FABRÍCIO DOS REIS NETO GUIMARÃES

DESCOBRINDO PADRÕES DE GÊNEROS DAS MENSAGENS EM FÓRUNS DE DISCUSSÃO DE AMBIENTES VIRTUAIS DE APRENDIZAGEM VIA MINERAÇÃO DE TEXTO

LAVRAS – MG

2015

FABRÍCIO DOS REIS NETO GUIMARÃES

DESCOBRINDO PADRÕES DE GÊNEROS DAS MENSAGENS EM FÓRUNS DE DISCUSSÃO DE AMBIENTES VIRTUAIS DE APRENDIZAGEM VIA MINERAÇÃO DE TEXTO

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Inteligência Computacional e Processamento Gráfico, para a obtenção do título de Mestre.

Orientador

Dr. Ahmed Ali Abdalla Esmin

LAVRAS – MG

2015

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Guimarães, Fabrício dos Reis Neto.
Descobrindo padrões de gêneros das mensagens em fóruns de discussão de ambientes virtuais de aprendizagem via mineração de texto / Fabrício dos Reis Neto Guimarães. – Lavras : UFLA, 2015.
114 p.

Dissertação (mestrado acadêmico)–Universidade Federal de Lavras, 2015.
Orientador(a): Ahmed Ali Abdalla Esmin.
Bibliografia.

1. Mineração de dados. 2. Classificação multi-classe. 3. Classificação multi-rótulo. 4. Desbalanceamento dos dados. I. Universidade Federal de Lavras. II. Título.

FABRÍCIO DOS REIS NETO GUIMARÃES

DESCOBRINDO PADRÕES DE GÊNEROS DAS MENSAGENS EM FÓRUNS DE DISCUSSÃO DE AMBIENTES VIRTUAIS DE APRENDIZAGEM VIA MINERAÇÃO DE TEXTO

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Inteligência Computacional e Processamento Gráfico, para a obtenção do título de Mestre.

APROVADA em 06 de Fevereiro de 2015.

Dr. Ahmed Ali Abdalla Esmin	UFLA
Dr. André Luiz Zambalde	UFLA
Dr. Cristiano Leite de Castro	UFMG

Dr. Ahmed Ali Abdalla Esmin
Orientador

LAVRAS – MG

2015

AGRADECIMENTOS

Agradeço e dedico esta conquista à memória de minha mãe que me auxilia de onde estiver e sua ausência é indescritível.

Agradeço à minha família por me apoiarem. Aos amigos que sempre fazem parte de qualquer conquista, auxiliando para que o caminho seja menos árduo.

Agradeço à Universidade Federal de Lavras, ao Departamento de Ciência da Computação e ao Programa de Pós-Graduação em Ciência da Computação da UFLA, pela estrutura oferecida e pela oportunidade de realização do Mestrado.

Agradeço ao meu orientador Prof. Ahmed pela confiança, orientação e amizade durante este trabalho.

A todos que contribuíram de forma direta ou indireta para a realização deste trabalho e conquista deste título.

RESUMO

Ambientes Virtuais de Aprendizagem fornecem um conjunto de ferramentas para auxiliar o processo de ensino e aprendizagem. Dentre elas, os fóruns de discussões são muito utilizados, pois permitem a troca de mensagens textuais entre alunos e tutores. O acompanhamento da grande quantidade de mensagens é uma tarefa que demanda grande quantidade de tempo e esforço tornando-se necessário o uso de técnicas para encontrar mensagens com mesmo conteúdo. Neste contexto a classificação de texto pode ser utilizada para agrupar as mensagens por gêneros, fornecendo uma nova visão sobre as mensagens para o acompanhamento destes fóruns. Para classificar a mensagem em um gênero, são utilizados algoritmos de classificação multi-classe, e para classificar mais de um gênero são utilizados algoritmos de classificação multi-rótulo. Um problema da classificação de texto conhecido e ocorrido neste trabalho foi o desbalanceamento da distribuição dos dados entre as classes, o que faz com que algoritmos de classificação tenham bons resultados para as classes com mais dados e resultados ruins para classes com menos dados. Para contornar este problema podem ser utilizados algoritmos de balanceamento dos dados, através da criação, remoção ou redistribuição dos dados e também abordagens de distribuição de modelos de classificação. Este trabalho realizou estudos e investigações com o objetivo de encontrar a melhor forma de classificar as mensagens de fóruns em gêneros. Como resultado principal é proposta uma abordagem que classifica a mensagem em um ou mais gêneros, com boas taxas de acerto comparadas com algoritmos de classificação da literatura. Com a vantagem de ser construída somente com algoritmos de classificação multi-classe, que já estão bem consolidados na literatura, e com o conjunto de mensagens de treino classificadas em um gênero.

Palavras-chave: Mineração de texto. Classificação multi-classe. Classificação multi-rótulo. Desbalanceamento de dados.

ABSTRACT

Virtual Learning Environments provide a set of tools to assist in teaching and learning processes. Among them, the discussion forums are widely used, given that they allow the exchange of text messages between students and tutors. The monitoring of a large number of messages is a task that requires a great amount of time and effort, making necessary the use of techniques for grouping messages with the same content. In this context, text classification can be used to group messages by genres, providing a new insight over the messages in order to monitor these discussion forums. To classify the message by a genre, multi-class classification algorithms are used and, to classify more than one genre, multi-label classification algorithms are used. A known issue of text classification, which occurred in this study, was the unbalance of data distribution between classes, which leads the classification algorithms to presenting good results for classes with more data and poor results for classes with less data. To solve this issue, data balancing algorithms can be used by means of creating, deleting or redistributing data, in addition to a few classification model distribution approaches. This work conducted studies and researches in order to find the best way to classify the messages from the forums into genres. As main result, we proposed an approach that classifies the message into one or more genres, with good success rates when compared with classification algorithms from literature. With the advantage of being built with only multi-class classification algorithms, which are already well established in the literature, and with a dataset of messages classified in one genre.

Keywords: Text mining. Multi-class classification. Multi-label classification. Unbalance Data.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

Nos dias atuais cada vez mais o processo de ensino e aprendizado vem sendo auxiliado por sistemas computacionais, que atuam como um complemento e, às vezes, até na substituição de um espaço físico. Estes sistemas são conhecidos como Ambientes Virtuais de Aprendizagem (AVA) ou *Learning Management Systems* (LMS) e oferecem uma grande variedade de canais e espaços para facilitar o trabalho de compartilhamento de informações e a comunicação entre os participantes de um curso, permitindo aos professores a distribuição de informações e materiais para os alunos, a preparação de trabalhos e testes, a participação de alunos e professores em fóruns de discussões, dentre outras ferramentas (ROMERO; VENTURA; GARCIA, 2008).

Atualmente, um destes sistemas comumente utilizado é o Moodle, acrônimo para *Modular Object-Oriented Developmental Learning Environment*, um sistema com licença *open source*, projetado para fornecer aos educadores e alunos um ambiente de aprendizagem personalizável e flexível e que possuí grande apoio de uma comunidade de desenvolvimento e evolução do sistema (RICE, 2006). Este ambiente utiliza um banco de dados relacional para guardar diversas e distintas informações sobre o uso do sistema tais como as atividades de leitura e escrita realizadas pelos alunos, o caminho de uso do sistema e as mensagens escritas em fóruns de discussão.

O fórum de discussões permite a comunicação entre os participantes, através da troca de mensagens de maneira síncrona e assíncrona. O acompanhamento destes fóruns de discussões é de extrema importância para o bom funcionamento das disciplinas e dos cursos especialmente em Ensino a Distância (EaD). Este acompanhamento deve ser feito através da intervenção de

tutores sendo uma tarefa que demanda uma grande quantidade de tempo e esforço. O tutor deve acompanhar e guiar uma discussão, e também responder rapidamente às mensagens que necessitem de maior atenção devido a seu conteúdo.

Acompanhar as discussões dos fóruns e encontrar mensagens que necessitam receber maior atenção ou uma rápida mediação por parte do tutor/avaliador é um trabalho complexo que demanda grande quantidade de tempo e esforço. Ao passo que ao encontrar a mensagem pode ser que já tenha passado a validade de uma discussão. No AVA Moodle as mensagens não estão agrupadas por seu conteúdo, mas de uma forma sequencial/temporal e assim, abordagens automáticas que diferenciam e agrupam as mensagens podem auxiliar os tutores e avaliadores.

A mineração de dados textuais estuda encontrar e agrupar dados, normalmente em um grande volume de dados, que possuam padrões de conteúdos semelhantes. Neste contexto a mineração de texto pode ser utilizada com o objetivo de agrupar mensagens com conteúdo semelhante. Este conteúdo pode ser o tipo do assunto da mensagem, ou seja, o gênero que representa a mensagem. Neste trabalho os gêneros considerados foram Anúncio, Dúvida, Esclarecimento, Interpretação e Outros. As mensagens podem possuir mais de um gênero, porém apenas um se sobressai.

Na classificação de texto são utilizados algoritmos de classificação juntamente com um conjunto de dados de treino para a construção de modelos de classificação. Estes modelos são mapeamentos das mensagens do conjunto de dados de treino com uma ou mais classe(s) que compõe este conjunto. Este conjunto de dados de treino são mensagens que já possuam um gênero, ou classe, associada a ela.

Nos fóruns de discussões, as mensagens podem estar distribuídas em quantidades com proporções muito discrepantes entre os gêneros, ou classes.

Quando se há esta discrepância em um conjunto de dados de treino, os algoritmos de mineração de dados podem se tornar tendenciosos para as classes que possuam maior quantidade de dados. Este problema é conhecido como desbalanceamento dos dados entre as classes (JAPKOWICZ; STEPHEN, 2002). Para contornar este problema, podem ser utilizados algoritmos para o balanceamento dos dados, através da criação de novos dados sintéticos para as classes que contem menos dados, a remoção de dados das classes que possuem maior quantidade de dados ou ainda uma redistribuição dos dados através da combinação de criação e remoção de dados. Além da criação de dados sintéticos, existem diferentes abordagens para tratar do problema do desbalanceamento dos dados, como a distribuição em cascata de modelos de classificação (LIN; HSIEH; CHUANG, 2009).

1.2 Problema e proposta deste trabalho

O problema deste trabalho foi encontrar a melhor abordagem de classificação de gêneros de mensagens em fóruns de discussão. Para encontrar a melhor abordagem foram utilizados diversos algoritmos de classificação multi-classe como *Support Vector Machine* (VAPNIK, 1995), *Sequential Minimal Optmization* (PLATT, 1998) e o Naïve Bayes (LEWIS, 1998), cada um induzido com três diferentes conjuntos de dados de treino, um desbalanceado um balanceado com o algoritmo *Synthetic Minority Oversampling Technique* (CHAWLA et al., 2002) e um balanceado com o algoritmo RESAMPLE.

As comparações foram feitas à procura da melhor composição entre algoritmo e tipo de conjunto de dados para contornar os problemas oriundos da distribuição desbalanceada dos dados, avaliados a partir das Acurácia do algoritmo e nas métricas Precisão, Revocação e Medida-F.

1.3 Objetivos gerais e específicos

O objetivo deste trabalho foi o desenvolvimento de uma abordagem de classificação de gêneros de mensagens de texto. Para tanto, foram coletadas mensagens de fóruns de ambientes virtuais de aprendizagem e estas foram classificadas manualmente por especialistas. Este conjunto de mensagens coletadas está desbalanceado, o que levou à necessidade de utilizar algoritmos de tratamento do desbalanceamento dos dados. Foram utilizados algoritmos de classificação conhecidos da literatura induzidos com diferentes conjuntos de dados de treino. Os modelos gerados foram distribuídos em uma abordagem em cascata baseada em Lin, Hsieh e Chuang, (2009)e em uma abordagem proposta neste trabalho.

Para alcançar o objetivo geral, buscou-se resolver os seguintes objetivos específicos:

a) selecionar mensagens de texto de ambiente virtual de aprendizagem, classificadas manualmente por especialistas;
b) compreender os algoritmos de classificação multi-classe com SVM, SMO e Naïve Bayes;
c) compreender o problema do desbalanceamento dos dados e os algoritmos SMOTE e RESAMPLE;
d) compreender os algoritmos de classificação multi-rótulo MLkNN e RAkEL;
e) avaliar a abordagem da proposta de classificação multi-classe e multi-rótulo;
f) desenvolver um módulo de classificação de gêneros de mensagens que permita sua integração em um ambiente real;

1.4 Tipo de pesquisa

A pesquisa deste trabalho pode ser classificada em relação ao seu objetivo como exploratória; quanto aos experimentos ela é experimental, por sua abordagem é quantitativa e executada em laboratório. Ela é considerada uma pesquisa aplicada quanto à sua natureza, pois utiliza conhecimento básico para gerar conhecimentos com finalidade de aplicação (WAINER, 2007).

A pesquisa também é classificada como *design science*, pois desenvolve novos conhecimentos e soluções para o problema em estudo (AKEN, 2005).

1.5 Resultados alcançados e as principais contribuições

A principal contribuição deste trabalho foi a construção de uma abordagem de classificação automática e eficiente de mensagens capaz de classificar, em um único processo, uma mensagem em um ou mais gênero (multi-classe / multi-rótulo).

Os resultados obtidos pela abordagem proposta foram compatíveis com os algoritmos de classificação multi-classe e multi-rótulo, com a vantagem de classificar uma mensagem em um único processo utilizando somente algoritmos de classificação multi-classe e com o conjunto de dados de treino rotulados em somente uma única classe.

1.6 Estrutura do trabalho

O presente trabalho está dividido em 5 capítulos. O capítulo 1 introduziu e contextualizou o problema, a proposta deste trabalho, os objetivos gerais e específicos, o tipo de pesquisa realizada, os resultados alcançados e as principais

contribuições. O Capítulo 2 apresenta os assuntos envolvidos e as referências para a realização deste trabalho. O capítulo 3 apresenta a descrição do problema trabalhado, a construção da base de dados e as métricas de avaliação de classificação. O Capítulo 4 descreve a construção de modelos de classificação multi-classe e binários, analisando o problema do desbalanceamento dos dados. O Capítulo 5 demonstra abordagens de distribuição de modelos de classificação juntamente com a comparação entre as abordagens. O Capítulo 6 apresenta a conclusão obtida e propostas de trabalhos futuros.

2 REFERENCIAL TEÓRICO

Nesse capítulo são apresentados os conceitos para entendimento deste trabalho sendo Ambientes Virtuais de Aprendizagem (AVA), a mineração de dados, a mineração de texto aplicada ao AVA, classificação de texto, o problema de desbalanceamento dos dados e soluções propostas, algoritmos de classificação multi-classe, algoritmos de classificação multi-rótulo, as bibliotecas Weka e Meka. Também são apresentados os trabalhos relacionados e a contribuição deste trabalho.

2.1 Ambientes Virtuais de Aprendizagem

Atualmente com o uso da tecnologia e da Internet, a educação está passando por mudanças significativas contemplando novas maneiras de ensino e aprendizado. O uso de sistemas educacionais na Web, conhecidos como *e-learning*, tem crescido exponencialmente nos últimos anos estimulado pelo fato de estudantes e professores não necessitarem estar vinculados a um lugar físico específico (BRUSILOVSKY; PEYLO, 2003). Estes sistemas oferecem uma grande variedade de canais e espaços de trabalho, facilitando a troca de informação e a comunicação entre os participantes em um curso, permitindo a educadores distribuírem informações a estudantes, produzir materiais, preparar testes e atribuições, elaborar discussões, gerenciar aulas a distância e também um espaço para aprendizado participativo e colaborativo como fóruns, chats, serviços de notícia, etc (ROMERO; VENTURA; GARCÍA, 2008).

Estes sistemas são conhecidos como *Learning Management Systems* - LMS ou Ambiente Virtual de Aprendizagem - AVA. Alguns exemplos de

sistemas comerciais são BlackBoard[1], TopClass[2] e WebCT[3]; e alguns gratuitos são o Moodle[4], Claroline[5]e o Ilias[6]. Estes sistemas se diferenciam em relação à linguagem de programação em que foram desenvolvidos, nos recursos e o tipo de conteúdo disponível, entretanto possuem o mesmo objetivo que é oferecer uma ferramenta de administração e gerenciamento de um ambiente de ensino.

Um ambiente bem difundido e que possui licença *open-source* é o Moodle, acrônimo para *Modular Object-Oriented Developmental Learning Environment* que foi desenvolvido sobre a filosofia de aprendizado interativo, que mostra que as pessoas aprendem melhor quando interagem com o material, através da construção de novos materiais para colegas e a utilização de materiais criados por colegas. A diferença entre uma sala de aula tradicional e esta filosofia pode ser considerada a mesma diferença entre uma palestra e uma discussão (RICE, 2006); na palestra tem-se uma figura central como orador assim como o professor para a sala de aula e em uma discussão há interação e participação de todos os envolvidos. O ambiente Moodle é composto por módulos de atividades e recursos como: criação de materiais estáticos como páginas web, links para outros conteúdos, envio de arquivos de texto ou imagens, criação de materiais interativos e avaliações em que se pode pontuar; e também a criação de atividades onde estudantes e professores interagem entre si como fóruns, chats e mensagens particulares (ROMERO; VENTURA; GARCÍA, 2008).

Estes ambientes acumulam uma grande quantidade de informações, como as mensagens escritas em fóruns e chats, a realização de atividades avaliativas, os materiais mais utilizados, e também como é a interação do

[1] Blackboard (2014)
[2] WBT Systems (2014)
[3] Blackboard (2014a)
[4] Moodle (2014)
[5] Consortion Claroline (2014)
[6] Ilias (2014)

usuário com o sistema através dos links navegados e o tempo gasto utilizando o sistema. Com o uso de técnicas de mineração de dados para a extração de padrões, estas informações possuem um grande potencial para explicar o comportamento de alunos usuários do sistema e podem fornecer informações para auxiliar os avaliadores nas tomadas de decisões (MOSTOW et al., 2005).

2.2 Mineração de dados

Mineração de Dados, ou *Data Mining,* é um processo de exploração e análise, geralmente em um grande volume de dados através do uso de algoritmos, para descoberta de padrões implícitos ou explícitos e desenvolvimento de modelos significativos que representem estes padrões (MAIMON; ROKACH, 2005). Os algoritmos de mineração de dados podem ser vistos como composições de algumas técnicas e princípios básicos, sendo geralmente divididos em três componentes: (1) um modelo, composto pela função (classificação, clusterização, árvores de decisões, etc) e a forma de representação deste modelo (por exemplo, função linear); (2) um critério de preferência, indicando o porquê da escolha de um modelo sobre outro ou a escolha do conjunto de parâmetros de entrada; (3) um algoritmo, para encontrar a melhor composição de modelos e parâmetros, dado um conjunto de dados, um modelo e um critério de preferência (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

A mineração de dados é uma etapa do processo descoberta de conhecimento em banco de dados, ou *Knowledge Discovery in Databases* (KDD). KDD é um processo não trivial de identificar, validar, modelar, encontrar novos, potencialmente úteis e compreensíveis padrões nos dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). De acordo com Romero e Ventura (2007), consiste em etapas que envolvem pré-processamento, técnicas

de mineração de dados e um pós-processamento. A Figura 1 ilustra este processo de KDD adaptada de (HAN; KAMBER, 2005).

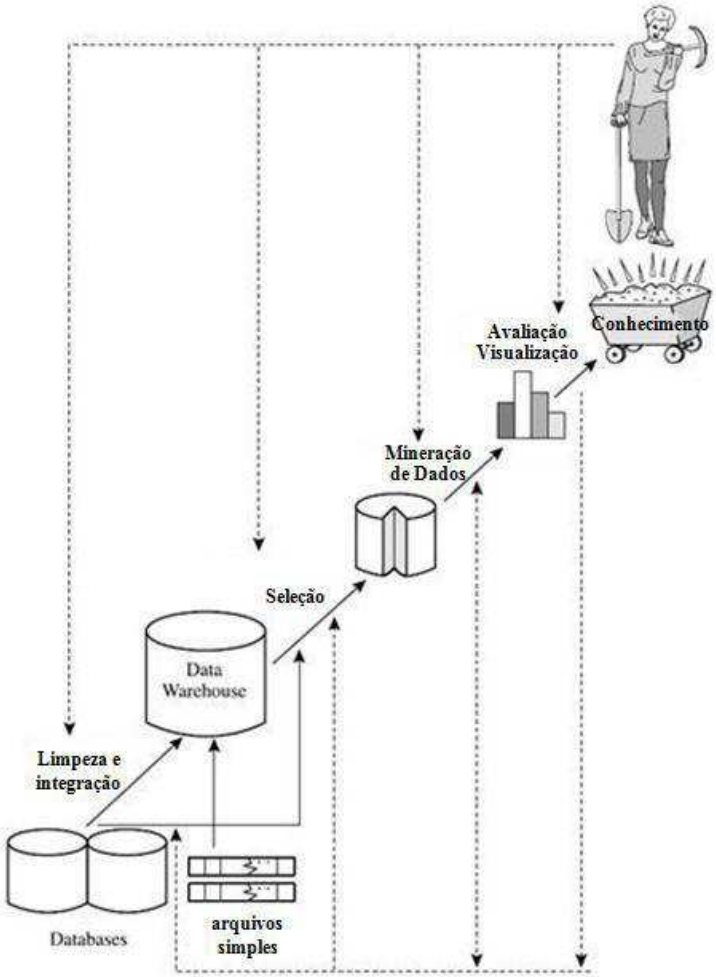

Figura 1 Processo de KDD extraído de Han e Kamber (2005)

Maimon e Rokach (2005) decompõe o processo de extração de conhecimento em etapas que iniciam com o entendimento do domínio do problema, a seleção e criação do conjunto de dados, etapas de pré-processamento, aplicações de métodos e algoritmos de mineração de dados e

finaliza com a validação e o uso do conhecimento extraído. Nesta abordagem, o processo de KDD é decomposto em nove passos (MAIMON; ROKACH, 2005):

a) desenvolve-se um entendimento sobre o domínio de uso da aplicação;
b) seleciona-se conjunto de dados;
c) pré-processamento e limpeza nos dados como remoção de ruídos;
d) transformação adequada dos dados para o uso por algoritmos;
e) escolhe-se o método mais apropriado para Mineração dos Dados;
f) escolha-se o algoritmo de Mineração de Dados;
g) utiliza-se o algoritmo no conjunto de dados;
h) validação e avaliação dos resultados obtidos;
i) uso do conhecimento descoberto.

A mineração de dados é uma área multidisciplinar para a qual muitos paradigmas de computação convergem, tais como a construção de árvores de decisão, regras de indução, redes neurais artificiais, etc. Algumas de suas tarefas e técnicas utilizadas são visualização, clusterização, classificação, regras de associação, mineração de textos, etc.(ROMERO; VENTURA; GARCÍA, 2008).

Um dos métodos da mineração de dados é o aprendizado supervisionado, em que um modelo é construído a partir de uma generalização de um conjunto de dados de treino de tamanho suficiente. Este modelo é então aplicado a dados não utilizados na construção do modelo. O aprendizado supervisionado pode gerar dois tipos de modelos: modelos de classificação ou modelos de regressão. Modelos de classificação mapeiam o conjunto de dados de entrada em classes pré-determinadas. Modelos de regressão mapeiam o conjunto de dados de entrada em valores do domínio do modelo (MAIMON; ROKACH, 2005).

Srivastava et al. (2000) mostrou o grande interesse do comércio eletrônico na utilização de técnicas de mineração de dados como mineração de uso da Web (SRIVASTAVA et al., 2000). De acordo com Romero et al. (2009), há um crescente interesse em aplicar mineração de dados em sistemas educacionais, fazendo este tema uma importante área de pesquisa (ROMERO et al., 2009; ROMERO; VENTURA, 2007; ROMERO; VENTURA; GARCÍA, 2008).

Uma parte destes dados é composta por mensagens de texto. Uma variação do campo da mineração de dados é a mineração de textos que procura encontrar padrões interessantes em grandes quantidades de textos (GUPTA; LEHAL, 2009).

2.3 Mineração de dados em AVA

Nas salas de aulas tradicionais a interação entre alunos e professores é feita de maneira mais próxima do que em salas de aulas virtuais, pois há a presença física dos envolvidos. O acompanhamento do desenvolvimento de um aluno pode ser facilmente observado e analisado quando se há esta presença, pois o professor pode observar as expressões dos alunos e adaptar rapidamente o processo de ensino. Entretanto, nos ambientes virtuais de aprendizagem não há a presença física constante sendo a interação entre alunos e professores feita por intermédio de um sistema de informação, o que faz necessário encontrar outras maneiras de acompanhar o desenvolvimento de um aluno. Uma maneira de se analisar o desenvolvimento do aluno é através da interação deste com o ambiente virtual de aprendizagem como o comportamento de acesso e navegação pelo ambiente, materiais mais utilizados e também as interações entre os alunos e professores nos fóruns de discussões (MAZZA; MILANI, 2005).

Métodos de mineração de dados e reconhecimento de padrões de navegação são utilizados em comércio eletrônico, e estas técnicas têm atraído interesse para serem aplicadas em AVA. Embora os métodos de extração do conhecimento sejam praticamente os mesmos, seus objetivos finais são diferentes. No comércio eletrônico, o objetivo é guiar o cliente a comprar produtos enquanto no AVA o objetivo é guiar o estudante ao aprendizado (ROMERO; VENTURA; BRA, 2005). O processo de mineração de dados pode ser utilizado para construir modelos para identificação de padrões e tendências do comportamento dos estudantes para auxiliar professores na melhoraria do aprendizado do estudante e na manutenção dos cursos (ROMERO; VENTURA; GARCIA, 2008).

A aplicação de mineração de dados em AVA é um ciclo iterativo, em que o conhecimento extraído deve retornar ao sistema para orientar, facilitar e melhorar o processo de aprendizagem como um todo, não apenas transformando os dados em conhecimento, mas auxiliando na tomada de decisões (ROMERO; VENTURA; GARCIA, 2008). A Figura 2 ilustra os quatro passos do processo de mineração de dados em AVA: a coleta dos dados, o pré-processamento dos dados, a aplicação de algoritmos de mineração de dados e a interpretação, avaliação e a implantação dos resultados.

Figura 2 Processo de mineração de dados no AVA Moodle
Fonte: Extraído de Romero, Ventura e Garcia (2008).

A etapa de coleta consiste em recuperar dados que podem conter informações pertinentes a serem mineradas do AVA. A etapa de pré-processamento consiste na limpeza e transformação dos dados para um formato apropriado para os algoritmos de mineração de dados. A etapa de aplicação de algoritmos de mineração de dados resulta na criação de modelos significativos que mapeiam o conhecimento descoberto de interesse dos avaliadores. Por fim, a etapa de interpretação, avaliação e implantação destes modelos para auxílio nas tomadas de decisões e melhorias no processo de ensino.

2.4 Mineração de textos

Mineração de Textos (*Text Mining*) trata sobre o processamento automatizado de dados textuais em um processo de exploração e análise, geralmente em um grande volume de dados, para descoberta de padrões implícitos ou explícitos e o desenvolvimento de modelos significativos que descrevem estes padrões. É uma área interdisciplinar envolvendo diversos

paradigmas de computação tais como Recuperação de Informação, Processamento de Linguagem Natural e Aprendizado de Máquina.

A Recuperação de Informação ou *Information Retrieval* (IR) procura recuperar os dados relevantes à determinada consulta através do uso de técnicas e métodos de indexação, de forma eficiente tanto em tempo quanto em espaço de armazenamento, sendo uma área que ganhou destaque com o grande crescimento da *World Wide Web*(www). O Processamento de Linguagem Natural ou *Natural Language Processing* (NLP) objetiva em fazer com que os computadores possam interpretar as linguagens dos seres humanos através do uso de técnicas que vão desde a simples manipulação das palavras até a análise semântica das formas gramaticais de cada língua. Para este fim, são aplicadas diversas formas de Aprendizado de Máquina ou *Machine Learning* (ML), sendo uma área da inteligência artificial dedicada ao desenvolvimento de técnicas de programação para que computadores possam realizar automaticamente análises sobre um conjunto de dados (HOTHO; NÜRNBERGER; PAAB, 2005).

A mineração de texto pode trabalhar com conjuntos de dados não estruturados ou semi estruturados tais como documentos de texto, arquivos HTML, e-mails, etc., e geralmente envolve um processo de estruturação do conjunto de dados de texto, juntamente com adição ou remoção de regras gramaticais e características linguísticas que são diferentes em cada idioma (ROMERO; VENTURA; GARCIA, 2008).

Mineração de Textos é uma parte do processo conhecido como *Knowledge Discovery in Textual Database* (KDT), como visto na Figura 3, podendo este processo ser dividido em etapas sequenciais iniciando pela coleta dos dados, seu pré-processamento, sua indexação, a mineração e a análise e interpretação dos resultados (ARANHA; VELLASCO, 2007).

Figura 3 Processo de KDT
Fonte: Extraído de Aranha e Vellasco (2007).

2.4.1 Etapa de coleta

A coleta é a busca por dados que formarão a base a ser analisada. Estes dados podem estar em diferentes formatos como em arquivos de texto ou bancos de dados. Para cada formato existem diferentes técnicas para sua coleta e extração. Um arquivo de texto pode ser lido linha a linha; enquanto que em um banco de dados, a coleta é feita através de consultas.

2.4.2 Etapa de pré-processamento

Nesta etapa são aplicadas técnicas para transformações dos dados, consistindo em identificar e tratar dados corrompidos, atributos irrelevantes e valores desconhecidos. Em dados textuais, algumas técnicas são a remoção de pontuações e caracteres especiais, a tokenização, remoção de *stop words*, a *lematization*, o *stemming*, um processamento linguístico, etc.

Tokenização é a quebra do texto em palavras conhecidas como *tokens*. O texto é quebrado a partir de um delimitador como espaços em branco ou vírgula.

As palavras resultantes da tokenização de um conjunto de dados textuais são o dicionário deste conjunto. Este dicionário pode conter um grande número de palavras; podendo, se necessário, utilizar outras técnicas para diminuir o tamanho deste dicionário, como a remoção de *stop words*, lematização e *stemming* (HOTHO; NÜRNBERGER; PAAB, 2005).

Stop Words são palavras que contém pouca ou nenhuma informação semântica para diferenciar os textos como artigos, preposições e conjunções. *Lematization* modifica as palavras através do mapeamento dos verbos para sua forma no infinitivo e dos substantivos para sua forma no singular. *Stemming* é a transformação morfológica das palavras para sua forma básica, como por exemplo, a retirada de prefixos, sufixos e gerúndio.

Na mineração de textos, o processamento linguístico fornece informações adicionais sobre o contexto/vocabulário estudado das palavras, através de técnicas como *Part of Speach* (POS) e *Word Sense Desambiguation* (WSD). *Part of Speach* determina se uma palavra é um substantivo, adjetivo ou verbo; e *Word Sense Desambiguation* determina o melhor significado de palavras ambíguas de acordo com o contexto em análise.

2.4.3 Etapa de indexação

Esta etapa é a fase de criação de estruturas que facilitem o armazenamento e a recuperação dos dados de maneira eficiente. Um dos modelos existentes é o Modelo de Espaço Vetorial (*Vector Space Model*). Este modelo faz uma representação dos dados em vetores, sendo que cada posição do vetor representa uma palavra ou conjunto de palavras. Para cada palavra é calculado um peso que é o grau de importância da palavra no conjunto de dados. Este peso é calculado pelas medidas *Term Frequency* (TF) e *Inverse Document Frequency* (IDF). A medida TF é a quantidade de vezes que a palavra aparece

em um texto (por exemplo em uma frase) e a medida IDF mede a raridade ou a alta frequência do termo na coleção de textos (todas as frases do conjunto de dados). A Fórmula 1 representa o *IDFt*, sendo *t* o termo em questão, *N* a quantidade de textos, *DFt* sendo a quantidade de textos que contém o termo *t*.

$$IDFt = \log \frac{N}{DFt} \quad (1)$$

A junção das medidas TF e IDF é o peso do termo *t* no documento *d*, como representado na Fórmula 2. Altos pesos são associados a termos usados frequentemente em documentos relevantes, mas são raros no resto dos documentos (HOTHO; NÜRNBERGER; PAAB, 2005).

$$TF - IDFt, d = TFt, d * IDFt \quad (2)$$

2.4.4 Etapa de mineração

Nesta etapa são aplicados algoritmos e técnicas para extrair informações sobre o conjunto de dados já estruturados. Maimon e Rokach (2005) dividem esta etapa em três fases: (1) escolha do método apropriado de mineração de dados como por exemplo, a classificação ou clusterização; (2) escolha de um algoritmo de acordo com o método escolhido, como por exemplo *Support Vector Machine* ou *Naïve Bayes* para classificação; (3) aplicação do algoritmo no conjunto de dados e análise do resultado.

A classificação é um método supervisionado que procura rotular um documento com uma classe ou mais classes, dentre um conjunto de classes previamente conhecidas. Por exemplo, dada uma sentença, os algoritmos são capazes de classificar seu conteúdo como sendo positivo ou negativo, ou

identificar o(s) gênero(s) da sentença, dentre um conjunto de gêneros previamente conhecidos. Este método é dividido em duas etapas: a etapa de treino, em que a um conjunto de dados são aplicadas técnicas e algoritmos para a geração de modelos de classificação, e a etapa de teste que utiliza o modelo de classificação gerado em dados cuja classe é desconhecida. A performance desta classificação é calculada por métricas como Precisão (*Precision*), Revocação (*Recall*) e Medida-F (*F-measure*)(HOTHO; NÜMBERGER; PAAB, 2005).

2.4.5 Etapa de análise

Nesta etapa de análise é feita a avaliação e interpretação dos resultados do processo de mineração de texto, envolvendo o julgamento dos resultados obtidos por especialistas. Algumas medidas supracitadas podem ser usadas para mensurações de qualidade na classificação. Este é um processo iterativo sendo que os resultados obtidos podem ser incorporados ao sistema e o processo de mineração de dados recomeça utilizando este conhecimento extraído, a fim de melhorar os resultados das métricas de avaliação.

2.5 Algoritmos de classificação multi-classe

Para a descrição da formulação de funcionamento dos algoritmos de classificação multi-classe e multi-rótulo é necessário introduzir algumas simbologias que serão utilizadas, que se encontram na Tabela 1.

Tabela 1 Símbolos e notações utilizadas na descrição dos algoritmos de classificação

Símbolo	Significado	Utilizado no Trabalho
C	Conjunto de classes	C = {A, D, E, I, O}
|C|	Número de classes	|C| = 5
CLP	Conjunto de classes resultantes do método *Label Powerset* (LP)	Tabela 3
|CLP|	Número de classes resultantes do método LP	|CLP| = 31

Classificação multi-classe, também conhecida como *single label* ou *multi-class classification*, é uma atividade de aprendizado de máquina em que um dado é associado a uma única classe de um conjunto finito de classes (READ, 2008). Dentre os algoritmos de classificação conhecidos na literatura vamos destacar três que foram utilizados nos experimentos, sendo o *Support Vector Machines* (SVM) proposto por Vapnik (1995), *Sequential Minimal Optimization* (SMO) proposto por Platt (1998) e o *Naïve Bayes* (NB) proposto por Lewis(1998).

Support Vector Machines são baseados no princípio *Structural Risk Minimization* oriundo da teoria do aprendizado computacional (VAPNIK, 1995). A idéia deste princípio é encontrar uma hipótese que garante o menor erro real (*true error*). Este erro real é a probabilidade desta hipótese estar errada quando aplicada em dados desconhecidos.

O SVM procura encontrar um hiperplano, dentre todas as superfícies de separação que podem ser desenhadas para a separação dos dados em duas classes, aquela que possua a maior margem de separação entre os dados. Um exemplo pode ser visto na Figura 4, onde as cruzes indicam dados positivos e os círculos dados negativos, enquanto as linhas representam as superfícies de decisão. Os dados nos quadrados representam os *support vectors*. A linha em

negrito representa a melhor superfície de separação entre os dados destas duas classes (SEBASTINI, 2002).

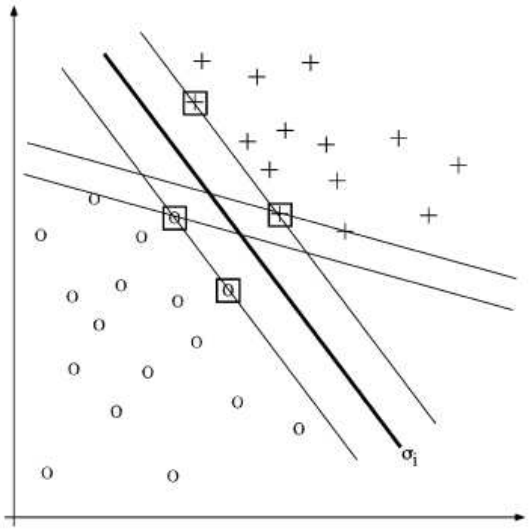

Figura 4 Hiperplanos de separação a melhor está em negrito e os *support vector* nos quadrados

Uma propriedade importante do SVM é que seu aprendizado é quase independente da dimensionalidade do espaço de características, ou *feature spaces*. Ele raramente necessita de uma seleção de características, visto que ele seleciona alguns pontos, conhecido como *support vectors*, que são utilizados para a classificação. Isto permite uma boa generalização mesmo na presença de um grande número de características e faz com que o SVM seja especialmente adequado para a classificação de texto (JOACHIMS, 1998). O SVM vem sendo aplicado para a classificação de texto em muitos trabalhos como em Joachims (1998), Dumais et al. (1998), Sebastini (2002) e Sebastini (2006).

As características que permitiram que o algoritmo SVM fosse bem aceito na classificação de texto são: (a) alta dimensionalidade do espaço de

características, (b) poucas características irrelevantes, (c) vetores das instâncias esparsos (JOACHIMS, 1998).

O problema de otimização tratado pelo SVM requer a solução de um problema na programação quadrática muito grande, ou *quadratic programming* (QP). Para diminuir este grande problema, baseado em métodos de decomposição de problemas, foi proposto o algoritmo *Sequential Minimal Optimization* (SMO) (PLATT, 1998). De acordo com Platt (1998), o SMO quebra este grande problema QP em uma série de problemas QP menores resolvidos separadamente. Isto faz com que a quantidade de memória que é necessária para o treino do SMO seja linear, o que permite que seja utilizado para uma quantidade ainda maior de dados. O algoritmo SMO é resolvido entre o tempo linear e o quadrático, enquanto que o algoritmo SVM padrão é resolvido entre o tempo linear e o cúbico (PLATT, 1998).

O método SVM pode ser utilizado para a resolução de problemas de classificação (VAPNIK, 1995) e foi estendido para lidar com problemas de regressão (SMOLA, 1996). O método SVM de classificação foi melhorado pelo algoritmo SMO (PLATT, 1998), e que por sua vez foi estendido e melhorado em Keerthi et al. (2001). O método SVM de regressão foi alvo de trabalho em Smola e Scholkopf (1998), e é uma extensão do algoritmo SMO para o método SVM de regressão, e posteriormente este método foi melhorado no trabalho de Keerthi et al.(2002).

O modelo de classificação Naïve Bayes assume que os atributos que formam as instâncias são independentes entre si para se determinar uma classe. Na classificação de texto, em que os atributos são as palavras, esse modelo "naive" possui bons resultados e vem sendo utilizado na classificação de texto (JOACHIMS, 1997; MCCALLUM; NIGAM, 1998; RENNIE, 2001).

Para explicar o funcionamento do algoritmo Naïve Bayes, considere a seguinte terminologia que foi baseada em Youn e Jeong (2009). Considere um

conjunto de instâncias de treino denominado M = {m_1, m_2, ..., m_i}, sendo *i* o tamanho do conjunto de treino, e uma instância particular a ser observada deste conjunto é denominada por *m*. Cada instância *m* é formada por um vetor de atributos, como m = {a_1, a_2, ..., a_j}, sendo *j* a quantidade de atributos da instância. O rótulo das classes é denotado pelo conjunto C = {c_1, c_2, ..., c_k} e uma classe particular a ser observada é denominada por *c*. Com o uso desta terminologia, a classificação de texto é designada como uma classe *c* do conjunto C para uma instância *m,* e podemos descrever o Teorema de Bayes adaptado de Youn e Jeong (2009):

$$P(c \mid m) = \frac{P(m \mid c) . P(c)}{P(m)} \quad (3)$$

P(*m* | *c*) é a probabilidade da instância *m* ser escolhida aleatoriamente do conjunto de instâncias que pertençam à classe *c*. Como exemplo, considere um problema de classificação com duas classes, C = {A, NA} portanto *c* = A ou *c* = NA. Utilizando o Teorema de Bayes para este problema de classificação, dada uma instância de teste *m* cuja classe é desconhecida, calcula-se:

$$P(A \mid m) = \frac{P(m \mid A) . P(A)}{P(m)}$$

$$P(NA \mid m) = \frac{P(m \mid NA) . P(NA)}{P(m)} \quad (4)$$

Como resultado, a instância *m* é classificada em A caso P(A | *m*) > P(NA | *m*) e classificada em NA, caso contrário. A probabilidade P(*c*| *m*) é chamada de probabilidade *a posteriori* da classe *c* e P(*m* | *c*) é chamada de verossimilhança, ou *likelihood*, da classe *c* para determinada instância *m*. O classificador Naïve

Bayes procura encontrar a classe que maximize a probabilidade *a posteriori* (MAP) para uma dada instância de teste cuja classe é desconhecida. Através do Teorema de Bayes, para calcular a probabilidade *a posteriori* de uma classe *c*, dada uma instância *m* temos:

$$P(c \mid m) = P(c \mid (a_1, a_2, \ldots, a_j)) = \frac{P(a_1,a_2,\ldots,a_j) \mid c).P(c)}{P(a_1,a_2,\ldots,a_j)} \quad (5)$$

Como iremos comparar as probabilidades *a posteriori* para diferentes classes (*c*'s), e o denominador é comum para estas diferentes classes (*c*'s), pode-se ignorar o denominador e calcular somente o numerador na Fórmula 3. No numerador, P(*c*), a probabilidade *a priori* pode ser calculada contando o número de instâncias cuja classe seja *c*, dividido pelo total de instâncias do conjunto de treino.

O problema de se usar a Fórmula 3 é que, em algumas situações tal como o número de atributos sendo poucos e distintos e o número de classes sendo grande, para calcular a $P((a_1, a_2, ..., a_j) \mid c)$ seriam necessárias muitas instâncias, combinações de atributos $(a_1, a_2, ..., a_j)$, para que houvesse mapeamento entre todos os atributos e todas as classes. Ainda há combinações de atributos raras e específicas para cada classe. Além disto, algumas combinações de atributos poderiam aparecer nas instâncias de teste (classe desconhecida) e estas combinações não existiriam no conjunto de instâncias de treino. Para tratar disto, a abordagem Naïve Bayes assume a independência entre os atributos para calcular a probabilidade de uma instância pertencer a uma classe. Com esta abordagem "naïve" (ingênua), o numerador da Fórmula 3 pode ser escrito como:

$$P((a_1, a_2, \ldots, a_j) \mid c) \,.\, P(c) = \prod_{i=i}^{j} P(m \mid c) \,.\, P(c) \quad (6)$$

Sumarizando as fórmulas, a abordagem Naïve Bayes classifica uma instância *m* como *c*, onde *c*:

$$c = argmax_c \prod_{i=1}^{j} \mathrm{P}(m \mid c) \,.\, \mathrm{P}(c) \qquad (7)$$

2.6 Algoritmos de classificação multi-rótulo

Classificação multi-rótulo é uma extensão do problema de classificação multi-classe, sendo que os dados são associados a uma ou mais classes. O aprendizado multi-rótulo originou da investigação da classificação de texto, sendo que um documento pode pertencer a vários tópicos simultaneamente (READ, 2008).

Classificação multi-rótulo é uma forma de aprendizado de máquina supervisionado em que um algoritmo classificador é treinado a partir de um conjunto de dados classificados em uma ou mais classes. Um problema do mundo real é a classificação de uma mensagem em pertencente a um ou mais gêneros. Os métodos de classificação multi-rótulo podem ser agrupados em dois métodos propostos em (TSOUMAKAS; KATAKIS, 2007).

a) Método de Transformação de Problema
b) Método de Adaptação de Algoritmos

O Método de Transformação do Problema (*Problem Transformation Method)* consiste em transformar os dados do conjunto de treino multi-rótulo em representações de dados utilizadas na classificação multi-classe e assim utilizar esta representação para treinar qualquer algoritmo de classificação multi-classe.

O resultado da classificação multi-classe é transformado de volta à representação de dados multi-rótulo de maneira reversa (READ, 2008). Alguns destes métodos são *Binary Relevance* (BR) (READ, 2008); *Label Powerset* (LP) (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010); *RAndom k-LabEL Sets* (RAkEL) (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011).

O Método de Adaptação de Algoritmos (*Algorithm Adaptation Method)* consiste em estender e adaptar algoritmos de classificação multi-classe para lidar diretamente com dados multi-rótulo. Dentre os métodos existentes temos o *Multi-Label* k *Nearest Neighbor* (MLkNN) (ZHANG; ZHOU, 2007), uma extensão do algoritmo k-*NearestNeighbor* (kNN); o AdaBoost.MH uma extensão do algoritmo AdaBoost para lidar com dados multi-rótulo (SCHAPIRE; SINGERY, 2000).

O classificador multi-rótulo, que compõe a abordagem de classificação multi-rótulo desenvolvida, é baseado no Método de Transformação do Problema *Binary Relevance*. A fim de comparar os resultados da abordagem desenvolvida foram utilizados dois métodos, o Método de Transformação de Problemas através do algoritmo RAKEL e o Método de Algoritmo Adaptado através do algoritmo MLKNN. Os algoritmos destes dois métodos são provenientes da ferramenta Meka.

Binary Relevance (BR) é uma das abordagens mais populares como Método de Transformação do Problema, que cria um modelo binário de classificação para cada classe do conjunto de dados de treino, atribuindo o valor positivo se o dado pertence à classe e negativo, caso contrário. Cada algoritmo classificador binário é treinado com um conjunto de dados composto da maneira "um-vs-resto", em que o conjunto de dados de cada classificador binário contém dois tipos de dados, dados que pertencem a sua classe e à união dos outros dados que não pertencem a sua classe. Embora a complexidade computacional deste

método seja linear, ele é criticado por assumir a independência entre as classes (READ, 2008; SOROWER, 2010).

Para a classificação de uma nova instância, as classes resultantes da classificação da abordagem *BR* é a união das classes classificadas positivamente por cada um dos |C| modelos classificadores binários (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010).

Os conjuntos de dados de treino que foram utilizados para criação de modelos binários estão representados na Tabela 2, em que os gêneros estão representados por suas siglas e a união dos outros gêneros representados pela letra N e a sigla do gênero.

Tabela 2 Conjuntos dos dados de treino utilizados no método *Binary Relevance*

Classe	**A**	**NA**	**E**	**NE**	**I**	**NI**	**O**	**NO**
Quantidade	576	5023	407	5192	2046	3553	2100	3499

Label Powerset (LP) é um simples e efetivo Método de Transformação do Problema que considera cada combinação de classes existentes no conjunto de dados de treino como uma nova classe. O método LP transforma o conjunto de dados de um problema de classificação multi-rótulo em multi-classe. Considere o conjunto C = {A, D, E, I, O}, o método LP aplicado a este conjunto resultaria nas combinações da Tabela 3, onde *k* representa o número de classes do conjunto. Este novo conjunto *label powerset* (CLP) da classe C contém 31 novas classes de um problema de classificação multi-classe, e este conjunto é utilizado por algoritmos de classificação multi-classe. Uma nova instância é classificada em uma destas novas classes, que representa um conjunto de uma ou mais classes (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011).

Tabela 3 Classes geradas pelo método LP no conjunto de classes C

K = 1	K = 2		K = 3		K = 4	K = 5
A	A,D	D,I	A,D,E	A,I,O	A,D,E,I	A,D,E,I,O
D	A,E	D,O	A,D,I	D,E,I	A,D,E,O	
E	A,I	E,I	A,D,O	D,E,O	A,D,I,O	
I	A,O	E,O	A,E,I	D,I,O	A,E,I,O	
O	D,E	I,O	A,E,O	E,I,O	D,E,I,O	

Este método tem a vantagem de levar em consideração as relações e dependências entre as classes, mas devido a sua complexidade computacional, delimitada superiormente por min(i, $2^{|C|}$) sendo *i* a quantidade de instâncias de treino e |*C*| o número de classes de treino antes da aplicação do método de transformação LP, e sua performance da classificação em domínios com grande quantidade de classes ou dados de treino levaram à criação de uma nova proposta, o algoritmo RAndom *k*-labEL sets - RAkEL (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011).

O objetivo do algoritmo RAkEL é diminuir o grande número de modelos necessários para a classificação das novas classes oriundas da aplicação do método LP, e escolher alguns modelos para comporem um conjunto de classificadores LP chamado *labelsets*. O algoritmo funciona selecionando, a cada uma das *m* interações, randomicamente e sem repetição (*without replacement*), um *k-labelset* e para cada um destes *m k-labelsets* é treinado um classificador LP. O número de interações *m* que varia entre 1 e |CLP| e o número de *labelsets k* que varia entre 2 e |C| - 1 são parâmetros especificados pelo usuário. Note que para $k = 1$ e $m = |CLP|$ é gerado um conjunto de classificadores binários (assim como o método *Binary Relevance*), e para $k = |CLP|$ e consequentemente $m = 1$, é gerado um único classificador multi-classe (assim como um classificador multi-classe LP de todas classes) (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011).

O algoritmo RAkEL possui a vantagem de levar em consideração as relações existentes entre as classes enquanto evita problemas relacionados ao método LP. Para a classificação multi-rótulo de uma nova instância, cada modelo LP do conjunto provê uma decisão binária classificando a instância como pertencente ou não pertencente às classes daquele modelo LP. A partir das classificações de todos modelos do conjunto,um ranking de cada classe é formado e aquelas classes que obtiverem uma votação maior que um *threshold t* são as classes da instância. Por conta desta votação, o algoritmo pode prever um *labelset* que não estava presente no conjunto de modelos LP e também possui a chance de corrigir eventuais classificações errôneas de um *labelset* que compõe o modelo. A Tabela 4 mostra um exemplo de processo do método RAkEL com os parâmetros de valor $k = 3$ e $m = 6$.

Tabela 4 Exemplo do processo do método RAkEL com k = 3 e m = 6

Labelsets	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$
$\{C_1, C_2, C_5\}$	1	0	-	-	1
$\{C_2, C_3, C_4\}$	-	0	0	1	1
$\{C_3, C_4, C_5\}$	-	-	1	1	1
$\{C_1, C_2, C_4\}$	1	1	-	0	-
$\{C_2, C_4, C_5\}$	-	0	-	0	0
$\{C_1, C_4, C_5\}$	0	-	-	0	1
Média das votações	2/3	1/4	1/2	2/5	4/5
Resultado Final	1	0	1	0	1

A grande desvantagem desta abordagem é sua complexidade algorítmica sendo de $O(2^k)$ além de necessitar de uma combinação de parâmetros *m* x *k* x *t* para obter bons resultados (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011).

O algoritmo adaptado Multi-Label *k*-Nearest Neighbor (MLkNN) é uma derivação do algoritmo preguiçoso k-Nearest Neighbor (kNN). A classificação de uma nova instância *t* funciona inicialmente com seus *k* vizinhos mais próximos do conjunto de teste sendo identificados (ZHANG; ZHOU, 2007).

Dada uma instância t e seu conjunto de classes $Y \subseteq C$ o vetor $\vec{y}_t$ representa as classes que compõe a instância t. Cada i-ésima posição deste vetor $\vec{y}_t = \{c_1, c_2, ..., c_i\}$ ($c_i \in C$), com $1 \leq i \leq |C|$, armazena o valor 1 se a classe $c \in Y$ e 0 caso contrário. A partir das classes de cada vizinho v com $v \in k$, o vetor que $\vec{V}_t(c)$ representa o número de vizinhos de t que pertencem à classe c, conhecido como *membership counting* é definido pela Fórmula 8.

$$\vec{V}_t(c) = \sum_{v \in k} \overrightarrow{y_v}(c), \; c \in Y \qquad (8)$$

Para cada nova instância t a ser classificada, o MLkNN primeiro identifica seus k vizinhos mais próximos do conjunto de treino. Seja $H_1^c(t)$ o evento que nova instância t possui a classe c, enquanto H_0^c (t) seja o evento que a instância t não possua a classe c. Seja E_j^1 (j $\in$ {0,1,...,k}) o evento que, entre os k vizinhos mais próximos de t, contenha exatamente j instâncias que possuam a classe c. Assim, baseado no vetor *membership counting* a fórmula de *Maximum a Posteriori* (MAP), usada para identificar o vetor $\vec{y}_t$ de classes da instância, é dada pela Fórmula 9.

$$\vec{y}_t(c) = argmax_{b \in \{0,1\}} P(H_b^c)\, P(E^c_{\vec{C}_{(t)}(c)} \mid H_b^c) \qquad (9)$$

2.7 Desbalanceamento dos dados

Na área de pesquisa em aprendizado de máquina, a distribuição desbalanceada do conjunto de dados ocorre quando uma das classes possui maior quantidade de dados do que a(s) outra(s) classe(s). As classes que possuem maior número de instâncias são chamadas classes majoritárias, enquanto que as classes que contêm relativamente menos instâncias são

chamadas classes minoritárias (CHAWLA; JAPKOWICZ; KOTICZ, 2004). Para resolver os problemas associados com o desbalanceamento dos dados existem métodos que são divididos em três categorias sendo, o tratamento do desbalanceamento durante a etapa de pré-processamento, o uso de algoritmos adaptados e a seleção de dados significativos (LONGADGE; DONGRE; MALIK, 2013).

O tratamento do desbalanceamento dos dados na etapa de pré-processamento envolve a redistribuição dos dados, que pode ser feita de três maneiras, com o *under-sampling* da classe majoritária, com o *over-sampling* da classe minoritária, ou a combinação destas duas técnicas (WASIKOWAKI; CHEN, 2010) (LONGADGE; DONGRE; MALIK, 2013). O método de *random under-sampling* procura balancear os dados através da remoção aleatória de dados da classe majoritária. *Random over-sampling* procura balancear os dados através da duplicação aleatória de dados da classe minoritária. O problema de usar estas técnicas aleatórias é que, em *random under-sampling* pode-se remover dados significativos ao contexto, e o *random over-sampling* pode levar ao *overfitting*, que é o classificador especializado em um conjunto específico de dados de treino (CHAWLA; JAPKOWICZ; KOTICZ, 2004; SEIFFERT et. al., 2008).

Chawla et al. (2002) propôs outro método de *over-sampling* chamado *Synthetic Minority Oversampling Technique* (SMOTE), que ao invés de duplicar dados aleatoriamente das classes minoritárias, SMOTE cria novos dados mais representativos para as classes minoritárias.

Os algoritmos adaptados levam em consideração o custo de a classificação estar certa ou errada, e procuram reduzir as classificações que estejam erradas. Este tipo aprendizado é conhecido como *cost sensitive learning*, que procura tratar o problema do desbalanceamento dos dados através do uso de

matrizes que descrevem o custo de classificar erroneamente (MUNTEAN et al., 2010; HAIBO; GARCIA, 2009).

O objetivo da seleção dos dados significativos é selecionar do conjunto de dados um subconjunto que melhor representem o contexto estudado, permitindo assim o classificador obter melhor performance (WASIKOWAKI; CHEN, 2010). A seleção de dados significativos é um passo importante para muitos algoritmos de aprendizado de máquina, especialmente quando o conjunto de dados possui uma alta dimensionalidade, como em processamento de imagem, categorização da Web ou mineração de textos. A alta dimensionalidade em conjuntos de dados reais é comumente acompanhada por outro problema que é a distribuição dos dados, geralmente com a classe de interesse contendo poucos dados; por isso a importância de selecionar os dados que melhor treinam o classificador (CHAWLA; JAPKOWICZ; KOTICZ, 2004). A seleção dos dados tem sido aplicada à categorização de texto para melhorar sua escalabilidade, eficiência e acurácia. Uma série de métricas de seleção de dados tem sido explorada na categorização de texto, entre as quais estão *information gain* (IG), *chi-squared* (CHI) e *oddsratios* (OR), consideradas mais eficientes (FORMAN, 2003).

Neste trabalho o tratamento do problema do desbalanceamento dos dados foi realizado durante a etapa de pré-processamento, através do uso de dois algoritmos que compõem a ferramenta Weka, sendo os filtros SMOTE e RESAMPLE, os quais estão descritos no capítulo 3, seções 3.4.1 e 3.4.2 respectivamente. O filtro SMOTE foi utilizado para o *over-sampling* das classes minoritárias e o filtro RESAMPLE foi utilizado tanto para o *over-sampling* das classes minoritárias quanto para o *under-sampling* das classes majoritárias.

2.8 Bibliotecas Weka e Meka

A biblioteca *open source* Weka (WITTEN; FRANK, 2005) contém algoritmos para auxiliar o todo processo de mineração de dados. Esta biblioteca contém algoritmos para a etapa de limpeza e pré-processamento como o *String To Word Vector*, algoritmos de tratamento do desbalanceamento dos dados, como o SMOTE e o RESAMPLE, implementações de algoritmos de classificação, como o SVM, o SMO e o Naïve Bayes, técnicas para geração e avaliação de modelos, como a 10-*fold cross-validation,* e métricas como a Precisão, Revocação e Medida-F.

A biblioteca *open source* Meka[7] fornece implementações de algoritmos de classificação multi-rótulo e métodos para avaliação de modelos multi-rótulo. Este projeto foi baseado na ferramenta Weka e incorpora a biblioteca Mulan[8] (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011), uma biblioteca desenvolvida em Java que contém algoritmos de classificação multi-rótulo, como o RAkEL e o MLKNN, além de métodos para avaliar modelos multi-rótulo como o *Hamming Loss* e *One-error*.

2.9 Trabalhos relacionados

Nessa seção são apresentados trabalhos que trataram dos mesmos problemas encontrados e que fornecem referências para a construção deste trabalho. Os trabalhos relacionados descrevem técnicas de mineração de dados em ambiente virtual de aprendizagem, o uso de mineração de texto em fóruns destes ambientes a fim de dar outra visão das mensagens postadas, o uso de

[7] Meka (2014)
[8] Mulan (2014)

abordagens para tratar as diferentes distribuições das classes das mensagens, o estudo de técnicas e abordagens para o tratamento de desbalanceamento dos dados e o uso de algoritmos para criação de mensagens sintéticas a fim de balancear os dados.

Romero, Ventura e Garcia (2008) descrevem de forma teórica e prática todo o processo de mineração de dados, da coleta dos dados à aplicação de diversas técnicas como classificação e clusterização, em um ambiente virtual de aprendizagem, o Moodle. Dentre estes processos foi utilizada a mineração de texto, a fim de extrair palavras chaves das discussões de fóruns para que o professor possa acompanhar a qualidade das mensagens postadas nestes fóruns.

No trabalho de Oliveira Júnior et. al. (2011) foi utilizada mineração de texto para a identificação de polaridades de mensagens em fóruns do Moodle, sendo positiva ou negativa, a fim de identificar mensagens que necessitem de maior atenção. Utilizando o algoritmo de classificação SVM e um conjunto de dados de treino desbalanceado, com maior quantidade de mensagens na polaridade negativa, como objetivo do modelo classificador ser tendencioso para a classificação de mensagens negativas, pois são aquelas que necessitam de maior atenção.

Em Lin, Hsieh e Chuang (2009) foi proposto um modelo em cascata para a classificação de gêneros de mensagens em fóruns do Moodle chamado *Genre Classification System* (GCS). O conjunto de dados de treino dos modelos que compõem esta abordagem é ajustado para treinar cada classe em separado. Segundo os autores, o modelo em cascata GCS pode lidar melhor com o desbalanceamento dos dados em comparação com os modelos multi-classe.

Para o tratamento do desbalanceamento dos dados, os trabalhos de R. Longadge, Dongre e Malik (2013), Chawla, Japkowicz e Koticz (2004), e Wasikowaki e Chen (2010) descrevem três metodologias utilizadas para melhorar as classificações dos algoritmos. Uma metodologia se refere à

distribuição dos dados entre as classes e maneiras de aumentar os dados da classe minoritária ou diminuir os dados das classes majoritárias. Uma metodologia que utiliza de algoritmos adaptados e utilizam medidas como ranking e técnicas conjuntas de modelos, a fim de diminuir os erros causados na classificação das classes minoritárias. E uma metodologia que procura selecionar alguns atributos mais relevantes para a classificação e com isso diminuir a quantidade de atributos para a construção dos modelos.

Chawla et al. (2002) propuseram um método mais eficiente de *over-sampling* do algoritmo *Synthetic Minority Oversampling Technique* (SMOTE), que ao invés de replicar dados da classe minoritária, cria novas instâncias sintéticas. A replicação de instâncias faz com que o classificador se torne ainda mais específico para aqueles dados minoritários; enquanto que, com a criação de novas instâncias sintéticas, são criadas novas combinações de atributos dos dados para as classes minoritárias.

Em Guimarães e Esmin (2014) foram utilizados dois algoritmos de tratamento do desbalanceamento de dados, o SMOTE e o RESAMPLE, a fim de melhorar a Acurácia e as métricas Precisão, Revocação e Medida-F dos algoritmos de classificação SVM, SMO e o Naïve Bayes. Os resultados comprovam que o uso destes dois algoritmos de balanceamento dos dados melhora significantemente o valor das métricas supracitadas. A comparação entre dois modelos de classificação multi-classe gerados pelo algoritmo SMO, um com o conjunto de dados desbalanceado e outro balanceado com RESAMPLE, resultou em uma diferença de 25,88% na Acurácia entre os modelos.

Este trabalho vem com o intuito de encontrar a melhor abordagem de classificação de gêneros das mensagens de fóruns de um ambiente virtual de aprendizagem. As mensagens destes fóruns estão distribuídas de maneira desbalanceada entre os gêneros, o que fez necessário o uso de técnicas para o

tratamento do desbalanceamento dos dados. Os resultados dos modelos gerados pelos algoritmos SVM, SMO e Naïve Bayes, com o uso dos algoritmos SMOTE e RESAMPLE para o balanceamento dos dados, resultou numa melhoria significante em relação ao conjunto de dados desbalanceados. Os modelos gerados pelos algoritmos SVM, SMO e Naïve Bayes, com três tipos de conjuntos de dados, um desbalanceado, um balanceado com SMOTE e um balanceado com RESAMPLE, foram distribuídos na abordagem em cascata proposta por (LIN; HSIEH; CHUANG, 2009).

Neste trabalho foi proposta uma nova abordagem em que a mensagem é disposta a dois classificadores simultaneamente e a composição destes classificadores pode resultar na mensagem sendo classificada como multi-classe ou multi-rótulo. Um classificador é formado por um conjunto de modelos de classificação binários e o outro classificador é formado por um modelo de classificação multi-classe. Esta nova abordagem é capaz de classificar a mensagem com um ou mais gêneros (multi-rótulo), e para tanto, são usadas mensagens classificadas manualmente em somente um gênero para a geração dos modelos de classificação.

3 METODOLOGIA PARA CLASSIFICAÇÃO DE GÊNEROS DE MENSAGENS

Este capítulo apresenta a descrição dos problemas encontrados durante a construção dos modelos de classificação e as abordagens utilizadas para contorná-los, o processo de construção das bases de dados, bem como as principais métricas de avaliação multi-classe e multi-rótulo empregadas para avaliar os resultados.

3.1 Descrição da metodologia e o processo de trabalho

O estudo para descoberta de padrões de gêneros das mensagens em fóruns de discussão de Ambientes Virtuais de Aprendizagem via mineração de texto iniciou-se com a utilização dos algoritmos de classificação, SVM, SMO e Naïve Bayes, juntamente com o conjunto de dados de treino coletados que está desbalanceado para a geração de modelos de classificação multi-classe. Os modelos gerados foram tendenciosos para as classes majoritárias. Por consequência desta tendência, foram utilizados os algoritmos SMOTE e RESAMPLE para o balanceamento dos dados do conjunto de treino. As medidas avaliadas dos modelos de classificação multi-classe gerados após o tratamento do desbalanceamento dos dados foram substancialmente melhores.

A segunda realização foi a criação de modelos de classificação binários, capazes de classificar uma mensagem como pertencente ou não a uma classe. Estes modelos foram combinados em uma abordagem em cascata como em (LIN; HSIEH; CHUANG, 2009), e o resultado desta abordagem em cascata foi comparado com os modelos de classificação multi-classe. Estas abordagens classificam a mensagem em somente um gênero.

A terceira realização foi uma abordagem de distribuição dos modelos de classificação binários em paralelo, sendo a mensagem classificada por todos os modelos, e o resultado da classificação é a união do resultado de cada modelo. Esta abordagem resultou na classificação da mensagem em mais de um gênero, multi-rótulo, o que necessitou de decidir qual gênero melhor classificaria a mensagem. Para tanto, foi utilizado um modelo de classificação multi-classe como oráculo a fim de decidir qual o gênero que melhor descreve a mensagem.

A abordagem proposta é resultado desta terceira realização, sendo a composição de dois classificadores, um sendo conjunto de modelos binários, e outro sendo um classificador multi-classe. A união dos resultados destes dois classificadores é o resultado da classificação da abordagem. A descrição desta abordagem se encontra no capítulo 4.

3.2 Construção das bases de dados

Neste trabalho foram utilizadas mensagens de texto escritas em português provenientes de fóruns de discussão do AVA Moodle de cursos a distância. Estas mensagens foram anonimizadas, ou seja, passaram pelo processo de retirada de informações que identificam pessoas e os autores envolvidos a fim de proteger sua privacidade (TAGG, 2012). Estas mensagens foram manualmente classificadas por três especialistas sendo que cada mensagem foi classificada uma única vez por um especialista.

Foi desenvolvida uma aplicação na qual as mensagens são dispostas aos especialistas para sua classificação, representado na Figura 5. Nesta aplicação os especialistas classificaram a mensagem em um, e apenas um, dos seguintes gêneros: Anúncio, Conflito, Dúvida, Esclarecimento, Interpretação e Outros. Esta classificação inicial dos especialistas é considerada como a classe principal da mensagem, sendo importante para a métrica de avaliação de classificação

multi-rótulo *One-error*. Como havia poucas mensagens classificadas no gênero Conflito, estas mensagens foram classificadas com o gênero Outros e o gênero Conflito foi removido.

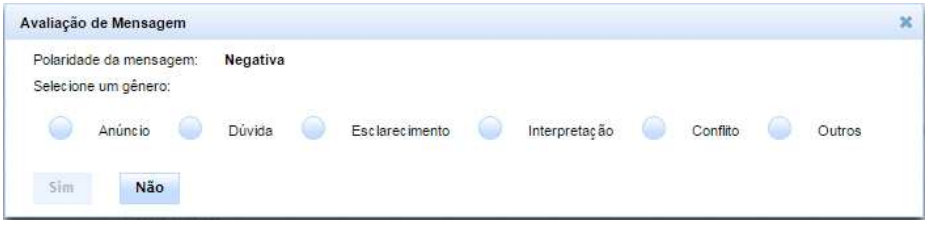

Figura 5 Interface para classificação inicial da mensagem em um gênero

A base de dados coletada possui 7367 mensagens classificadas em um dos cinco gêneros e foi dividida em dois conjuntos, um conjunto de dados de treino contendo 6367 mensagens e um conjunto de dados de teste contendo 1000 mensagens. O conjunto de dados de treino é utilizado para o aprendizado dos algoritmos de classificação e geração de modelos de classificação. O conjunto de dados de teste é utilizado para avaliar os modelos gerados pelos algoritmos classificação. A distribuição desta base de mensagens coletadas, classificadas em um gênero, se encontra na Tabela 5 dividida entre os gêneros e agrupadas pelo tipo do conjunto de dados.

Tabela 5 Distribuição do conjunto de dados coletados

Conjunto	*Treino*		*Teste*	
Gêneros	**Quant.**	**%**	**Quant.**	**%**
Anúncio	699	10,98	200	20
Dúvida	421	**6,61**	**200**	**20**
Esclarecimento	487	7,65	200	20
Interpretação	2220	34,87	200	20
Outros	2540	**39,89**	**200**	**20**
Total	6367	100	1000	100

Com o desenrolar dos experimentos foi desenvolvida uma abordagem que possibilitou a classificação de uma mensagem em mais de um gênero. Por conta disto, tornou-se necessário avaliar os resultados desta abordagem de classificação multi-rótulo. Para tanto, é necessário um conjunto de dados classificados multi-rótulo.

As mensagens do conjunto de dados de teste, que estão classificadas em um gênero, foram classificadas pela abordagem proposta de classificação multi-rótulo. Os resultados da classificação, isto é, as mensagens e o(s) gênero(s) em que elas foram classificadas automaticamente pela abordagem, foram analisados e corrigidos pelos especialistas, através da interface ilustrada na Figura 6. Estas mensagens corrigidas pelos especialistas formaram o conjunto de dados de treino para algoritmos de classificação multi-rótulo.

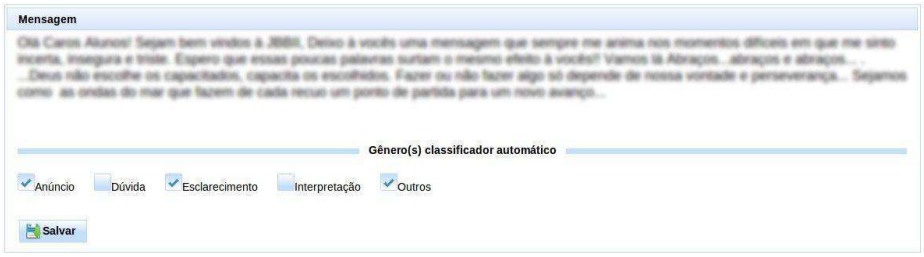

Figura 6 Interface para classificação multi-rótulo da mensagem

A distribuição deste conjunto de dados multi-rótulo está na Tabela 6, sendo que os nomes dos gêneros estão sendo representados pela sua primeira letra. Estes dados também já haviam passado pela etapa de limpeza e pré-processamento, que está descrita na seção a seguir, e contém um total de 902 mensagens.

Tabela 6 Distribuição do conjunto de dados multi-rótulo

Um Gênero		*Dois Gêneros*		*Três Gêneros*		*Quatro Gêneros*		*Cinco Gêneros*	
Gênero	**Qte.**	**Gêneros**	**Qte.**	**Gêneros**	**Qte.**	**Gêneros**	**Qte.**	**Gêneros**	**Qte.**
A	68	A,D	24	A,D,E	3	A,D,E,I	1	A,D,E,I,O	0
D	68	A,E	12	A,D,I	4	A,D,E,O	0		
E	115	A,I	36	A,D,O	13	A,D,I,O	2		
I	69	A,O	71	A,E,I	3	A,E,I,O	1		
O	128	D,E	42	A,E,O	9	D,E,I,O	2		
		D,I	1	A,I,O	20				
		D,O	45	D,E,I	7				
		E,I	55	D,E,O	9				
		E,O	42	D,I,O	8				
		I,O	27	E,I,O	17				

Este conjunto de dados multi-rótulo foi utilizado como conjunto de dados de treino para a indução de dois algoritmos de classificação multi-rótulo, para que seus resultados fossem comparados com a abordagem desenvolvida.

Os conjuntos de dados coletados precisam ser formatados para serem utilizados por algoritmos de classificação. Esta formatação é feita na etapa de limpeza e pré-processamento estando descrita na seção a seguir.

3.3 Limpeza e pré-processamento

Esta etapa no processo de mineração de dados consiste na transformação do arquivo de treino para o formato de entrada para algoritmos de classificação.

A etapa de limpeza consiste na retirada de dados que contenham algum tipo de erro ou ruído, como mensagens em branco ou repetidas, palavras que não possuem significado semântico conhecidos como *stop words.* Neste trabalho também foram retiradas mensagens que fossem menor que duas palavras, a fim de retirar aquelas mensagens que possuíam pouco ou nenhum significado, removendo por exemplo mensagens que continham agradecimentos ou apenas apresentações como, “obrigado”, “tchau”, “bom dia”, “por favor”. A Figura 7

ilustra um trecho de arquivo de treino após a limpeza. Este arquivo possui o formato *Attribute-Relation File Format*(.arff), onde as mensagens foram abreviadas conforme abaixo.

```
@relation conjunto_dados_treino
@attribute message string
@attribute class {A, D, E, I, O}
@data
'dia cursista perguntou ... orienta los abracos obrigada', D,
'ideal desejavel seria ... gente colaborar discussoes', E,
'ola cursistas prazer ... perguntem abracos', A,
```

Figura 7 Arquivo .arff após limpeza

Neste trabalho, na etapa de pré-processamento, o arquivo foi processado pelo filtro *String To Word Vector* da ferramenta Weka. Este filtro faz a conversão das palavras das mensagens em uma representação numérica da palavra e um peso associado a ela. Este peso significa o quanto o atributo (palavra) é importante no conjunto das instâncias (mensagens). O cálculo do peso é através do uso da estatística TF-IDF, *Term Frequency* (TF) x *Inverse Document Frequency* (IDF), sendo que o TF considera a frequência do atributo na instância e o IDF a frequência do atributo em relação ao conjunto de todas as instâncias. A Figura 8 ilustra um exemplo de trecho de arquivo ".arff" após o filtro *String To Word Vector,* este formato de arquivo é utilizado como parâmetro de entrada para algoritmos de classificação.

```
@relation conjunto_dados_treino_pos_string_to_word_vector
@attribute class {A, D, E, I, O}
@attribute abaixo numeric
@attribute aberto numeric
...
@attribute vivacidade numeric
@data
{6 1.002618,28 3.558505, ... ,1861 3.67239}
{0 D,240 3.697598,368 2.430437, ... ,2241 4.018264}
{0 E, 531 2.302687,551 2.762852}
```

Figura 8 Trecho de arquivo .arff após o filtro *String To Word Vector*

Após esta etapa de limpeza e pré-processamento, algumas mensagens foram removidas, diminuindo o tamanho do conjunto de dados de treino e de teste. A distribuição do conjunto de dados de treino e teste resultante destas etapas está na Tabela 7, estando destacado em negrito a porcentagem de dados das classes minoritária e majoritária do conjunto de treino. O destaque tem como intuito frisar a diferença de tamanho da classe com mais dados, majoritária, para a classe com menos dados, minoritária.

Tabela 7 Distribuição dos dados após a limpeza e o pré-processamento

Conjunto	*Treino*		*Teste*	
Gêneros	**Quant.**	**%**	**Quant.**	**%**
Anúncio	576	10,29	143	15,85
Dúvida	407	**7,27**	191	21,18
Esclarecimento	470	8,39	192	21,29
Interpretação	2046	36,54	199	22,06
Outros	2100	**37,51**	177	19,62
Total	5599	100	902	100

Este conjunto de treino resultante desta etapa é referido nas próximas seções como conjunto de dados de treino desbalanceado.

3.4 Desbalanceamento dos dados entre as classes

Como destacado na Tabela 7 há um desbalanceamento da distribuição dos dados entre as classes do conjunto de treino. Neste conjunto podemos considerar que temos três classes minoritárias e duas classes majoritárias, sendo que a porcentagem de dados das classes minoritárias são 7,27%, 8,39% e 10,29% e das classes majoritárias são 36,54% e 37,51%, havendo uma diferença de 26,25% entre a menor porcentagem da classe majoritária e a maior porcentagem da classe minoritária. Se compararmos o maior porcentual da classe majoritária com o menor porcentual da classe minoritária, sendo a porcentagem de 37,51% e 7,27%, temos uma diferença de aproximadamente 30,24%. O desbalanceamento dos dados é um problema a ser considerado elaboração de metodologias para classificação de dados (JAPKOWICZ; STEPHEN, 2002).

Os experimentos foram realizados utilizando três diferentes conjuntos de dados de treino. A partir do conjunto de dados de treino desbalanceado, foram gerados dois novos conjuntos utilizando técnicas para balanceamento da distribuição dos dados entre as classes. Um conjunto com criação de dados sintéticos para as classes minoritárias e sem remoção de dados das classes majoritárias utilizando o algoritmo *Synthetic Minority Over-Sampling Technique* (SMOTE) (CHAWLA et al., 2002). Um conjunto com redistribuição de dados através da criação de dados sintéticos para classes minoritárias e da remoção de dados da classe majoritária utilizando o algoritmo RESAMPLE.

3.4.1 Balanceamento dos dados com SMOTE

A técnica *Synthetic Minority Over-Sampling Technique* (SMOTE) foi utilizada para o balanceamento de dados através da criação de dados sintéticos

para as classes minoritárias Anúncio, Dúvida e Esclarecimento. Nas classes majoritárias o algoritmo não foi aplicado. O algoritmo SMOTE, pode melhorar a acurácia de classificadores para classes minoritárias. Essa técnica pode envolver a remoção de amostragem (*under-sampling*) da classe majoritária juntamente com a adição de amostragem nas classes minoritárias. A adição é feita a partir da criação de dados sintéticos para cada classe minoritária. A remoção é feita removendo aleatoriamente algumas amostras. Esta combinação levou a se obter melhores resultados do que somente a remoção de amostragem nas classes majoritárias (CHAWLA et al., 2002).

O *over-sampling* dos dados da classe minoritária é feito para cada amostra desta classe, em que são criados novos dados sintéticos através do uso de alguns de seus *k* vizinhos mais próximos. Neste trabalho são utilizados os cinco vizinhos mais próximos. Por exemplo, se o total de dados criados por *over-sampling* for de 200%, somente dois dos cinco vizinhos são escolhidos e uma nova amostra é gerada.

3.4.2 Balanceamento dos dados com RESAMPLE

A técnica RESAMPLE foi utilizada para o balanceamento de dados através da criação de dados sintéticos para as classes minoritárias Anúncio, Dúvida e Esclarecimento e com a remoção de dados das classes majoritárias Interpretação e Outros. Como descrito na ferramenta Weka, este algoritmo pode ser utilizado para criar uma nova amostra com os dados distribuídos de maneira mais uniforme entre as classes e/ou aumentar ou diminuir o tamanho da amostra (HALL et al., 2009). Neste trabalho o algoritmo foi configurado com o parâmetro "bias", indicando que os dados entre as classes deveriam estar balanceados, e com o parâmetro "*sample Size Percent*", indicando que não houvesse aumento e nem diminuição no tamanho do conjunto de dados.

3.5 Métricas de avaliação de modelos de classificação

Avaliar um modelo de classificação é mensurar o quão distante estão as predições do modelo em relação as classes reais de dados. Na classificação de gêneros de mensagens de texto, a avaliação é feita comparando os gêneros que a mensagem foi classificada pelo modelo por especialistas humanos. A classificação da mensagem realizada pelos especialistas é considerada a classe correta da mensagem, conhecido como *ground truth*.

As abordagens desenvolvidas neste trabalho são capazes de classificar uma mensagem em um gênero (multi-classe) ou classificar uma mensagem em um ou mais gêneros (multi-rótulo). Estas abordagens foram avaliadas através de métricas de avaliação conhecidas na literatura e que estão descritas nas subseções seguintes. Os resultados da abordagem desenvolvida de classificação multi-rótulo também foram comparados com os resultados obtidos por algoritmos e métodos adaptados para classificação multi-rótulo.

3.5.1 Métricas de avaliação de modelos de classificação multi-classe

Para avaliar os modelos de classificação multi-classe foram analisadas algumas métricas conhecidas na literatura, sendo elas: a Acurácia, Precisão, Revocação e a Medida-F (HOTHO; NÜRNBERGERA; PAAB, 2005; LIN; HSIEH; CHUANG, 2009).

Para entendimento do cálculo das métricas será feita uma introdução com alguns conceitos e símbolos. Como exemplo, considere um conjunto de classes C, sendo C = {A, D, E, I, O} e um conjunto de mensagens M, sendo M = $\{M_1, M_2, M_3, M_4\}$. Para a avaliação de um modelo de classificação, para cada classe do conjunto C é calculado o valor das métricas Acurácia, Precisão,

Revocação e a Medida-F. Estas métricas são calculadas através da Tabela de Contingência. Na Tabela 8, a Tabela de Contingência para a classe A.

Tabela 8 Tabela de Contingência para a classe A

Tipo Classificação		*Especialista*	
	Classe	**A**	**NA**
Modelo de	**A**	TP	FP
Classificação	**NA**	FN	TN

Como exemplo considere a classe A do conjunto de classes C. Para o cálculo do valor das métricas para esta classe temos que: *True Positive* (TP), mensagem classificada pelo Modelo de Classificação como A e pelo Especialista como A; *True Negative* (TN), mensagem classificada pelo Modelo de Classificação como Não A e pelo Especialista como Não A; *False Positive* (FP), mensagem classificada pelo Modelo de Classificação como A e pelo Especialista como Não A; *False Negative* (FN), mensagem classificada pelo Modelo de Classificação como Não A e pelo Especialista como A. Para ilustrar estas combinações da Tabela de Contingência, considere o conjunto de mensagens M sendo classificadas em uma classe do conjunto de classes C por Especialistas e por um Modelo de Classificação, como encontra-se na Tabela 9.

Tabela 9 Exemplo de resultado de Tabela de Contingência

Mensagem	**Especialista (correta)**	**Modelo Classificação**	**Resultado classe A**
M_1	A	A	**TP**
M_2	A	NA	**FN**
M_3	NA	A	**FP**
M_4	NA	NA	**TN**

Acurácia (Ac): a quantidade de dados classificados corretamente em relação ao conjunto total de dados classificados. Neste trabalho é o gênero

corretamente classificado das mensagens pelo total de mensagens classificadas automaticamente.

$$\mathrm{Ac} = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

Precisão (P): é mensurada por classe, sendo a quantidade de dados corretamente classificados em uma classe dividida pela soma de dados classificados naquela classe pelo Modelo de Classificação. Neste trabalho é a fração de mensagens corretamente classificadas pelo total de mensagens classificadas em cada gênero.

$$\mathrm{P} = \frac{TP}{TP+FP} \quad (11)$$

Revocação (R): é mensurada para cada classe, sendo quantidade de dados corretamente classificados em uma classe dividida pela soma de dados classificados naquela classe pelo Especialista. Neste trabalho é a fração de mensagens corretamente classificadas dividida pelo total de mensagens classificadas neste gênero pelos Especialistas.

$$\mathrm{R} = \frac{TP}{TP+FN} \quad (12)$$

Medida-F (MF): é a média harmônica entre a precisão e revocação.

$$\mathrm{MF} = \frac{2*(P*R)}{P+R} \quad (13)$$

3.5.2 Métricas de avaliação de modelos de classificação multi-rótulo

O resultado da previsão de um modelo de classificação multi-rótulo pode estar totalmente correto, parcialmente correto ou totalmente incorreto; o que torna a avaliação de classificadores multi-rótulo mais desafiador que a avaliação de modelos classificadores binários ou classificadores multi-classe (SOROWER, 2010).

Para avaliar a abordagem de classificação multi-rótulo desenvolvida, foram utilizadas métricas de avaliação de algoritmos para classificação multi-rótulo. Os resultados da abordagem desenvolvida também foram comparados com resultados de modelos gerados por algoritmos e métodos adaptados para a classificação multi-rótulo, sendo o algoritmo adaptado MLkNN e o método de transformação RAkEL. Para entendimento das fórmulas das métricas multi-rótulo avaliadas, a Tabela 10 introduz uma série de simbologias e notações que serão encontradas nas fórmulas baseados em (SOROWER, 2010).

Tabela 10 Símbolos e notações das fórmulas das métricas multi-rótulo

Símbolo	Significado
n	Número de instâncias
I	Instância
C	Conjunto de classes
|C|	Número de classes
$Y = \{y_1, y_2, .., y_k\}$	Conjunto de classes rotuladas por Especialistas (corretas)
$Z = \{z_1, z_2, .., z_k\}$	Conjunto de classes previstas pelo Modelo de Classificação
c_p	Classe principal da instância

***Hamming Loss* (HL):** avalia na média entre todas as instâncias, quantas classes foram classificadas erroneamente; isto é, uma classe que pertence à

instância e não foi prevista, ou uma classe que não pertence à instância e foi prevista. Em resumo, é a quantidade de classes classificadas erradas.

$$\mathrm{HL} = \frac{1}{|C|n}\sum_{i=1}^{n}\sum_{l=1}^{|C|}[I(l \in Zi \wedge l \notin Yi) + I(l \notin Zi \wedge l \in Yi)] \quad (14)$$

Idealmente o valor da métrica *Hamming Loss* é 0, o qual implicaria que está tudo correto e não há erro. Quanto menor o *Hamming Loss*, melhor a performance do algoritmo.

***One-error* (OE):** esta métrica é baseada em ranking de classes, isto é, para cada uma das classes rotuladas em uma instância, um valor é associado identificando o quanto aquela instância pertence àquela classe. A classe com maior valor é a classe principal da instância c_p. *One-error* e avalia quantas vezes a classe principal da instância não foi prevista pelo classificador. Neste trabalho o gênero principal da mensagem é aquele que foi classificado pelos especialistas na primeira etapa de classificação, portanto a métrica *One-Error*, mensura quantas vezes este gênero não foi classificado pelo classificador.

$$\mathrm{OE} = \frac{1}{n}\sum_{i=1}^{n} I(argmin_{\ c \in C} r_i(c) \notin Y_i) \quad (15)$$

Em que r_i (c) é a classe com maior rank, isto é, a probabilidade de estar correta. Neste trabalho, esta classe é a classe principal da mensagem, como referido na seção 3.2. Idealmente o valor da métrica One-error é 0, o que significa quanto menor o valor desta métrica melhor a performance do algoritmo.

4 CONSTRUÇÃO DE MODELOS E O DESBALANCEAMENTO DOS DADOS

A fim de entender, demonstrar e mensurar o efeito do problema de desbalanceamento dos dados nos algoritmos classificação, este capítulo apresenta os resultados da indução de algoritmos de classificação com diferentes bases de treinamento.

Neste capítulo são descritas as configurações destes experimentos na construção de modelos de classificação multi-classe e de modelos de classificação binários.

A seção 4.1 descreve as configurações dos experimentos e nas seções 4.2 e 4.3 são criados modelos de classificação multi-classe e binários respectivamente, utilizando os três conjuntos de dados de treino.

4.1 Configurações dos experimentos

Neste capítulo os resultados mostrados são referentes aos algoritmos SVM e sua implementação LibSVM com o *kernel* linear, o SMO e o Naïve Bayes da ferramenta Weka. Nos experimentos de criação de modelos de classificação foram utilizados três conjuntos de dados de treino, um conjunto desbalanceado, um conjunto balanceado com o SMOTE e um conjunto balanceado com o RESAMPLE.

Para a avaliação da criação dos modelos foi utilizada a técnica 10-*fold cross-validation*, técnica que separa nove partes do conjunto de dados para treino do algoritmo e uma parte do conjunto dos dados para teste e avaliação do modelo gerado pela indução do algoritmo.

Os resultados avaliados nos experimentos multi-classe foram a Acurácia e as métricas Precisão, Revocação e Medida-F.

4.2 Modelo de classificação multi-classe

Um modelo gerado por um algoritmo de classificação multi-classe é um classificador com a capacidade de rotular um novo dado em uma, e apenas uma, das classes do conjunto de dados de treino utilizado para indução do algoritmo e geração do modelo. No contexto deste trabalho, o modelo de classificação multi-classe classifica a mensagem em um, e apenas um, dos cinco gêneros: Anúncio, Dúvida, Esclarecimento, Interpretação e Outros. A Figura 9 ilustra o funcionamento deste modelo de classificação multi-classe.

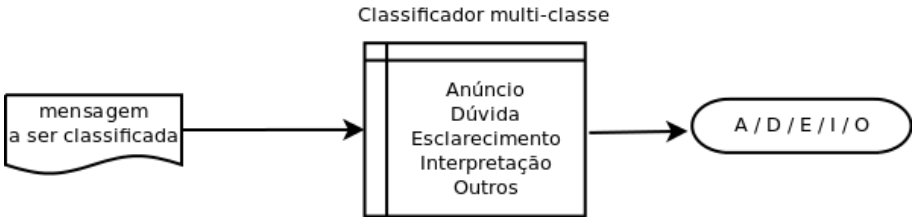

Figura 9 Funcionamento do modelo de classificação multi-classe

4.2.1 Conjunto de dados de treino

Os conjuntos de dados de treino foram gerados a partir dos dados originalmente coletados detalhados na Tabela 5. A partir deste conjunto de dados coletados e que está desbalanceado, foram gerados dois novos conjuntos de dados balanceados com o uso dos algoritmos SMOTE e RESAMPLE.

A descrição destes novos conjuntos de dados (*dataset*) balanceados se encontra na Tabela 11, estando destacado em negrito o percentual da classe

minoritária e da classe majoritária para cada tipo de conjunto de dados, uma coluna indica a quantidade de dados (Qte.), uma coluna indicando o percentual desta quantidade de dados em relação ao total (%), e uma coluna (Dif.) indicando o percentual aproximado de aumento (símbolo ↑) ou diminuição (símbolo ↓) quantidade dos dados em relação ao conjunto coletado.

Tabela 11 Conjunto de dados de treino para classificação multi-classe

Dataset	*Desbalanceados*		*SMOTE*			*RESAMPLE*		
Gênero	**Qte**	**%**	**Qte**	**%**	**Dif.**	**Qte**	**%**	**Dif.**
Anúncio	576	10,29	2016	**19,73**	↑ 250	1025	**18,31**	↑ 77,95
Dúvida	407	**7,27**	2035	19,92	↑ 400	1160	**20,72**	↑ 185
Escla.	470	8,39	2020	19,77	↑ 330	1122	20,03	↑ 138,72
Inter.	2046	36,54	2046	20,03	-	1140	20,36	↓ 79,47
Outros	2100	**37,51**	2100	**20,55**	-	1152	20,57	↓ 82,29
Total	5599	100	10217	100	↑82,5	5599	100	0

O algoritmo SMOTE foi aplicado somente às classes minoritárias Anúncio, Dúvida e Esclarecimento para que fossem criados dados sintéticos para o balanceamento com as classes majoritárias; já as classes majoritárias Interpretação e Outros não foram modificadas. Com a aplicação deste filtro houve um aumento no tamanho do conjunto de dados de treino em 82,48% e as mensagens estão distribuídas de maneira balanceada entre os gêneros, sendo que a porcentagem da classe minoritária Anúncio é 19,73% e a porcentagem da classe majoritária Outros é 20,55% dos dados, uma diferença de apenas 0,82%.

O algoritmo RESAMPLE foi aplicado a todo conjunto de treino criando um novo conjunto de dados de treino balanceado, através da criação de dados para as classes minoritárias Anúncio, Dúvida e Esclarecimento e remoção de dados para as classes majoritárias Interpretação e Outros. A porcentagem dos dados da classe minoritária Anúncio é 18,31% e a da classe majoritária Dúvida é 20,72% sendo uma diferença de apenas 2,41%, sem que houvesse aumento no tamanho do conjunto de dados de treino.

4.2.2 Resultados dos modelos de classificação multi-classe

Nesta seção estão os resultados das métricas Precisão (P), Revocação (R) e Medida-F (MF) dos modelos gerados pelos algoritmos SVM, SMO e Naïve Bayes com a utilização dos três conjuntos de dados de treino, o desbalanceado, o balanceado com SMOTE e o balanceado com RESAMPLE.

Os resultados da métrica Precisão se encontram na Tabela 12, destacado em negrito o melhor valor de cada algoritmo em cada tipo de conjunto de dados, e sombreada está a média da Precisão comparada na Figura 10.

Tabela 12 Métrica Precisão por algoritmo e conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Algoritmo	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Anúncio	0,549	**0,599**	0,541	0,862	0,873	**0,999**	**0,9**	0,898	0,681
Dúvida	**0,449**	0,429	0,376	0,877	0,88	**1**	**0,897**	0,882	0,612
Escla.	0,262	**0,274**	0,272	0,808	0,834	**1**	**0,847**	0,844	0,523
Inter.	0,649	**0,687**	0,647	0,656	**0,679**	0,57	0,815	**0,818**	0,555
Outros	0,573	**0,594**	0,653	0,576	**0,596**	0,497	0,741	**0,75**	0,506
Média	0,496	**0,516**	0,497	0,755	0,772	**0,813**	**0,84**	0,838	0,575

Os resultados da média da métrica Precisão da Tabela 12 demonstram o quanto o desbalanceamento dos dados afeta o valor das métricas dos algoritmos de classificação multi-classe SVM, SMO e Naïve Bayes.

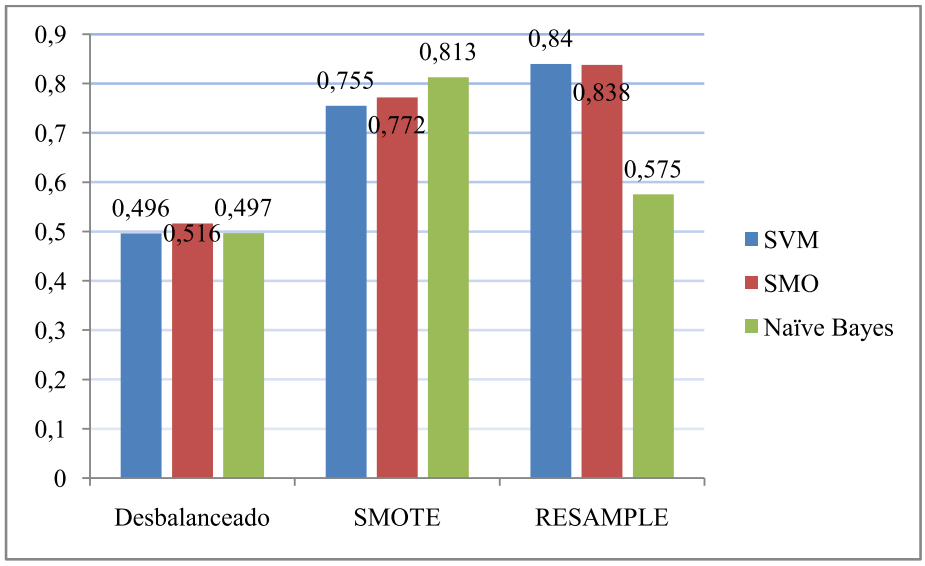

Figura 10 Média da métrica Precisão por algoritmos por conjunto de dados

Para a métrica Precisão, o algoritmo SMO foi melhor no conjunto de dados desbalanceado, o algoritmo NB para o conjunto balanceado com SMOTE e o SVM para o conjunto balanceado com RESAMPLE.

Os resultados da métrica Revocação se encontram na Tabela 13, destacado em negrito o melhor valor de cada algoritmo em cada tipo de conjunto de dados, e sombreada está a média da Revocação comparada na Figura 11.

Tabela 13 Métrica Revocação dos algoritmos por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Anúncio	0,569	0,625	**0,681**	0,87	**0,89**	0,718	0,918	**0,919**	0,715
Dúvida	0,369	**0,435**	0,43	0,883	**0,884**	0,8	0,955	**0,962**	0,592
Escla.	0,23	0,279	**0,383**	0,834	**0,836**	0,768	**0,913**	0,908	0,517
Inter.	0,651	0,63	**0,755**	0,617	0,629	**0,806**	0,719	0,732	**0,761**
Outros	0,602	**0,631**	0,422	0,582	0,624	**0,638**	**0,709**	0,689	0,32
Média	0,484	0,52	**0,534**	0,757	**0,772**	0,746	**0,842**	**0,842**	0,581

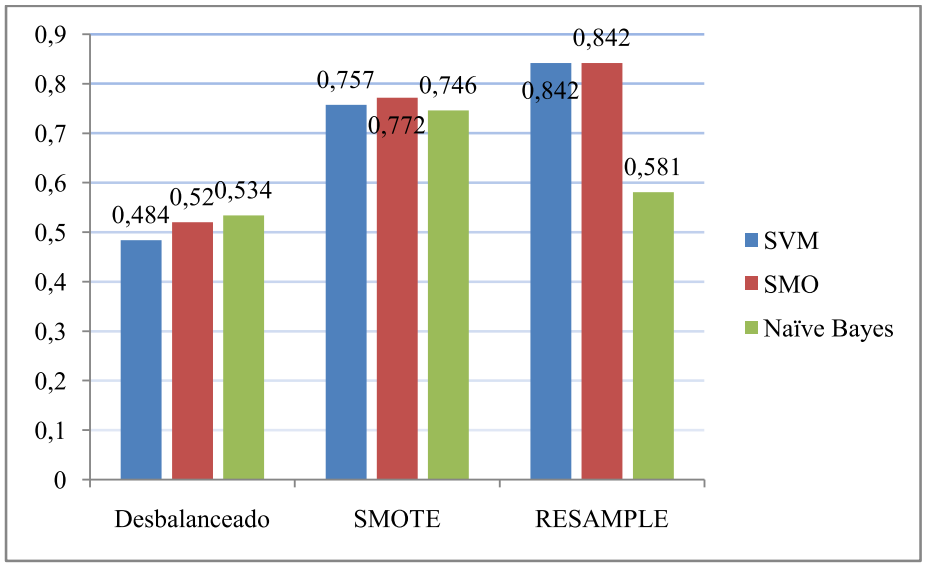

Figura 11 Média da Revocação dos algoritmos por conjunto de dados

Para a métrica Revocação o algoritmo SMO foi melhor no conjunto de dados desbalanceados e para o conjunto balanceado com SMOTE e o SVM para o conjunto balanceado com RESAMPLE.

Os resultados da métrica Medida-F se encontram na Tabela 14, destacado em negrito o melhor valor de cada algoritmo em cada tipo de conjunto de dados, e sombreada está a média da Medida-F comparada na Figura 12.

Tabela 14 Métrica Medida-F dos algoritmos por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Anúncio	0,559	**0,612**	0,603	0,866	**0,882**	0,835	**0,909**	0,908	0,698
Dúvida	0,405	**0,432**	0,401	0,88	0,882	**0,889**	**0,925**	0,92	0,602
Escla.	0,245	0,276	**0,318**	0,82	0,835	**0,869**	**0,879**	0,875	0,52
Inter.	0,65	0,657	**0,697**	0,636	0,653	**0,668**	0,764	**0,773**	0,642
Outros	0,587	**0,612**	0,513	0,579	**0,61**	0,559	**0,725**	0,718	0,392
Média	0,489	**0,517**	0,506	0,756	**0,772**	0,764	**0,84**	0,838	0,57

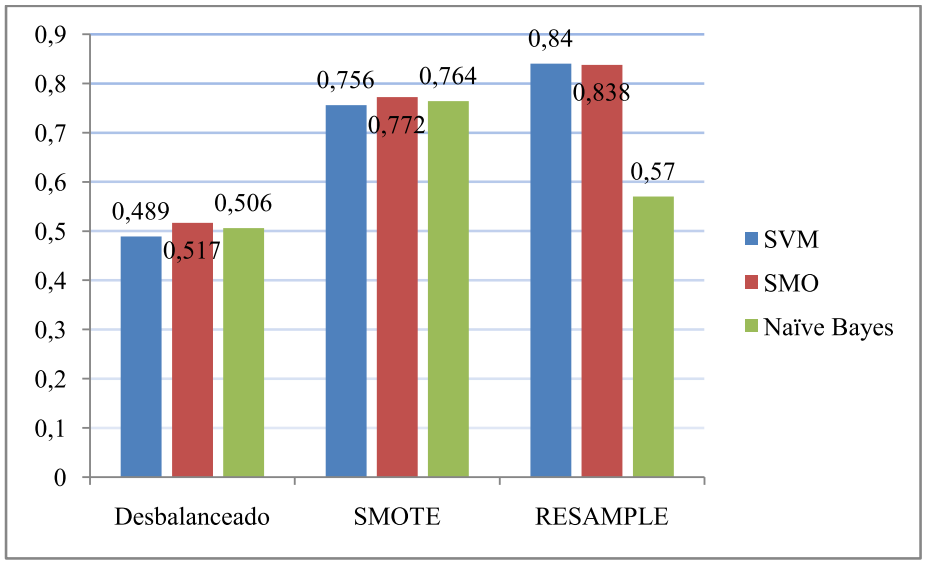

Figura 12 Média da Medida-F dos algoritmos por conjunto de dados

Para a métrica Medida-F o algoritmo SMO foi melhor no conjunto de dados desbalanceados e para o conjunto balanceado com SMOTE e o SVM para o conjunto balanceado com RESAMPLE.

A Tabela 15 apresenta sombreada a Acurácia dos modelos multi-classe, e destacado em negrito o melhor valor de Acurácia por tipo de conjunto de dados. Esta Acurácia esta comparada na Figura 13.

Tabela 15 Acurácia dos algoritmos por conjunto de dados

Algo.	*Dataset*	*Desbalanceado*		*SMOTE*		*RESAMPLE*	
	Acurácia	**Quant.**	**%**	**Quant.**	**%**	**Quant.**	**%**
SVM	Corretas	3181	56,8137	7720	75,5603	4710	**84,1222**
	Incorretas	2418	43,1863	2497	24,4397	889	15,8778
SMO	Corretas	3283	**58,6355**	7878	**77,1068**	4705	84,0329
	Incorretas	2316	41,3645	2339	22,8932	894	15,9671
NB	Corretas	3179	56,778	7616	74,5424	3236	57,796
	Incorretas	2420	43,222	2601	25,4576	2363	42,204

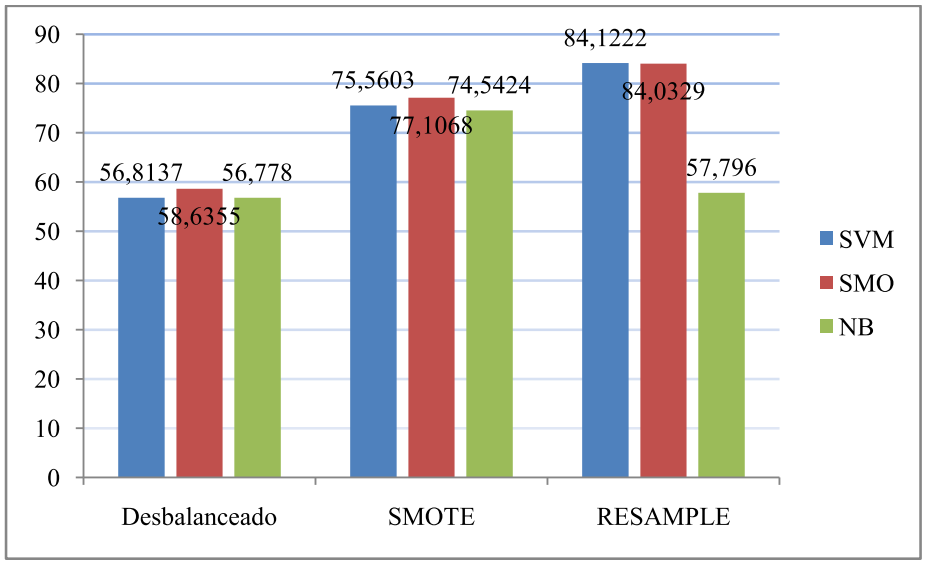

Figura 13 Acurácia dos algoritmos por conjunto de dados

Para a Acurácia o algoritmo SMO foi melhor no conjunto de dados desbalanceados e para o conjunto balanceado com SMOTE e o SVM para o conjunto balanceado com RESAMPLE.

4.3 Modelos de classificação binários

Um modelo de classificação binário é um classificador capaz de rotular um dado como pertencente ou não pertencente a uma classe, ou seja, classificar uma mensagem em pertencente ou não pertencente a um gênero. A Figura 14 ilustra um exemplo de funcionamento de dois modelos de classificação binário, um especializado em classificar o gênero Anúncio (A, NA) e um especializado em classificar o gênero Dúvida (D, ND).

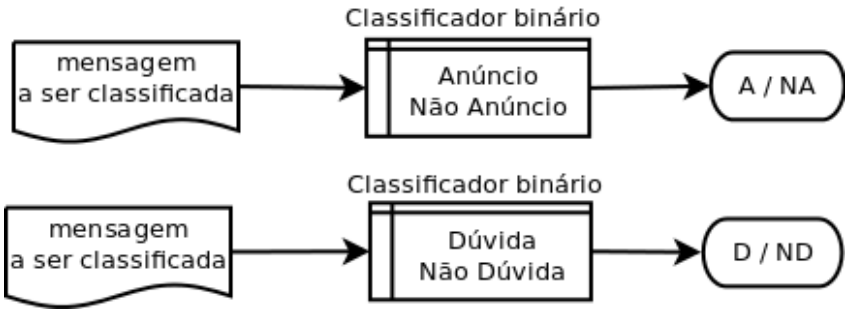

Figura 14 Exemplo de funcionamento de dois classificadores binários

4.3.1 Conjunto de dados de treino

Para treinar um algoritmo de classificação binário é necessário que o conjunto de dados de treino contenha duas classes. A partir dos dados coletados foram criados novos conjuntos adaptados para a indução de algoritmos de classificação multi-classe para a criação de modelos de classificação binário.Cada novo conjunto adaptado contém mensagens pertencentes a um gênero e mensagens pertencentes à união dos gêneros restantes sendo os conjuntos contendo Anúncio e Não Anúncio, Dúvida e Não Dúvida, Esclarecimento e Não Esclarecimento, Interpretação e Não Interpretação, Outros e Não Outros.

Por exemplo, considere A o conjunto de mensagens classificadas com o gênero Anúncio e Não Anúncio, sendo NA o conjunto de mensagens composto pela união dos outros gêneros Dúvida, Esclarecimento, Interpretação e Outros. O conjunto de dados de treino para a geração do modelo de classificação binário Anúncio é {A,NA}, de maneira análoga para as outras classes, resultando em cinco conjuntos de treino com duas classes, representados como {A,NA}, {D,ND}, {E, NE}, {I,NI}, {O, NO}.

A partir destes novos conjuntos contendo duas classes e que estão desbalanceados, foram aplicados os algoritmos SMOTE e RESAMPLE para a

criação de novos conjuntos de dados de treino balanceados. Os novos conjuntos de dados binários criados estão na Tabela 16, e nela contêm a quantidade de dados de cada classe e o percentual de aumento(↑) ou diminuição(↓) do novo conjunto em relação ao conjunto de dados desbalanceados.

Tabela 16 Distribuição dos dados de treino dos modelos binários

Dataset	*Desbalanceado*	*SMOTE*		*RESAMPLE*	
Gênero	**Quant.**	**Quant.**	**%**	**Quant.**	**%**
Anúncio	576	5184	↑ 800 %	2793	↑385%
Não Anúncio	5023	5023	---	2806	↓55,86%
Dúvida	407	5291	↑1200%	2793	↑586%
Não Dúvida	5192	5192	---	2806	↓54%
Esclarecimento	470	5170	↑ 1000%	2793	↑494%
Não Esclarecimento	5129	5129	---	2806	↓54,7%
Interpretação	2046	3682	↑80%	2793	↑36,51%
Não Interpretação	3553	3553	---	2806	↓78,97%
Outros	2100	3780	↑80%	2793	↑33%
Não Outros	3499	3499	---	2806	↓24,7%

4.3.2 Resultados dos modelos de classificação binário

Nesta seção estão os resultados comparativos dos modelos de classificação binários. Os resultados da métrica Precisão se encontram na Tabela 17, na qual está destacado em negrito o algoritmo que obteve o melhor valor para cada tipo de conjunto de dados. Nesta Tabela está sombreada a média geral dos modelos e ilustrada na Figura 15.

Tabela 17 Métricas Precisão, Revocação e Medida F, por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**P**	**R**	**M-F**	**P**	**R**	**M-F**	**P**	**R**	**M-F**
Anúncio	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Não Anúncio	0,473	**0,606**	0,427	0,938	0,959	**1**	0,929	**0,933**	0,834
Média	0,945	**0,95**	0,973	0,947	**0,949**	0,897	**0,992**	0,975	0,856
Dúvida	0,709	0,778	**0, 7**	0,942	**0,954**	0,948	**0,96**	0,954	0,845
Não Dúvida	0,343	**0,425**	0,286	0,947	0,963	**1**	**0,94**	0,936	0,758
Média	**0,95**	0,948	0,96	**0,951**	0,95	0,927	**0,994**	0,991	0,782
Escla.	0,646	0,686	**0,623**	0,949	0,956	**0,963**	**0,967**	0,963	0,77
Não Escla.	0,254	**0,311**	0,16	0,912	0,949	**1**	0,902	**0,905**	0,677
Média	0,937	0,933	**0,948**	**0,936**	0,934	0,916	**0,985**	0,968	0,728
Inter.	0,595	0,622	**0,554**	0,924	0,941	**0,958**	**0,943**	0,936	0,702
Não Inter.	0,657	**0,691**	0,608	0,809	0,84	**0,978**	**0,881**	0,873	0,728
Média	0,792	0,799	**0,862**	0,795	**0,801**	0,647	**0,9**	0,876	0,788
Outros	0,724	0,745	**0,735**	0,802	**0,82**	0,812	**0,89**	0,874	0,758
Não Outros	0,566	0,583	**0,627**	0,738	0,761	**0,926**	**0,791**	0,763	0,725
Média	0,756	**0,759**	0,726	0,746	**0,753**	0,678	**0,839**	0,813	0,627
Média Geral	0,661	**0,671**	0,676	0,742	0,757	**0,802**	**0,815**	0,788	0,676

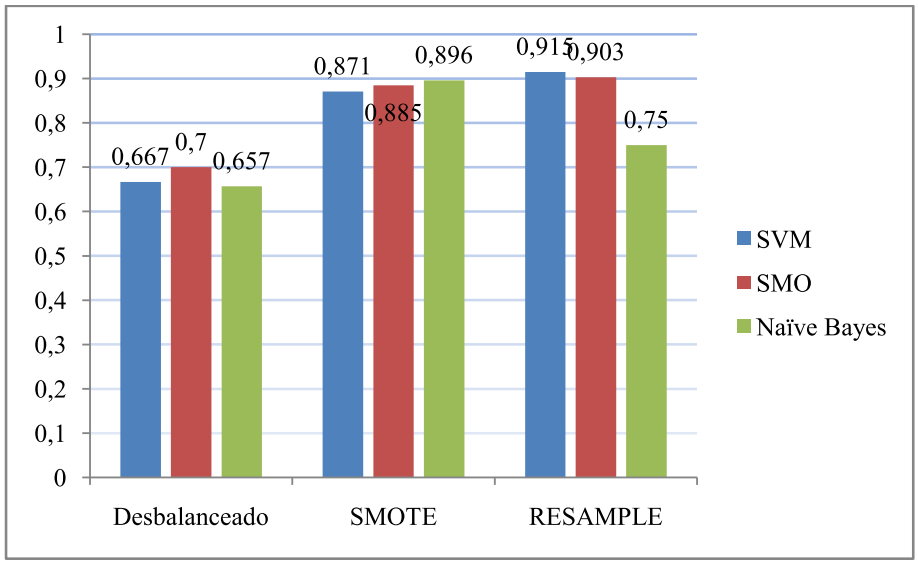

Figura 15 Média da Precisão dos modelos binários por conjunto de dados

Os resultados da métrica Revocação se encontram na Tabela 18, na qual está destacado em negrito o algoritmo que obteve o melhor valor para cada tipo de conjunto de dados. Nesta Tabela está sombreada a média geral dos modelos e ilustrada na Figura 16.

Tabela 18 Métrica Revocação dos modelos binários por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
A	0,524	0,561	**0,786**	0,949	**0,95**	0,889	**0,992**	0,976	0,86
NA	0,933	**0,958**	0,879	0,935	0,958	**1**	**0,925**	0,93	0,829
Média	0,728	**0,759**	0,832	0,942	**0,954**	0,944	**0,958**	0,953	0,844
D	0,369	0,329	**0,528**	**0,952**	0,95	0,923	0,995	**0,992**	0,79
ND	0,945	**0,965**	0,896	0,946	0,963	**1**	**0,937**	0,932	0,749
Média	0,657	**0,647**	0,712	0,949	0,956	**0,961**	**0,966**	**0,962**	0,77
E	0,328	0,26	**0,567**	**0,938**	0,934	0,909	**0,986**	0,971	0,76
NE	0,912	**0,947**	0,729	0,908	0,95	**1**	0,893	**0,898**	0,639
Média	0,62	**0,603**	0,648	0,923	0,942	**0,954**	**0,94**	0,934	0,7
I	0,631	0,634	**0,805**	**0,8**	0,798	0,479	**0,902**	0,866	0,811
NI	0,81	**0,837**	0,701	0,804	0,842	**0,989**	**0,879**	0,875	0,699
Média	0,72	**0,735**	0,753	0,802	**0,82**	0,734	**0,891**	0,87	0,755
O	**0,614**	0,61	0,479	**0,779**	0,776	0,583	**0,85**	0,828	0,521
NO	0,717	0,738	**0,829**	0,702	0,737	**0,95**	0,776	0,743	**0,803**
Média	0,665	0,674	**0,654**	0,74	0,756	**0,766**	**0,813**	0,786	0,662
Média Geral	0,678	0,683	0,719	0,871	0,885	**0,871**	**0,913**	**0,901**	0,746

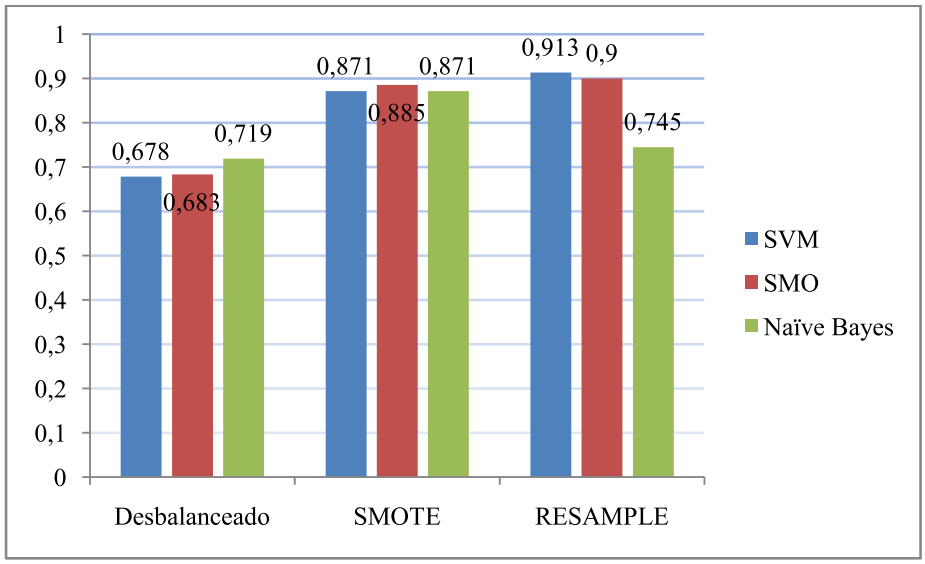

Figura 16 Média da Revocação dos modelos binários por conjunto de dados

Na Tabela 19 estão os resultados da métrica Medida-F e está sombreada a média geral de cada algoritmo em cada conjunto de dados e a comparação entre eles está destacada na Figura 17.

Tabela 19 Métrica Medida-F dos modelos binários por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
A	0,498	**0,583**	0,553	0,943	**0,954**	0,941	**0,96**	0,954	0,846
NA	0,939	**0,954**	0,923	0,941	**0,953**	0,946	**0,957**	0,952	0,842
Média	0,718	**0,768**	0,738	0,942	**0,953**	0,943	**0,958**	0,953	0,844
D	0,355	**0,371**	**0,371**	0,95	0,956	**0,96**	**0,967**	0,963	0,774
ND	0,947	**0,957**	0,927	0,948	0,956	**0,962**	**0,965**	0,961	0,765
Média	0,651	**0,664**	0,649	0,949	0,956	**0,961**	**0,966**	0,962	0,769
E	**0,286**	0,283	0,249	0,925	0,941	**0,952**	**0,942**	0,936	0,716
NE	0,924	**0,94**	0,824	0,922	0,942	**0,956**	**0,937**	0,932	0,681
Média	0,605	**0,611**	0,412	0,923	0,941	**0,954**	**0,939**	0,934	0,698
I	0,644	0,661	**0,693**	0,805	**0,818**	0,643	**0,892**	0,869	0,767
NI	0,801	**0,817**	0,773	0,8	**0,821**	0,782	**0,889**	0,871	0,741
Média	0,722	**0,739**	0,733	0,802	**0,819**	0,712	**0,89**	0,87	0,754
O	0,589	**0,596**	0,543	0,758	**0,769**	0,715	**0,819**	0,794	0,606
NO	0,736	0,749	**0,774**	0,723	0,745	**0,791**	**0,806**	0,777	0,704
Média	0,662	**0,672**	0,658	0,74	**0,757**	0,753	**0,812**	0,785	0,655
Média Geral	0,671	**0,69**	0,638	0,871	**0,885**	0,864	**0,913**	0,900	0,744

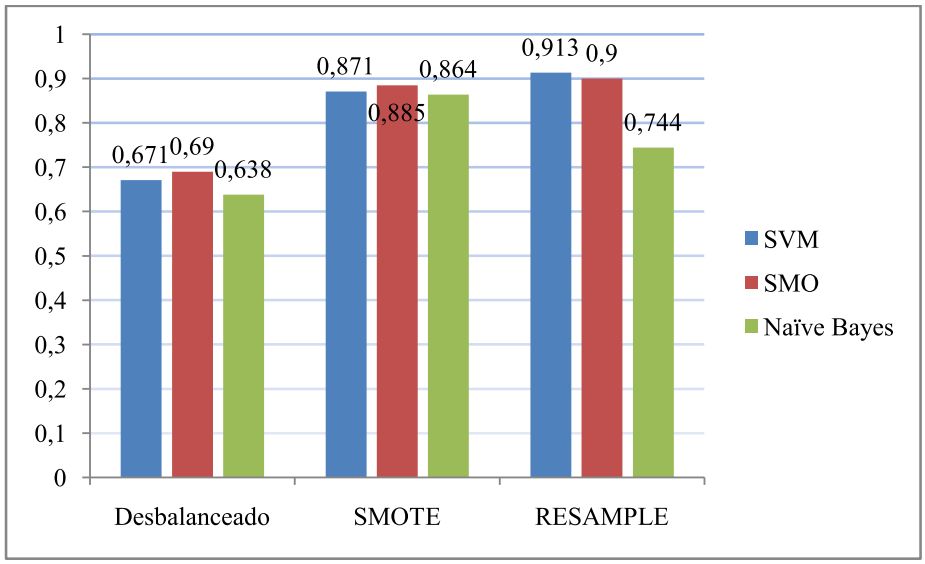

Figura 17 Média da Medida-F dos modelos binários por conjunto de dados

Os resultados da métrica Acurácia de cada algoritmo em cada conjunto de dados se encontram na Tabela 20, na qual está destacado em negrito o algoritmo que obteve o melhor valor para cada tipo de conjunto de dados de treino, e sombreada a média geral, que está comparada na Figura 18.

Tabela 20 Acurácia dos modelos binários por conjunto de dados

	Desbalanceado			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Anúncio	89,1	**91,73**	86,92	94,21	**95,39**	94,36	**95,83**	95,26	84,44
Dúvida	90,28	**91,89**	86,96	94,90	95,63	**96,12**	**96,58**	96,19	76,96
Escla.	86,28	**88,96**	71,51	92,31	94,15	**95,44**	**93,98**	93,42	69,95
Inter.	74,47	**76,28**	73,88	80,22	**81,94**	72,96	**89,05**	87,03	75,47
Outros	67,85	**69,03**	69,76	74,19	75,75	**75,93**	**81,30**	78,56	66,22
Média	81,60	**83,57**	77,82	87,17	**88,57**	86,96	**91,35**	90,09	74,61

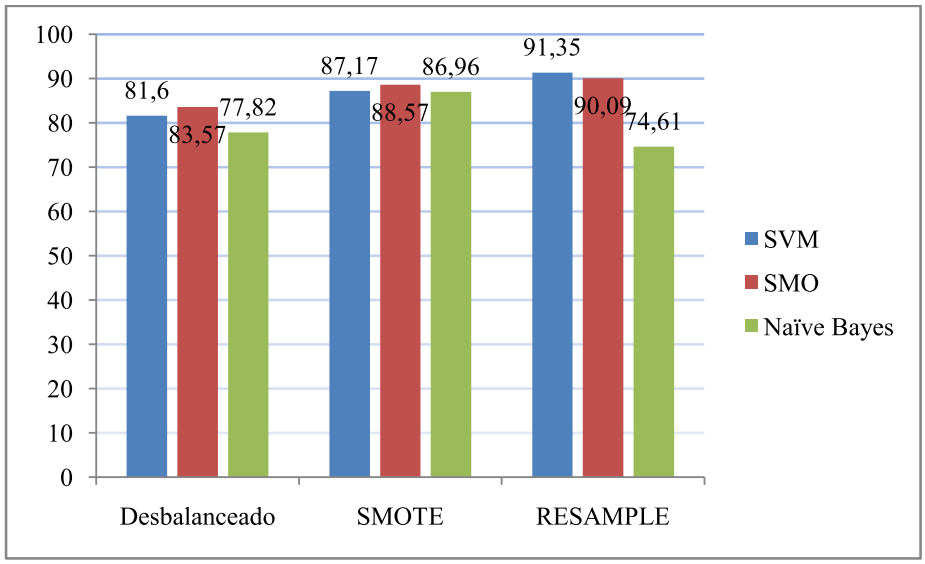

Figura 18 Média da Acurácia dos modelos binários por conjunto de dados

Em relação à média da Acurácia, o algoritmo SMO obteve melhores resultados no conjunto de dados desbalanceados e no balanceado com SMOTE. Já o SVM obteve melhores resultados com o conjunto de dados balanceados com RESAMPLE.

Os resultados deste capítulo demonstram o quão importante é o tratamento do desbalanceamento do conjunto de dados de treino para a indução de algoritmos de classificação de texto. Este tratamento pode ser feito na etapa de pré-processamento através do uso de algoritmos para criação e/ou remoção de dados, ou através da distribuição de modelos de classificação binários. Nos experimentos realizados, o tratamento do desbalanceamento de dados na etapa de pré-processamento obteve melhores resultados.

Os experimentos utilizaram três algoritmos com três diferentes bases de dados para a geração de modelos, utilizando a técnica 10-*fold cross-validation*. O resultado destes experimentos foi 9 modelos de classificação multi-classe (flat) e 45 modelos de classificação binários (especialistas). Estes modelos foram

utilizados em duas abordagens de classificação. Na próxima seção, estes modelos serão dispostos em três abordagens de classificação de mensagem.

5 ABORDAGENS DE DISTRIBUIÇÃO DE MODELOS DE CLASSIFICAÇÃO

Neste capítulo, os modelos gerados no capítulo anterior são distribuídos em três abordagens distintas de classificação multi-classe, uma sendo composta por um modelo de classificação multi-classe, uma abordagem de classificação em cascata utilizando modelos binários e uma abordagem que é proposta neste trabalho.

A abordagem proposta também é capaz de classificar multi-rótulo e seus resultados foram comparados a dois algoritmos adaptados para classificação multi-rótulo.

5.1 Configuração dos experimentos

Neste capítulo os resultados mostrados são referentes aos algoritmos modelos gerados no capítulo anterior. Para a avaliação da criação das abordagens foi utilizado o conjunto de dados de teste.

Os resultados avaliados nos experimentos multi-classe foram a Acurácia. Nos experimentos multi-rótulo foram avaliadas as métricas *Hamming Loss* e *One-Error* e os resultados da abordagem também foram comparados com o resultado dos algoritmos adaptados para classificação multi-rótulo,o RAkEL e o MLkNN.

5.1.1 Conjunto de dados de teste

Para avaliação das abordagens desenvolvidas foi utilizado o conjunto de dados de teste. As mensagens deste conjunto não foram utilizadas para a indução

dos algoritmos e construção dos modelos de classificação. Este conjunto está representado na Tabela 21.

Tabela 21 Conjunto de dados de teste

Conjunto	*Teste*	
Gêneros	**Quantidade**	**%**
Anúncio	143	15,85
Dúvida	191	21,18
Esclarecimento	192	21,29
Interpretação	199	22,06
Outros	177	19,62
Total	902	100

5.2 Modelos de classificação multi-classe

Um modelo de classificação multi-classe é um classificador capaz de classificar a mensagem em uma classe como descrito na seção 4.2. O resultado da avaliação da Acurácia dos modelos gerados com o conjunto de dados de teste está na Tabela 22 e na Figura 19.

Tabela 22 Acurácia dos modelos multi-classe com o conjunto de dados de teste

Algo.	*Dataset*	*Desbalanceado*		*SMOTE*		*RESAMPLE*	
	Acurácia	**Quant.**	**%**	**Quant.**	**%**	**Quant.**	**%**
SVM	Corretas	362	40,133	355	39,3569	392	43,4589
	Incorretas	540	59,867	547	69,643	510	56,541
SMO	Corretas	391	43,348	393	43,5698	403	44,6785
	Incorretas	511	56,652	509	56,4301	499	55,3215
NB	Corretas	517	**57,317**	463	**51,3303**	514	**56,9845**
	Incorretas	385	42,683	439	48,6696	388	43,0155

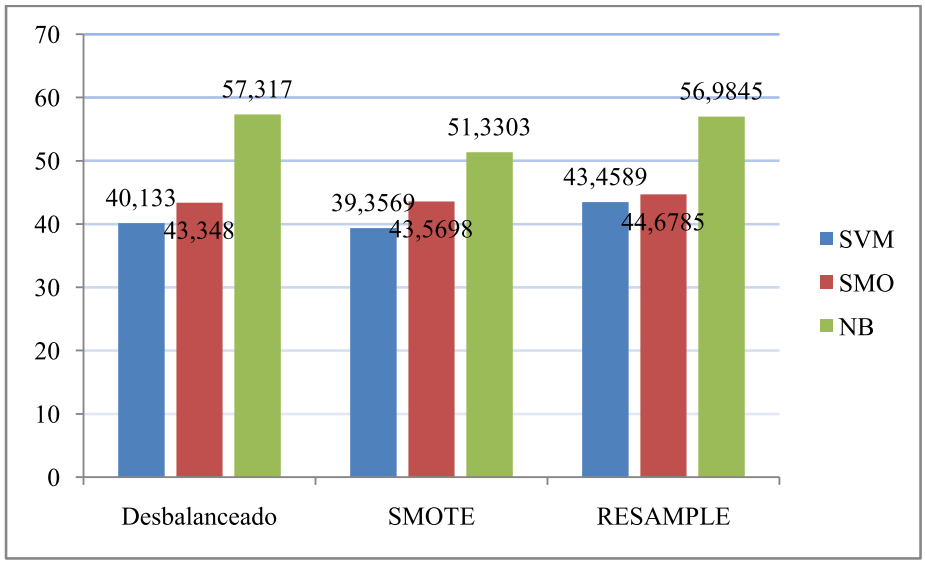

Figura 19 Acurácia dos modelos multi-classe com o conjunto de dados de teste

Os resultados da Acurácia desta avaliação demonstram que com o uso de um conjunto de dados de teste, que não foi utilizado para o treino dos algoritmos de classificação, não houve necessariamente melhora nos resultados. Com o algoritmo SMO houve aumento de 0,22% com o uso do SMOTE e de 1,3305% com o uso do RESAMPLE. Com o algoritmo SVM houve diminuição de 0,7761% com o uso do SMOTE e um aumento de 3,3259% com o uso do RESAMPLE. Com o algoritmo NB houve diminuição de 5,9867% com o uso de SMOTE e de 0,3325% com o uso do RESAMPLE.

Procurando encontrar a melhor abordagem de classificação multi-classe de gêneros de mensagens, os modelos binários de classificação foram combinados em duas abordagens distintas. Uma abordagem em cascata baseada em (LIN; HSIEH; CHUANG, 2009), em que a mensagem passa por um classificador binário por vez, seção 5.3; e uma abordagem em paralelo em que a mensagem é disposta a todos os classificadores binários ao mesmo tempo, seção 5.4.

5.3 Abordagem de distribuição em cascata dos modelos binários

No funcionamento da abordagem em cascata, a mensagem passa por um modelo binário de classificação de forma sequencial. Caso o modelo binário classifique a mensagem em uma classe, a classificação da mensagem termina e o resultado da classificação é a classe do modelo binário. Caso contrário, a mensagem é levada ao próximo modelo binário e assim por diante, até que algum modelo classifique a mensagem como pertencente a sua classe. Caso nenhum dos modelos binários classifique a mensagem, então a mensagem é classificada como pertencente ao gênero Outros. A Figura 20 ilustra o funcionamento desta abordagem em cascata desenvolvida neste experimento.

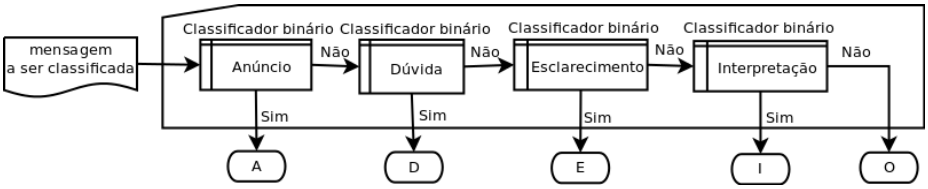

Figura 20 Abordagem de classificação em cascata utilizando modelos binários

Para a construção da abordagem de classificação em cascata, os modelos binários foram dispostos na ordem do modelo que obteve o maior para o menor resultado na medida Acurácia; sendo a ordem dos modelos Anúncio, Dúvida, Esclarecimento e Interpretação.

5.3.1 Avaliação da distribuição em cascata dos modelos binários

Nesta seção estão os resultados da abordagem de distribuição em cascata dos modelos binários. Os modelos binários foram dispostos em ordem de maior

para menor Acurácia, sendo a seguinte ordem: Anúncio, Dúvida, Esclarecimento e Interpretação. O resultado desta avaliação será comparado com a abordagem proposta de classificação multi-classe.

Esta avaliação foi feita com a utilização do conjunto de dados de teste. Os resultados da métrica Precisão se encontram na Tabela 23, destacado em negrito o melhor valor de cada algoritmo em cada tipo de conjunto de dados, e sombreada está a média da Precisão, comparada na Figura 21.

Tabela 23 Métrica Precisão dos modelos por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Anúncio	0,467	**0,504**	0,415	0,445	0,508	**0,85**	0,392	**0,448**	0,359
Dúvida	0,513	**0,652**	0,507	0,513	**0,652**	0	0,429	**0,529**	0,346
Escla.	0,262	**0,379**	0,192	0,261	**0,4**	0	0,316	**0,364**	0,178
Inter.	0,385	0,462	**0,482**	0,418	0,525	**1**	0,449	**0,529**	0,388
Outros	0,257	0,268	**0,436**	0,258	**0,266**	0,199	0,277	0,298	**0,47**
Média	0,376	**0,453**	0,406	0,379	**0,47**	0,409	0,372	**0,433**	0,348

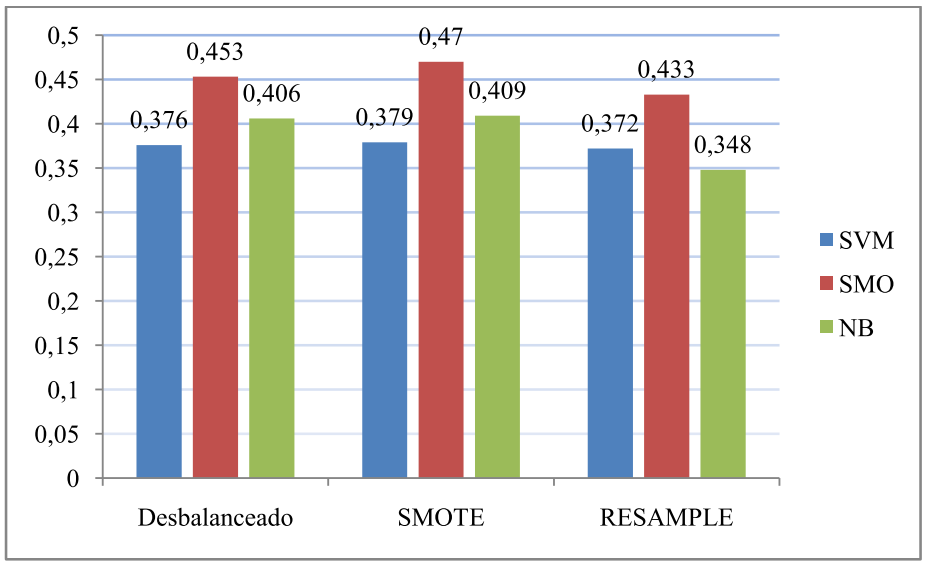

Figura 21 Média da métrica Precisão dos modelos por conjunto de dados

Os resultados da métrica Revocação se encontram na Tabela 24, destacado em negrito o melhor valor de cada algoritmo em cada tipo de conjunto de dados, e sombreada está a média da Revocação, comparada na Figura 22.

Tabela 24 Métrica Revocação dos modelos por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Anúncio	0,454	0,412	0,72	0,426	0,412	0,125	0,51	0,545	0,748
Dúvida	0,204	0,235	0,34	0,198	0,235	0	0,35	0,424	0,45
Escla.	0,109	0,114	0,177	0,114	0,114	0	0,192	0,203	0,187
Inter.	0,346	0,376	0,351	0,371	0,422	0,01	0,381	0,452	0,201
Outros	0,621	0,745	0,502	0,627	0,74	0,988	0,429	0,502	0,135
Média	0,346	0,376	0,418	0,347	0,384	0,222	0,372	0,425	0,344

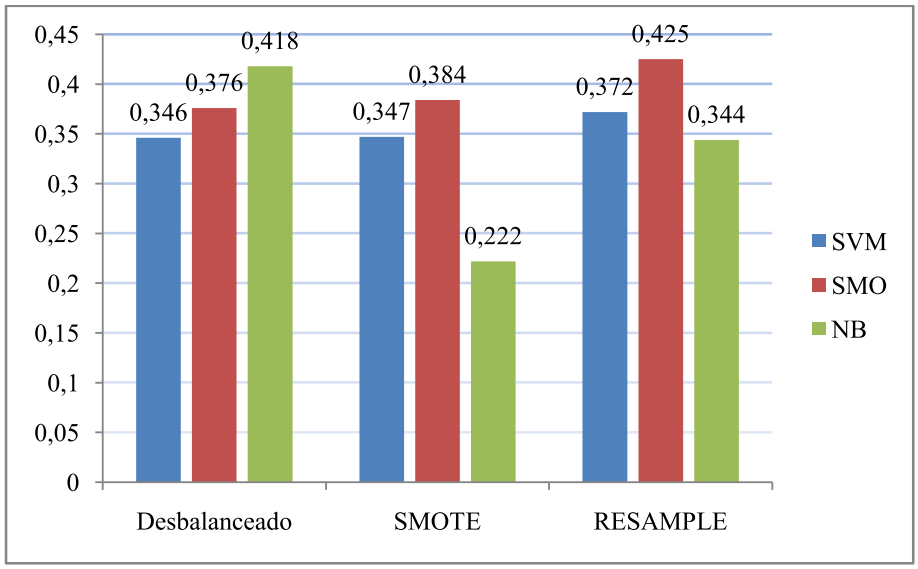

Figura 22 Média da métrica Revocação dos modelos por conjunto de dados

Os resultados da métrica Medida-F se encontram na Tabela 25, destacado em negrito o melhor valor de cada algoritmo em cada tipo de conjunto de dados, e sombreada está a média da Medida-F, comparada na Figura 23.

Tabela 25 Métrica Medida-F dos modelos por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Anúncio	0,46	0,453	0,526	0,435	0,684	0,217	0,443	0,491	0,485
Dúvida	0,291	0,345	0,407	0,285	0,692	0	0,385	0,47	0,391
Escla.	0,153	0,175	0,184	0,158	0,795	0	0,238	0,26	0,182
Inter.	0,364	0,414	0,406	0,393	0,678	0,181	0,412	0,487	0,264
Outros	0,363	0,392	0,466	0,365	0,665	0,331	0,336	0,373	0,209
Média	0,326	0,355	0,397	0,327	0,702	0,145	0,362	0,416	0,306

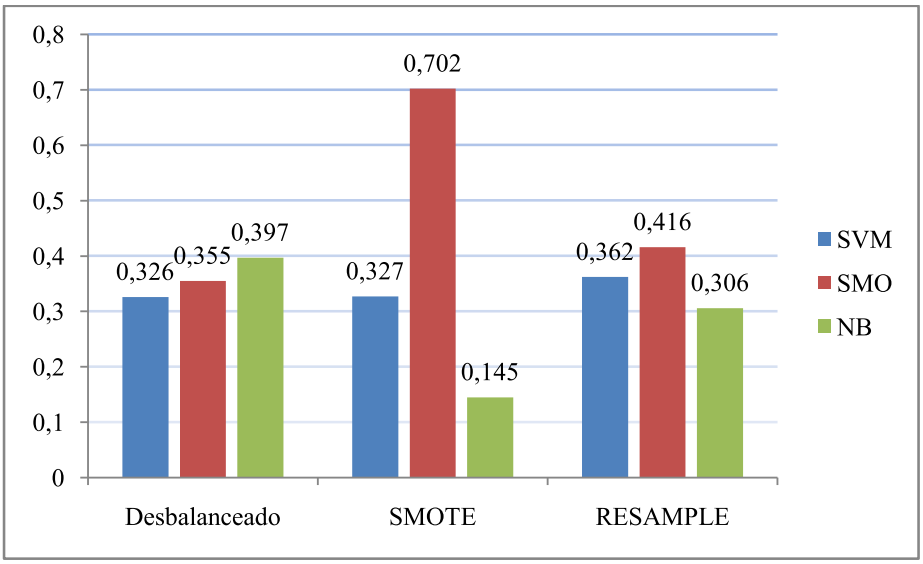

Figura 23 Média da métrica Medida-F dos modelos por conjunto de dados

Analisando a Figura 23, a comparação da Medida-F, podemos dizer que para o algoritmo NB o uso de algoritmos para balanceamento dos dados piorou o resultado. Para o algoritmo SVM, o uso tanto de SMOTE quanto de

RESAMPLE apresentou uma melhora de 3,6%. Para o algoritmo SMO, houve uma grande melhora com o balanceamento, sendo de 34,5% com SMOTE e 6,1% com RESAMPLE.

A Tabela 26 exibe em negrito a Acurácia do melhor modelo gerado por cada tipo de conjunto de treino, e ilustrada na Figura 24.

Tabela 26 Acurácia da abordagem em cascata

Algo.	*Dataset*	*Desbalanceado*		*SMOTE*		*RESAMPLE*	
	Acurácia	**Quant.**	**%**	**Quant.**	**%**	**Quant.**	**%**
SVM	Corretas	304	33,7029	306	33,9246	329	36,4745
	Incorretas	598	66,2971	596	66,0753	573	63,5254
SMO	Corretas	333	36,918	341	**37,8048**	377	**41,796**
	Incorretas	569	63,082	561	62,1951	525	58,204
NB	Corretas	361	**40,022**	195	21,6186	293	32,4833
	Incorretas	541	59,978	707	78,3813	609	67,5164

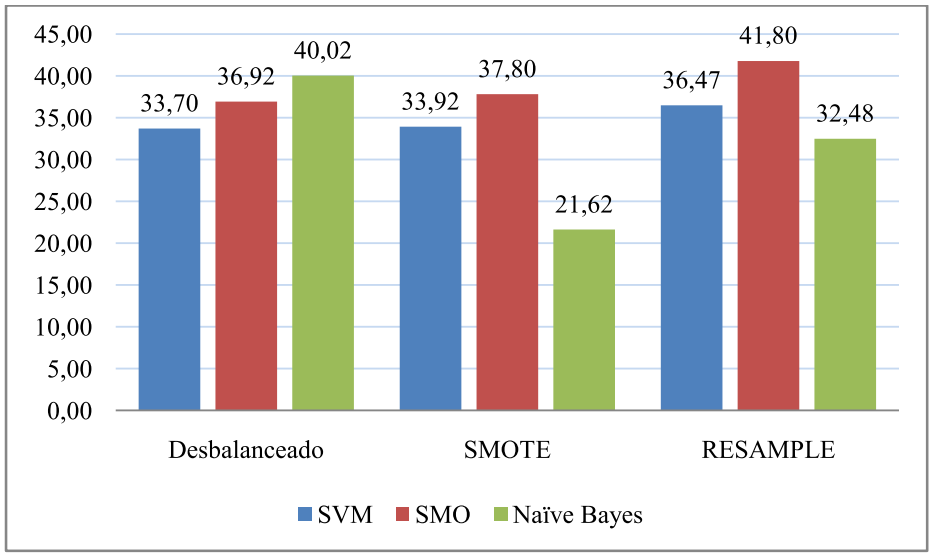

Figura 24 Acurácia da abordagem em cascata

A Acurácia, assim como a Medida-F, demonstra que com o algoritmo NB houve uma piora nos resultados. Com o algoritmo SVM, houve uma ligeira

melhora com os dois algoritmos de balanceamento dos dados. Para o algoritmo SMO, houve uma melhora de 0,88% utilizando o SMOTE e de 4,88% com RESAMPLE.

Esta abordagem de classificação em cascata é muito sensível aos erros cometidos em qualquer nível de classificação, pois ela pode errar, por exemplo, no primeiro nível e assim nenhum dos outros modelos de classificação tem a oportunidade de classificar corretamente a mensagem.

5.4 Abordagem de distribuição em paralelo dos modelos binários

Na abordagem em paralelo, os modelos de classificação binários são agrupados formando um conjunto de classificação com um único resultado. Uma mensagem é classificada por todos os modelos binários ao mesmo tempo e o resultado da classificação é a união da classificação de cada modelo binário como pertencente ou não à classe em que foi treinado e especializado. Esta união de resultados de classificação pode resultar na mensagem sendo classificada em um gênero, ou em mais de um gênero, conforme ilustrado nas Figuras 25 e 26 respectivamente.

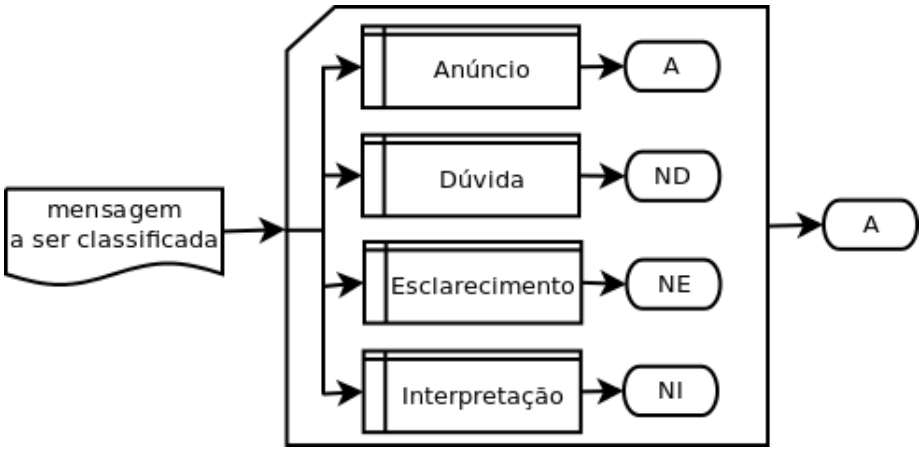

Figura 25 Abordagem em paralelo com resultado multi-classe

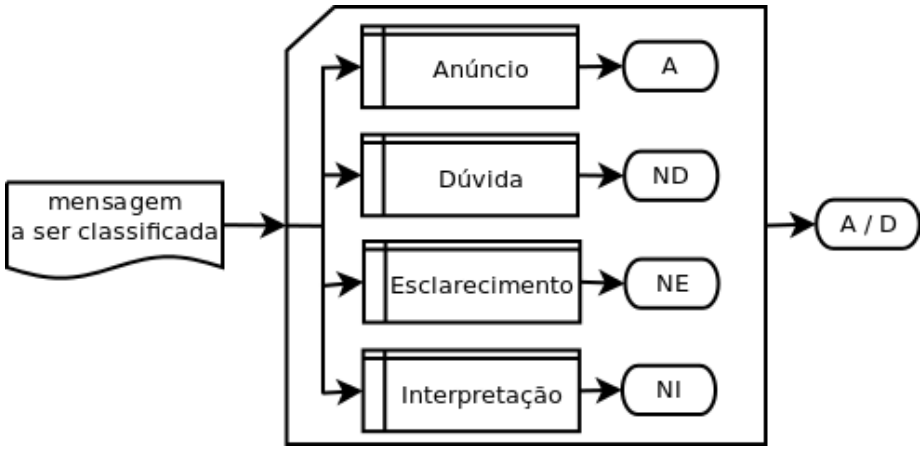

Figura 26 Abordagem em paralelo com resultado multi-rótulo

No início do trabalho o intuito era classificar a mensagem em um gênero. A abordagem de distribuição do conjunto de modelos binários em paralelo resultou na classificação da mensagem em um ou mais gêneros. Nos resultados com mais de um gênero houve a necessidade de escolher qual destes gêneros estava correto ou o que melhor representasse a mensagem, levando em consideração que cada mensagem possuía apenas um gênero. A partir do

resultado de classificação multi-rótulo, foi idealizado e utilizado o modelo classificador multi-classe do experimento da seção 5.2 como "oráculo", com a finalidade de escolher dentre os gêneros classificados pela abordagem em paralelo, aquele que mais representa a mensagem.

Portanto, a abordagem desenvolvida é composta por dois classificadores, um classificador formado por um conjunto de modelos binários e um classificador formado por um modelo multi-classe. A Figura 27 ilustra os classificadores que compõem esta abordagem.

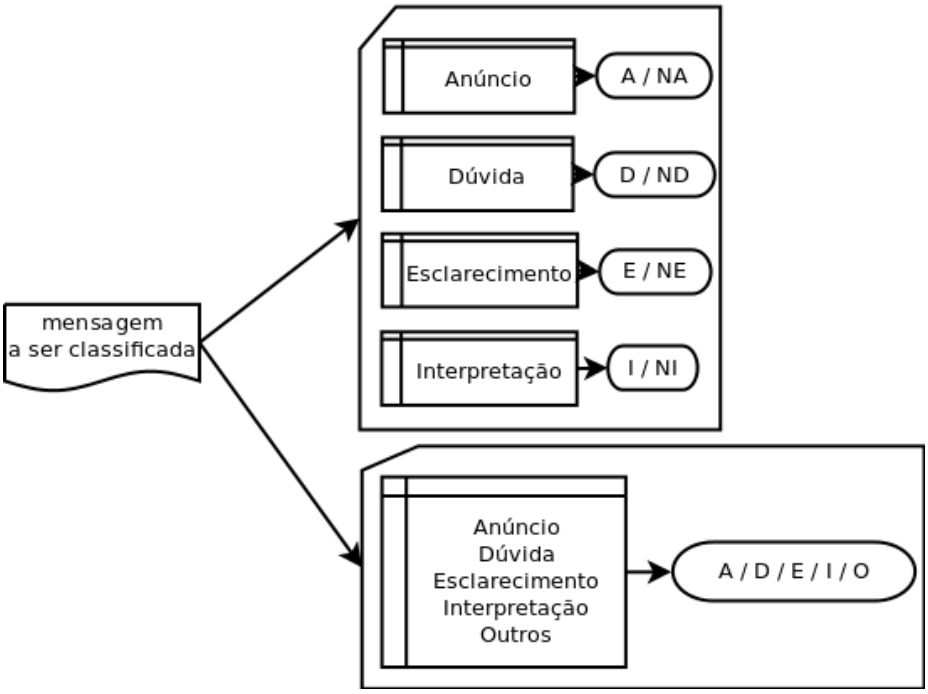

Figura 27 Abordagem proposta de classificação de gêneros de mensagem

Nesta abordagem proposta de classificação de gêneros da mensagem, a mensagem a ser classificada é entregue simultaneamente aos dois classificadores que compõem a abordagem. A união da classificação do modelo multi-classe e

de cada modelo binário é o resultado da classificação da mensagem. A abordagem proposta pode ser utilizada como multi-classe, que classifica a mensagem em apenas um gênero e está descrita na seção 5.4.1, ou como multi-rótulo, que classifica a mensagem em um ou mais gêneros e está descrita na seção 5.5.1.

5.4.1 Abordagem de classificação multi-classe de gêneros de mensagens

Nesta abordagem de classificação multi-classe, a mensagem é classificada somente em um dos cinco gêneros. Esta abordagem é dividida em duas etapas sequenciais, sendo que a classificação pode finalizar na primeira etapa e pular a segunda. Na primeira etapa a mensagem é classificada somente pelo conjunto de modelos binários e se todos os modelos classificarem a mensagem corretamente, a classificação finaliza nesta etapa. Na segunda etapa é utilizado o classificador multi-classe para decidir sobre o resultado da classificação do conjunto de modelos binários na primeira etapa, sendo esta segunda etapa dividida em dois casos. No primeiro caso, o gênero classificado pelo classificador multi-classe está inserido no conjunto de gêneros classificado pelo conjunto de modelos binários e no segundo caso, o gênero não está inserido. Estas etapas e casos estão descritos detalhadamente a seguir.

Primeira Etapa:

Na primeira etapa a mensagem é classificada pelo classificador formado pelo conjunto de modelos binários. O resultado desta classificação pode ser um ou mais gêneros, veja nas Figuras 25 e 26, respectivamente. Com o resultado sendo apenas um gênero, a classificação desta abordagem termina na primeira etapa.

Para o resultado da classificação da mensagem ser apenas um gênero, um dos modelos binários do conjunto deve classificar a mensagem como pertencente à sua classe e todos os outros modelos binários devem classificar a mensagem como não pertencente às suas respectivas classes. Por exemplo, para a mensagem ser classificada em Anúncio na primeira etapa, o resultado de cada modelo do conjunto de modelos binários deve ser: o modelo Anúncio classificar como Anúncio, o modelo Dúvida classificar como Não Dúvida, o modelo Esclarecimento classificar como Não Esclarecimento e o modelo Interpretação classificar como Não Interpretação. Com este resultado dos modelos de classificação binários, a abordagem termina nesta etapa e a mensagem é classificada como gênero Anúncio. A combinação de resultados para que a abordagem termine na primeira etapa está indicada em cada linha da Tabela 27.

Tabela 27 Combinação de resultados para finalizar na primeira etapa

MODELOS DE CLASSIFICAÇÃO				*RESULTADO*
Anúncio	**Dúvida**	**Esclarecimento**	**Interpretação**	**Gênero Classificado**
Anúncio	Não Dúvida	Não Esclarecimento	Não Interpretação	Anúncio
Não Anúncio	Dúvida	Não Esclarecimento	Não Interpretação	Dúvida
Não Anúncio	Não Dúvida	Esclarecimento	Não Interpretação	Esclarecimento
Não Anúncio	Não Dúvida	Não Esclarecimento	Interpretação	Interpretação
Não Anúncio	Não Dúvida	Não Esclarecimento	Não Interpretação	Outros

Segunda etapa:

A segunda etapa ocorre no caso de mais de um modelo binário classificar a mensagem como pertencente à classe em que foi treinado. Neste caso o classificador multi-classe é utilizado como "oráculo" a fim de decidir a qual classe a mensagem pertence.

A abordagem de classificação desenvolvida é composta por dois classificadores, um capaz de rotular a mensagem em somente um gênero, multi-classe; e outro capaz de rotular a mensagem em um ou mais gêneros, multi-rótulo. As classes rotuladas por estes dois classificadores podem ser distintas, sendo que a classe rotulada pelo classificador multi-classe pode ou não estar inserida no conjunto de classes rotuladas pelo classificador multi-rótulo. Esta segunda etapa é dividida nestes dois casos. O primeiro caso é quando o gênero do classificador multi-classe está inserido no conjunto de gêneros classificados pelo classificador multi-rótulo, e o segundo caso é quando o gênero não está inserido.

Segunda etapa: 1- Primeiro caso

O primeiro caso é quando o gênero rotulado pelo classificador multi-classe estiver contido no conjunto de gêneros rotulados pelos modelos binários e assim, este gênero em comum é o resultado da classificação da mensagem. Veja como exemplo na Figura 28 a classificação resultante sendo o gênero Anúncio.

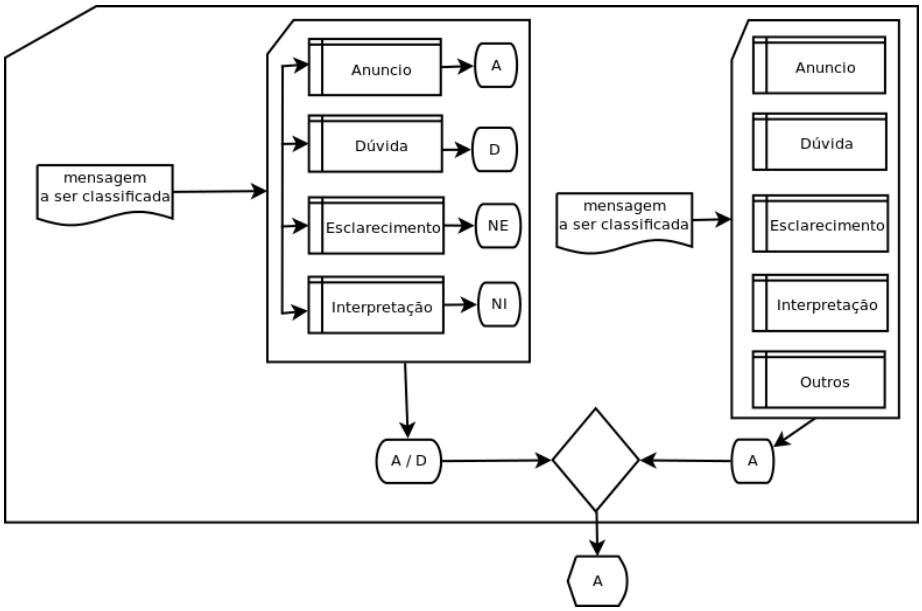

Figura 28 Abordagem proposta de classificação multi-classe classificando uma mensagem com o gênero Anúncio

Segunda etapa: 2- Segundo caso

O segundo caso é quando o gênero classificado pelo modelo de classificação multi-classe não está contido no conjunto de gêneros classificados pelos modelos binários de classificação. Neste caso, o resultado da abordagem de classificação é o gênero classificado pelo classificador multi-classe. Esta decisão foi tomada, pois foi decidido que todos os gêneros classificados pelos modelos binários estão corretos e não poderíamos escolher ao acaso, e nem escolher aquele classificador binário cujo modelo possui melhores resultados nas medidas analisadas, pois não foram considerados nenhum tipo de ranking. A Figura 29 ilustra um exemplo deste caso, sendo o resultado da classificação o gênero Esclarecimento.

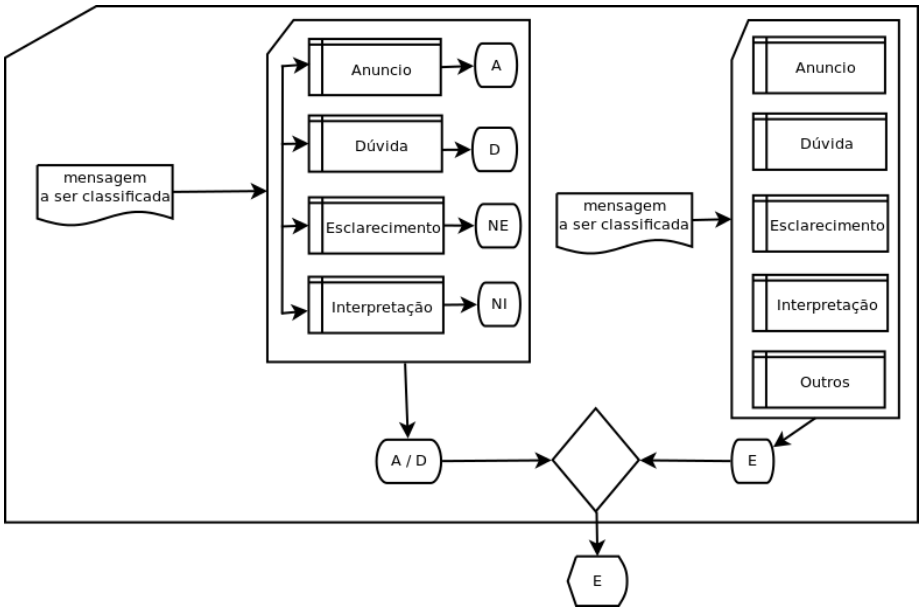

Figura 29 Abordagem proposta de classificação multi-classe classificando uma mensagem com o gênero Esclarecimento

Para a avaliação desta abordagem, que classifica a mensagem em somente um gênero, o gênero considerado das mensagens do conjunto de dados de teste foi o gênero classificado inicialmente pelos especialistas, aquele considerado como classe principal da mensagem.

5.4.2 Resultados de classificação multi-classe de gêneros de mensagens

Os resultados da métrica Precisão se encontram na Tabela 28, destacado em negrito o melhor valor de cada algoritmo em cada tipo de conjunto de dados, e sombreada está a média da Precisão, comparada na Figura 30.

Tabela 28 Métrica Precisão dos algoritmos por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Anúncio	0,462	0,468	**0,564**	0,441	0,472	**0,511**	0,479	0,496	**0,584**
Dúvida	0,574	**0,614**	0,535	0,563	**0,62**	0	0,475	0,563	**0,648**
Escla.	0,4	0,42	**0,472**	0,386	**0,437**	0,427	0,352	0,401	**0,444**
Inter.	0,366	0,437	**0,656**	0,415	0,447	**0,664**	0,487	0,521	**0,642**
Outros	0,329	0,347	**0,494**	0,324	0,347	**0,433**	0,366	0,398	**0,565**
Média	0,426	0,457	**0,544**	0,425	**0,464**	0,407	0,431	0,475	**0,576**

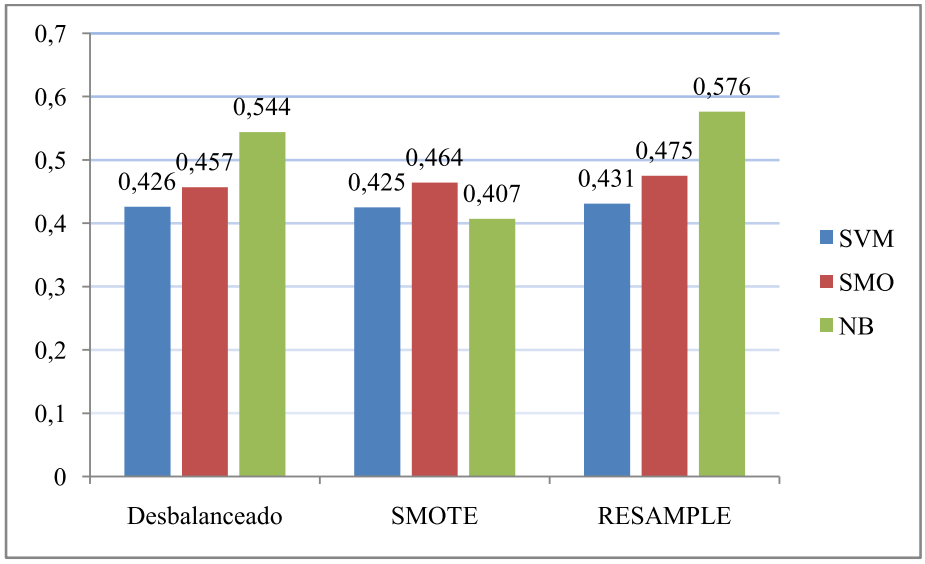

Figura 30 Média da métrica Precisão dos algoritmos por conjunto de dados

Os resultados da métrica Revocação se encontram na Tabela 29, destacado em negrito o melhor valor de cada algoritmo em cada tipo de conjunto de dados, e sombreada está a média da Revocação, comparada na Figura 31.

Tabela 29 Métrica Revocação dos algoritmos por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Anúncio	0,51	0,468	**0,615**	0,475	0,475	**0,608**	0,489	0,531	**0,58**
Dúvida	0,324	0,350	**0,429**	0,303	**0,35**	0	0,403	0,486	**0,612**
Escla.	0,26	0,218	**0,395**	0,265	0,218	**0,447**	0,291	0,328	**0,437**
Inter.	0,537	0,507	**0,864**	0,542	0,532	**0,864**	0,577	0,603	**0,843**
Outros	0,463	**0,627**	0,474	0,463	0,621	**0,666**	0,412	**0,435**	0,146
Média	0,418	0,434	0,555	0,409	0,439	**0,517**	0,434	0,476	**0,523**

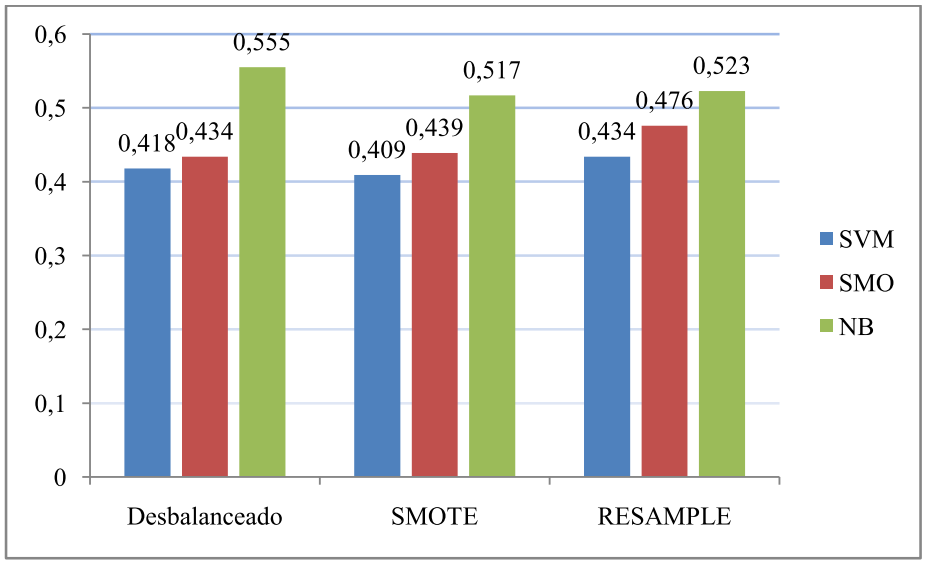

Figura 31 Média da Métrica Revocação dos algoritmos por conjunto de dados

Os resultados da métrica Medida-F se encontram na Tabela 30, destacado em negrito o melhor valor de cada algoritmo em cada tipo de conjunto de dados, e sombreada está a média da Precisão, comparada na Figura 32.

Tabela 30 Métrica Medida-F dos algoritmos por conjunto de dados

Dataset	*Desbalanceado*			*SMOTE*			*RESAMPLE*		
Gênero	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**	**SVM**	**SMO**	**NB**
Anúncio	0,484	0,468	**0,588**	0,457	0,473	**0,555**	0,483	0,512	**0,581**
Dúvida	0,414	0,445	**0,476**	0,393	**0,447**	0	0,436	0,521	**0,629**
Escla.	0,315	0,287	**0,43**	0,314	0,29	**0,436**	0,318	0,36	**0,44**
Inter.	0,435	0,469	**0,745**	0,47	0,485	**0,75**	0,528	0,559	**0,728**
Outros	0,384	0,446	**0,483**	0,381	0,445	**0,524**	0,387	**0,415**	0,232
Média	0,406	0,423	**0,544**	0,403	0,428	**0,453**	0,430	0,473	**0,522**

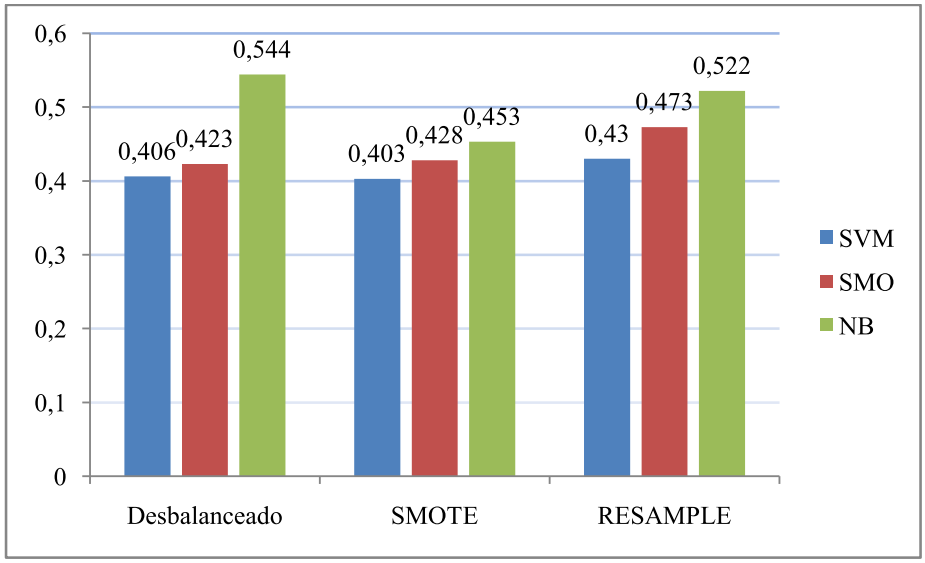

Figura 32 Média da métrica Medida-F dos algoritmos por conjunto de dados

A Tabela 31 exibe destacado em negrito a Acurácia de cada algoritmo em cada conjunto de dado de treino e a Figura 33 ilustra a Acurácia.

Tabela 31 Acurácia da classificação multi-classe

Algo.	*Dataset*	*Desbalanceado*		*SMOTE*		*RESAMPLE*	
	Acurácia	**Quant.**	**%**	**Quant.**	**%**	**Quant.**	**%**
SVM	Corretas	374	41,4634	367	40,6873	391	43,3481
	Incorretas	528	58,5366	535	59,3126	511	56,6518
SMO	Corretas	388	43,0155	393	43,5698	429	47,561
	Incorretas	514	56,9845	509	56,4301	473	52,439
NB	Corretas	502	**55,654**	463	**51,3303**	472	**52,3282**
	Incorretas	400	44,346	439	48,6696	430	47,6718

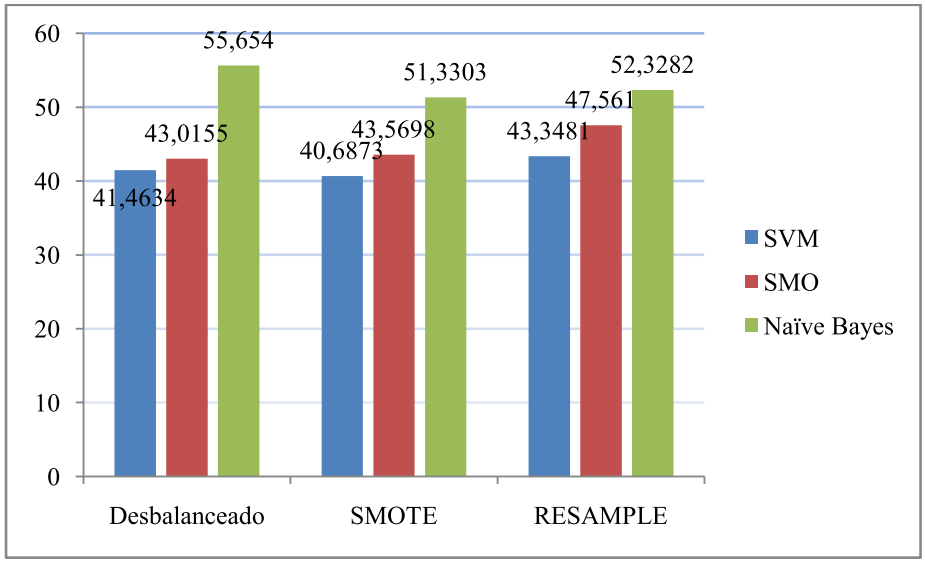

Figura 33 Acurácia da abordagem multi-classe por algoritmo e conjunto de dados

Observando o valor da Medida-F na Figura 32 e a Acurácia na Figura 33, os resultados do algoritmo Naïve Bayes foram piores com o uso dos algoritmos de desbalanceamento dos dados. Em relação à métrica Medida-F, os resultados do algoritmo SVM melhoraram com o uso do SMOTE e RESAMPLE, e em relação à Acurácia, o SVM melhorou com o uso do RESAMPLE mas piorou com o uso do SMOTE. Os resultados da Medida-F

para o algoritmo SMO melhoraram com o uso do SMOTE em 2,8% e com o uso do RESAMPLE melhorou em 5%, e na Acurácia, um aumento de 0,5543% e com o uso de SMOTE e de 4,5455% com o uso do RESAMPLE.

Os resultados da abordagem proposta de classificação multi-classe utilizando modelos gerados pelo algoritmo SVM, tanto com dados desbalanceados quanto com dados balanceados com o algoritmo SMOTE, foi melhor que um modelo gerado pelo algoritmo SVM e também da abordagem de distribuição de modelos em cascata. Utilizando o algoritmo SVM, o resultado da Acurácia dos modelos gerados com dados desbalanceados da abordagem proposta foi de 41,4634% contra 40,133% do modelo gerado pelo algoritmo SVM e 33,7029% da abordagem em cascata. O resultado dos modelos gerados com dados balanceados com SMOTE da abordagem proposta foi de 40,6873% contra 39,3569% do modelo gerado pelo algoritmo SVM e 33,9246% da abordagem em cascata. A abordagem proposta composta por modelos gerados pelo algoritmo SMO com o conjunto de dados balanceados com RESAMPLE foi de 47,561% contra 44,6785% do modelo gerado pelo algoritmo SMO e 41,796% da abordagem em cascata.

5.4.3 Resultados da classificação multi-rótulo de gêneros de mensagens

Uma mensagem de texto pode conter mais de um assunto, referenciar a diversas pessoas ou lugares, e cada parte do texto pode conter um sentimento diferente. Esta mensagem de texto sendo disponibilizada a diferentes especialistas humanos poderia receber diferentes gêneros. Então, em uma mensagem que contenha mais de uma classe distinta, é interessante que classificadores possam identificar todas as várias classes. Agora considere a Figura 29, cada um dos dois classificadores que compõem a abordagem desenvolvida classificou a mensagem com gêneros distintos; um total de três

gêneros. Isto foi possível pois a abordagem é composta de dois classificadores, um modelo de classificação multi-classe e um conjunto de modelos binários. O resultado do conjunto de modelos binários de classificação é a união dos resultados de cada modelo; e este resultado é formado por uma das combinações de um conjunto formado por cinco gêneros {A, D, E, I, O}, como por exemplo: {(A,D) , (A,E), (A,D,E), …, (I,O) ,(O)}, observe a Tabela 6.

O funcionamento da abordagem de classificação multi-rótulo acontece da seguinte maneira. A mensagem é disposta simultaneamente para os dois classificadores que compõem a abordagem e a união dos gêneros resultante dos dois classificadores é o resultado da classificação de uma mensagem. Como exemplo, na Figura 29 a mensagem seria classificada como pertencente aos gêneros Anúncio, Dúvida e Esclarecimento.

Para avaliar esta abordagem de classificação multi-rótulo, as mensagens do conjunto de teste foram classificadas por especialistas em um ou mais gêneros, resultando no conjunto da Tabela 6. Assim foi possível calcular o resultado de métricas de classificação multi-rótulo como a *Hamming Loss* e o *One-error*. Além destas métricas, as mensagens do conjunto de teste foram utilizadas como conjunto de treino (*golden-set* multi-rótulo) para algoritmos adaptados para a classificação multi-rótulo. Os modelos de classificação gerados por algoritmos de classificação multi-rótulo também foram avaliados pelas métricas *Hamming Loss* e *One-error*, e seus resultados foram comparados com os resultados da abordagem proposta.

Foram utilizados dois diferentes algoritmos, sendo o MLkNN (ZHANG; ZHOU, 2007) e o RAkEL (TSOUMAKAS; KATAKIS; VLAHAVAS, 2011).Os algoritmos de classificação multi-rótulo utilizados a fim de comparação foram validados com a técnica 10-*fold cross-validation*, e assim sendo, os resultados de suas métricas estão demonstrados por sua média, havendo uma variância no

menor/maior valor das métricas. Os valores das métricas observadas estão na Tabela 32.

Tabela 32 Resultado da abordagem multi-rótulo

Modelos	**Algoritmos**	***HammingLoss***	***One-error***
Desbalanceados	SVM	0,2738	0,6374
	SMO	0,2687	0,6496
	NB	**0,2676**	**0,3425**
SMOTE	SVM	0,2764	**0,6385**
	SMO	**0,2691**	0,6419
	NB	0,3068	0,9113
RESAMPLE	SVM	**0,2082**	0,5177
	SMO	0,2359	0,4911
	NB	0,3110	**0,2427**
Algoritmo Transformação	MLkNN	0,3152±0,0201	0,4539±0,0429
	RAkEL	0,2819±0,0234	0,4028±0,0623

Os resultados da Tabela 32 demonstram que o algoritmo de balanceamento SMOTE não foi interessante, pois a maioria dos resultados foi pior após a aplicação deste algoritmo. Com o algoritmo RESAMPLE, houve uma melhora na métrica *Hamming Loss* e uma melhora significativa da métrica *One-error*.

Na Figura 34 podemos verificar que o uso do algoritmo de balanceamento SMOTE houve uma ligeira piora para o SVM em 0,26%, para o SMO em 0,04% e para o NB em 3,92%. Com o uso do RESAMPLE, o NB piorou em 4,34%, já o SVM melhorou em 6,56% e o SMO em 3,28%.

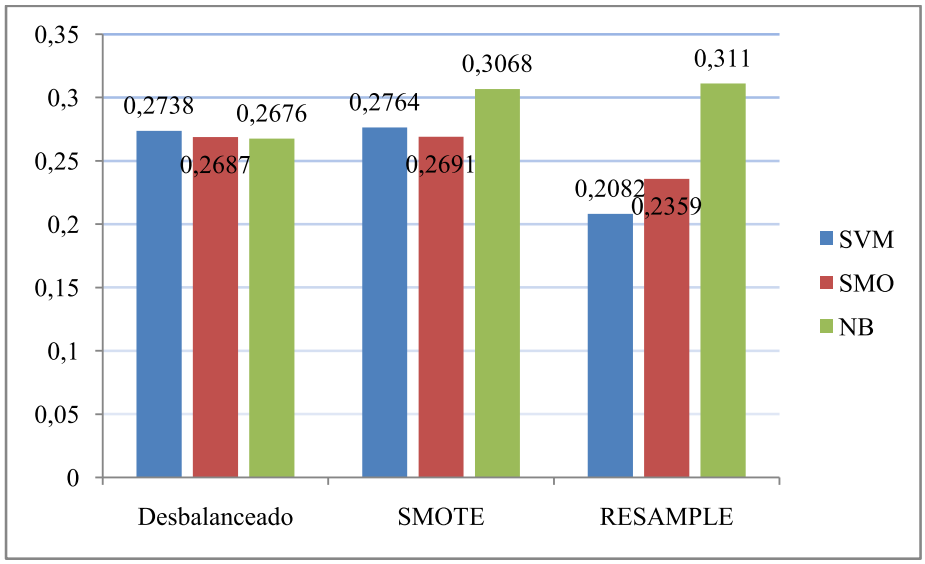

Figura 34 *Hamming Loss* dos algoritmos por conjunto de dados

Na Figura 35 podemos verificar que o uso do SMOTE piorou o resultado do SVM em 0,11% e do NB em 56,88%. Já o balanceamento com o algoritmo RESAMPLE melhorou o resultado do SVM em 11,97%, do SMO em 15,84%, e do NB em 9,98%.

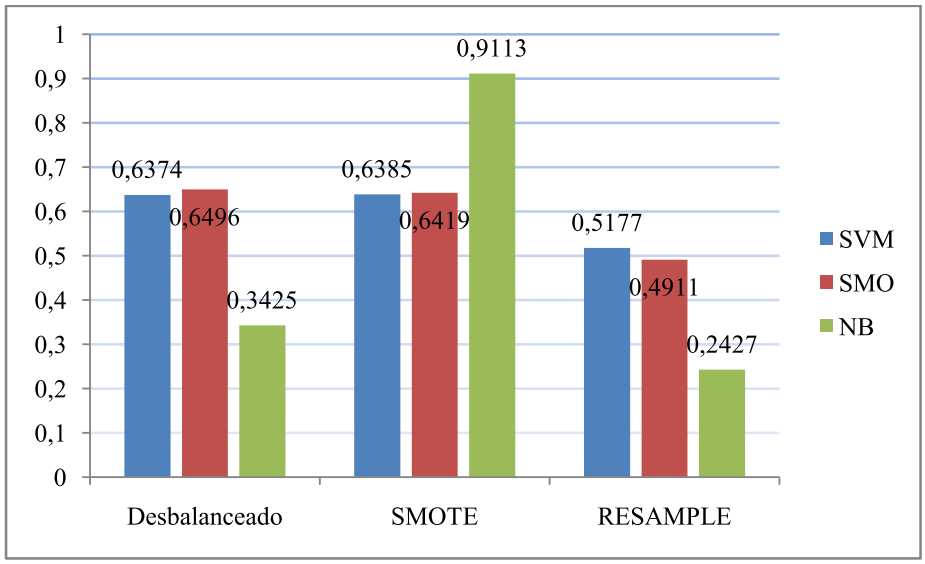

Figura 35 *One-error* dos algoritmos por conjunto de dados

Os resultados da abordagem de classificação multi-rótulo proposta são tão bons quanto os resultados dos algoritmos adaptados para a classificação multi-rótulo propostos na literatura. Uma característica interessante observada nesta abordagem proposta de classificação multi-rótulo é que, para a geração dos modelos de classificação que compõem a abordagem, foram utilizados algoritmos de classificação multi-classe e o conjunto de dados com mensagens classificadas em um gênero.

6 CONCLUSÃO E TRABALHOS FUTUROS

Com o intuito de encontrar a melhor maneira de classificar os gêneros das mensagens de fóruns de discussão em Ambientes Virtuais de Aprendizagem, foram realizados diversos experimentos utilizando algoritmos de classificação multi-classe conhecidos na literatura. Durante a criação de modelos de classificação multi-classe foi validado o quanto o desbalanceamento dos dados afeta os resultados das métricas e o uso dos algoritmos de balanceamento de dados SMOTE e RESAMPLE mostrou uma melhora significativa nos resultados das métricas.

Avaliando os resultados dos experimentos, envolvendo um modelo de classificação multi-classe, a abordagem em cascata e a abordagem proposta, concluímos que os resultados da abordagem em cascata são piores em todos os casos. Utilizando o algoritmo SVM, a abordagem proposta é melhor que um modelo multi-classe quando se utiliza os dados desbalanceados ou balanceados com SMOTE. Utilizando o algoritmo SMO, a abordagem proposta é melhor que um modelo multi-classe quando se utiliza os dados balanceados com RESAMPLE; para os dados balanceados com SMOTE o resultado foi o mesmo, e para os dados desbalanceados o modelo multi-classe foi melhor. Para o algoritmo SVM, o modelo multi-classe é melhor que a abordagem proposta. Como os algoritmos SVM e SMO são mais utilizados na literatura para a classificação de texto, podemos afirmar que a abordagem proposta obteve melhores resultados e ainda é capaz de classificar a mensagem em mais de uma classe.

Outra vantagem da abordagem proposta é que classifica a mensagem em mais de um gênero, ou seja, multi-rótulo. Os resultados da abordagem proposta multi-rótulo foram comparados com algoritmos especializados adaptados para a classificação multi-rótulo.

Avaliando a métrica *Hamming Loss*, o resultado do algoritmo SMO utilizando os três conjuntos de dados ficou entre a variância dos resultados do algoritmo MLkNN e RAkEL. Para a métrica *One-error*, os resultados com o algoritmo balanceado com o algoritmo RESAMPLE ficou aproximadamente 4% pior do que a variância de resultados do algoritmo MLkNN e aproximadamente 9% pior que a variância de resultados do algoritmo RAkEL.

A abordagem proposta de classificação multi-rótulo foi desenvolvida a partir de modelos gerados por algoritmos de classificação multi-classe com um conjunto de dados classificados em apenas uma classe. Desta forma podemos concluir que a abordagem proposta apresenta um bom desempenho neste contexto, permitindo a classificação da mensagem em um ou mais gêneros.

Como trabalho futuro será desenvolvido um módulo que implemente esta abordagem de classificação para ser utilizada em um ambiente real.

REFERÊNCIAS

AKEN J. E. V. Management research as a design science: articulating the research products of mode 2 knowledge production in management. **British Journal of Management**, Chichester, v. 16, n. 1, p. 19-36, Mar. 2005.

ARANHA, C. N.; VELLASCO, M. M. B. R. **Uma abordagem de pré-processamento automático para mineração de textos em português**: sob o enfoque da inteligência computacional. 2007. 144 p. Tese (Doutorado em Engenharia Elétrica) - Pontífica Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

BLACKBOARD. Disponível em: <http://www.blackboard.com/>. Acesso em: 14 mar. 2014.

BLACKBOARD. Disponível em: <http://www.webct.com>. Acesso em: 14 mar. 2014.

BRUSILOVSKY, P.; PEYLO, C. Adaptive and intelligent web-based educational systems. **International Journal of Artificial Intelligence in Education**, Easton, v. 13, n. 2-4, p. 156-169, Apr. 2003.

CHAWLA N. V. et al. SMOTE: synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research**, Washington, v. 16, n. 1, p. 321-357, Jan. 2002.

CHAWLA N. V.; JAPKOWICZ, N.; KOTICZ, A. Special issue on learning from imbalanced data sets. **ACM SIGKDD Explorations Newsletter**, New York, v. 6, n. 1, p. 1-6, June 2004.

CONSORTION CLAROLINE. **Easy e flexibly learning solutions**. France: [s.n.], 2014. Disponível em: <http://www.claroline.net>. Acesso em: 14 mar. 2014.

DUMAIS, S. et al. Inductive learning algorithms and representations for text categorization. In: INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 17., 1998., **Proceedings…**Washington: CIKM, 1998. p. 148 -55.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. The kdd process for extracting useful knowledge from volumes of data. **Communications of the ACM**, New York, v. 39, n. 11, p. 27-34, Nov.1996.

FORMAN, G. An extensive empirical study f feature selection metrics for text classification. **Journal of Machine Learning Research**, Washington, v. 3, p. 1289 - 1305, Mar. 2003.

GUIMARÃES, F. R. N.; ESMIN A. A. A. Identificação automática de gêneros das mensagens em fóruns de discussões do AVA. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL, 2014, São Carlos. **Anais...** São Carlos: ENIAC, 2014. p. 176-181.

GUPTA, V.; LEHAL, G. S. A survey of text mining techniques and applications. **Journal of Emerging Technologies in Web Intelligence**, Amsterdam, v. 1, n. 1, p. 60-76, Aug. 2009.

HAIBO, H.; GARCIA, E. A. Learning from imbalanced data. **IEEE Transactions on Knowledge and Data Engineering,** New York, v. 21, n. 9, p. 1263-1284, Sept. 2009.

HALL, M. et al. The WEKA data mining software: an update. **ACM SIGKDD Explorations Newsletter**, New York, v. 11, n. 1, p. 10-18, Nov. 2009.

HAN, J.; KAMBER, M. **Data mining:** concepts and techniques. San Francisco: Morgan Kaufmann, 2005.

HOTHO A.; NÜMBERGER, A.; PAAB, G. A brief survey of text mining. **Journal for Computational Linguistics and Language Technology**, Washington, v. 20, n. 1, p. 09-63, May 2005.

ILIAS. Disponível em: <http://www.ilias.de>. Acesso em: 14 mar. 2014.

JAPKOWICZ, N.; STEPHEN, S. The class imbalance problem: a systematic study. **Intelligent Data Analysis**, Amsterdam, v. 6, n. 5, p. 429-449, Oct. 2002.

JOACHIMS, T. Text categorization with support vector machines: learning with many relevant features. EUROPEAN CONFERENCE ON MACHINE LEARNING, 10., 1998, Chemnitz. **Proceedings…** Chemnitz: Machine Learning, 1998. p. 137-142.

KEERTHI, S. S. et al. Improvements to Platt's SMO algorithm for SVM classifier design. **Journal of Neural Computation**, Oxford, v. 13, n. 3, p. 637-649, Mar. 2001.

KEERTHI, S. S. et al. Improvements to the SMO algorithm for SVM regression. **IEEE Transactions on Neural Networks**, New York, v. 11, n. 5, p. 1188-1193, Aug. 2002.

LEWIS, D. D. Naive (Bayes) at forty: the independence assumption in information retrieval. In: EUROPEAN CONFERENCE ON MACHINE LEARNING, 10., 1998, Belin. **Proceedings…** Belin: Heidelberg, 1998. p. 04-15, 1998.

LIN, F. R.; HSIEH, L. S.; CHUANG, F. T. Discovering genres of online discussion threads via text mining. **Computers & Education**, New York, v. 52, n. 2, p. 481-495, Feb. 2009.

LIU, P. et al. Classifying skewed data streams based on reusing data. In: INTERNATIONAL CONFERENCE ON COMPUTER APPLICATION AND SYSTEM MODELING, 2010, China. **Proceedings…** China: IEEE, 2010. p. 90-93.

LONGADGE, R.; DONGRE, S. S.; MALIK, L. Class imbalance problem in data mining: review. **International Journal of Computer Science and Network**, New York, v. 2, n. 1, p. 01-06, Feb. 2013.

LOPES, A. P.Teaching with moodle in higher education. In: INTERNATIONAL TECHNOLOGY, EDUCATION AND DEVELOPMENT CONFERENCE, 2011, Valencia. **Proceedings…**Valencia: INTED, 2011. p. 970-976.

MAIMON, O.; ROKACH, L. **Data mining and knowledge discovery handbook.** New York: Springer-Verlag, 2005.

MAZZA R.; MILANI, C. Exploring usage analysis in learning systems: gaining insights from visualizations. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 12., 2005, Amsterdam. **Proceedings...** Amsterdam: AIED, 2005.

MCALLUM A.; NIGAM, K. A comparison of event models for Naive Bayes text classification. In: WORKSHOP ON LEARNING FOR TEXT

CATEGORIZATION, 1998, Madison. **Proceedings…** Madison: AAAI, 1998. p. 41-48.

MEKA: a multi-label extension to WEKA. Disponível em: <http://meka.sourceforge.net>. Acesso em: 14 mar. 2014.

MOODLE. **Community driven, globally supported.** [S.l.: s.n.], 2014. Disponível em: <http://www.moodle.org>. Acesso em: 14 mar. 2014.

MOR, E.; MINGUILLÓN, J. E-learning personalization based on itineraries and long-term navigational behavior. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE ON ALTERNATE TRACK PAPERS & POSTERS, 13., 2004, New York. **Proceedings…** New York: ACM, 2004. p. 264-265.

MOSTOW, J. et al. An educational data mining tool to browse tutor-student interaction: time will tell!. In: WORKSHOP ON EDUCATIONAL DATA MINING, 2005, Pitsburgh. **Proceedings…** Pitsburgh: AAI Press, 2005. p. 15-22.

MULAN: a java library for multi-label learning. Disponível em: <http://mulan.sourceforge.net>. Acesso em: 14 mar. 2014.

MUNTEAN, M. et al.**Improving classification with support vector machine for unbalanced data**. Cluj-Napoca: Automation Quality and Testing Robotics, 2010.

OLIVEIRA JÚNIOR, R. et al. Uma ferramenta de monitoramento automático de mensagens de fóruns em ambientes virtuais de aprendizagem. In: O SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 22.; WORKSHOP DE INFORMÁTICA NA ESCOLA, 17., 2011, Aracaju. **Anais…** Aracaju: SBC, 2011. p. 60-69.

PAHL, C.; DONNELLAN, C. Data mining technology for the evaluation of web-based teaching and learning systems. In: WORLD CONFERENCE ON E-LEARNING, 2002, Montreal. **Proceedings...** Montreal: AACE, 2002. p. 747-752.

PENA-SHAFT, J. B.; NICHOLLS, C. Analyzing student interactions and meaning construction in computer bulletin board discussions. **Computers & Education**, New York, v. 43, n. 3, p. 243-265, Apr. 2004.

PLATT, J. C. Sequential minimal optimization: a fast algorithm for training support vector machines. **CiteSeer**, Pennsylvania, 1998.

READ, J. A pruned problem transformation method for multi-label classification. In: COMPUTER SCIENCE RESEARCH STUDENT CONFERENCE, 2008, New Zealand. **Proceedings…** New Zealand, 2008. p. 143-150.

RENNIE, J. D. M. et al. Tackling the poor assumptions of naive bayes text classifiers. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 20., 2003, Washington. **Proceedings…** Washington: AAAI Press, 2003. 2003. p. 616-623.

RENNIE, J. D. M. Improving multi-class text classification with naive bayes. AI **Technical Report**, Massachusetts, p. 01-43, Sept. 2001.

RICE, W. **Moodle E-learning course development.** Oxford: Packt Publishing, 2006.

ROMERO, C. et al. Applying Web usage mining for personalizing hyperlinks in Web-based adaptive educational systems. **Computers & Education**, New York, v. 53, n. 3, p. 828-840, Nov. 2009.

ROMERO, C.; VENTURA, S. Educational data mining: a survey from 1995 to 2005. **Expert Systems with Applications**, New York, v. 33, n. 1, p. 135-146, July 2007.

ROMERO, C.; VENTURA, S.; BRA, P. D. Knowledge discovery with genetic programming for providing to courseware authors. **User Modeling and User-Adapted Interaction**, Dordrecht, v. 14, n. 5, p. 425-464, Jan. 2005.

ROMERO, C.; VENTURA, S.; GARCÍA, E. Data mining in course management systems: moodle case study and tutorial. **Computers & Education**, New York, v. 51, n. 1, p. 368-384, Aug. 2008.

SCHAPIRE R. E.; SINGER Y. Boostexter: a boosting-based system for text categorization. **Machining Learning**, Oxford, v. 39, n. 2-3, p. 135-168, May 2000.

SEBASTINI, F. Classification of text, automatic. **The Encyclopedia of Language and Linguistics**, Oxford, v. 14, p. 457-462, 2006.

SEBASTINI, F. Machine learning in automated text categorization. **ACM Computing Surveys**, New York, v. 34, n. 1, p. 01-47, Mar. 2002.

SEIFFERT, C. et al. A comparative study of data sampling and cost sensitive learning. In: IEEE INTERNATIONAL CONFERENCE ON DATA MINING WORKSHOPS, 2008, Los Alamitos. **Proceedings…** Los Alamitos: IEEE Computer, 2008. p. 46-52.

SMOLA, J. A. **Regression estimation with support vector learning machines.** Berlin: GMD, 1996.

SMOLA, J. A.; SCHÖLKOPF, B. **A tutorial on support vector regression.**Hingham: Kluwer Academic, 1998.

SOROWER, M. S. A literature survey on algorithms for multi-label learning. **CiteSeer**, Pennsylvania, 2010.

SRIVASTAVA, J. et al. Web usage mining: discovery and applications of usage patterns from web data. **ACM SIGKDD Explorations Newsletter**, New York, v. 1, n. 2, p. 12 - 23, Jan. 2000.

STEINBACH, M.; ERTOZ, L.; KUMAR, V. The challenges of clustering high dimensional data. In: STEINBACH,M.; ERTÖZ, L.; KUMAR, V**. In new vistas in statistical physics:** applications in econophysics, bioinformatics, and pattern recognition. Berlin: Springer-Verlag, 2003. p. 273-309.

TAGG, C. **Discourse of text messaging:** analysis of SMS communication. Londres: Bloomsbury Academic, 2012.

TAVALERA, L.; GAUDIOSO, E. Mining student data to characterize similar behavior groups in unstructured collaboration spaces. In: WORKSHOP ARTIFICIAL INTELLIGENCE METHODS IN COMPUTER-SUPPORTED COLLABORATIVE LEARNING, 2004, Valencia. **Proceedings…**Valencia: CSCL, 2004. p. 17-23.

TSOUMAKAS, G. et al. Mulan: a java library for multi-label learning. **Journal of Machine Learning Research**, Washington, v. 12, p. 2411-2414, July 2011.

TSOUMAKAS, G.; KATAKIS, I. Multi-label classification: an overview. **International Journal of Data Warehousing and Mining**, Netherlands, v. 3, n. 3, p. 01-13, July 2007.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Mining multi-label data. In: MAIMON, O.; ROKACH, L. (Ed.). **Data mining and knowledge discovery handbook.** New York: Springer, 2010. p. 667-685.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Random k-Label sets for multilabel classification. **IEEE Transactions on Knowledge and Data Engineering**, New York, v. 23, n. 7, p. 1079-1089, July 2011.

VAPNIK, V. N. **The nature of statistical learning theory**. 2 nd ed. New York: Springer, 1995.

WAINER, J. **Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação**. Rio de Janeiro: Sociedade Brasileira da Computação, 2007.

WASIKOWAKI, M.; CHEN, X. Combating the small sample class imbalance problem using feature selection. **IEEE Transactions on Knowledge and Data Engineering**, New York, v. 22, n. 10, p. 1388–1400, Oct. 2010.

WBT SYSTNS. Disponível em: <http://www.wbtsystems.com/>. Acesso em: 14 mar. 2014.

WITTEN, I. H.; FRANK, E. **Data mining: practical machine learning tools and techniques**. 2nd. ed. San Francisco: Morgan Kaufmann, 2005.

YOUN, E.; JEONG, M. K. Class dependent feature scaling method using naive bayes classifier for text datamining. **Pattern Recognition Letters**, Amsterdam, v. 30, n. 5, p. 477-485, Apr. 2009.

ZHANG, M. L.; ZHOU, Z. H. ML-KNN: a lazy learning approach to multi-label learning. **Pattern Recognition**, Ezmsford, v. 40, n. 7, p. 2038-2048, July 2007.