desenvolvidos para dar suporte a processos de aprendizagem, tanto nas modalidades de ensino presencial como a distância. Um AVA agrupa em um espaço virtual, recursos definidos em meios eletrônicos, como: fórum, wiki, bate-papo, conferências, envios de mensagens, material de leitura, banco de questões, e outras tecnologias que

colaboram para o processo de ensino e aprendizagem. Este ambiente é configurado com a meta de alcançar determinados objetivos definidos pelo tutor do ambiente (ROSEMANN et al., 2014).

O desenvolvimento de um AVA exige a criação de uma plataforma na qual o aluno interaja, e que a partir desta interação o ambiente reaja. Esta reatividade pode ser realizada por um professor/tutor ou por ferramentas inteligentes. Já existem vários AVAs consolidados no mercado, entre eles é possível identificar alguns com maior aplicabilidade perante outras soluções. Um AVA que se destaca devido a sua particularidade de ser de código aberto e possuir bibliotecas adicionais para adaptação é o Moodle (ROSEMANN et al., 2014).

O SOLAR foi desenvolvido pelo Instituto Universidade Virtual, da Universidade Federal do Ceará. É baseado no modelo de três camadas, cujo modelo de participação é orientado ao professor e ao aluno (COUTINHO et al., 2013a). Quanto ao processamento do sistema, caracteriza-se por ser um sistema distribuído. Além de possibilitar a publicação de cursos e interação com professores e alunos, o SOLAR foi desenvolvido potencializando o aprendizado a partir da relação com a própria interface gráfica do ambiente, sendo desenvolvido para que o usuário tenha rapidez no acesso às páginas e ao conteúdo, fácil navegabilidade e compatibilidade com navegadores. Nele, o interagente se sente seguro para explorar os espaços disponibilizados. O ambiente é apoiado numa filosofia de interação e não de controle (COUTINHO et al., 2013b).

3 O AMBIENTE SOLAR

Neste capítulo são descritas algumas informações gerais da plataforma SOLAR e também a sua API (*Application Programming Interface*).

3.1 Interface web

O SOLAR é um ambiente virtual de aprendizagem desenvolvido pelo Instituto Universidade Virtual, pertencente à Universidade Federal do Ceará. Foi projetado de forma a permitir a criação de dversos espaços virtuais de forma a se atender tanto cursos presenciais quanto semipresenciais. Encontra-se disponível na internet e seu acesso é feito via apresentação de usuário e senha, como é possível ver na Figura 1.

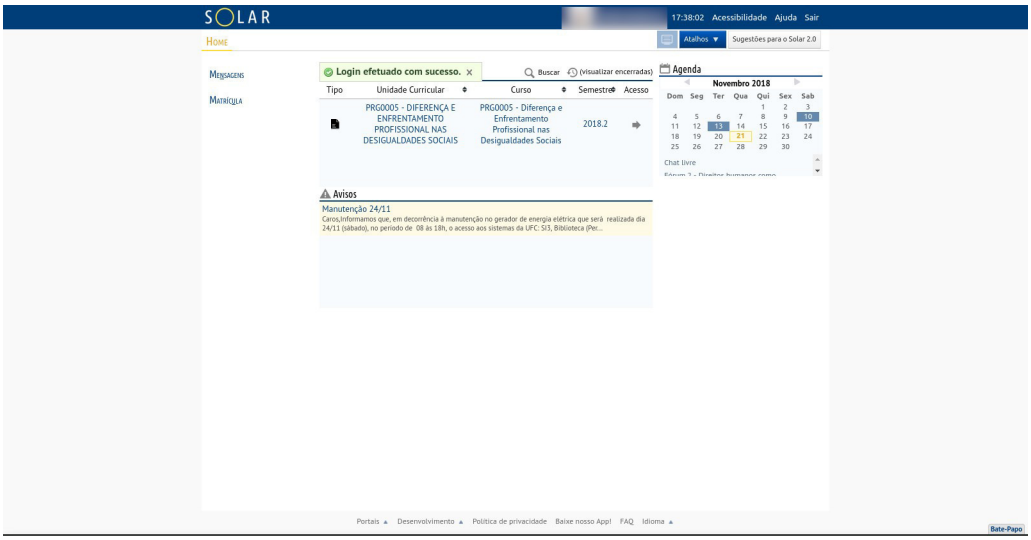
Figura 1 — Tela de login do SOLAR

Fonte: autor.

O usuário aqui demonstrado nas imagens é um aluno matriculado em uma disciplina, então a interface apresentada a ele possui diferenças em relação à dos professores/tutores.

Após o login, o usuário é então apresentado à tela inicial do sistema (Figura 2), que conta com 3 divisões: uma barra lateral esquerda, onde se encontram as mensagens de chat enviadas ou recebidas pelo usuário e a opção de gerenciar as disciplinas que o usuário fez ou irá fazer; uma barra lateral direita, onde o usuário encontra um calendário/agenda onde o usuário pode, além de ver as datas pertinentes do mês, quais os dias que ocorrem atividades relacionadas às disciplinas que está matriculado; por fim, uma coluna central, onde o usuário pode visualizar cada uma de suas disciplinas ativas no momento e, caso clique no botão correspondente, as disciplinas passadas.

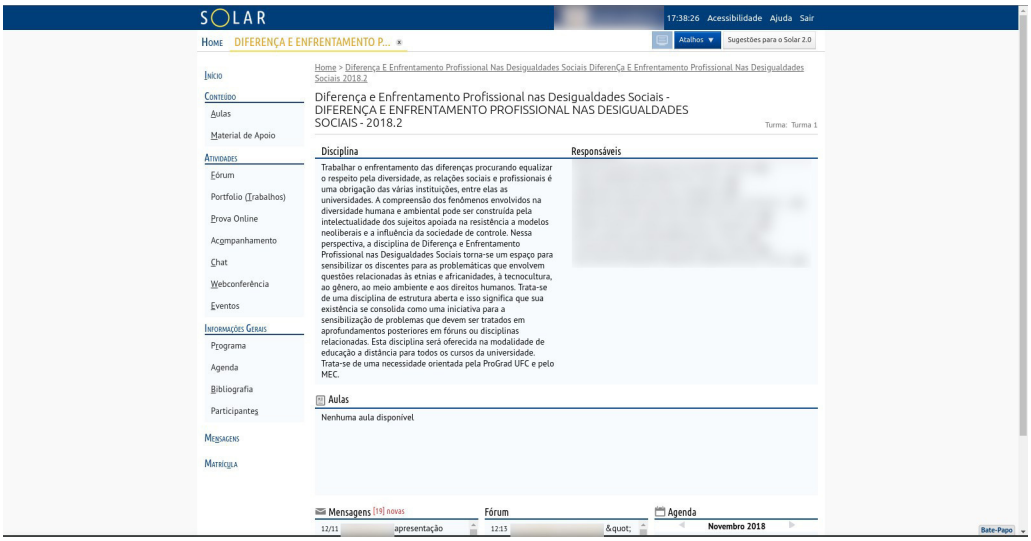
Figura 2 — Tela inicial do SOLAR



Fonte: autor.

Ao clicar em uma disciplina, o usuário é então direcionado à página (Figura 3) que concentra todas as informações disponíveis sobre a mesma. A página é composta de duas colunas: a coluna da direita, que possui todos os atalhos para as principais formas de interação da disciplina – como por exemplo conteúdo, atividades e informações gerais; a coluna do meio por si possui várias subdivisões, sendo elas o título da disciplina, a descrição da disciplina, os professores/tutores responsáveis pela mesma, as aulas planejadas, as mensagens direcionadas à toda a turma, os fóruns da disciplina e a mesma agenda presente na terceira coluna da tela inicial.

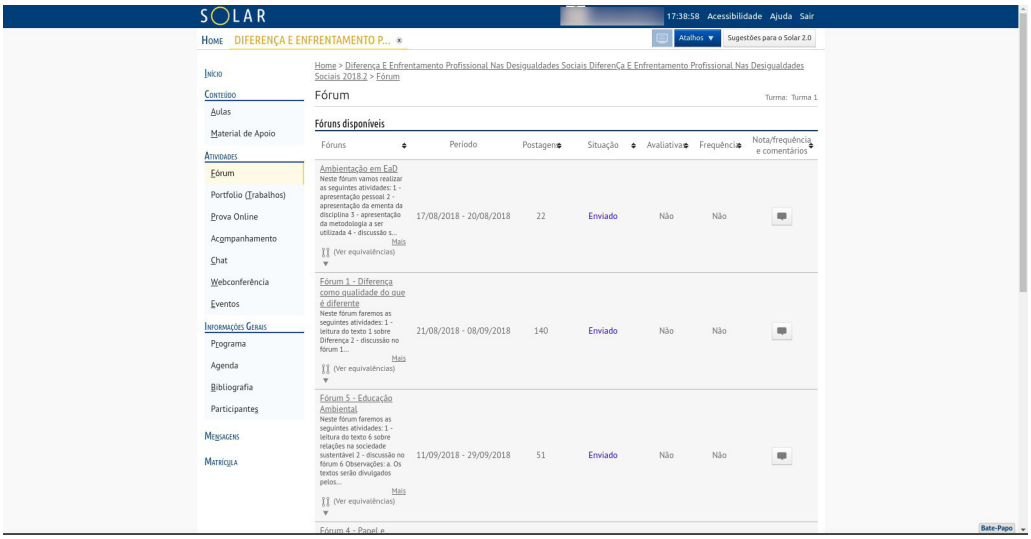
Figura 3 — Tela de disciplina do SOLAR



Fonte: autor.

Ao clicar em fóruns, o usuário é direcionado para a tela que lista todos os fóruns para a dada disciplina (Figura 4). Nela, o usuário pode escolher com qual dos fóruns disponíveis deseja interagir (contanto que o fórum ainda tenha tempo disponível) ou apenas ler (disponível para qualquer fórum).

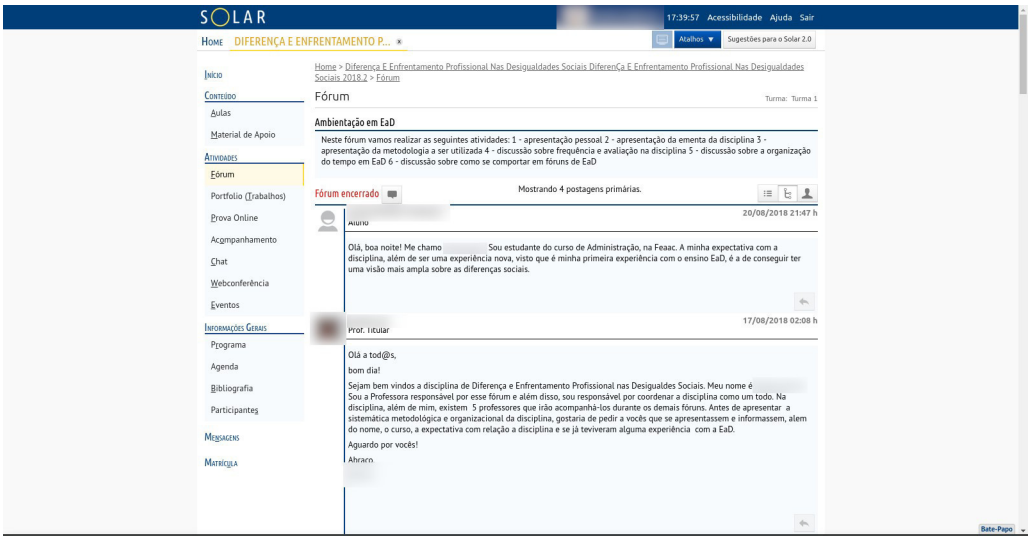
Figura 4 — Tela de fóruns da disciplina do SOLAR



Fonte: autor.

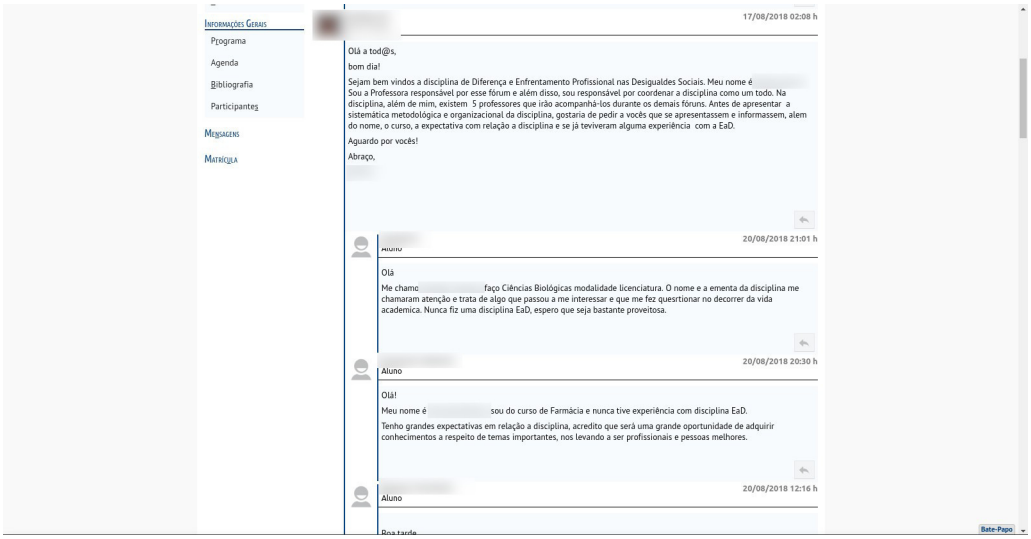
Por fim, ao clicar num desejado fórum o usuário é redirecionado para uma página (Figura 5) que contém todo o conteúdo (seja este texto, imagens ou vídeos) do fórum. O fórum é organizado por padrão de forma hierárquica (Figura 6), de forma a se garantir uma melhor visualização do conteúdo, mas o usuário tem a opção de decidir se prefere ver todas as informações num mesmo nível hierárquico.

Figura 5 — Textos do fórum de disciplina



Fonte: autor.

Figura 6 — Hierarquia dos textos de fórum de disciplina



Fonte: autor.

3.2 API do SOLAR

O SOLAR é um projeto de código aberto e livre, sob a licença GPL 3.0. Possui uma API (*Application Programming Interface*) que permite que aplicações de entidades independentes possam interagir com o seu banco de dados, de forma a se utilizar dos mesmos como desejar.

Essa API se baseia em tecnologias HTTP e sempre retorna as informações no formato JSON. A API no momento é utilizada oficialmente na implementação do aplicativo Mobilis, que permite que os usuários da plataforma acessem o sistema via celular.

A API é disponibilizada de forma aberta para o público no serviço de hospedagem do código, mas até o momento é de forma incompleta. Para a execução deste trabalho, foi utilizada uma documentação interna (ANEXO A) que a equipe do SOLAR pretende disponibilizar na íntegra em um futuro próximo.

Todos os *requests* (requisição) feitos ao servidor do SOLAR são via método GET, exceto o primeiro, que é via POST. Um *request* HTTP é um pacote de dados binários de informação que um computador, geralmente um cliente, envia para outro, geralmente um servidor, para comunicar alguma coisa.

Um primeiro *request*, feito para se obter o *token* do usuário, é executado via POST para se obter o *token* de identificação do usuário. O *token* de identificação é uma sequência de caracteres alfanuméricos gerado pelo servidor do SOLAR que serve como identificação única para um único usuário, e que possui uma validade de tempo definida, ou seja, depois de algum tempo ele não será mais válido. Na documentação, é referenciado como “autenticação de usuário”. Não é possível a execução de tarefa qualquer via a API do SOLAR sem o devido *token* válido.

O próximo *request* é feito de forma a se obter todas as disciplinas que o usuário tem acesso. Na documentação, é referenciado como “lista de disciplinas de um usuário com as turmas ativas”. Um fato a se observar é que, ainda que a API diga de forma clara que esse método captura os dados de todos os semestres, na realidade ele apenas captura as disciplinas ativas, o que, por consequência, significa somente as do semestre atual. Existe um método anterior a este, descrito na documentação, que diz que consegue capturar todas as disciplinas de todos os semestres, ativas ou não, mas esta não funciona no momento da escrita do trabalho – apenas retorna as mesmas informações que a utilizada aqui.

Em seguida, um *request* é feito de forma a se obter todas as turmas ativas que o usuário possui em determinada disciplina. Na documentação é chamada de “lista de turmas de um usuário”. Esta informação é necessária pois só com ela é possível o acesso aos fóruns.

A seguir, outro *request* é feito, de forma a se obter todos os fóruns das turmas conseguidas anteriormente. Na documentação, é chamada de “lista de fóruns de uma turma”.

Por fim, o último método utilizado é chamado de “lista de posts para um fórum (nível 1)”, onde todos os textos de um fórum são obtidos.

4 TRABALHOS RELACIONADOS

Existem diversos trabalhos sobre Análise de Sentimento na língua inglesa, mas há uma falta sobre os mesmos na língua portuguesa, ainda mais especificamente na versão brasileira (KANSAON et al., 2018). Este capítulo é então destinado a demonstrar e reconhecer alguns outros trabalhos sobre o tema da Análise de Sentimentos, além de comparar os métodos utilizados por outros autores com os aqui aplicados. Os trabalhos aqui citados foram recolhidos em diversos anais de eventos, e foram selecionados pelo critério de semelhança de tema (Análise de Sentimentos em português brasileiro) ou de semelhança de objeto de estudo (Análise de Sentimentos em *tweets*).

Como citado, Kansaon et al. afirmam que existem diversos trabalhos sobre a análise de sentimentos para a língua inglesa, mas no caso do português brasileiro a quantidade de trabalhos é menor por não existirem muitas bases de dados disponíveis e métodos para realizar a análise. Realizaram então uma série de testes com a ferramenta WEKA (*Waikato Environment for Knowledge Analysis*) em cima de milhares de *tweets* em português brasileiro e como resultado chegou-se à conclusão de que na ferramenta existem algoritmos que conseguem atingir até 85% de acerto na análise, mas ao mesmo tempo também foi observado que a relação de acertos entre sentimentos próximos possuem taxa de acerto menor que 70% (KANSAON et al., 2018).

Christie et al. constataram que, da grande quantidade de dados compartilhados na *web*, boa parte desses possuem posicionamentos expressos, sejam eles direta ou indiretamente, e que a detecção deste posicionamento pode identificar qual a polaridade de uma desta em relação a uma ideia-alvo. A partir dessa ideia criaram um experimento baseado em uma grande amostra de *tweets* em português brasileiro sobre a corrida presidencial brasileira de 2018, onde se utiliza desses dados para a detecção automática de posicionamento através de uma abordagem semi-supervisionada. Como conclusão foi possível apontarem a existência de *bots* no *twitter*, e também que com o uso de três classificadores foi possível a

obtenção de um *F-Measure* (*F-Measure* pode ser definido como a média harmônica entre a *precision* e o *recall*. Ambos são formas de se aferir a acuidade estatística de uma predição) acima de 94% (CHRISTIE et. al., 2017).

Dosciatti et al. apontam que a identificação automática de emoções em textos tem se provado como uma boa forma de obtenção de resultados em diversas aplicações, e então proporam uma abordagem utilizando máquinas de vetores de suporte para a identificação de emoções em textos escritos em português brasileiro. Os dados utilizados para a extração de emoções foram compostos de notícias extraídas de um jornal *online*. Como resultado, obtiveram que textos previamente rotulados e submetidos a um classificador SVM (*Support Vector Machine*) em configuração multiclasse possui uma taxa de acerto de 61% (DOSCIATTI et al., 2018).

Já Caetano et al. definem que a homofilia é a tendência de um indivíduo possuir características semelhantes aos de seus pares. Com base nisto, foi proposta uma análise da homofilia política entre os usuários do *twitter* durante as eleições americanas de 2016. A base de dados utilizada constitui-se de *tweets* coletados num período de 122 dias e foram analisadas a homofilia das mensagens em relação à Trump e Hillary. Os resultados mostraram que existe maior homofilia entre usuários que compartilham sentimentos negativos em relação aos dois candidatos, e que existe maior heterofilia entre usuários que não manifestam sentimento em relação a nenhum candidato (CAETANO et al., 2017).

A Tabela 1 a seguir demonstra os autores, técnicas utilizadas, se houve experimento ou não, as ferramentas utilizadas e a área de atuação de cada uma das pesquisas supracitadas.

Tabela 1 — Características dos trabalhos relacionados

Autores	**Técnica**	**Exp.**	**Algoritmos**	**Ferramentas**	**Área de Atuação**
Kansaon et al.	Análise de sentimento	Sim	Naive Bayes	WEKA	Análise de sentimentos em redes sociais
Christie et	Análise de	Sim	Naive Bayes;	LIWC;	Análise de

al.	sentimento		SVM; Random Forest	WEKA	sentimentos e política em redes sociais
Dosciatti et al.	SVM	Sim	Naive Bayes; SVM; KNN	LibSVM; WEKA	Análise de sentimentos
Caetano et al.	Lexemas de dicionários	Sim	Não citado	SentiStrenght; Stanford Parser	Análise de sentimentos e política em redes sociais

Fonte: autor.

Em relação aos trabalhos apresentados acima, é possível observar que o trabalho aqui apresentado apresenta semelhança ao fazer uso de *Multinomial Naive Bayes* para a classificação de frases. Outra semelhança é que todos os trabalhos apresentados fazem uso de experimento.

Por fim, como maior diferença pode-se apontar que todos os outros trabalhos se utilizam de aplicações pré-existentes para a análise de dados, a maioria das vezes tendo sido o WEKA. O trabalho aqui apresentado faz uso da distribuição Anaconda, que junta diversas ferramentas em Python. Fazendo o uso do Anaconda, desenvolveu-se uma aplicação totalmente customizada, que atende exclusivamente ao problema proposto.

5 METODOLOGIA

A natureza de uma pesquisa pode ser de dois tipos: básica, ou pura, onde se busca entender e desvelar fenômenos, assim gerando conhecimento básico; já a pesquisa aplicada se destina a aplicar leis, teorias e modelos na solução de problemas que exigem a ação e/ou diagnóstico de uma realidade (POLAK et. al., 2011).

Na abordagem do problema, existem dois grupos: a pesquisa quantitativa, que trabalha com dados mensuráveis, auxiliada geralmente pelo uso de estatística; a pesquisa

qualitativa, por outro lado, busca compreender fenômenos onde é possível uma interpretação subjetiva e é utilizado geralmente para descobrir e refinar as questões da pesquisa. Por fim, há também o subtipo quali-quanti, que agrupa qualidades tanto qualitativas quanto quantitativas (POLAK et. al., 2011).

Pelos objetivos, é possível a classificação em três tipos: a pesquisa descritiva visa dar uma explicação sistemática de um ou mais fenômenos ou aprofundar um tema e busca especificar propriedades e características importantes de qualquer fenômeno que se analise; a pesquisa exploratória é aplicada de forma a se estudar problemas novos ou pouco conhecidos, sempre respondendo questões do tipo “o quê?, como? e por que?”; já os explicativos vão além da descrição de conceitos ou fenômenos ou do estabelecimento de relações entre conceitos, estabelecendo as causas dos acontecimentos, fatos, fenômenos físicos ou sociais estudados (POLAK et. al., 2011).

Por fim, existem vários tipos de pesquisas segundo os procedimentos técnicos, que podem variar bastante de autor para autor. De forma a manter uma definição mais sucinta, neste caso será demonstrado apenas o tipo que este trabalho se encaixa: na pesquisa experimental o pesquisador intervém na realidade e há manipulações de uma variável enquanto as demais são controladas, de modo que qualquer variação no comportamento do fenômeno estudado será associada ao elemento manipulado, de forma a verificar hipóteses de pesquisa e procurar generalizações empíricas (POLAK et. al., 2011).

Seguindo os objetivos específicos apresentados na introdução e com as informações apresentadas nos parágrafos anteriores, é possível então a classificação deste trabalho, em formato de monografia, nas seguintes categorias:

a) natureza: pesquisa aplicada, já que se procura aplicar algoritmos de Aprendizagem de Máquina e Análise de Sentimento nas amostras, definidas mais à frente;
b) abordagem: quali-quanti. Quantitativa já que métodos estatísticos serão aplicados nos resultados obtidos, de forma a se medir objetivamente a quantidade de “erros” e “acertos” do sistema a ser desenvolvido e analisado; qualitativa pois suposições de como os tutores/professores podem utilizar essas informações serão sugeridas;
c) objetivos: descritiva, já que se planeja um aprofundamento na temática de Análise de Sentimento;

d) procedimentos: pesquisa experimental. O estudo aqui feito não pode ser generalizado para outros sistemas, mas pode ser generalizado para qualquer usuário que pertença ao AVA SOLAR.

A metodologia deste trabalho começa a ser discutida e demonstrada a partir deste ponto, e foi dividida em 3 grandes áreas, que serão tratadas como subtópicos.

5.1 Preparações iniciais

No início faz-se uma pesquisa bibliográfica básica, de forma a se compreender o atual estado das técnicas de Mineração de Dados e de Análise de sentimento, dando foco nas técnicas anteriormente citadas no referencial teórico. Neste caso, fez-se uso do algoritmo de classificação *Naive Bayes*, por sua relativa simplicidade e acurácia.

Estudam-se então formas de aplicar este algoritmo nas tecnologias selecionadas para o trabalho. Foi feito uso da plataforma Anaconda (Python; CLI), conhecida por ser a maior e mais popular distribuição para *Data Science* em Python. Na parte dos dados coletados, foi definido que eles seriam salvos em um arquivo JSON, de forma que pudesse facilmente ser lido por humanos. O arquivo de dicionário foi feito no formato de arquivo CSV, que ainda é relativamente legível por humanos mas que ocupa menos espaço em disco em relação à arquivos JSON.

Faz-se então um estudo sobre a documentação da API do SOLAR, de forma que seja possível a obtenção das amostras de textos de fóruns para a execução da Análise de Sentimentos.

Para a execução do passo anterior, se faz necessária a obtenção de contas de usuários ativos e com turmas ativas no semestre atual no SOLAR, sejam eles professores/tutores ou alunos.

São criados os formulários em papel e eletrônico onde se faz o pedido de autorização para o uso de *login* e senha dos usuários supracitados, de forma a se obter acesso aos dados para a amostra descrita no parágrafo seguinte. Neste formulário, fica claro que os dados são utilizados somente para fins estritamente acadêmicos, de forma que o dono da conta a ceder os dados fique ciente de que suas informações não serão utilizadas de má-fé.

As amostras colhidas para o processamento serão todos os *posts* de fórum dos usuários em específico que aceitem os termos propostos no formulário.

5.2 Planejamento e projeto

Primeiramente, todas as funcionalidades obrigatórias para a criação da aplicação são definidas. Estas são: um arquivo de base de dados de onde possam ser extraídos automaticamente pela aplicação os dados de usuários do SOLAR; um primeiro módulo da aplicação que faça o registro em arquivo de todos os textos obtidos do SOLAR e os guarde em uma base de dados; um segundo módulo da aplicação que faça a Análise de Sentimentos baseado no arquivo gerado pelo primeiro módulo; e por fim um menu, que disponibiliza ao usuário a opção de escolha de que módulo quer que seja executado.

Em seguida, são criadas diversas aplicações separadas e com interface bastante simplificada, de forma a se testar todos os módulos citados anteriormente. O primeiro módulo em especial é testado diversas vezes até que retorne o resultado esperado descrito pela API do SOLAR.

5.3 Experimento

Esta fase começa com a implementação da versão definitiva da ferramenta aqui proposta e descrita, num formato que seja possível a aplicação da mesma diretamente com qualquer usuário do SOLAR e que possa ser aplicada por qualquer pessoa com treinamento mínimo. No momento, espera-se que seja uma aplicação *stand-alone*, ou seja, que funcione de forma isolada, mas que pode facilmente ser portada e implementada dentro do próprio AVA e utilizada por qualquer pessoa sem treinamento.

O passo a passo se dá desta forma:

a) coleta de login e senha: os formulários eletrônico e de papel são divulgados, pedindo o acesso às contas de usuários ativos e com disciplinas ativas do SOLAR;
b) coleta dos textos: com ao menos 10 contas ativas do SOLAR no caso de alunos, ou 1 aluno com pelo menos 10 fóruns, ou 1 conta ativa de professor/tutor com ao menos 5 disciplinas, os *logins* são inseridos num banco de dados e o primeiro módulo da aplicação é executado. Um segundo banco de dados com todos os textos coletados é gerado pela primeira aplicação;

c) classificação dos textos: de posse do arquivo de textos, o segundo módulo da aplicação é executado. Com base no método de Análise de Sentimentos *Multinomial Naive Bayes*, os textos são analisados e classificados em **Positivos**, **Negativos** ou **Neutros**. Essas informações são então exibidas em uma interface *web*, de forma a melhorar a visualização dos dados;
d) referência cruzada: com as classificações geradas no passo anterior em mãos, é escolhida uma porcentagem das amostras de texto e as mesmas são então mostradas a uma ou mais pessoas (que não possuem conhecimento dos resultados apontados pela aplicação), onde elas farão seu julgamento próprio. Após isso as classificações humanas são comparadas com as mesmas feitas pelo algoritmo. Este passo serve para apontar falsos-positivos e atestar a capacidade do sistema de funcionar sem supervisão posterior.

6 PROJETO E EXECUÇÃO

Neste capítulo são descritas as partes mais técnicas do trabalho, de forma a agregar completude e aprofundar o conhecimento sobre como a aplicação funciona internamente. Basicamente trata-se da metodologia em execução, e divide-se exatamente na mesma quantidade de passos, aqui tratados como subtópicos.

6.1 Coleta de *login* e senha

A coleta de *login* e senha é feita através de formulário eletrônico (APÊNDICE A) e de papel (APÊNDICE B). Os dados são então manualmente adicionados em um arquivo no formato JSON (*login.json*). Esse arquivo então passa a ser o banco de dados de usuários, utilizados na execução da API do SOLAR.

6.2 Coleta dos textos

Para este experimento, os dados de *login* pelo menos 10 alunos com um fórum disponível, ou 1 aluno com pelo menos 10 fóruns, ou ao menos 1 professor com pelo menos 5

disciplinas ministradas são então coletados e armazenados no banco de dados citado anteriormente, de forma que a automatizar o passo a passo a seguir.

Todo o processo da coleta de textos se dá dentro do módulo Python *requisita_*.py*.[3]

Primeiramente, o arquivo *login.json* é aberto e é executada a seguinte ordem de passos para a obtenção dos textos, para cada entrada de usuário:

a) **postTokenUsuario**: faz uma requisição HTTP POST ao servidor com os dados de login e senha do usuário. Caso a requisição seja bem sucedida, a aplicação recebe uma resposta que contém o **access_token** daquele usuário. Access_Token é um identificador único que, internamente ao SOLAR, representa a sessão de um usuário. Sem ele, torna-se impossível alterar e até mesmo visualizar qualquer informação da plataforma;

b) **getUserGroups**: faz uma requisição HTTP GET ao servidor com o **access_token** obtido no passo anterior. Caso a requisição seja bem sucedida, a aplicação recebe um *array* de **userGroups**, onde cada membro é chamado de **group_id**. Group_ID são a forma que o SOLAR classifica internamente cada disciplina que o usuário participa ou já participou na plataforma;

c) **getForumId**: faz uma requisição HTTP GET ao servidor com o **access_token** e **group_id** para cada membro de **userGroups**. Caso a requisição seja bem sucedida, a aplicação recebe um **forum_id**. Forum_ID é como o SOLAR chama internamente cada fórum de discussão criado por um professor ou tutor; são neles que acontecem as interações textuais que importam a esse trabalho;

d) **getForumContent**: faz uma requisição HTTP GET ao servidor com o **access_token**, **group_id** e **forum_id**. Caso a requisição seja bem sucedida, a aplicação recebe texto e nome de usuário de todos os usuários que participaram daquele fórum em específico.

Após os textos serem obtidos com sucesso, a aplicação então escreve em disco um arquivo chamado *data_file.json*, onde todos os textos são salvos junto com nome do usuário

---

[3] todas as respostas do servidor do SOLAR são em formato JSON, que possuem diversas chaves com diversos dados diferentes, não necessários para a execução desta aplicação. Assim sendo, serão citados apenas aqueles que são importantes para a ferramenta.

que postou e o fórum ao qual o texto pertencia. Embora no escopo desta aplicação os dados exceto o texto não sejam necessários, são salvos de forma a manter o arquivo mais organizado e legível.

6.3 Classificação dos textos

Todo o processo da classificação de textos se dá dentro do módulo Python *classifica_*.py*.

6.3.1 Tipos de sentimento

Após os textos serem salvos, a classificação dos textos é feita. Para os fins deste trabalho, os sentimentos a serem analisados são **Positivo**, **Negativo** e **Neutro**.

Sentimentos **positivos** são aqui definidos como aqueles que demonstram apoio a uma ideia, causa ou que simplesmente passe a sensação de que o sentido daquela frase é agradar a alguém ou a alguma coisa.

Sentimentos **negativos** são definidos como oposto dos positivos: demonstram discordância com ideia, causa ou passam a sensação de desagrado com algo ou alguém.

Sentimentos **neutros** são apenas aqueles que não se encaixam em nenhuma das duas classificações anteriores.

6.3.2 Dicionários

Para a classificação dos textos, o método *Naive Bayes* necessita aprender o que deve ser classificado. Por isso, é necessária a existência de um **dicionário** prévio, onde já tenha sido feita a classificação, geralmente manual, de textos.

Um dicionário nada mais é que um banco de dados qualquer onde é presente uma quantidade substancial de exemplos que possam ser utilizados pelo algoritmo para serem aprendidas as regras intrínsecas da classificação a ser feita.

Neste caso o dicionário é o arquivo *dicionario.csv*, que foi composto por três bases de dados. A primeira é uma grande base com *tweets* recolhidos por um usuário anônimo

sobre o governo do estado de Minas Gerais (SILVA, 2017) por um tempo desconhecido, que possui vários exemplos de textos classificados tanto como positivos, negativos e neutros.

A segunda e terceira provém do mesmo local, uma base de palavras retiradas do *site* Kaggle (TATMAN, 2017). Ao contrário da primeira, que contém vários textos classificados, ambas são apenas palavras avulsas mas ainda assim classificadas.

6.3.3 Classificação

Primeiramente, o arquivo *dicionario.csv* é aberto, lido e alimentado ao algoritmo de Análise de Sentimentos *Multinomial Naive Bayes*, que é incluído no pacote scikit-learn. Em seguida, o arquivo *data_file.json*, que contém os textos salvos é aberto e carregado para a aplicação. Logo após, um *array* de textos presente direto no código para propósitos de teste é compilado ao conteúdo de texto do arquivo carregado anteriormente. A classificação é então feita pelo algoritmo e fica pronta para ser exibida pela aplicação.

6.3.4 Exibição dos dados

Depois de feita a classificação, todos os textos são exibidos na CLI (*Command-line interface*), apenas como forma de *debug* caso algum problema ocorra na visualização principal. Após isso, é mostrado um *link* na tela, que caso seja clicado abrirá o navegador numa página *web* formatada para a visualização final e correta dos dados obtidos.

A tela no navegador possui as seguintes informações:

a) **precisão:** a precisão geral média do algoritmo baseada nos dicionários que recebeu. Valores são no intervalo de 0~1;

b) estatísticas: Tabela completa com várias estatísticas geradas automaticamente pelo algoritmo. Possui os campos ***precision***, ***recall***, ***f1-score*** e várias médias para cada tipo de sentimento classificado. *Precision* é definida como o número total de resultados relevantes obtidos dividido pelo número total de resultados obtidos; *recall* é definido como o total de resultados relevantes obtidos dividido pelo número total de resultados relevantes na base de dados (TING, 2017). *F1-Score* é usado para analisar a acurácia da predição e é definido como a média harmônica da *precision* e do *recall*; assim sendo, se localiza entre os

dois, mas é mais próximo do menor valor; um sistema com alto F1 tem então boa *precision* e bom *recall* (SAMMUT; WEBB, 2017).Valores são no intervalo de 0~1, exceto em *support*;

c) **matriz de confusão:** Tabela que demonstra a acurácia para cada classificação em relação a si mesma e a outras. Valores inteiros;
d) **frases classificadas:** Tabela que mostra todos os textos analisados pelo algoritmo e a sua classificação.

6.4 Cross-reference

Após a análise dos dados, uma amostra de 10% das frases é analisada por um humano que não tem conhecimento da classificação atribuída pelo algoritmo a cada frase, de forma a se testar a coerência da classificação pela máquina com a classificação feita por um humano.

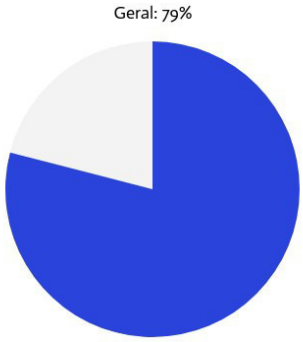
Após a classificação pelo humano, os dados são guardados em um banco de dados também em formato JSON para a referência cruzada com os dados obtidos pelo algoritmo, e será conferido manualmente por um terceiro.

Esse passo é apenas uma forma adicional de se atestar a qualidade da classificação automática de textos pela aplicação.

7 RESULTADOS

Neste capítulo os resultados são apresentados e discutidos. Todas as figuras aqui mostradas são geradas e se encontram disponíveis na interface *web* gerada pela segunda parte da aplicação.

Gráfico 1 — Gráfico da precisão geral



Fonte: dados do trabalho

Figura 7 — Precisão geral com 16 casas decimais

Fonte: dados do trabalho

Como é possível ver no gráfico 1 e na figura 7, a aplicação do algoritmo atingiu uma taxa de sucesso de aproximadamente 79%, o que pode ser considerado um sucesso.

Tabela 2 — Tabela de estatísticas

	Precision	**Recall**	**F1-Score**	**Support**
Positivo	0.93	0.64	0.76	5084
Negativo	0.75	0.90	0.81	5227
Neutro	0.70	0.85	0.77	2453
micro avg	0.79	0.79	0.79	12764
macro avg	0.79	0.80	0.78	12764
weighted avg	0.81	0.79	0.78	12764

Fonte: dados do trabalho.

Da Tabela 2 é possível se observar que a maior taxa de acertos é em relação aos textos com teor **positivo**, atingindo 93% de chance de classificação correta; em segundo lugar se coloca a classificação de textos com teor **negativo**, com 75% de chance de acerto; por último, com a menor chance mas ainda assim com um valor considerado aceitável vem a classificação de textos com teor **neutro**, com taxa de acerto de apenas 70%.

Também pode-se observar que a melhor taxa de **recall** pertence à classificação de frases com teor **negativo**, com 90% das instâncias indicadas sendo corretas; frases com teor **neutro** em segundo lugar, com 85% das instâncias indicadas sendo apontadas corretamente; por último, frases com teor **positivo** com apenas 64% das instâncias indicadas apontadas de forma correta.

Por fim, dela podemos ainda podemos conferir que a precisão média de todas as classificações ficou em excelentes 81%, enquanto que o recall ficou com 79%.

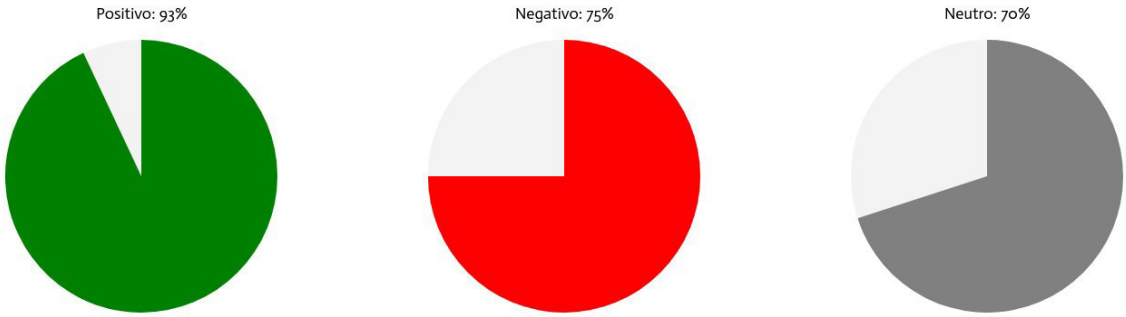
Tabela 3 — Matriz de confusão

		Predito		
		Negativo	**Neutro**	**Positivo**
Real	**Negativo**	**4693**	405	129
	Neutro	253	**2088**	112
	Positivo	1350	488	**3246**

Fonte: dados do trabalho.

Na Tabela 3, podemos observar de forma quantificável os resultados do **recall**; aqui se mostram as quantidades de “positivos” e “falso-positivos” de cada classe, em relação de uma com a outra. Como podemos ver, a classe de textos **negativos** é a que realmente possui a maior taxa de reais positivos, enquanto que a classe de textos **positivos** é a que tem a maior chance de ativar um falso-positivo.

Gráfico 2 — Gráfico da precisão de cada uma das classificações



Fonte: dados do trabalho

O Gráfico 2 é apenas uma representação gráfica da precisão apontada pela Tabela 1, e que demonstra que precisão não é a medida mais importante a ser considerada. Se olhada de forma isolada, pode levar ao erro de que as frases com conotação **positiva** são as que o algoritmo mais acerta, sendo que na verdade o caso é o oposto.

Figura 8 — Exemplo de tabela de classificação geral gerada pelo sistema

Frases classificadas

Totais	
Positivos	58
Negativos	159
Neutros	398
Total	615

Fonte: dados do trabalho

A Figura 8 é resultado de um teste com um único usuário, mas com uma disciplina bastante ativa. Foram classificados 615 textos, sendo que 398 foram considerados neutros, 159 negativos e 58 positivos. Essa informação é disponibilizada para que o usuário possa ter uma visão geral rápida da situação, e é a informação que seria mostrada ao professor/tutor caso fosse implementada no SOLAR.

Figura 9 — Exemplos de frases classificadas, exibidas na interface web

#	Classificação	Texto
1	Negativo	Esse governo está no início, vamos ver o que vai dar
2	Neutro	Estou muito feliz com o governo de Minas esse ano
3	Negativo	O estado de Minas Gerais decretou calamidade financeira!!!
4	Negativo	A segurança desse país está deixando a desejar
5	Negativo	Repugnante a atitude desse senhor

Fonte: dados do trabalho

A Figura 9 mostra alguns exemplos de frases classificadas e como a tabela é organizada na interface *web*, dispondo das colunas de número, classificação e texto. A coluna de número serve para que o usuário do sistema possa ir direto para a frase que desejar, a de classificação além de texto indicativo possui esquema de cores para rápida identificação do conteúdo (verde = positivo, vermelho = negativo, cinza = neutro), e a coluna de texto traz a sentença completa que foi analisada. As frases aqui mostradas são apenas exemplos já disponíveis no código, então não são retiradas de textos de fórum do SOLAR.

Figura 10 — Texto corretamente classificado como positivo

101	Positivo	Quando o movimento feminista trabalha com mulheres acima de 18 anos e encontram crianças de 11 ou 12 anos, que estão em situação de exploração sexual, eles encaminham para a secretaria de Direitos Humanos e para os movimentos em defesa da criança

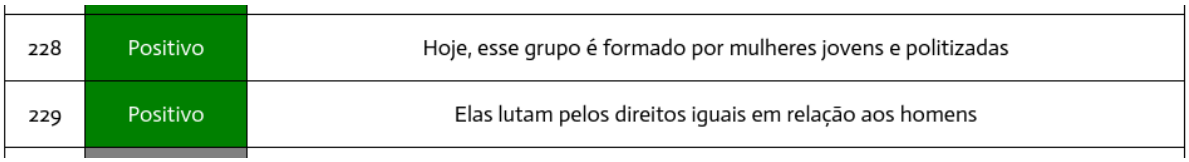
Fonte: dados do trabalho

Figura 11 — Texto corretamente classificado como positivo

223	Positivo	O conhecimento é a chave para acabar com o preconceito

Fonte: dados do trabalho

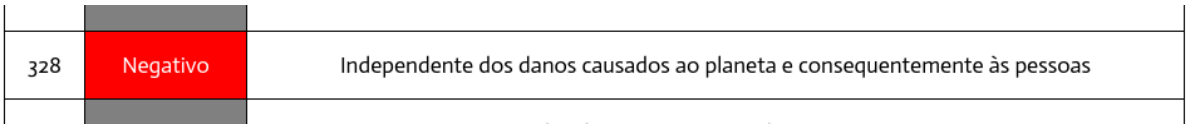
Figura 12 — Texto corretamente classificado como positivo



Fonte: dados do trabalho

As Figuras 10, 11 e 12 mostram exemplos de textos que foram classificados corretamente como positivos. Como explicado anteriormente, positivo aqui se considera como uma frase que apoie uma ideia desejada. A disciplina de onde esses textos foram extraídos se chama Diferença e Enfrentamento Profissional nas Desigualdades Sociais, e o fórum em específico continha discussões que falavam exatamente sobre este assunto. Levando em conta o tema do fórum, é possível observar que os textos corroboram positivamente com o tema discutido.

Figura 13 — Texto corretamente classificado como negativo

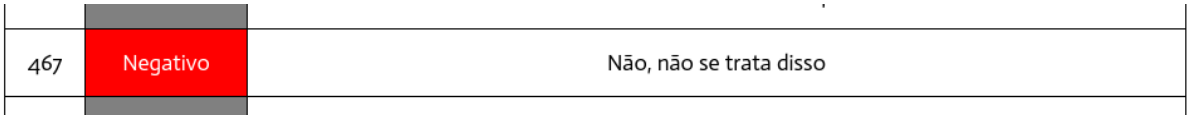


Fonte: dados do trabalho

Figura 14 — Texto corretamente classificado como negativo

Fonte: dados do trabalho

Figura 15 — Texto corretamente classificado como negativo

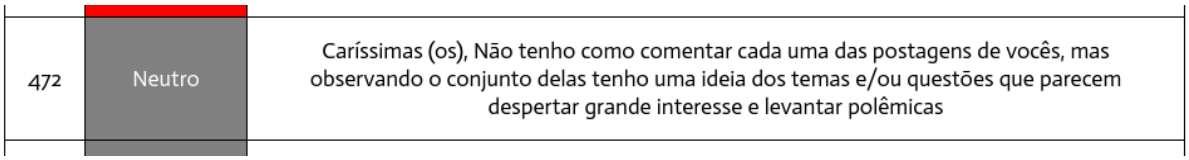


Fonte: dados do trabalho

As Figuras 13, 14 e 15 demonstram textos que tem a conotação negativa, que mostram discordância com o que foi falado anteriormente e que possuem o atributo de

negarem qualquer argumento antes proposto. Assim sendo, demonstram que o algoritmo possui a capacidade de julgar corretamente este tipo de informação.

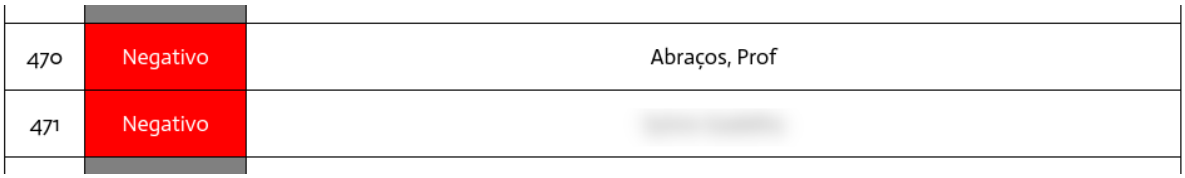
Figura 16 — Texto subjetivamente classificado como neutro



Fonte: dados do trabalho

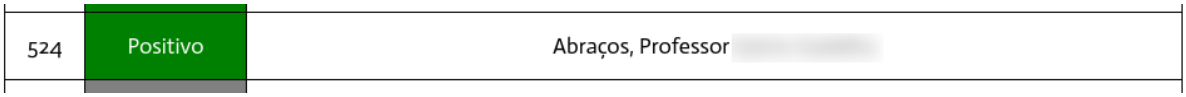
A Figura 16 demonstra um texto que pode muito bem ser considerado neutro, mas é um caso um pouco mais complicado de ser julgado por um humano, já que ao mesmo tempo que neutras, existem conotações positivas, como o professor elogiar o interesse que os textos despertam.

Figura 17 — Textos erroneamente classificados como negativos



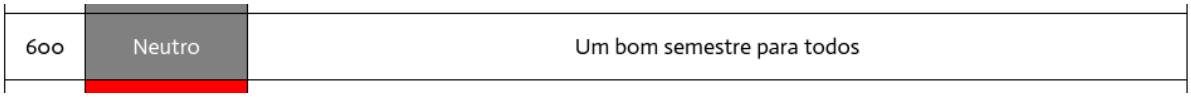
Fonte: dados do trabalho

Figura 18 — Texto erroneamente classificado como positivo



Fonte: dados do trabalho

Figura 19 — Texto erroneamente classificado como neutro

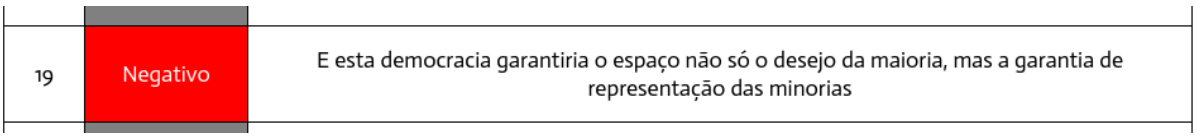


Fonte: dados do trabalho

As Figuras 17, 18 e 19 mostram o tipo de falha que pode ocorrer na classificação de textos. Na Figura 17, por algum motivo, o algoritmo declara que abraços e o nome do

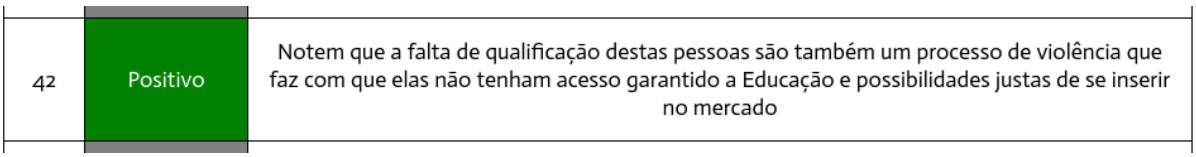
professor tem a conotação negativa, mas na Figura 18 o mesmo texto é considerado como positivo, apenas por ter sido colocado por extenso e na mesma linha. Entretanto, sendo analisado por um humano, esses dois trechos além de terem a mesma conotação, neste caso deveriam se encaixar em Neutro, pois temos o conhecimento que esse tipo de encerramento é mera formalidade. Já a Figura 19 poderia encaixar melhor na classificação positiva, afinal é por nós vista como uma mensagem de encorajamento.

Figura 20 — Classificação incorreta do texto como negativo por dicionário falho



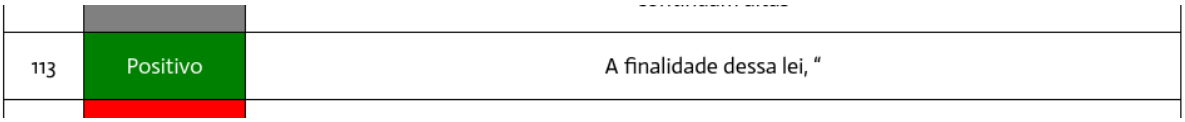
Fonte: dados do trabalho

Figura 21 — Classificação incorreta do texto como positivo por dicionário falho



Fonte: dados do trabalho

Figura 22 — Classificação incorreta do texto como positivo por dicionário falho



Fonte: dados do trabalho

As Figuras 20, 21 e 22 demonstram outro tipo de falha que pode ocorrer, mas que parece mais estar relacionada ao fato do dicionário utilizado ser de um escopo limitado. As frases que compõem o dicionário foram extraídas do *twitter* por um usuário desconhecido, mas todas se referem a tuítes sobre o Governo do Estado de Minas Gerais. Por observação, é possível perceber que o algoritmo demonstra uma certa tendência em relação à frases negativas e neutras, já que geralmente é este tipo de teor de mensagem que as pessoas que se engajam com *social media* de órgãos de governo utilizam.

Figura 23 — Tipo de erro que pode ocorrer ao separar os grandes posts de fórum em textos menores.

114	Negativo	
115	Negativo	
116	Negativo	
117	Negativo	

Fonte: dados do trabalho

Por fim, a Figura 23 mostra um tipo de erro que acontece ao separar os longos textos em frases menores: alguns caracteres utilizados exclusivamente por alguns sistemas operacionais, como por exemplo o retorno de carro (caracteres \r nos sistemas Mac antes das versões 10.x; \n nos sistemas a partir das versões 10.x e Unix; \r\n em todos os sistemas Windows), podem acabar não sendo eliminados no processo de limpeza do texto. Assim, o algoritmo de classificação não entende como se portar perante esse tipo de informação e atribui uma classificação qualquer, neste caso a negativa.

8 CONCLUSÃO

O trabalho aqui representado teve o objetivo de demonstrar que, mesmo com técnicas simples, é possível se atingir um nível satisfatório de análise de sentimentos em textos de fóruns de AVAs, em específico o SOLAR, de forma a se ajudar os professores/tutores a, por exemplo, direcionarem uma turma ou descobrirem se a turma em geral concorda ou discorda de um dado tema.

Sobre as dificuldades, ocorreram diversas com variados graus de complexidade no decorrer do trabalho. Análise de Sentimento de textos em português brasileiro ainda não é um tema devidamente abordado em grande quantidade, o que leva a uma limitação de referências, algoritmos, técnicas e até mesmo de dicionários. Possivelmente a maior dificuldade no decorrer da execução do trabalho pode ser atribuída ao fato de que não se encontra facilmente um dicionário de textos em português já classificado e de boa qualidade –

o dicionário utilizado nesta aplicação constitui-se de nada menos que três outros concatenados quase que manualmente.

Outra dificuldade foi a escolha de ferramentas e linguagens para a execução. Originalmente planejava-se fazer uso da ferramenta WEKA, que economizaria uma quantidade de trabalho por já ter implementada vários algoritmos de Análise de Sentimento, mas logo foi descoberto que não era possível encontrar um dicionário em português que fosse compatível, então ficou decidido pela implementação manual da aplicação.

Como o SOLAR é uma plataforma *web*, o primeiro pensamento foi o de desenvolver um cliente que pudesse ser utilizado via navegador, de forma que qualquer usuário por mais leigo que fosse pudesse testar sem o menor problema; assim sendo, a aplicação foi desenvolvida mas logo surgiu a descoberta de que os navegadores modernos barravam a execução dela devido à violação de uma regra de *cross-domain*. Então, ainda na ideia de manter nas tecnologias baseadas em JavaScript, o código foi migrado para o *framework* Node.JS, que é excelente e supria totalmente a necessidade do projeto de obter os textos e o salvarem em um arquivo JSON. Entretanto, nenhuma das bibliotecas que implementavam *Naive Bayes* funcionavam com o dicionário montado para a aplicação, e o trabalho para torná-lo utilizável com as tecnologias existentes seria considerável. Assim, a implementação do projeto passou por outras linguagens, mais notavelmente por R e Python.

Com Python foi possível executar todas as funcionalidades que o Node.JS conseguia e mais, como por exemplo a execuçã da análise de sentimentos com *frameworks* bem documentados. Com isso, o código teve que ser completamente reescrito e uma nova sintaxe e paradigmas de linguagem tiveram que ser aplicados.

Como apontado nos resultados, o dicionário utilizado tem uma certa tendência com classificações neutras e negativas justamente por parte dele, aproximadamente 40%, ter origem de tweets sobre o governo de um estado. Ainda assim, os resultados mostraram que mesmo com a aparente “deficiência”, as previsões ainda se mostram corretas em sua maioria.

Este trabalho surgiu de uma ideia do professor Emanuel Coutinho, que é a de tentar ajudar no trabalho dos professores/tutores que geralmente acumulam muitas atividades relacionadas com a mídia escrita. Em nossa época atual, muitas tarefas podem ser automatizadas e relegadas à computadores sem muitos riscos de problemas serem gerados.

Como demonstrado nos resultados, a aplicação possui uma taxa substancialmente satisfatória de acertos, indicando que ela é o adequada o suficiente pro objetivo que foi proposto.

Todo o algoritmo utilizado aqui pode ser facilmente importado, portado ou implementado na plataforma SOLAR, o que o torna útil caso os desenvolvedores do mesmo demonstrem interesse de implementar alguma funcionalidade similar.

Também, este trabalho pode ser usado como base para outros na mesma área, e até talvez condensado em um artigo para publicação em revista ou periódico da área de mineração de dados, seja como incentivo a outros para a ampliação da Análise de Sentimentos em português brasileiro ou apenas como mais uma base para outros trabalhos que possam se beneficiar do conteúdo aqui descrito.

Por fim, como possíveis trabalhos futuros imaginam-se: a criação uma ferramenta que seja mais abrangente e que funcione em mais de um AVA; a criação de *plugins* que funcionem como extensões diretas de outros AVAs; um estudo semelhante ao aqui apresentado no tocante ao tema de Análise de Sentimentos em português brasileiro, mas que envolva a utilização de redes sociais como o *Facebook* ou *Twitter*, por possuírem uma base de dados disponível bem mais ampla; e por fim a aplicação de outros métodos e algoritmos de Análise de Sentimento, sejam estes supervisionados ou não.

REFERÊNCIAS

CAETANO, Josemar Alves; LIMA, Hélder Seixas; SANTOS, Mateus Freira dos; MARQUES-NETO, Humberto Torres. Utilizando Análise de Sentimentos para Definição da Homofilia Política dos Usuários do Twitter durante a Eleição Presidencial Americana de 2016. **XXXVII Congresso da Sociedade Brasileira de Computação (CSBC), São Paulo,** 2017.

CHRISTHIE, William; REIS, Julio C. S.; BENEVUNUTO, Fabrício; MORO, Mirella M.; ALMEIDA, Virgílio. Detecção de Posicionamento em Tweets sobre Política no Contexto Brasileiro. **Brazilian Workshop on Social Network Analysis and Mining (BraSNAM_CSBC)**, [S.l.], v. 7, n. 1/2018, july 2018. ISSN 2595-6094. Disponível em: <http://portaldeconteudo.sbc.org.br/index.php/brasnam/article/view/3583>. Acesso em: 05 nov. 2018.

COUTINHO, Emanuel Ferreira; JUNIOR, Antônio de Lisboa Coutinho; SARMENTO, Wellington Wagner Ferreira. Desenvolvimento de Aplicações para Educação à Distância: O Ambiente Virtual de Aprendizagem SOLAR. **CBSOFT, Brasília**, 2013a.

COUTINHO, Emanuel F.; MOREIRA, Leonardo O.; SARMENTO, Wellington WF. MAAT-Sistema de Avaliação de Alunos e Tutores para um Ambiente Virtual de Aprendizagem. **IX Simpósio Brasileiro de Sistemas de Informação (SBSI2013)**, 2013b.

DILLENBOURG, Pierre; SCHNEIDER, Daniel; SYNTETA, Paraskevi. Virtual learning environments. In: **3rd Hellenic Conference" Information & Communication Technologies in Education"**. Kastaniotis Editions, Greece, 2002. p. 3-18.

DOSCIATTI, Mariza Miola; FERREIRA, Lohann Paterno Coutinho; PARAISO, Emerson Cabrera. Identificando Emoções em Textos em Português do Brasil usando Máquina de Vetores de Suporte em Solução Multiclasse. **X Encontro Nacional de Inteligência Artificial e Computacional (ENIAC), Fortaleza,** 2013.

ESULI, Andrea; SEBASTIANI, Fabrizio. Determining term subjectivity and term orientation for opinion mining. In: **11th Conference of the European Chapter of the Association for Computational Linguistics**. 2006.

HAND, DJ; SMYTH, Mannila H. Principles of data mining. 2001, MIT Press.

IBM. Ibm. 2013. Disponível em: <https://www-03.ibm.com/press/br/pt/pressrelease/41300.wss>. Acesso em abril 2018.

KANSAON, Daniel P.; BRANDÃO, Michele A.; PINTO, Saulo A. de Paula. Análise de Sentimentos em Tweets em Português Brasileiro.**Brazilian Workshop on Social Network Analysis and Mining (BraSNAM_CSBC)**, [S.l.], v. 7, n. 1/2018, july 2018. ISSN 2595-6094. Disponível

em: <http://portaldeconteudo.sbc.org.br/index.php/brasnam/article/view/3578>. Acesso em: 05 nov. 2018.

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

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. **An introduction to Information Retrieval**. Cambridge University Press, 2009.

MICHALSKI, Ryszard S.; CARBONELL, Jaime G.; MITCHELL, Tom M. (Ed.). **Machine learning: An artificial intelligence approach**. Springer Science & Business Media, 2013.

O'LEARY, Ros; RAMSDEN, Andy. Virtual learning environments. **Learning and Teaching Support Network Generic Centre/ALT Guides, LTSN. Retrieved July**, v. 12, p. 2005, 2002.

POLAK, Ymiracy N. de Souza; DINIZ, José Alves; SANTANA, José Rogério. **Dialogando sobre metodologia científica**. Fortaleza: Editora Universidade Federal do Ceará, 2011. 177 p.

RENNIE, Jason D. et al. Tackling the poor assumptions of naive bayes text classifiers. In: **Proceedings of the 20th international conference on machine learning (ICML-03)**. 2003. p. 616-623.

ROSEMANN, Douglas; RAABE, André LA; TEIVE, Raimundo C. Ghizoni. Personalização de Conteúdo e Avaliação Multicritério em Ambiente Virtual de Aprendizagem de Código Aberto. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2014. p. 203.

SAMMUT, Claude; WEBB, Geoffrey I. **Encyclopedia of Machine Learning.** 2ª ed. Boston, MA: Springer, 2017. 513 p.

SILVA, Leandro. 2017. **Tweets from MG/BR.** Disponível em: <https://www.kaggle.com/leandrodoze/tweets-from-mgbr>. Acesso em setembro 2018.

SOLAR. Solar. 2018. Disponível em: <http://www.solar.virtual.ufc.br/faq>. Acesso em abril 2018.

TATMAN, Rachael. 2017. **Sentiment Lexicons for 81 Languages.** Disponível em: <https://www.kaggle.com/rtatman/sentiment-lexicons-for-81-languages#sentiment-lexicons.zip>. Acesso em setembro 2018.

TING, Kai Ming. Precision and Recall. In: SAMMUT, Claude; WEBB, Geoffrey I. **Encyclopedia of Machine Learning.** 2ª ed. Boston, MA: Springer, 2017. p. 990-991.

WITTEN, Ian H. et al. **Data Mining: Practical machine learning tools and techniques**. Morgan Kaufmann, 2016.

YU, Hong; HATZIVASSILOGLOU, Vasileios. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In: **Proceedings of the 2003**

conference on Empirical methods in natural language processing. Association for Computational Linguistics, 2003. p. 129-136.

APÊNDICE A — FORMULÁRIO ELETRÔNICO

Coleta de dados de login do SOLAR

Este formulário tem por objetivo a coleta de dados válidos de login na plataforma SOLAR, de usuários que possuam disciplinas ativas no semestre corrente.

Os dados aqui recolhidos serão mantidos em sigilo e serão utilizados estritamente para fins acadêmicos. Este formulário é parte do Trabalho de Conlusão de Curso de Mário Silva Ribeiro, estudante do curso de Sistemas e Mídias Digitais da Universidade Federal do Ceará.

*Obrigatório

1. **Você está ciente de que a plataforma Google não aconselha o envio de senhas por formulários?** *

   Marcar apenas uma oval.

   - ◯ Sim *Ir para a pergunta 2.*
   - ◯ Não *Ir para "Obrigado pela sua participação!".*

Confirmação de participação

2. **Você concorda em disponibilizar seus dados de login na plataforma SOLAR para uso estritamente acadêmico?** *

   Marcar apenas uma oval.

   - ◯ Sim
   - ◯ Não *Ir para "Obrigado pela sua participação!".*

Dados a serem enviados

Por favor, insira seu nome de usuário e senha nos respectivos campos

3. **Login** *

4. **Senha** *

Obrigado pela sua participação!

Mesmo enviando ou não seus dados, obrigado por ter tido tempo de verificar e ter pensado em ajudar neste trabalho.

APÊNDICE B — FORMULÁRIO IMPRESSO

TERMO DE AUTORIZAÇÃO PARA O USO DE DADOS DE LOGIN NA PLATAFORMA SOLAR

Eu, _____, [ ] Professor [ ] Aluno, portador da matrícula _____ na Universidade Federal do Ceará, estou ciente da utilização, para fins exclusivamente acadêmicos, do uso de meus dados de login no ambiente de aprendizagem virtual SOLAR pelo aluno da Universidade Federal do Ceará, MÁRIO SILVA RIBEIRO, matrícula de número 376825, estudante do curso de SISTEMAS E MÍDIAS DIGITAIS. Concordo com o uso dos dados para ajudar na execução da aplicação de TRABALHO DE CONCLUSÃO DE CURSO, desde que os mesmos permaneçam privados e sejam utilizados exclusivamente no semestre 2018.1.

Login/Nome de usuário: _____

Senha: _____

Fortaleza, _____ de _____ de 2018.

_____

Assinatura do Cedente

_____

Mário Silva Ribeiro

ANEXO A — DOCUMENTAÇÃO DA API DO SOLAR

Autenticação de usuário (obtendo access_token)

Chamada

POST /oauth/token

Parâmetros

grant_type: obrigatório, valor: password
login: obrigatório
password: obrigatório

Retorno

{access_token: “XXXXXX”, tonek_type: "XXX"}

Exemplo de chamada

POST /oauth/token?grant_type=password&login=aluno&password=senha123

Exemplo de retorno


```
{
  "access_token": "ffeebabd16bf6f6679764a776fe9c190e36416c7aac08b0d82707844f1963a4b",
  "token_type": "bearer"
}
```


Lista de disciplinas de um usuário

Chamada

GET /api/v1/curriculum_units

Parâmetros

access_token: obrigatório

Retorno

array de disciplinas que o usuário tem acesso (todos os semestres)

```
[
   {id: X, code: "XXX", name: "XXXX"}, ...
]
```

Exemplo de chamada (access token omitido)

GET /api/v1/curriculum_units

Exemplo de retorno


```
[
   {
      "id": 1,
      "code": "RM404",
      "name": "Introducao a Linguistica"
   },
   {
      "id": 3,
      "code": "RM301",
      "name": "Quimica I"
   },
   {
      "id": 2,
      "code": "RM405",
      "name": "Teoria da Literatura I"
   }
]
```


Lista de disciplinas de um usuário com as turmas ativas

Chamada

GET /api/v1/curriculum_units/groups

Parâmetros

access_token: obrigatório

Retorno

array de disciplinas que o usuário tem acesso com as turmas ativas (todos os semestres)

```
[
   {id: X, code: "XXX", name: "XXXX",
    groups: [
          {id: X, code: "XXX", name: "XXXX", semester: "XXXX"},
    ] }, ...
]
```

Exemplo de chamada (access token omitido)

GET /api/v1/curriculum_units/groups

Exemplo de retorno


```
[
  {
    "id": 1,
    "code": "RM404",
    "name": "Introducao a Linguistica",
    "groups": [
      {
        "id": 1,
        "code": "IL-FOR",
        "name": nil,
        "semester": "2011.1"
      }
    ]
  },
  {
    "id": 3,
    "code": "RM301",
    "name": "Quimica I",
    "groups": [
      {
        "id": 3,
        "code": "QM-CAU",
        "name": nil,
        "semester": "2011.1"
      }
    ]
  },
  {
    "id": 2,
    "code": "RM405",
    "name": "Teoria da Literatura I",
    "groups": [
      {
        "id": 2,
        "code": "TL-CAU",
```

```
            "name": nil,
            "semester": "2011.1"
        }
    ]
  }
]
```


Lista de turmas de um usuário

Chamada

GET /api/v1/curriculum_units/**curriculum_unit_id**/groups

Parâmetros

curriculum_unit_id: id da disciplina, obrigatório, integer
access_token: obrigatório

Retorno

array de turmas ativas que o usuário tem acesso

```
[
  {id: X, code: "XXX", name: "XXXX", semester: X }, ...
```

]

Exemplo de chamada (access token omitido)

GET /api/v1/curriculum_units/3/groups

Exemplo de retorno


```
[
  {
    "id": 3,
    "code": "QM-CAU",
    "name": null,
    "semester": "2011.1"
  }
]
```


Lista de turmas de um usuário a partir de semestre

Chamada

GET /api/v1/user/groups

Parâmetros

curriculum_unit_id: id da disciplina, opcional, integer
course_id: id do curso, opcional, integer
curriculum_unit_type_id: id do tipo da disciplina, opcional, integer
profiles_ids: ids dos perfis a serem retornados, opcional, array
semester: nome do semestre, opcional, string
access_token: obrigatório

Retorno

array de turmas ativas que o usuário tem acesso considerando os possíveis dados informados; se o semestre não é informado, retorna apenas as turmas dos semestres correntes

[
{id: X, code: "XXX", uc_code: "XXXX", uc_name: “XXX“, course_code: “XXX“, course_name: “XXX“, semester_name: “XXX“, type: “XXX“, profiles: [X, X, X] }, ...

Lista de fóruns de uma turma

Chamada

GET /api/v1/groups/**group_id**/discussions

Parâmetros

group_id: id da turma, obrigatório, integer
access_token: obrigatório

Retorno

{id: X, name: “XXX”, description: “XXX”, start_date: “XXX”, end_date: “XXX”, files: [{id: X, name: “XXX”, content_type: “XXX”, updated_at: “XXX”, size: X, url: “XXX”}], status: X, last_post_date: “XXX”, researcher: true/false, can_post: true/false }

Valores para o parâmetro status:

0: fórum não iniciado
1: fórum aberto (professores e tutores ficam com o fórum aberto 3 dias a mais que alunos)
2: fórum encerrado

Valores para o parâmetro files[:url]:

A url dos possíveis arquivos deve ser acessada enviando o access_token do mesmo modo que a lista de fóruns.

Exemplo de chamada (access token omitido)

GET /api/v1/groups/3/discussions

Exemplo de retorno


```
[
  {
    "id": 2,
    "name": "Forum 2",
    "description": "O empenho em analisar o novo modelo estrutural aqui preconizado agrega valor ao estabelecimento do sistema de participação geral.",
    "start_date": "2011-07-25",
    "end_date": "2018-06-08",
    "files": [],
    "status": "1",
    "last_post_date": "2018-03-20T11:22:13.710-03:00",
    "researcher": false,
    "can_post": true
  },
  {
```

```
    "id": 1,
    "name": "Forum 1",
    "description": "No mundo atual, o fenômeno da Internet prepara-nos para enfrentar situações atípicas decorrentes dos conhecimentos estratégicos para atingir a excelência.",
    "start_date": "2011-09-20",
    "end_date": "2018-04-06",
    "files": [
      {
        "id": 1,
        "name": "pdf_branco.pdf",
        "content_type": "application/pdf",
        "updated_at": "2018-04-12T10:14:30.793-03:00",
        "size": 1114,
        "url": "http://localhost/discussions/api_download?file_id=1"
      }
    ],
    "status": "2",
    "last_post_date": "2018-03-20T11:22:13.709-03:00",
    "researcher": false,
    "can_post": true
  },
  {
    "id": 7,
    "name": "Forum 5",
    "description": "Todas estas questões, devidamente ponderadas, levantam dúvidas sobre se a adoção de políticas descentralizadoras faz parte de um processo de gerenciamento do processo de comunicação como um todo.",
    "start_date": "2018-05-09",
    "end_date": "2018-09-09",
    "files": [],
    "status": "0",
    "last_post_date": null,
    "researcher": false,
    "can_post": true
  },
  {
    "id": 8,
    "name": "Forum 6",
    "description": "É importante questionar o quanto o fenômeno da Internet estende o alcance e a importância das condições financeiras e administrativas exigidas.",
    "start_date": "2018-07-09",
    "end_date": "2018-11-09",
    "files": [],
    "status": "0",
    "last_post_date": null,
    "researcher": false,
    "can_post": true
  }
]
```


Lista de posts para um fórum (nível 1)

Chamada

GET /api/v1/discussions/discussion_id/posts

Parâmetros

discussion_id: id do fórum, obrigatório, integer
group_id: id da turma, obrigatório, integer
access_token: obrigatório
limit: limite de posts por página, padrão são 20
page: página, default: 1
ignore_drafts: se os rascunhos do usuário devem ser ignorados ou não, default: true

Obs

Posts de rascunho só podem ser visualizados pelo autor.
Este método retorna apenas o 1º nível de posts. Para visualizar o nível seguinte ao post (respostas a ele) deve-se usar o método de lista de posts para um post.

Retorno

{id: X, parent_id: X, profile_id: X, user_id: X, level: X, content: "XXX", created_at: "XXX", children_count: X, draft: true/false, files: [id: X, name: "XXX", content_type: "XXX", updated_at: "XXX", size: X, url: "XXX"], user_nick: "XXX"}

Valores para o parâmetro files[:url]:
A url dos possíveis arquivos deve ser acessada enviando o access_token do mesmo modo que a lista de posts.

Valores para o parâmetro parent_id: id do post pai
Valores para o parâmetro profile_id: id do usuário autor
Valores para o parâmetro level: nível do post
Valores para o parâmetro children_count: quantidade de posts na árvore no nível imediatamente seguinte

Exemplo de chamada (access token omitido)

GET /api/v1/discussions/2/posts?group_id=3

Exemplo de retorno


```
[
  {
    "id": 11,
    "parent_id": null,
    "profile_id": 4,
    "user_id": 1,
    "level": 1,
```