ANÁLISE ADAPTATIVA DE FLUXOS DE SENTIMENTO

ISMAEL SANTANA SILVA

ANÁLISE ADAPTATIVA DE FLUXOS DE SENTIMENTO

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais - Departamento de Ciência da Computação. como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Renato Antônio Celso Ferreira
Coorientador: Adriano Alonso Veloso

Belo Horizonte

Março de 2012

Silva, Ismael Santana
S586a Análise adaptativa de fluxos de sentimento / Ismael Santana Silva. — Belo Horizonte, 2012
xxii, 66 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais - Departamento de Ciência da Computação.
Orientador: Renato Antônio Celso Ferreira
Coorientador: Adriano Alonso Veloso

1. Computação - Teses. 2. Redes sociais on-line - Teses. 3. Twitter - Teses. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*04(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Análise adaptativa de fluxos de sentimento

ISMAEL SANTANA SILVA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. RENATO ANTÔNIO CELSO FERREIRA - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ADRIANO ALONSO VELOSO - Co-orientador
Departamento de Ciência da Computação - UFMG

PROFA. GISELE LOBO PAPPA
Departamento de Ciência da Computação - UFMG

PROF. NIVIO ZIVIANI
Departamento de Ciência da Computação - UFMG

PROF. WAGNER MEIRA JÚNIOR
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 26 de março de 2012.

Dedico este trabalho a minha família, Samuel, Sebastiana, Tatiana, Cristiane e todos Tios e Primos. Exemplos de superação. Tudo isso foi por vocês! Dedico também a Glívia (minha namorada) por todo companheirismo e ajuda durante todo esse tempo.

Agradecimentos

Muito Obrigado,

Pai (Samuel), Mãe (Sebastiana), Irmãs (Tatiana e Cristiane), todos os Tios e Tias. Sem vocês para me espelhar provavelmente não teria caminhado até aqui. Vocês são fontes de inspiração, carinho e amor. Exemplos de superação! A melhor família do mundo. Um agradecimento especial para meu Pai (Samuel) e minha Mãe (Sebastiana) que souberam me educar e me guiar até aqui com muito amor.

Glívia (minha namorada), sempre companheira e sempre agraciando a todos com sua atenção e empenho. Muito obrigado por me acompanhar pelos mais longos caminhos e pelas incansáveis revisões nos nossos trabalhos. Agradeço também, por aprender tanto sobre aprendizado de máquina e mineração de dados durante as minhas repedidas explicações em busca de uma inspiração.

A todos os colegas que conheci durante o meu mestrado. Soraia e Daniel, por revisar vários dos meus textos e muitos momentos de conversa. Rodrigo Silva e Thatyene pelo apoio quando eu iniciei o mestrado. Janaína pela participação em alguns trabalhos. Walter e toda equipe do Observatório da Web pela ajuda na obtenção de dados e construção de uma aplicação para demonstração. Aos companheiros, Flávio Roberto e Luiz Damilton. A todos os integrantes do laboratório e-speed e PENSI.

Aos meus orientadores Renato Ferreira e Adriano Veloso, pela confiança, incentivo e ajuda em todo o trabalho, foi um honra trabalhar com vocês. Muito obrigado Professor Wagner, pela ajuda na pesquisa e revisões nos trabalhos enviados para conferências. Agradeço ao Professor Mohammed Zaki, por orientar minhas pesquisas durante período no Rensselaer Polytechnic Institute (RPI) e viabilizar minha estadia nos EUA. Obrigado Professora Gisele por ter sido a primeira pessoa que me ajudou com um tema de pesquisa.

Membros da banca, (Nivio, Wagner e Gisele), obrigada por aceitar o convite em colaborar com essa pesquisa. Finalmente, a todos os funcionários da UFMG e aos órgãos CAPES e UOL pela ajuda financeira durante o mestrado.

"Reconhecimento de padrões está ligado à busca de regularidades. Desde tempos pré-históricos, o homem buscou regularidades em que pudesse confiar e que lhe desse uma sensação de segurança em um mundo hostil."

(Autor desconhecido)

Resumo

Nos últimos anos, a tarefa de análise de sentimentos tem atraído muito interesse por parte da comunidade de Aprendizado de Máquina. Com o advento dos canais de mídias sociais essa análise tem ganhado força, isso porque nessas aplicações, os usuários são convidados, continuamente, a compartilhar suas opiniões e sentimento sobre diferentes tópicos de discussão. Como consequência, uma grande massa de conteúdo opinativo é gerada em tempo real. Muitas técnicas de classificação automática têm sido utilizadas para realizar a análise de sentimento, contudo é consenso que o modelo de chegada de mensagens a partir de mídias sociais segue o paradigma de fluxo de dados e as técnicas de classificação tradicionais não estão adequadas para tratar as características especificas desse fluxo de sentimento. Entre os desafios impostos às técnicas classificação podemos destacar: (1) o *sentiment drift* (i.e., constantes mudanças nas características dos dados), (2) a necessidade de atualização em tempo real do modelo de classificação a partir de mensagens mais recentes e (3) a quantidade limitada de dados para treinamento dos algoritmos.

Neste trabalho, esses problemas foram estudados a partir de uma proposta de aprendizado semi-supervisionado. O algoritmo proposto auto expande o conjunto de treino, a partir de novas mensagens do fluxo e uma pequena semente de treinamento inicial. Modelos de classificação são produzidos em tempo real a partir de regras de associação, mantendo o modelo atualizado de maneira incremental. Dessa forma em qualquer momento do evento, o modelo reflete o sentimento que está sendo transmitido. Com intuito de tratar o *sentiment drift*, mensagens de treinamento são projetadas sob-demanda, de acordo com o conteúdo da mensagem sendo classificada. A projeção de dados de treinamento oferece uma série de vantagens, incluindo a habilidade de rapidamente detectar tópicos de informação emergindo no fluxo.

Um estudo de caso foi realizado a partir das mensagens do Twitter, postadas em tempo real em relação a três importantes eventos de 2010: (1) Copa do Mundo de Futebol; (2) Eleições Presidenciais do Brasil; e (3) Escolha da personalidade do ano pela revista TIME. Através desse experimento observou-se que o desempenho da predição se

mantem, ou até aumenta, com o decorrer do fluxo e a inclusão de novas mensagens no conjunto de treinamento. Estes resultados são assegurados para diferentes linguagens, em casos onde a distribuição do sentimento muda de diferentes maneiras com o decorrer do tempo ou em casos onde a semente de treinamento inicial é extremamente pequena.

Além disso, uma análise comparativa foi realizada onde as versões estáticas (i.e., não ocorre atualizações no modelo após o treinamento inicial) dos mais populares algoritmos foram consideradas como um limite inferior. Nessa comparação, verificou-se que nossa abordagem é eficiente nos cenários analisados, provendo ganhos de 14% a 41%. Posteriormente, um limite superior foi definido, a partir de/da: (1) experimentos utilizando a metodologia que considera todos os dados rotulados, (2) busca pelo melhor tamanho de janela deslizante de treinamento para cada coleção de dados e (3) proposta de uma nova técnica de esquecimento em fluxo de dados, baseada em um processo de amostragem ativa, denominada Janela Deslizante Ativa (JDA). A JDA foi capaz de alcançar resultados equivalentes ao melhor tamanho de janela de treino (definido por uma busca exaustiva) sem necessitar da configuração prévia do tamanho da janela. Dessa forma, avaliou-se o quanto nossa abordagem auto treinamento se aproxima do limite superior e verificamos que a solução proposta atinge até 87% do limite superior, utilizando somente uma pequena semente de dados rotulados.

Palavras-chave: Análise de Sentimento, Fluxo de Dados, *Concept Drift*, Mídia Social.

Abstract

Over the past years, the sentiment analysis task has attracted the interest of the machine learning researchers. This interest has grown significantly due to the large volume of opinionative content generated and shared via social media. Considering the benefits of knowing the sentiment of the population regarding different topics and entities, the analysis of the content generated by social medias it is a promising and necessary task. Many automatic classification techniques have been used to perform sentiment analysis, however it is consensus that the arrival pattern of messages from social medias follows the data stream paradigm and the traditional classification techniques are not adequate to address the specific characteristics of this sentiment stream. Among the challenges imposed to classification techniques we can be highlight: (1) concept drift (i.e., constant changes in data characteristics, which in this study was approached as sentiment drift), (2) the need of real-time update of the classification model from the most recent messages and (3) the limitation of computing and training resource, which makes the two firsts challenges mentioned more difficult.

We analyzed these problems from a semi-supervised learning proposal. Our algorithm adapts the training set, to the changes in the data, from a self-augmenting training process with the passes of the stream. It uses a small seed as an initial training and then classification models are produced in real-time using association rules. This strategy keeps the model up-to-date incrementally, so that at any time of the event the model reflects the sentiment that is being transmitted. In order to address the sentiment drift, messages to training are projected on-demand, according to the message content that is being classified. Projection of the training data offers a number of advantages including the ability to quickly detect emerging trends in the information stream. We conducted a case study using the Twitter messages, posted in real-time, related to major events in 2010. In these experiments the performance of the prediction kept the same or increased, with the passes of the stream and the inclusion of new messages in the training set. We evaluated the proposed solution in different languages, in cases where the sentiment distribution changed in different way

over time and in cases where the initial training seed is rather small.

In order to complement our experiments, we performed a comparative analysis with the static version (i.e., the model is not updated after the initial training) of the most popular classification algorithms, which were defined as a lower bound. Our approach showed extreme effective in these analyzed scenarios, providing gains from 14% to 41%. We define an upper bound from experiments using the evaluation methodology "interleaved test-then-train"and the search for the best sliding window size to train each data set. Thereafter, we proposed a new forgetting technique in the data stream, based on an active sampling process, called Active Sliding Window (ASW). ASW was able to achieve results equivalent to the best window size (set by exhaustive search) without prior setting of the window size, since the configuration of this parameter by the user can be unfeasible, as shown in the experiments. We analyzed how our semi-supervised approach approaches to the upper bound and we found that it achieves up to 87% of the upper bound, using only a small seed of labeled data.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

A Web deixou de ser um espaço que interliga exclusivamente documentos, páginas ou recursos para tornar um ambiente de comunicação, no qual produtores e consumidores de conteúdo se misturam e interagem, estabelecendo assim novas formas de criar, organizar, compartilhar e utilizar o conhecimento [Easley & Kleinberg, 2010]. Uma das fontes para geração desse conteúdo são as mídias sociais (e.g., Orkut, Twitter e Facebook), onde milhões de usuários são encorajados a compartilhar e expressar suas opiniões e sentimentos sobre os mais diversos tópicos e entidades. A ascensão destes canais de mídias sociais baseados em texto tem atraído atenção da comunidade cientifica. Isso porque a descoberta de conhecimento a partir do conteúdo expresso pelos usuários da web oferece oportunidades estratégicas para diferentes áreas [Bermingham & Smeaton, 2010].

Nos últimos anos a tarefa de analisar conteúdo, a partir dos sentimentos nele expressos, tem sido amplamente estudada [Pang et al., 2002; Pang & Lee, 2008; Bifet & Frank, 2010]. Nessa tarefa, a mensagem é basicamente classificada de acordo com o sentimento expresso pelo escritor a respeito de um tópico ou entidade. Considerando os benefícios da análise de sentimento e o potencial dos canais de mídias social como fonte de conteúdo opinativo, cada vez mais se faz necessário realizar análise de sentimentos a partir dos dados gerados nas mídias sociais.

A partir dessas aplicações é possível mensurar o sentimento compartilhado continuamente pela população online sobre diversos assuntos (e.g., epidemias [Chew & Eysenbach, 2010], eleições [Diakopoulos & Shamma, 2010] e eventos esportivos [Silva et al., 2011c]), e o conhecimento obtido podem ser utilizado por diversas áreas no processo de tomada de decisão.

Existe uma crescente tendência em executar análise de sentimentos utilizando técnicas relacionadas a classificação [Pang et al., 2002; Bifet & Frank, 2010], um

processo que automaticamente constrói um modelo de classificação por aprendizado da característica subjacente do texto, a partir de um conjunto de mensagens previamente rotuladas (i.e., dados de treinamento), que distingue um sentimento de outro (e.g., felicidade, tristeza, fúria, surpresa e desconfiança). O sucesso destes classificadores se deve pela sua habilidade de julgar atitudes pelo significado de padrões textuais presentes nas mensagens, as quais geralmente aparecem na forma de expressões idiomáticas e combinações de palavras. É bem aceito que a qualidade dos dados de treinamento, que é fornecido para o classificador, é crucial para a efetividade da análise.

Apesar de não existir um consenso de como os dados de treinamento devem ser produzidos, o custo de rotular manualmente uma vasta quantidade de mensagens é proibitivo, uma vez que a aquisição destas mensagens de exemplo pode requerer a inspeção de um humano qualificado. Diante disso, formas alternativas para desse treino têm sido propostas (e.g., técnicas que fazem uso de alternativas semi-supervisionadas [Chapelle et al., 2006] e de aprendizagem ativo [Settles, 2009]). Contudo, estas técnicas envolvem complexos procedimentos que são inviáveis a serem realizados na velocidade de chegada das mensagens.

Além disso, as técnicas de classificação tradicionais não estão adequadas para a análise de sentimento a partir do conteúdo compartilhado em mídias sociais. Isso porque, na maioria dos casos essas técnicas assumem que uma grande quantidade de dados é amostrada de uma distribuição estacionária para geração do conjunto de treinamento [Hulten et al., 2001]. Porém, essa consideração é violada, porque nessas aplicações o processo de chegada dos dados segue o paradigma de fluxo de dados, os quais estão sujeitos a constantes mudanças na distribuição e na maneira que os sentimentos são expressos com o passar do tempo. Nesse trabalho este cenário é chamado de *fluxo de sentimento* e a tarefa de separar os sentimentos nele contido de *análise de fluxo de sentimento*.

Gama & Mohamed [2007] definem fluxos de dados como uma sequência de dados ilimitada que chegam em tempo real, continuamente e em alta velocidade. Em relação as fontes de fluxos de dados existentes é possível citar: monitoramento de redes, rede de sensores, gerenciamento de dados de telecomunicações, aplicações financeiras e aplicações web. Pesquisadores apontam a análise deste tipo dado como um dos maiores desafios enfrentados atualmente em áreas como aprendizado de máquina e mineração de dados [Bifet, 2010; Bifet & Frank, 2010]. Dentre esses desafios é possível destacar:

- A necessidade dos classificadores se adaptarem às constantes mudanças no fluxo [Zhang et al., 2008] (conhecidas como: *concept drift* e que neste trabalho será tratado como *sentiment drift*) e;

- O fato dos classificadores operarem com limitações de memória, tempo de processamento e dados rotulados para treinamento.

Embora, existam propostas baseadas em aprendizado de máquina para análise de sentimento, elas não são suficientes para capturar e tratar o *sentiment drift*. Nesse contexto é necessário propor soluções para atualização do modelo de classificação em tempo real, e essa atualização deve ocorrer de forma a permitir a adaptação do modelo com a inclusão e remoção de exemplos do conjunto de treinamento. Em outras palavras, em análise de fluxo de sentimento, aprender com o decorrer do fluxo é tão relevante quanto esquecer o que já não descreve o sentimento atual.

1.1 Motivação e Objetivo

Conforme apresentado anteriormente, um modelo de classificação estático não é adequado para realizar a tarefa de análise do fluxo de sentimento gerado por mídias sociais. Isso porque os dados mudam constantemente e estas técnicas não dispõem de recursos para manter seu modelo atualizado. Além disto, existe uma escassez de dados rotulados para treinamento contínuo do classificador.

Sabendo que as técnicas de aprendizado semi-supervisionado permitem melhorar a eficácia de classificadores utilizando os dados não rotulados [Chapelle et al., 2006; Balcan & Blum, 2005], é possível enraizar a hipótese de que: *técnicas de aprendizado semi-supervisionado podem ser adequadas para a tarefa de análise de fluxos sentimento de forma a adaptar o classificador aos fenômenos de sentiment drift.*

Diante disso, considerando a hipótese apresentada e motivados tanto pelos desafios impostos pela análise de sentimento em fluxo de dados, quanto pela a carência de técnicas para resolver os problemas existentes nesse cenário, o objetivo deste trabalho consistiu em analisar e propor soluções para o problema relacionado a falta de dados rotulados para treinamento do classificador, bem como, minimizar os desafios em relação a identificação, caracterização e tratamento do *sentiment drift*.

1.2 Contribuições

Em termos de contribuições este trabalho apresenta uma solução para aprendizado de sentimentos utilizando um modelo de classificação composto por regras de associação. Após uma pequena semente de treinamento ser fornecida para o classificador, ele é capaz de extrair regras que consistem em mapeamentos locais que relacionam os sentimentos a padrões textuais nas mensagens. Além disso, três novas características

fazem a nossa proposta uma solução diferenciada para tratar com diferentes configurações de *sentiment drift*, operando com recursos limitados:

1. A aplicação de um procedimento de auto aumento de treino, o qual incorpora confiáveis predições como novas informações de treinamento, e como resultado os dados de treinamento são automaticamente aumentados como o passar das mensagens do fluxo. O modelo de classificação é imediatamente renovado mantendo as regras atualizadas incrementalmente, então a próxima mensagem no fluxo pode ter vantagens pela informação incluída recentemente;

2. Uma estrátegia de *sub judice*, de acordo com a dificuldade do fluxo, o classificador pode abster de predições duvidosas, criando um bloco de mensagens que estão temporariamente esperando por uma predição de confiança. Estas mensagens para as quais nenhuma predição é possível com somente as mensagens de treinamento disponível até o momento, se beneficiam de informações de treinamento futuras, adquiridas enquanto elas estavam esperando;

3. Execução de projeção dos dados de treinamento sob demanda [Veloso & Meira Jr., 2011], o que determina uma específica (potencialmente diferente) projeção de treinamento para cada mensagem que chega através do fluxo. As mensagens de treinamento que compõem cada projeção são automaticamente selecionadas de acordo com o conteúdo da mensagem que está sendo analisada. Argumentamos que esta é uma poderosa estratégia para tratar diferentes tipos de *sentiment drift*, uma vez que elimina mensagens de treinamento que não significativas para a mensagem que está sendo analisada. Demonstramos que as mensagens removidas não são prejudiciais para o desempenho de predição, e que o número de regras extraídas a partir de cada projeção de treino cresce polinomialmente com o tamanho do vocabulário, não importando o valor de suporte mínimo. Isso permite que o classificador concentre-se em tendências de informação que estão emergindo no fluxo, enquanto permanece livre de uma enorme quantidade de informações inúteis.

Essas características foram propostas de forma a se complementar. Isso porque elas produzem um efeito sinérgico no sentido de que ambas precisam uma da outra para funcionar corretamente. A capacidade de auto aumentar o treino assegura a inclusão de novas mensagens no conjunto de treinamento, que são necessárias para produzir projeções de treinamento atualizadas. Neste ponto, o uso de restrições típicas de custo computacional baseadas em limiares de suporte mínimo poderia comprometer

o processo, uma vez que padrões importantes que estão emergindo no fluxo seriam podados, e como resultado, o modelo de classificação se tornaria obsoleto e incapaz de responder às tendências de opinião.

Portanto, a fim de detectar rapidamente o surgimento de novas informações no fluxo de mensagens, o classificador deve extrair regras sem empregar restrições de frequência com base no suporte. Como será demonstrado, a abordagem de projeção sob demanda assegura que regras são eficientemente extraídas a partir da projeção de treinamento, mesmo sem aplicar restrições de suporte. Além disso, o custo associado com a extração de regras é grandemente amortizado devido a uma abordagem incremental sem perdas, o que reduz drasticamente o número de acessos para os dados de treino.

Para avaliar a eficácia da solução proposta, um conjunto de experimentos sistemáticos utilizando coleções de mensagens do Twitter (ricas em sentimentos que retratam três eventos importantes no ano de 2010) foi utilizado. Analisamos diferentes cenários de aprendizagem (i.e., diferentes sentimentos, idiomas, tamanhos de sementes treinamento e diferentes tipos de mudanças nos sentimentos). Comparamos nossa abordagem com as versões estáticas de algoritmos estado da arte e os resultados mostram que nossa proposta é mais efetiva sobre diversos cenários de aprendizado, alcançando ganhos que variam de 14% a 41%. Embora tenhamos realizado um estudo de caso no Twitter, as técnicas propostas são aplicáveis a qualquer fonte de dado que compartilhe as mesmas caracteristicas deste sistema.

Definimos uma aproximação do limite superior para o problema abordado no trabalho com a executamos uma série de experimentos utilizando a metodologia de avaliação *interleaved test-then-train* [Gama et al., 2009; Dawid, 1984; Bifet & Frank, 2010], a qual considera que todas as mensagens estão disponíveis com o seu rótulo correto para atualização do modelo logo após o seu processamento.

Durante a definição do limite superior, foi apresentada uma contribuição que se trata de uma nova proposta para remoção de exemplos desatualizados do conjunto de treinamento (i.e., esquecimento) na análise de sentimento em fluxo de dados.

Essa proposta é baseada na teoria de Aprendizado Ativo e foi denominada de Janela Deslizante Ativa (JDA). Ela pode ser aplicada em uma outra perspectiva do problema de análise de fluxos de sentimento, ou seja, em cenários que os exemplos para treinamento são disponibilizado ao decorrer do fluxo (i.e., mensagens chegam rotuladas). O objetivo dessa proposta foi prover ao classificador a capacidade de manter os exemplos mais significativos no conjunto de treino com um viés temporal. Isto porque, conforme descrito anteriormente, neste cenário, aprender com o decorrer do fluxo é tão relevante quanto esquecer o que já não descreve o sentimento atual.

Verificamos experimentalmente que JDA foi capaz de alcançar resultados equivalentes ao melhor tamanho de janela de treino (definido com exaustivos experimentos para cada coleção de dados) sem necessitar da configuração prévia deste parâmetro. Uma vez que a configuração do tamanho de janela treinamento pelo usuário pode ser inviável como mostrado nos experimentos.

Finalmente, nosso trabalho apresenta uma caracterização do *sentiment drift* que ocorre em diferentes eventos do mundo real. Tal caracterização se faz relevante pois permite uma melhor compreensão de como ocorrem as mudanças no fluxo de sentimento em vários cenários e, consequentemente, favorece a criação de técnicas que permitam tratar essas mudanças durante o processo de classificação automática em fluxo de dados.

As soluções apresentadas nesse trabalho, resultados preliminares e aplicações foram apresentados em [Silva et al., 2011a] [Silva et al., 2011b] [Silva et al., 2011c].

1.3 Organização do texto

Esta dissertação está organizada em 5 Seções. Na próxima seção, apresentamos os trabalhos relacionados e estratégias existentes para análise de fluxo de dados. Em seguida, as soluções propostas e a avaliação experimental realizada. Na quarta seção definimos um limite superior para a abordagem de auto treinamento e verificamos o quanto os resultados dessa abordagem se aproximam deste limite. Finalmente, a última seção apresenta as conclusões alcançadas.

Capítulo 2

Trabalhos Relacionados

Nos últimos anos a tarefa de analisar conteúdo, a partir dos sentimentos nele expressos, tem sido amplamente estudada [Bifet & Frank, 2010]. Considerando os benefícios da análise de sentimento e o potencial dos canais de mídias social como fonte de conteúdo opinativo, cada vez mais se faz necessário realizar análise de sentimentos a partir do fluxo de dados gerados através dessas aplicações. No entanto, esse tipo de análise apresenta vários desafios a serem contornados pelas técnicas de mineração de dado. Isto porque classificadores devem operar com recursos limitados de computação e treinamento e tratar do *concept drift*. A seguir são apresentadas diferentes estratégias propostas para lidar com *concept drift* em fluxos de dados.

Na tentativa de contornar os problemas advindos do *concept drift*, muitos trabalhos abordam técnicas de manutenção de reservatório de amostragem em fluxo de dados. Essa amostragem é realizada com uma função probabilística de um registro ser incluído ou retirado do reservatório. Aggarwal [2006] apresenta uma solução para o viés temporal das amostras em um reservatório. Os autores fazem uso de uma função temporal para regular a escolha da amostra no fluxo de dados. Tal solução é eficiente em casos onde é desejável obter resultados com e sem viés temporal. No artigo foram apresentadas as vantagens do método para o problema de estimativa de consultas, análise de evolução e classificação.

Al-Kateb et al. [2007] estudaram reservatórios de amostragens em fluxos de dados com tamanho adaptativo sob duas perspectivas: (1) tamanho do reservatório e a (2) uniformidade da amostra. A partir desse estudo, os autores apresentaram 4 contribuições. Primeiro, os resultados baseados em estudos teóricos mostraram que o ajuste do tamanho de um reservatório de amostragem pode impactar de forma negativa sobre a probabilidade da amostra se tornar uniforme. Em seguida, eles propõem um novo algoritmo para manutenção da amostra após o ajuste no reservatório. A terceira

contribuição consiste na ampliação desse algoritmo para um algoritmo adaptativo que trabalha com multi reservatórios para amostragem e, finalmente, a quarta contribuição refere-se aos resultados empíricos do algoritmo adaptativo em relação ao tamanho do reservatório e a uniformidade da amostragem. Esses resultados mostraram que os métodos de amostragem propostos são eficientes na mineração sequencial.

Li et al. [2007] por sua vez, introduzem um método de detecção de mudanças. Esse método assume que os pontos no fluxo de dados são gerados de forma independente, mas também assume a natureza do processo de geração dos mesmos. Esta abordagem utiliza uma função para determinar a distância entre dois exemplos, um valor limite para determinar se ocorreram mudanças e técnicas de amostragem para decidir quais pontos dos dados serão analisados.

No trabalho de Bifet & Gavaldà [2007] é apresentado um arcabouço para desenvolvimento de algoritmos que podem aprender adaptativamente em fluxos de dados com o tempo. Este método é baseado no uso de detectores de mudanças e estimadores de módulos em locais altos. É apresentado um algoritmo de janela deslizante adaptativo (Adaptive Sliding WINdow - ADWIN), uma janela de tamanho variável ao decorrer do tempo, para detectar mudanças e manter estatísticas do fluxo de dados, ele é utilizado em algoritmos não projetados para dados com *concept drift*.

Chu et al. [2004] propõe uma abordagem para o problema de mineração em fluxo de dados a partir de um modelo discriminativo para mineração e aprendizagem rápida em fluxos com ruído. Nesse trabalho foi construído um comitê de classificadores que é adaptado de maneira ponderada de forma a maximizar a vizinhança dos dados. O objetivo foi combinar adaptação ao *concept drift* e robustez ao ruído, empregando técnicas estatísticas para diminuir o problema de sensibilidade a estes. Os autores formularam o problema de classificação ponderada como um problema de regressão, objetivando maximizar a vizinhança dos dados para o conceito atual. Este método de detecção foi integrado ao modelo global de aprendizado.

Zhang et al. [2008] categoriza o *concept drift* em 2 cenários: *concept drift* livre (do inglês, *Loose Concept Drifting* - LCD) e *concept drift* rigoroso (do inglês, *Rigorous Concept Drifting* - RCD), e então propõe uma solução para lidar com cada um separadamente. Para o LCD, dado que o conceito adjacência das partes dos dados são suficientemente próximos, foi aplicado o *Kernel Mean Matching* (KMM) para minimizar a discrepância das partes dos dados. Cada processo de minimização produz instâncias ponderadas para construção de um conjunto de classificadores e para lidar com o *concept drfit* em fluxo de dados.

Para o RCD, dado que o conceito dos dados pode mudar de forma aleatória e rápida, foi proposto um o método *Optimal Weights Adjustment* (OWA) para determinar

o valor de peso ótimo para os classificadores treinados com os dados mais recentes. Dessa forma, classificadores podem formar um conjunto preciso para prever os próximos casos nos dados. Experimentos em bases de dados reais e sintéticas mostram que a abordagem de instância ponderada é preferível quando o *concept drift* é a principal causa pela mudança na distribuição dos dados.

Zhang et al. [2009] propõe um SVM de aprendizado cooperativo baseado na estratégia de sistema de multi-agente de acordo com as características cooperativas e de sistemas distribuídos. Isso porque o SVM normalmente não é escalável para problemas de classificação em grandes conjuntos de dados devido a sua alta complexidade no processo de treino. Neste arcabouço o fluxo de dados chega aos agentes mestres que o particionam em pequenas seções, as quais podem ser atribuídas aos agentes escravos. As seções de dados em cada agente escravo são utilizadas para treinar os SVMs, esses por sua vez, são combinados de acordo com suas compatibilidades.

Outros trabalhos utilizam algoritmos de aprendizado incremental. Esses algoritmos tendem a evoluir ao longo do tempo de acordo com o fluxo de dados que eles recebem e, além disso, visam detectar e reagir a mudanças no ambiente de geração de dados. Gama et al. [2009] realizam um trabalho cujo objetivo consistiu em propor um arcabouço que visa avaliar a qualidade de algoritmos de aprendizagem de fluxo de dados. Os autores utilizaram estimativas de erro *sequencial preditiva* (ou *Interleaved Test-Then-Train*), através de uma janela deslizante para avaliar o desempenho desses algoritmos.

Seidl et al. [2009] propõem uma técnica baseada em índice para lidar com 3 desafios utilizando um classificador Bayesiano, os desafios consistem em: (1) lidar com um grande montante de dados, (2) o tempo variado entre dois itens do fluxo deve ser utilizado da melhor forma possível e (3) dados adicionais no treino devem ser aprendidos de forma incremental.

Outros trabalhos como os realizados por [Fukuhara et al., 2007] e [Bifet & Frank, 2010] tratam de análise de sentimento em fluxos de dados. O artigo de Fukuhara et al. [2007] apresenta uma abordagem que avalia tendências temporais de sentimentos e de tópicos. O método é baseado na presença de palavras chaves tais como "happy"ou "delighted at". Bifet & Frank [2010] apresentam uma discussão sobre o desafio de mineração em fluxo de dados do Twitter focando no problema de análise de sentimentos. Neste trabalho foram avaliados os algoritmos *Multinomial Naive Bayes* (MNB), *Stochastic Gradient Descent* (SGD) e *Hoeffding Tree*. Para o primeiro experimento foi utilizado um conjunto de treino rotulado a partir dos emoticons contidos na mensagem (e.g. :) :( ) e o conjunto de teste foi rotulado manualmente e o algoritmo que mostrou maior eficácia foi o MNB. Um segundo experimento foi

realizado com o conjunto de dados rotulados somente a partir dos emoticons contidos nas mensagens, nesse experimento o algoritmo que apresentou melhor resultado foi o SGD.

Embora os trabalhos acima citados estejam utilizando técnicas de comitê (*ensemble*), aprendizado incremental, janela deslizante e amostragem adaptativas para adaptar as técnicas de aprendizado de máquina às mudanças em fluxos de dados, o alto custo da rotulagem ainda persiste, principalmente se considerarmos aplicações onde o tempo de resposta é pequeno (e.g. a cada 1 minuto a classificação de todos os dados do minuto anterior deve ser realizada) ou imediato. Além disso, algumas dessas técnicas apresentam um custo computacional elevado [Hulten et al., 2001], uma vez que utilizam mais de um classificador para classificação dos dados ou detectores de mudanças.

Com o intuito de contornar esse problemas, Masud et al. [2008] apresentam uma técnica para classificação a partir de um conjunto de treinamento com dados não rotulados e um pequeno número de exemplos rotulados de cada uma das partes do fluxo. Esta técnica constrói micro-grupos com agrupamento semi-supervisionado e a classificação é realizada com o algoritmo de k vizinhos mais próximos (*K-Nearest Neighbor* - KNN). Um conjunto desses modelos é utilizado para classificar os dados não rotulados. Nas avaliações realizadas utilizando apenas uma pequena quantidade de dados rotulados, essa abordagem superou o algoritmo para classificação em fluxo de dados *On Demand Stream*, o qual utilizou vinte vezes mais dados rotulados.

Zhu et al. [2010] propõe um arcabouço de aprendizado ativo baseado em um comitê de classificadores, com o objetivo de reduzir o custo na rotulação de exemplos para treinamento. Esse arcabouço rotula seletivamente instâncias do fluxo de dados para construir um conjunto de classificadores. Os autores argumentam que a variância do conjunto de classificadores corresponde diretamente a taxa de erro, então reduzir a variância do conjunto é equivalente a melhorar a precisão das predições. Sendo assim as instâncias que devem ser rotuladas são aquelas que minimizam a variância do comitê de classificadores.

No trabalho é proposto o princípio de variância mínima para guiar a rotulação dos exemplos, além disto, um método de cálculo de peso ótimo é derivado, para determinar os valores dos pesos para cada classificador no conjunto. Estes são combinados para formar um arcabouço de aprendizado ativo para fluxo de dados. Os autores avaliaram o desempenho do classificador comparando-o com outras abordagens utilizando dados sintéticos e reais. Os resultados mostraram que a natureza dinâmica do fluxo de dados impõe desafios significativos aos algoritmos de aprendizado ativo.

O trabalho aqui proposto se diferencia dos apresentados anteriormente uma vez que apresenta uma abordagem para geração de treino automaticamente minimizando

intervenção humana de forma a capturar o *concept drift*. Além disso, essa abordagem não gera o custo computacional proveniente das técnicas de *ensemble*, já que utiliza apenas um classificador e um algoritmo, baseado em regras de associação, para aprendizado incremental.

Em termos de trabalhos relacionados a aprendizado semi-supervisionado em fluxo de dados podemos citar os realizados por [Wu et al., 2006; Seidl et al., 2009; Li et al., 2010]. Wu et al. [2006] propõem o *clustering-training*, um algoritmo semi-supervisionado que tem como objetivo agrupar exemplos não rotulados do fluxo, em uma das suas partes, após a classificação dos mesmos. Nesta abordagem, se o exemplo é agrupado em um grupo equivalente a classe atribuída ao mesmo pelo classificador, esse se torna um exemplo de confiança e passa a ser utilizado para treinar novamente o classificador de uma maneira incremental.

Li et al. [2010] apresentam um algoritmo semi-supervisionado para classificação em fluxo de dados com *concept drift* e dados não rotulados (SUN) baseado em árvore de decisão. Nessa abordagem os dados no fluxo são rotulados a partir do agrupamento com dados rotulados e a classe atribuída a eles é majoritária no grupo ao qual o exemplo é atribuído. O *concept drift* é dividido nos tipos: potencial, plausível e brusco, de acordo com o nível de ruído do fluxo identificado pela diferença entre o histórico de *clusters* gerados e o *cluster* dos dados que chegam. Os resultados do algoritmo são bons se comparados aos algoritmos *Concept-adapting Very Fast Decision Trees* (CVFDT) [Hulten et al., 2001] e *Concept drifting Detection based on an ensembling model of Random Decision Trees* (CDRDT).

Em trabalhos como os realizados por [Jansen et al., 2009; Bermingham & Smeaton, 2010; Diakopoulos & Shamma, 2010], são realizadas caracterizações dos sentimentos expressos através de mensagens do Twitter em relação a marcas e eventos do mundo real (e.g., debates). Jansen et al. [2009] apresentam os resultados de uma pesquisa realizada em micro-blogs. O objetivo foi verificar se esses micro-blogs podem ser considerados um mecanismo para propaganda online a ser explorado por marcas de produtos.

No trabalho os autores analisaram os comentários, sentimentos e opiniões expressos sobre algumas marcas. Como resultado Jansen et al. [2009] identificaram que os micro-blogs podem ser utilizados pelas marcas tanto para obter quanto para distribuir informações aos clientes. Constatou-se que, através dos micro-blogs, eles podem identificar a aceitação do seu produto, divulgá-los e fazer campanha de marketing a partir dos sentimentos expressos pelos consumidores. Logo, os autores concluem que é evidente a contribuição dos micro-blogs para esse seguimento e que esses podem se tornar uma fonte promissora de inteligência competitiva.

Bermingham & Smeaton [2010] apresentam a web em tempo real como uma nova área de foco para análise de sentimentos e discute as motivações e desafios por trás desta. Os autores apontam aplicações como o Twitter como um bom exemplo de fonte para obtenção dessas informações. O trabalho apresenta uma revisão da análise de sentimentos na web em tempo real como uma futura orientação para pesquisas.

Diakopoulos & Shamma [2010] caracterizam as mensagens enviadas através do Twitter em relação a um debate político de acordo com a forma com que as pessoas estão reagindo a ele e demonstram metodologias analíticas e representações visuais que podem ajudar a compreender melhor a dinâmica temporal do sentimento em relação ao debate. No entanto, os autores não estão interessados na detecção automática dos sentimentos ou de um vencedor ou perdedor. Para cada minuto do debate a agregação dos tweets foi realizada como o número de tweets positivos menos o número de tweets negativos.

Os trabalhos acima apresentados mostram que o Twitter é utilizado como mídia de expressão de sentimento. Contudo, eles apontam a necessidade de técnicas de classificação automática das mensagens em relação aos sentimentos. Diante desse cenário, no trabalho aqui proposto, apresentamos uma solução para esta necessidade, tratando peculiaridades da tarefa como o *sentiment drift* e necessidade de resposta de tempo real.

Embora alguns trabalhos abordem análise de sentimento em fluxo de dados (e.g., [Bifet & Frank, 2010; Fukuhara et al., 2007]), nenhum deles apresentou uma solução como a aqui proposta. Solução essa que não utiliza emoticons ou expressões, como em [Bifet & Frank, 2010; Fukuhara et al., 2007], para a classificação das mensagens, uma vez que essas abordagens são propensas a erros de interpretação e não são capazes de capturar tipos de sentimentos aos quais nenhum emoticon (ou expressão) é associado.

Capítulo 3

Aprendizado de Fluxo de Sentimento

Neste capítulo será apresentada a solução proposta para aprendizado de sentimentos que são expressos em um fluxo de mensagens.

3.1 Análise de Fluxo de Sentimentos

A tarefa de aprendizado de fluxos de sentimento, no contexto deste trabalho, é definida como a seguir. Temos como entrada uma pequena semente de treinamento (referenciado como $\mathcal{D}$), o qual consiste de um conjunto de registros na forma de $< d, s_i >$, onde d é uma mensagem (representada como uma lista de termos, q_1, q_2, ..., q_n) e s_i é o sentimento implícito em d. Mensagens em $\mathcal{D}$ são unicamente identificada e a variável sentimento s assume seus valores de um conjunto pré-definido e discreto de possibilidades (e.g., s_1, s_2, ..., s_k).

A semente de treinamento é utilizada para construir uma função relacionando padrões textuais nas mensagens aos seus respectivos sentimentos. Uma sequência de mensagens futuras, ordenadas cronologicamente, (referenciadas como $\mathcal{T}$) consiste de um registro $< t, ? >$ para o qual somente os termos na mensagem t são conhecidos, enquanto o sentimento expresso em t é desconhecido.

Os modelos de classificação obtidos a partir de $\mathcal{D}$ são utilizados para mensurar os sentimentos para cada mensagem em $\mathcal{T}$. Contudo, mensagens em $\mathcal{T}$ chegam em um fluxo contínuo, desta forma o classificador deve operar com recursos limitados de computação enquanto produz modelos de classificação. Além disso, o classificador deve se auto adaptar devido ao *sentiment drift*, sendo capaz de adquirir novas informações para treinamento como o passar do fluxo, e selecionar as mensagens de treinamento

que são relevantes para cada mensagem em $\mathcal{T}$.

Existem muitos paradigmas e estratégias para elaboração de classificadores para análise de sentimentos [Pang et al., 2002; Pang & Lee, 2008]. A maioria destas estratégias de classificação não é adequada para lidar com dados em tempo real chegando através de fluxos. Algumas estratégias [Breiman et al., 1984; Cortes & Vapnik, 1995] são especificamente concebidas para classificação *offline*, e isso é um problema porque, nestes casos, a produção de modelos de classificação em tempo real pode ser inaceitavelmente custosa. Diante dessas circunstâncias, estratégias de classificação alternativas podem se tornar mais convenientes. A seguir descrevemos modelos de classificação compostos de regras de associação [Veloso et al., 2006], e como estes modelos são utilizados para quantificar sentimentos.

Definição 1. Uma regra de sentimento é uma regra de associação especializada $\mathcal{X} \rightarrow s_i$, onde o antecedente $\mathcal{X}$ é um conjunto de termos, e o consequente s_i é o sentimento previsto. O domínio para $\mathcal{X}$ é o vocabulário de $\mathcal{D}$. A cardinalidade da regra $\mathcal{X} \rightarrow s_i$ é dada pelo número de termos no antecedente, que é $|\mathcal{X}|$. O suporte de $\mathcal{X}$, que é denotado como $\sigma(\mathcal{X})$, é o número de mensagens em $\mathcal{D}$ tendo $\mathcal{X}$ como um subconjunto. A confiança da regra $\mathcal{X} \rightarrow s_i$, denotada como $\theta(\mathcal{X} \rightarrow s_i)$, é a probabilidade condicional do sentimento s_i dados os termos em $\mathcal{X}$, que é calculada de acordo com a Equação 3.1.

$$\theta(\mathcal{X} \rightarrow s_i) = \frac{\sigma(\mathcal{X} \cup s_i)}{\sigma(\mathcal{X})} \tag{3.1}$$

3.1.1 Extração de Regras *Offline*

A abordagem mais simples para a aprendizagem de sentimento utilizando regras de sentimento é a *offline*, onde um conjunto de regras é extraído a partir dos dados de treinamento $\mathcal{D}$, e então, essas regras compõem o modelo de classificação.

Definição 2. O modelo de classificação é denotado como $\mathcal{R}$ e este é composto por um conjunto de regras $\mathcal{X} \rightarrow s_i$ extraída de $\mathcal{D}$. O modelo é representado como um conjunto de entidades na forma $<$*chave*, *valor*$>$, onde *chave*$=\{\mathcal{X}, s_i\}$ e *valor*$=\{\sigma(\mathcal{X}), \sigma(\mathcal{X} \cup s_i), \theta(X \rightarrow s_i)\}$. Cada entidade no conjunto corresponde a uma regra e a *chave* é utilizada para viabilizar o acesso rápido às propriedades das regras.

O processo de extração é dividido em 2 passos: contagem para definição do suporte e cálculo da confiança. Uma vez que o suporte $\sigma(\mathcal{X})$ é conhecido, é simples computar a confiança $\theta(\mathcal{X} \rightarrow s_i)$ para a regra correspondente [Zaki et al., 1997].

Existem várias estratégias de suporte inteligentes [Han et al., 2004; Agrawal et al., 1993; Zaki et al., 1997] e muitas extratégias para reduzir a ordem de complexidade [Bayardo et al., 2004] podem ser utilizadas.

Geralmente, o cálculo do suporte para o conjunto de termos em $\mathcal{D}$ inicia com a exploração de todas as mensagens em $\mathcal{D}$ e cálculo do suporte de cada termo isoladamente. Na próxima iteração, conjuntos de termos 2 (i.e., conjunto de tamanho 2) são enumerados utilizando os conjuntos de termos de tamanho 1, e seus valores de suporte são calculados acessando os dados de treinamento. A pesquisa pelos conjuntos de termos prossegue, e o processo de enumeração é repetido até os valores de suporte, para todos os conjuntos de termos em $\mathcal{D}$, serem finalmente calculados.

Obviamente, o número de regras aumenta exponencialmente com o tamanho do vocabulário (i.e., o número de termos distintos em $\mathcal{D}$) e restrições de custo computacional devem ser impostas durante a extração de regras. Tipicamente, a espaço de pesquisa para regras é restringido a partir de poda de regras que não aparecem frequentemente em $\mathcal{D}$ (i.e., abordagem de suporte mínimo). Enquanto tais restrições fazem a extração de regra factível, elas também levam a perdas nos modelos de classificação, uma vez que algumas regras são podadas e não são incluísas em $\mathcal{R}$.

3.1.2 Predição de Sentimentos

Uma vez que o modelo de classificação $\mathcal{R}$ é extraído a partir de $\mathcal{D}$, regras são coletivamente utilizadas para mensurar os sentimentos das novas mensagens que chegam através de $\mathcal{T}$. Basicamente, o modelo é interpretado como uma votação, na qual cada regra $\{\mathcal{X} \rightarrow s_i\} \in \mathcal{R}$ é um voto dado por $\mathcal{X}$ para o sentimento s_i. Dada uma mensagem $t \in \mathcal{T}$, uma regra $\mathcal{X} \rightarrow s_i$ somente é considerada como um voto válido se esta regra é aplicável para t.

Definição 3. Uma regra $\{\mathcal{X} \rightarrow s_i\} \in \mathcal{R}$ é dita ser aplicável para a mensagem $t \in \mathcal{T}$ se $\mathcal{X} \subseteq t$. Ou seja, se todos termos em $\mathcal{X}$ estão presentes em t.

Nem toda regra em $\mathcal{R}$ é aplicável a uma específica mensagem $t \in \mathcal{T}$. Eventualmente, o modelo pode conter muitas regras que não são aplicáveis à nenhuma mensagem em $\mathcal{T}$. Estas regras são ditas inúteis, e o conjunto de todas as regras inúteis em $\mathcal{R}$ são denotas como $\mathcal{R}_\emptyset$.

Denotamos como $\mathcal{R}_t$ o conjunto de todas as regras em $\mathcal{R}$ que são aplicáveis para a mensagem $t \in \mathcal{T}$. Assim, somente e todas as regras em $\mathcal{R}_t$ são consideradas como votos válidos quando estiverem mensurando os sentimentos na mensagem t. Portanto,

para m mensagens futuras em $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ o modelo de classificação $\mathcal{R}$ pode ser decomposto como $\{\mathcal{R}_{t_1} \cup \mathcal{R}_{t_2} \cup \ldots \cup \mathcal{R}_{t_m} \cup \mathcal{R}_\emptyset\}$. Regras em $\mathcal{R}_\emptyset$ representam um desperdício de recurso computacional, e podem poluir o modelo de classificação com informações irrelevantes, idealmente $|\mathcal{R}_\emptyset| = 0$.

Além disso, denotamos como $\mathcal{R}_t^{s_i}$ o subconjunto de $\mathcal{R}_t$ contendo apenas regras predizendo o sentimento s_i. Votos em $\mathcal{R}_t^{s_i}$ têm pesos diferentes, dependente da confiança das regras correspondentes. É calculada a média dos votos ponderados, por $\theta(\mathcal{X} \rightarrow s_i)$, para o sentimento s_i, dando uma pontuação para o sentimento s_i a respeito à mensagem t, como mostrado na Equação 3.2:

$$s(t, s_i) = \frac{\sum \theta(\mathcal{X} \rightarrow s_i)}{|\mathcal{R}_t^{s_i}|} \tag{3.2}$$

Finalmente, os pontos são normalizados, conforme expresso pela função $\hat{p}(s_i|t)$, como mostrado na Equação 3.3. A função de pontuação estima a probabilidade do sentimento s_i como a atitude implícita na mensagem t.

$$\hat{p}(s_i|t) = \frac{s(t, s_i)}{\sum_{j=0}^{k} s(t, s_j)} \tag{3.3}$$

3.1.3 Adaptação do Modelo Classificação em Tempo Real

O desempenho associado aos modelos de classificação estáticos tende a se deteriorar ao longo do tempo [Hulten et al., 2001]. Isto ocorre principalmente devido ao *sentiment drift* [Widmer & Kubat, 1996], o qual acontece quando a distribuição de dados em $\mathcal{T}$ é diferente do que em $\mathcal{D}$. A diferença geralmente aumenta ao longo do tempo, e em algum ponto no tempo os dados de treinamento podem eventualmente se tornar sem sentido e o modelo de classificação obsoleto [Bifet, 2010].

Drift é comumente observado em ambientes de fluxo contínuo, sendo evidenciado, quando: (1) ocorrem mudanças na distribuição dos sentimentos; ou (2) quando a relação entre padrões textuais e sentimentos muda [Zhang et al., 2008; Forman, 2006], i.e., ocorre mudança na probabilidade condicional ($\theta(\mathcal{X} \rightarrow s_i)$) de um sentimento s_i dado um conjunto de termos $\mathcal{X}$. Nestes casos a adaptação do modelo é essencial para controlar as mudanças nos sentimentos ao longo do tempo.

Existem várias estratégias para utilizar dados não rotulados no auxílio do aprendizado Blum & Mitchell [1998]; Chapelle et al. [2006]; Wu et al. [2006]. Na próxima seção serão apresentadas as estratégias propostas neste trabalho para atualização em tempo real do modelo de classificação utilizando os dados não rotulados

que chegam a partir de $\mathcal{T}$.

3.1.3.1 Inclusão de Novos Dados

A fim de adaptar o modelo de classificação adequadamente, é mandatório coletar as informações mais atuais emergindo no fluxo. As mais recentes mensagens de treinamento podem ser obtidas através da exploração das previsões realizadas utilizando a função de pontuação de sentimentos mostrada na Equação 3.3. Estas previsões podem ser utilizadas para atribuir sentimentos às mensagens, gerando mensagens rotuladas. Além disso, as previsões confiáveis podem ser consideradas como corretas e gerar mensagens rotuladas cofiáveis, as quais podem ser incluídas em $\mathcal{D}$.

Definição 4. Dada uma mensagem arbitrária $t_j \in \mathcal{T}$, dizemos que $< t_j, s_i >$ é uma mensagem rotulada confiável se $\hat{p}(s_i|t_j) \geq l(s_i, t)$. $l(s_i, t)$ representa um limiar adaptativo definido na Equação 3.4. Na equação a constante 0.5 é utilizada para especificar que a primeira mensagem inserida em $\mathcal{D}$ tenha a confiança, na predição, maior que a soma das confianças dos outros sentimentos acompanhados, $m(t)$ é o número de mensagens processadas até a mensagem t e δ_{min} é um fator especificado pelo usuário ($k \geq \delta_{min} \geq 1, 0$), onde k é o número de sentimentos rastreados.

$$l(s_i, t) = \delta_{min} \times \frac{0.5 + \sum_{u=0}^{m(t)-1} \hat{p}(s_i|t_u)}{m(t)} \tag{3.4}$$

O proposito é utilizar $l(s_i, t)$ como um limiar adaptativo, que equivale ao desvio da média de $\hat{p}(s_i|t)$ nas predições anteriores a t, que indica a confiabilidade da predição. Consideramos que o esperado é que a confiança ($\hat{p}(s_i|t)$) seja próxima da média, uma vez que $\hat{p}(s_i|t)$ alcança um valor acima da média existem indícios que a predição está correta e que a mensagem t carrega informações importantes.

Utilizamos o δ_{min} como um fator para indicar o quanto o $\hat{p}(s_i|t)$ deve desviar da média para ser considerada como uma mensagem rotulada confiável e inclui-la nos dados de treinamento $\mathcal{D}$. Esta estratégia visa também evitar o alto viés para um sentimento, em $\mathcal{D}$, durante a classificação das mensagens. Isso porque o surgimento repentino de um grande volume de mensagens com um mesmo sentimento pode fazer com que o treino $\mathcal{D}$ seja composto, na sua maioria, por mensagens da uma mesma classe, tornando inviável a auto inclusão de mensagens com algum sentimento contrário.

Intuitivamente, se as previsões são de fato confiavelmente corretas, os dados de treinamento serão continuamente aumentados com novas informações de treinamento, mantendo os dados de treinamento atualizados como a evolução do fluxo. No entanto,

o uso de suporte baseado em poda durante a extração das regras impede o pleno potencial do auto aumento do treino, uma vez que é altamente provável que o modelo de classificação $\mathcal{R}$ será composto apenas das regras mais gerais em $\mathcal{D}$, e a maioria destas regras podem não ser aplicáveis a mensagens futuras que transportam informações de tendências.

Definição 5. Dada uma mensagem $t \in \mathcal{T}$, dizemos que o modelo $\mathcal{R}$ é agnóstico para t, se $\mathcal{R}_t = \emptyset$. Ou seja, se $\mathcal{R}$ não contém regras que são aplicáveis a t. Um modelo $\mathcal{R}$ é dito gnóstico se $\mathcal{R}_t \neq \emptyset \ \forall t \in \mathcal{T}$.

Considerando que um valor de suporte mínimo ideal que garante um modelo de classificação gnóstico é improvável de existir, como discutiremos na próxima seção, nossa extração de regras deve ser livre de suporte, a fim de produzir modelos gnósticos e explorar todo potencial e benefícios do auto aumento de treinamento.

3.1.3.2 Extração de Regras *Online* com Projeção de Dados

Até agora discutimos a extração de regras *offline*, entretanto, a extração de regras *online* (em tempo real) oferece várias vantagens. Uma dessas vantagens é que os classificadores se tornam capazes de extrair eficientemente regras a partir de $\mathcal{D}$ sem a aplicação de poda baseada em suporte. A ideia por trás de extração de regras *online* é evitar completamente a extração de regras inúteis, projetando os dados de treinamento sob demanda [Veloso & Meira Jr., 2011]. Mais especificamente, a extração de regras é adiada até que uma mensagem $t \in \mathcal{T}$ é dada. Em seguida, os termos em t são utilizados como um filtro que configura os dados de treino em $\mathcal{D}$ de uma maneira que apenas regras que são aplicáveis a t podem ser extraídas. Este processo de filtragem produz um conjunto de treinamento projetado, denotado por $\mathcal{D}_t$, que contém apenas termos que estão presentes na mensagem t.

Lema 1 Todas as regras extraídas de $\mathcal{D}_t$ são aplicáveis a t.

Prova Uma vez que todas as mensagens de treinamento em $\mathcal{D}_t$ contem apenas termos que estão presentes na mensagem t, a existência de uma regra $\mathcal{X} \rightarrow s_i$ extraída a partir de $\mathcal{D}_t$, tal que $\mathcal{X} \nsubseteq t$, é impossível. ■

O Lema 1 implica que a projeção de treino sob demanda assegura que $|\mathcal{R}_\emptyset| = 0$, evidenciando que apenas as regras inúteis não estão incluídas no modelo de classificação $\mathcal{R}$. O próximo teorema afirma que o classificador extrai de forma eficiente regras a partir

de $\mathcal{D}$, não importando o valor de suporte mínimo (que pode ser arbitrariamente baixo). A intuição chave é que o classificador atua somente em termos que são conhecidos por estarem associados uns aos outros, diminuindo drasticamente o espaço de busca das regras.

Teorema 1 O número de regras extraídas a partir de $\mathcal{D}_t$ cresce polinomialmente com o número de termos distintos em $\mathcal{D}$.

Prova Seja n o número de termos distintos em $\mathcal{D}$. Uma vez que uma mensagem arbitrária $t \in \mathcal{T}$ contém no máximo l termos (com $l \ll n$), então qualquer regra aplicável a t pode ter no máximo l termos no seu antecedente. Isto é, para qualquer regra $\{\mathcal{X} \rightarrow s_i\}$, tal que $\mathcal{X} \subseteq t$, $|\mathcal{X}| \leq l$. Consequentemente, o número de possíveis regras que são aplicáveis a t é $l + \binom{l}{2} + \ldots + \binom{l}{l} = O(2^l) \ll O(n^l)$. Assim, o número de regras aplicáveis aumenta polinomialmente em n. ■.

Outra vantagem proporcionada pela projeção de treino sob demanda vem do fato de que a projeção também explora, como um efeito colateral, a localidade temporal associada aos termos em $\mathcal{D}$. Isto é particularmente importante para lidar com o *sentiment drift*, uma vez que, projetando os dados de treinamento de acordo com o conteúdo de uma mensagem $t \in \mathcal{T}$, o classificador está essencialmente concentrando-se nas informações para treinamento representativas em quadros temporais de dados praticamente contínuos. No entanto, decidir o quão recentes estes quadros devem ser é uma questão complicada, uma vez que diferentes mensagens em $\mathcal{T}$ podem exigir quadros de treinamento posicionados em diferentes pontos da linha do tempo do fluxo.

Ou seja, algumas mensagens em $\mathcal{T}$ podem exigir quadros de treinamento mais recentes, enquanto as outras mensagens em $\mathcal{T}$ pode exigir mais antigas. Assim, em vez de empregar um tamanho fixo de quadros temporais para todas as mensagens em $\mathcal{T}$ (o que equivale ao uso de uma janela deslizante de tamanho fixo), nosso classificador emprega um quadro diferente (ou seja, $\mathcal{D}_t$) para cada mensagem $t \in \mathcal{T}$. A proximidade no tempo dos quadros de treinamento para uma mensagem arbitrária t é decidido com base nos termos que estão na própria mensagem. Uma vez que estes termos são cronologicamente relacionados de alguma forma, os dados de treinamento projetados a partir de $\mathcal{D}_t$ são susceptíveis a conter mensagens de treinamento representativas para mensurar os sentimentos na mensagem t.

3.1.3.3 Estendendo Modelos de Classificação Dinamicamente

Com a extração de regras *online*, nós estendemos o modelo de classificação $\mathcal{R}$ dinamicamente à medida que as mensagens em $\mathcal{T}$ são processadas. Inicialmente $\mathcal{R}$ está vazio, um submodelo $\mathcal{R}_{t_i}$ é anexado a $\mathcal{R}$ a cada momento que o classificador processa uma mensagem t_i. Assim, após o processamento de uma sequência de m mensagens $\{t_1, t_2, \ldots, t_m\}$, o modelo $\mathcal{R}$ é $\{\mathcal{R}_{t_1} \cup \mathcal{R}_{t_2} \cup \ldots \cup \mathcal{R}_{t_m}\}$, e, portanto, $\mathcal{R}$ é gnóstico a todas as m mensagens.

Produzir um submodelo $\mathcal{R}_t$ envolve extração de regras a partir de $\mathcal{D}_t$. Esta operação tem um custo computacional significativo, uma vez que é necessário executar vários acessos a $\mathcal{D}$. Diferentes mensagens em $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ podem exigir diferentes submodelos $\{\mathcal{R}_{t_1}, \mathcal{R}_{t_2}, \ldots, \mathcal{R}_{t_m}\}$, mas diferentes submodelos podem compartilhar algumas regras (i.e., $\{\mathcal{R}_{t_i} \cap \mathcal{R}_{t_j}\} \neq \emptyset$). Neste caso, a memorização é muito eficaz para evitar a replicação de trabalho, reduzindo o número de operações de acesso a dados.

Dessa forma, antes de extrair a regra $\mathcal{X} \rightarrow s_i$, o classificador verifica se esta regra já está em $\mathcal{R}$. Se uma entrada é encontrada com a chave correspondente a $\{\mathcal{X}, s_i\}$, então a regra em $\mathcal{R}$ é utilizada ao invés de extraí-la a partir de $\mathcal{D}_t$. Se ela não for encontrada, a regra é extraída a partir de $\mathcal{D}_t$ e depois ela é inserida em $\mathcal{R}$. As principais etapas deste processo são resumidas no Algoritmo 1.

Algoritmo 1 Extração de Regras *Online*

Require: mensagem $t \in \mathcal{T}$ e $\mathcal{D}$
Ensure: $\mathcal{R}_t$ e $\mathcal{R}$

1: $\mathcal{D}_t \Leftarrow \mathcal{D}$ projetado de acordo com os termos em t
2: $\mathcal{R}_t \Leftarrow$ regras $\{\mathcal{X} \rightarrow s_i\} \notin \mathcal{R}$, extraídas a partir de $\mathcal{D}_t$
3: $\mathcal{R} \Leftarrow \mathcal{R}_t \cup \mathcal{R}$

3.1.3.4 Manutenção Incremental do Modelo

As entidades no modelo de classificação $\mathcal{R}$ podem se tornar inválidas quando mensagens rotuladas confiáveis $< t, s_i >$ são incluídas em $\mathcal{D}$. Como resultado, $\mathcal{R}$ deve ser atualizado corretamente. Propomos manter o modelo atualizado de forma incremental, de modo que o modelo atualizado seja exatamente o mesmo que seria obtido através da sua reconstrução total.

A velocidade de atualização é uma questão chave na manutenção do modelo e um desafio que ameaça a eficácia de nossa abordagem, uma vez que, o modelo pode ser composto de um número potencialmente grande de regras, e atualizar todas essas

regras podem ser inaceitavelmente caro em um ambiente de fluxo de dados. Entretanto, nem todas as regras em $\mathcal{R}$ precisam ser atualizadas.

Lema 2. A inclusão de uma mensagem rotulada $< t, s_i >$ em $\mathcal{D}$ não altera o valor de $\sigma(\mathcal{X})$, para qualquer conjunto de termos $\mathcal{X} \not\subset t$

Prova. Como $\mathcal{X} \not\subset t$, o número de mensagens em $\mathcal{D}$ tendo $\mathcal{X}$ como um subconjunto é essencialmente o mesmo que em $\{\mathcal{D} \cup t\}$. ■

Lema 3. A inclusão de uma mensagem rotulada $< t, s_i >$ em $\mathcal{D}$ não alterar o valor de $\theta(\mathcal{X} \rightarrow s)$, para qualquer regra de $\{\mathcal{X} \rightarrow s\} \in \mathcal{R}$ para o qual $\mathcal{X} \not\subset t$, $\forall s \in \{s_1, s_2, \ldots, s_k\}$.

Prova. Está diretamente relacionada ao fato de que a confiança é invariante sob a operação de adição nula [Tan et al., 2002]. ■

A partir dos Lemas 2 e 3, o número de regras que devem de ser atualizadas, devido à inclusão de uma mensagem rotulada $< t, s_i >$, é limitado pelo número de possíveis termos em t. Como a maioria das mensagens que são incluídas em $\mathcal{D}$ contém apenas uma fração muito pequena do total de termos possíveis, a inclusão de uma mensagem arbitrária t corresponde a uma adição nula para a maioria das regras em $\mathcal{R}$. O lema a seguir estabelece exatamente as regras em $\mathcal{R}$ que devem que ser atualizadas.

Lema 4. As únicas e todas as regras em $\mathcal{R}$ que devem ser atualizadas devido à inclusão de uma mensagem rotulada $< t, s_i >$ são aquelas em $\mathcal{R}_t$.

Prova. Todas as regras $\{\mathcal{X} \rightarrow s_i\} \in \mathcal{R}$ que devem que ser atualizado devido à inclusão de $< t, s_i >$ são aquelas que $\mathcal{X} \subseteq t$. Por definição, $\mathcal{R}_t$ contém apenas e todas as regras extraídas para t. ■

Uma vez que as regras $\{\mathcal{X} \rightarrow s\} \in \mathcal{R}_t$ são recuperadas a partir de $\mathcal{R}$, a atualização dos valores correspondentes a $\sigma(\mathcal{X})$ e $\theta(\mathcal{X} \rightarrow s)$ é uma operação simples. Basta iterar sobre $\mathcal{R}_t$ e incrementar os valores de $\sigma(\mathcal{X})$ e $\sigma(\mathcal{X} \cup s)$. Os valores correspondentes a $\theta(\mathcal{X} \rightarrow s)$ são obtidos através do cálculo de $\frac{\sigma(\mathcal{X} \cup s)}{\sigma(\mathcal{X})}$ $\forall s \in \{s_1, s_2, \ldots, s_k\}$. Essas etapas estão resumidas no Algoritmo 2.

Algoritmo 2 Manutenção Incremental do Modelo

Require: mensagem rotulada $< t, s_i >$, $\mathcal{D}$, e $\mathcal{R}_i$
Ensure: $\mathcal{R}$

1: **for all** regras $\{\mathcal{X} \rightarrow s\} \in \mathcal{R}_t$ **do**
2: incremente $\sigma(\mathcal{X})$
3: incremente $\sigma(\mathcal{X} \cup s_i)$
4: $\theta(\mathcal{X} \rightarrow s_i) \Leftarrow \frac{\sigma(\mathcal{X} \cup s_i)}{\sigma(\mathcal{X})}$
5: **end for**

3.1.3.5 Estratégia de Sub judice

Dada a restrição de que $\hat{p}(s_i|t)$ deve ser maior ou igual a $l(s_i, t)$ para que a mensagem t seja inserida em $\mathcal{D}$, algumas predições podem não ser confiáveis o suficiente, dado certo valor de $l(s_i, t)$ (i.e., $\hat{p}(s_i|t) < \delta_{min}$). A estratégia de *sub judice*[1] consiste em não utilizar tais predições duvidosas e não informar a predição realizada para a mensagem imediatamente, como o classificador não tem evidências suficientes para realizar um julgamento confiável, as mensagens são mantidas em uma fila de espera $\mathcal{S}$, ou seja, mantidas em *sub judice*.

Quando novas mensagens confiáveis são inclusas em $\mathcal{D}$, novas evidências de sentimentos são exploradas, esperançosamente aumentando a confiança das mensagens em $\mathcal{S}$ que foram previamente mantidas em *sub judice*, fazendo que elas possam ser classificadas. Mais especificamente, quando uma mensagem rotulada confiável é incluísa em $\mathcal{D}$, o classificador reavalia todas as mensagens que estão em $\mathcal{S}$ (*sub judice*). No fim do processo, ou as mensagens duvidosas se tornam confiáveis (provavelmente melhorando a desempenho da predição), ou não existem mais mensagens rotuladas confiáveis para serem incluídas em $\mathcal{D}$ e, portanto, as mensagens restantes que estão em *sub judice* devem ser processadas normalmente. O processo termina quando todas as mensagens em $\mathcal{T}$ são processadas pelo classificador. Os passos principais estão resumidos no Algoritmo 3.

As mensagens são mantidas em *sub judice* em uma fila de tamanho $|\mathcal{S}|$, e são retiradas de *sub judice* em dois casos: (1) Quando $\hat{p}(s_i|t)$, $t \in \mathcal{S}$, se torna maior ou igual a $l(s_i, t)$; e (2) Quando o número de mensagens em *sub judice* for igual a $|\mathcal{S}|$, necessariamente, a primeira mensagem da fila é removida e processada.

A estratégia de *sub judice* é uma importante ferramenta para tratar o *sentiment drift*, uma vez que, na ocorrência de um *drift* é esperado que a confiança na predição diminua, uma vez que a entropia dos dados aumenta nesse momento como demonstrado

[1] *Sub judice* é uma expressão do latim que designa um caso ou problema que ainda está sob a apreciação jurídica (sem uma sentença final)

Algoritmo 3 Procedimento de *Sub Judice*

Require: message $t \in \mathcal{T}$, δ_{min}
Ensure: $\mathcal{D}$

1: **if** $\hat{p}(s_i|t) < \delta_{min}$
2: manter t *sub judice* até outra mensagem rotulada ser inclusa em $\mathcal{D}$
3: **else**
4: incluir a mensagem rotulada $< t, s_i >$ em $\mathcal{D}$
5: **end if**

em [Vorburger & Bernstein, 2006; Abdulsalam et al., 2008] e evidenciaremos na Seção 3.2.2. E assim o classificador pode aguardar até que o conceito se estabilize para realizar a predição.

3.2 Avaliação Experimental

Nesta seção analisamos empiricamente a eficácia da solução proposta na pontuação de sentimentos. A seguir serão apresentadas as coleções de dados utilizadas nos experimentos.

3.2.1 Coleções de Dados

Esta seção apresenta a caracterização das coleções de dados criadas para avaliação da solução proposta. As principais características das coleções são apresentadas na Tabela 3.1. A partir destas coleções foi possível avaliar a solução proposta em diferentes idiomas, tipos de sentimentos rastreados e em cenários que os sentimentos mudam de diferentes maneiras no decorrer do tempo (i.e., diferentes tipos de *sentiment drift*).

Os gráficos da Figura 3.2 evidenciam a ocorrência de mudanças na distribuição de sentimentos nas coleções analisadas. Nestes gráficos em cada janela de tempo t (minutos e dias) calculamos a porcentagem de mensagens positivas em t. Essas mudanças podem afetar fortemente o desempenho de muitos classificadores [Bifet, 2010]. A seguir é detalhada cada coleção de dados e o evento a que ela se refere.

Tabela 3.1. Coleções de Dados

Coleção	Idioma	# Mensagens	Sentimentos Rastreados	Taxa de Chegada de Mensagens por Segundo
Eleições Presidenciais no Brasil	Português	66.643	Positivo e negativo	0.02
Personalidade do Ano	Inglês	5.616	Aprovação, surpresa, sarcasmo, reprovação e revolta	0.2
Copa do Mundo de Futebol	Português	3.214	Positivo e negativo	1.12
Copa do Mundo de Futebol	Inglês	1.432	Positivo e negativo	

Copa do Mundo de Futebol. Na Copa do Mundo de 2010 foram convocadas 32 equipes para competir pelo título. Após um jogo polêmico, a equipe brasileira foi derrotada pela equipe holandesa em 02/07/2010. O Brasil marcou um gol no início da partida, mas logo após a equipe Holandesa marcou duas vezes e venceu o jogo. Um jogador específico, Felipe Melo, teve participação decisiva nos três gols.

Desta forma, a coleção sobre a copa do mundo de futebol em 2010 reflete a opinião das pessoas em relação ao jogador Felipe Melo no último jogo da seleção Brasileira contra a Holanda. Foram coletadas 12.020 mensagens referentes ao Felipe Melo. Selecionamos aleatoriamente 3.214 mensagens em português (8.101 termos distintos) e

1.432 mensagens em Inglês (4.963 termos distintos) e as rotulamos manualmente. A distribuição de frequência dos termos é apresenta na Figura 3.1.

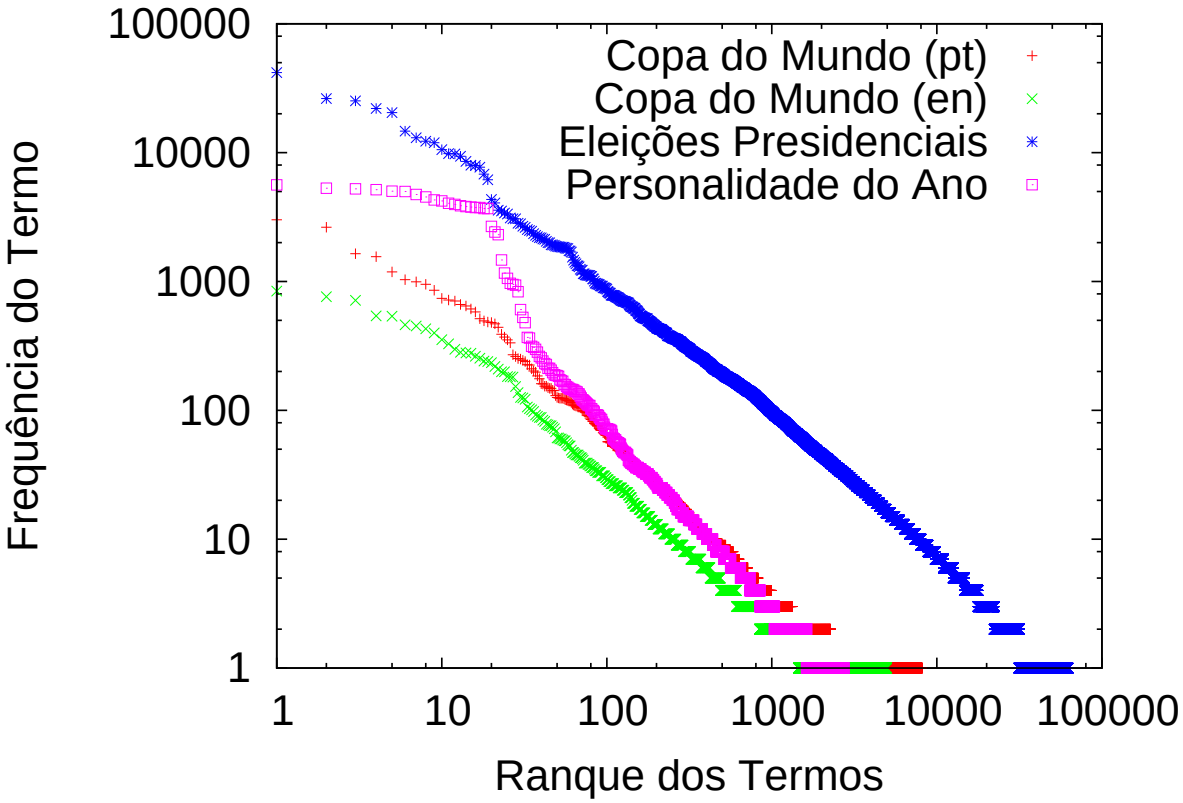
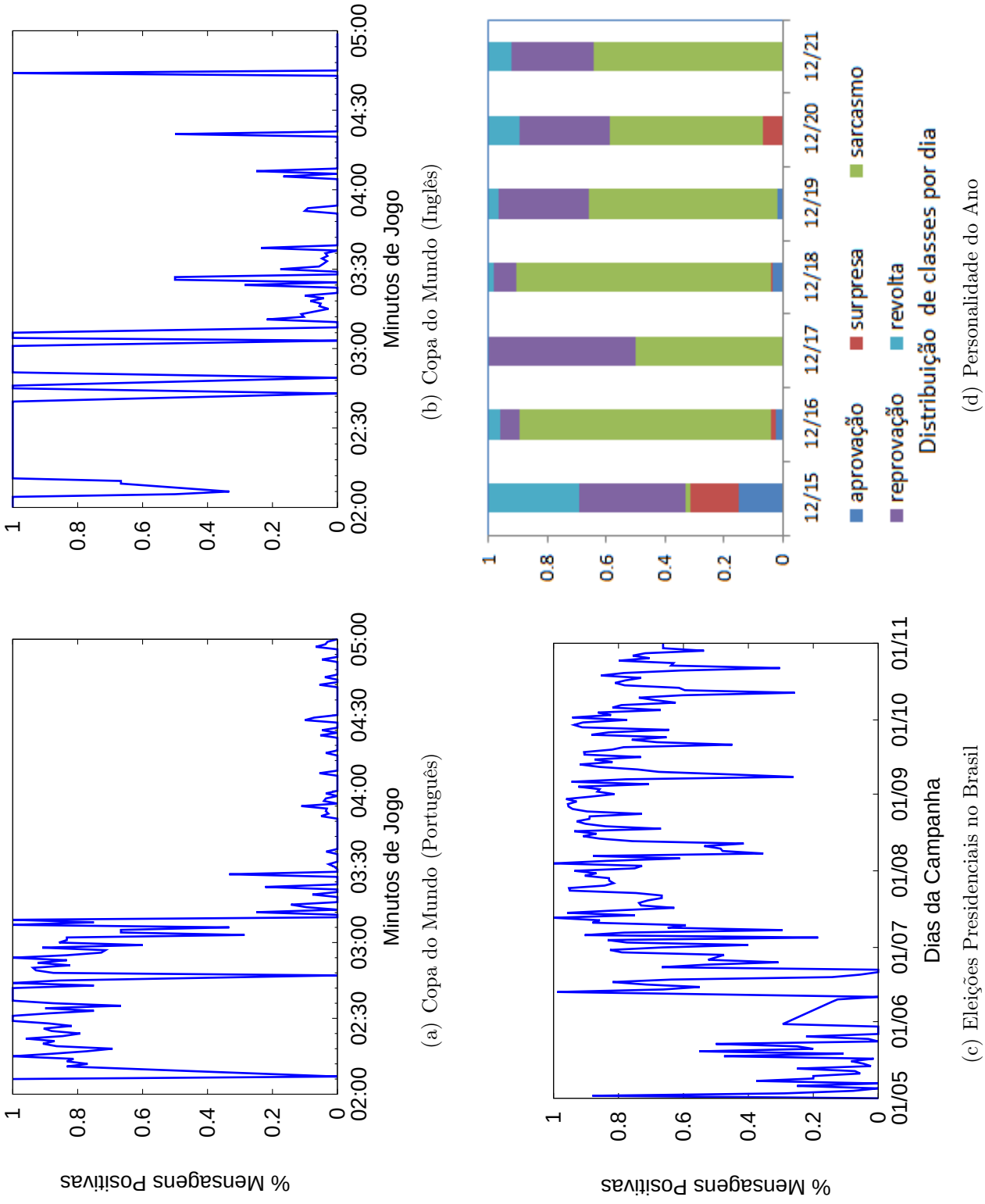


Figura 3.1. Distribuição de Frequência dos Termos

Na Figura 3.2(a), por exemplo, é possível verificar que a opinião dos usuários do Twitter, que postaram mensagens em português, em relação ao jogador Felipe Melo mudou durante a partida. No primeiro tempo do jogo é possível observar uma maior porcentagem de mensagens positivas em relação ao jogador uma vez que ele foi o responsável pelo passe que finalizou em um gol a favor da seleção Brasileira. Contudo, no segundo tempo, houve uma mudança inesperada de opinião - aumento de mensagens negativas. Essa mudança reflete o sentimento das pessoas a partir do momento em que o jogador faz um gol contra e posteriormente é expulso. O sentimento negativo tende a se intensificar no final da partida, isso porque o Brasil foi eliminado da copa nessa ocasião e a responsabilidade da derrota foi atribuída ao jogador.

Na Figura 3.2(b), é possível verificar a distribuição dos sentimentos nas mensagens em inglês. É possível perceber que a variação dos sentimentos é bastante ruidosa, em outras palavras, ao mesmo tempo em que muitas pessoas se mostram

felizes a respeito da derrota do Brasil, outras se declaram tristes, ou vice-versa. Isso pode ser explicado porque as pessoas que postam mensagem em inglês não têm uma opinião forte a respeito da partida, ou seja, a opnião da população de língua inglesa é, de certa forma, mais diversificada ou não existe um consentimento gobal e então o sentimento pode variar de maneira inconsistente.



(a) Copa do Mundo (Português)

(b) Copa do Mundo (Inglês)

(c) Eleições Presidenciais no Brasil

(d) Personalidade do Ano

Figura 3.2. Distribuição de classes ao longo do tempo

Eleições Presidenciais no Brasil. A campanha eleitoral a presidência do Brasil ocorreu no período de Junho a Outubro de 2010. Nesta base, o sentimento monitorado foi em relação a candidata Dilma Rousseff, que utilizou o Twitter como uma das principais fontes de informação para seus eleitores. A campanha atraiu mais de 500.000 seguidores e a Dilma foi a segunda pessoa mais citada no Twitter em 2010 [2]. As eleições chegaram ao segundo turno e Dilma Rousseff ganhou as eleições com 56% de votos.

Nessa coleção as mensagens foram classificadas como negativas ou positivas em função da candidata Dilma Rousseff. Nós coletamos 466.724 mensagens e destas anotamos 66.643 mensagens em português referentes a então candidata para acompanhar o sentimento de aprovação da população durante esse período. Este sentimento de aprovação varia fortemente durante este período devido a várias polêmicas e ataques políticos. Por isso nosso objetivo foi medir a aprovação da Dilma durante sua campanha. A coleção de dados contem 62.089 termos distintos (a distribuição de frequência é apresentada na Figura 3.1). As mensagens no fluxo chegam em uma taxa de 0,02 mensagens por segundo.

Nas eleições presidenciais de 2010 no Brasil Dilma começou com uma menor pretensão de votos, segundo as pesquisas, e sua popularidade aumentou gradativamente até o fim das eleições que terminou com a Dilma vencedora. Podemos ver comportamento semelhante no gráfico da Figura 3.2(c), onde a porcentagem de mensagens positivas começa baixa e aumenta ao decorrer do tempo.

Personalidade do Ano. No fim de todos os anos a revista TIME seleciona a personalidade do ano ou um grupo de pessoas que mais influenciaram eventos durante o ano. A personalidade do ano para 2010 foi Mark Zuckerberg (fundador do Facebook[3]). O leitor, contudo, escolheu o Julian Assange (criador do Wikileaks[4]) com uma grande superioridade de votos. Esta divergência causou o que ficou conhecido como a *Twittersphere Battle: Zuckerberg vs. Assange*[5].

Nesta base as mensagens foram classificadas em relação ao sentimento das pessoas diante da escolha de Mark Zuckerberg como merecedor do prêmio ao invés de Julian Assange, foram acompanhados os sentimentos de aprovação, reprovação, surpresa, revolta e sarcasmo. Coletamos 93.411 mensagens em inglês referentes ao Julian Assange e o Mark Zuckerberg de 15/12/2010 a 21/12/2010. Selecionamos aleatoriamente 5.616 destas mensagens e anotamos manualmente com intuito de acompanhar diferentes

[2] http://yearinreview.twitter.com/2010/trends/

[3] https://www.facebook.com

[4] http://www.wikileaks.org

[5] http://www.cbsnews.com/8301-505123_162-40043859/twittersphere-battle-is-still-on-zuckerberg-vs-assange/

sentimentos a respeito da decisão da revista para a personalidade do ano.

Na Figura 3.2(d) é possível verificar que no primeiro dia o percentual de mensagens que aprovam e reprovam a decisão da TIME é próximo (15% e 36% respectivamente), isso pode indicar uma disputa entre as pessoas com diferentes opiniões sobre a decisão da revista. Contudo, nos dias seguintes o sentimento de sarcasmo aparece com maior frequência, seguido pelo sentimento de reprovação. Isso nos faz crer que no primeiro dia, quando a noticia começava a ser propagada, houve um número semelhante de pessoas que aprovavam e reprovavam a decisão, contudo com o passar do tempo mais pessoas assumem a posição de reprovação sobre a decisão da TIME se considerarmos as mensagens de sarcasmos como um tipo de reprovação (e.g., *sorry julian assange I guess you didn't violate enough peoples privacy*).

Essa variação do sentimento (Figura 3.2) nos permite ilustrar a correlação entre as mensagens do Twitter e os sentimentos no mundo real e nos permite levantar a hipótese da existência do fenômeno conhecido como *concept drift* (que neste no cenário de analise de fluxos de sentimentos estamos chamando de *sentiment drift*) nos dados. Na próxima seção apresentamos uma caracterização das mudanças e *sentiment drift* ocorrido no fluxo durante os eventos.

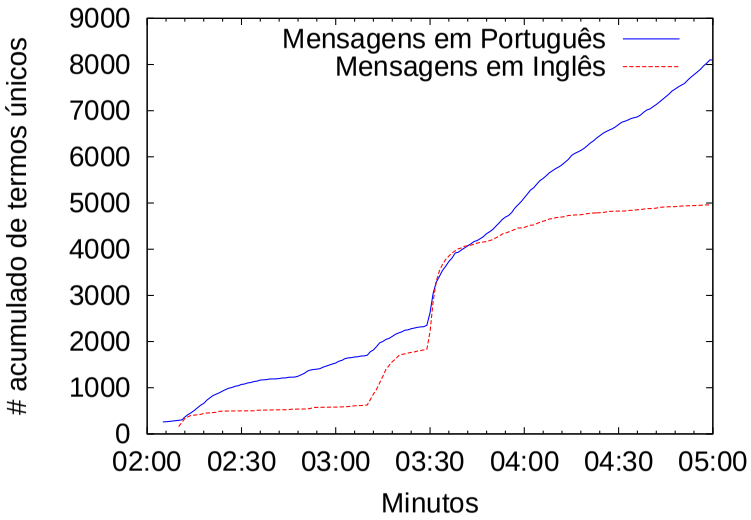
3.2.2 Caracterização do *Sentiment Drift*

Conforme informado anteriormente, *sentiment drift* é o termo que introduzimos para definir o *concept drift* em fluxos de sentimento. O *concept drift* é definido como mudanças ocorridas na probablidade prévia de um sentimento ($p(s_i)$), na probabilidade condicional ($p(s_i|X)$) ou em ambas ao mesmo tempo ao decorrer das mensagens do fluxo $\mathcal{T}$ [Zhu et al., 2010; Zhang et al., 2008].

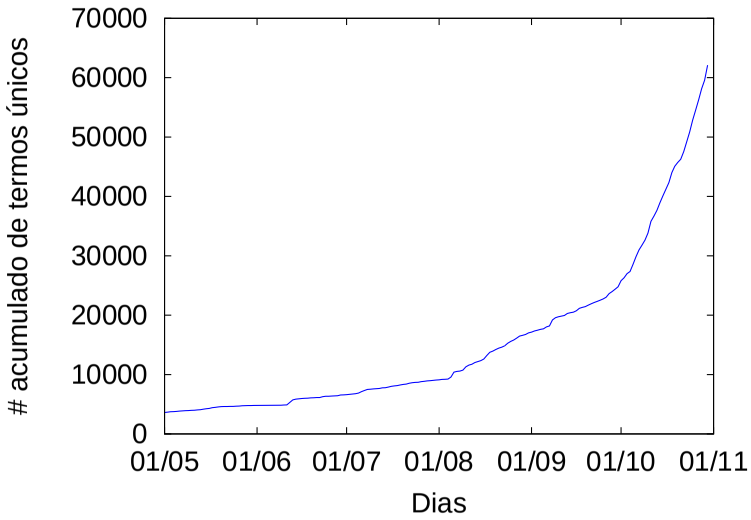
Na seção 3.2.1 discutimos, além das características dos eventos analisados, como a distribuição dos sentimentos mudou em relação as entidades e tópicos. Essas mudanças se tornam explícitas ao analisarmos os gráficos da Figura 3.2, que ilustram a mudança na probabilidade do sentimento ($p(s_i)$). Nesta seção vamos demostrar as mudanças ocorridas na probabilidade condicional (denotada aqui como $\theta(\mathcal{X} \rightarrow s_i)$), i.e., probabilidade do sentimento s_i dado um termo q.

Iniciamos nossa análise verificando como é o crescimento do vocabulário ($\mathcal{V}$) com o chegar das mensagens a partir de $\mathcal{T}$, ou seja, o número de termos únicos em cada coleção ao longo do tempo. Podemos ver na Figura 3.3(a) o crescimento do número de termos das coleção referente à Copa do Mundo. Notamos que nos momentos em que ocorrem mudanças na distribuição dos dados, demonstrado na Figura 3.2, ocorre também o surgimento de um grande número de novos termos (ver Figura 3.3(a)), então

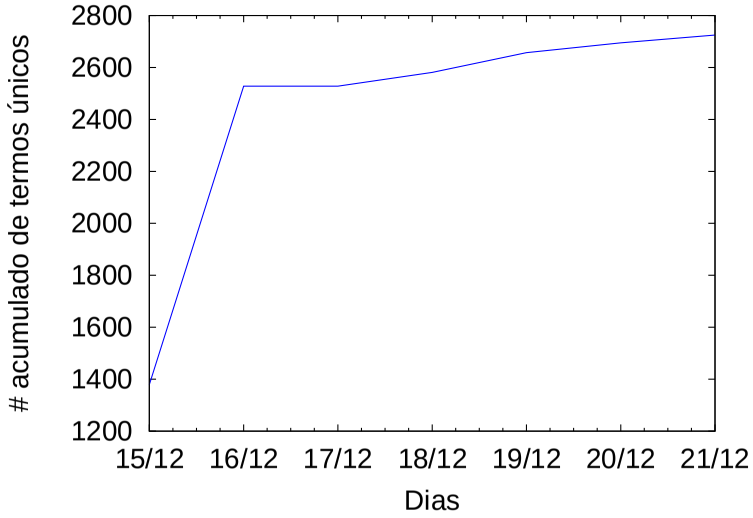
além do classificador se deparar com o *sentiment drift*, existe um outro problema que é a falta de informação sobre esse grande volume de termos que surgiram repentinamente, essa evidência reforça na necessidade de atualização continua do modelo com mensagens mais recentes de $\mathcal{T}$.



(a) Copa do Mundo



(b) Eleições Presidenciais no Brasil



(c) Personalidade do Ano

Figura 3.3. Número de Termos Únicos Acumulado ao Decorrer do Tempo

Esse surgimento de muitos termos novos é uma evidência da necessidade da estratégia de *sub judice*, uma vez que quando um novo termo surge o classificador não possui as informações necessárias para jugar o sentimento da mensagem. Mas a medida em que novas mensagens chegam, o contexto em torno do termo desconhecido pode ficar mais claro e aumentar a confiança da predição.

Nas Figuras 3.3(b) e 3.3(c), que representam respectivamente as bases referentes as Eleições Presidências e Personalidade do Ano, é possível verificar que existem dois pontos em que o número de termos cresce subitamente, na Figura 3.3(b) isto acontece no início do mês 10 e na Figura 3.3(c) no primeiro dia. Se analisarmos as Figuras 3.2(c) e (d), é possível observar claramente que existe uma mudança na distribuição dos sentimentos do primeiro para o segundo dia em relação a personalidade do ano, na Figura 3.2(d), e no que se refere as eleições presidenciais, mesmo que esse mudança não seja muito brusca, existe um diminuição dos sentimentos positivos no início do mês dez, conforme demonstrada na Figura 3.2(c).

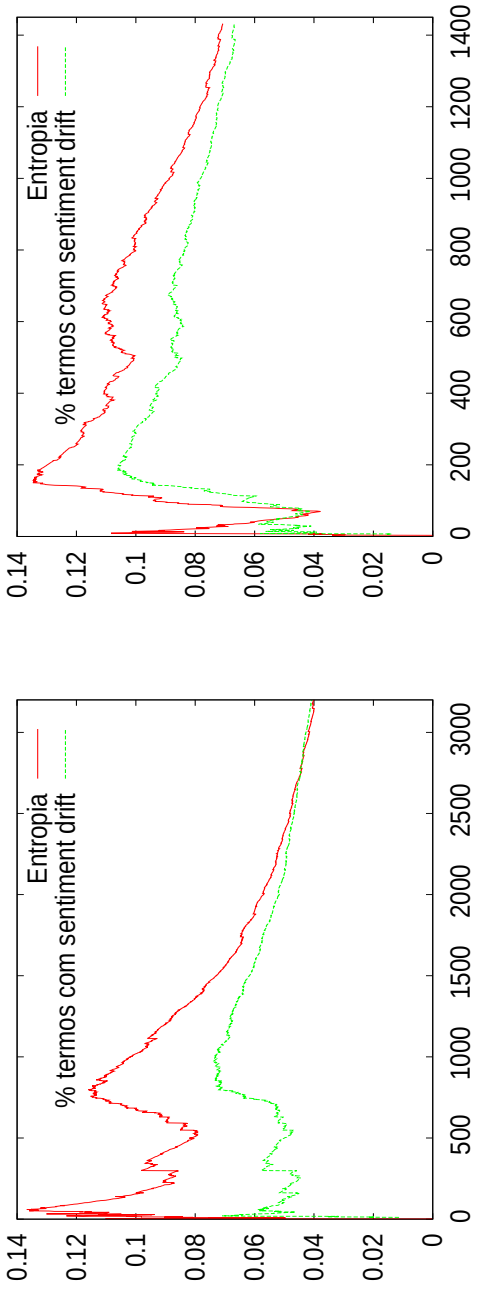
Na Figura 3.4 ilustramos como a probabilidade condicional, de um sentimento dado um termo q, muda ao passar do fluxo. Para isso calculamos a probabilidade de um sentimento dado um termo, utilizando $\theta(q \rightarrow s_i)$, e definimos para cada termo o seu sentimento a partir do sentimento com maior $\theta(q \rightarrow s_i)$. Desta forma, apresentamos a porcentagem de termos (q) que mudaram o sentimento mais provável, com o chegar das mensagens de $\mathcal{T}$ e o surgimento novos termos.

Na Figura 3.4 é apresentado também a entropia das coleções ao longo do tempo. A entropia [Shannon, 2001] é utilizada para medir a desordem ou aleatoriedade associada a uma variável. Para encontrar a entropia de uma coleção de dados com a distribuição conhecida é utilizada a Equação 3.5. Calculamos a entropia de $\mathcal{T}$ a cada nova mensagem processada utilizando a Equação 3.6, como realizado em [Vorburger & Bernstein, 2006; Abdulsalam et al., 2008], a qual consiste basicamente da média da entropia de todos os termos visto até o momento.

Através do gráfico da Figura 3.4(a) é possível observar que existe um pico próximo a mensagem 1.000, já na Figura 3.4(b) podemos ver esse comportamento próximo a mensagem 250. Este é o momento que ocorreu o súbito *sentiment drift* mostrado nas Figuras 3.2(a) e 3.2(b), onde uma grande mudança em $p(s_i)$ acarretou também mudança na probabilidade da mensagem pertencer a um sentimento dado um termo e crescimento da entropia.

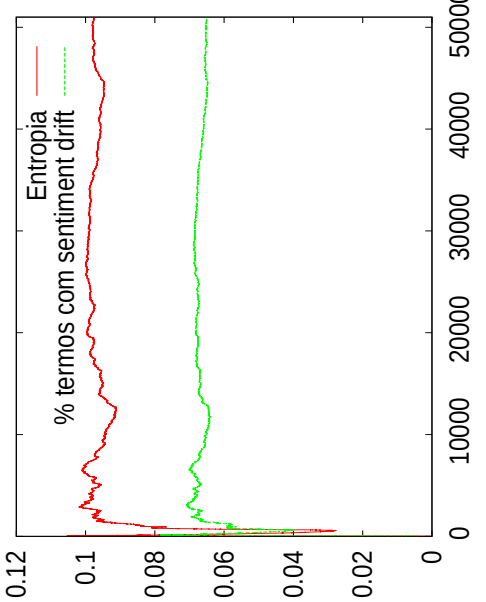
Neste caso, para as coleções da Copa do Mundo (Português e Inglês), um classificador treinado com uma amostra anterior a mensagem 1.000 (na versão em Português) e a mensagem 250 (na base em Inglês) teria sua eficácia fortemente abalada na predição do sentimento das mensagens após essas mensagens. Ao considerarmos o

caso da coleção em Português, após a mensagem 1.000, novos termos surgem tornando necessária a atualização do modelo com mensagens mais recentes para que a eficácia do classificador não se deteriore rapidamente. Em contraste, como apresentado na entropia, se novas mensagens forem inseridas a predição dos sentimentos das mensagens após mensagem 1.000 se torna cada vez mais fácil.

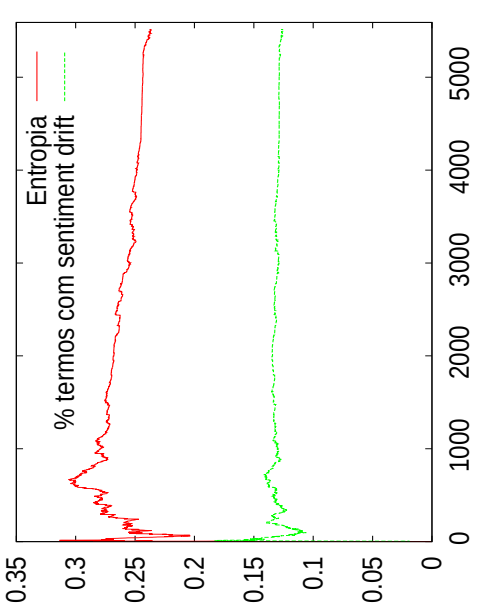


(a) Copa do Mundo (Português)

(b) Copa do Mundo (Inglês)



(c) Eleições Presidenciais no Brasil



(d) Personalidade do Ano

Figura 3.4. % termos que o maior valor de $\theta(q \rightarrow s_i)$ variou entre diferentes sentimentos e entropia ao decorrer longo do fluxo

Nas Figuras 3.4(c) e (d), é possível verificar que existe uma variação na quantidade de termos que mudaram de sentimento no começo do evento. Contudo, não existe um acontecimento que faça essa porcentagem mudar repentinamente. É preciso ressaltar que mesmo assim pelo menos 7% dos termos mudaram de sentimento pelo menos uma vez durante as Eleições Presidenciais no Brasil e 14% na coleção Personalidade do Ano. Na coleção Personalidade do Ano a entropia é bem mais alta nas primeiras mensagens de $\mathcal{T}$, isto indica que este será o intervalo que apresentará maior dificuldade para predição dos sentimentos, uma vez que após a mensagem 1.000 a entropia inicia um decaimento.

$$H(p(\mathcal{T}|q)) = -\sum_{i=1}^{k} p(s_i|q) \times \log p(s_i|q) \tag{3.5}$$

$$entropia = \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} H(p(\mathcal{T}|q_i)) \tag{3.6}$$

Em resumo foi possível observar que um *sentiment drift* vem acompanhado de surgimentos de muitos novos termos únicos, aumento da entropia e que realmente a probabilidade condicional é afetada com o passar do fluxo. Esta última característica é muito relevante uma vez que a probabilidade condicional é utilizada em muitos classificadores para realizar a predição, além disso, ela é a evidencia mais consistente de que ocorreu uma mudança nos padrões relacionados a um sentimento.

3.2.3 Resultados Obtidos e Discussão dos Resultados

Nesta seção discutimos os resultados obtidos com a solução proposta e a comparamos com versões estáticas de alguns dos classificadores mais populares encontrados na literatura, os quais consideramos como um limite inferior. Empregamos o erro quadrático médio (do inglês *Mean-Squared Error* - MSE [Ben-Haim & Eldar, 2009]) como medida básica de avaliação nos experimentos, o erro em cada mensagem foi calculado como $1 - \hat{p}(s_c|t)$ tal que s_c é o sentimento contido em t. O MSE foi utilizado para verificar a capacidade do algoritmo de quantificar a proporção de cada sentimento presente na mensagem analisada, uma vez que esse é o objetivo da análise de sentimento e, além disto, uma mensagem pode conter mais de um sentimento.

Os algoritmos utilizados como linha de base foram: (1) *Support Vector Machine* (SVM) modelado para realizar classificação [Platt, 1999] e regressão [Chang & Lin, 2011], neste último caso ele é chamado de *Support Vector Regression* (SVR); (2) *k Nearest Neighbor* (kNN) [Aha et al., 1991]; (3) *Naive Bayes* (NB) [John & Langley,

1995]; (4) Árvore de Decisão [Quinlan, 1993]; e (5) o *Lazy Associative Classifier* (LAC) [Veloso et al., 2006], este último consiste basicamente da versão estática da solução proposta. Para realizarmos os experimentos, otimizamos os parâmetros de cada um destes algoritmos. Para o SVM avaliamos o *kernel*, o custo C, gamma, coeficiente (coef0) e grau, além desses parâmetros para o SRV avaliamos suas formas ϵ-SVR e ν-SVR. Para o kNN o números de vizinhos (k), na Tabela 3.2 são apresentados os melhores parâmetros para cada algoritmo nas coleção analisadas.

Tabela 3.2. Melhores parâmetros para cada algoritmo avaliado

Coleção de Dados	kNN	SVM	SVR
Copa do Mundo de Futebol (Português)	k=6	kernel=polinomial, C=11, expoênte=6	tipo=ν-SVR, C=11, kernel=polinomial, coef0=9, expoênte=14
Copa do Mundo de Futebol (Inglês)	k=16	kernel=polinomial, C=11, expoênte=1	tipo=ϵ-SVR, C=16, kernel=radial, gamma=1/ *# termos*
Eleições Presidenciais no Brasil	k=26	kernel=RBF, C=11, gamma=0.01	tipo=ν-SVR, C=21, kernel=polinomial, coef0=0, expoênte=3
Personalidade do Ano	k=1	kernel=RBF, C=10, gamma=0.11	tipo=ν-SVR, C=15, kernel=radial, gamma=0.1

Neste experimento os algoritmos foram treinados com apenas uma pequena semente das primeiras mensagens do evento. Para o coleção Copa do Mundo de Futebol (pt) foram utilizada 36 mensagens, e para a versão em inglês (en) foram utilizadas 35 mensagens. Para as coleções Campanha Eleitoral da Dilma Rousseff e Personalidade do Ano foram utilizadas as 100 primeiras mensagens.

Experimentalmente foram configurados o tamanho da fila do *sub judice* ($|\mathcal{S}|$) e o fator δ_{min} do limiar. Para as coleções sobre a copa do mundo em Português os melhores parâmetros foram $|\mathcal{S}|$ 10 e δ_{min} =1, e em Inglês foram $|\mathcal{S}|$=10, δ_{min} igual 1.8 (*sub judice*) e 1.5 (julgamento imediato). Para a coleção Personalidade do Ano foi $|\mathcal{S}|$=350 e δ_{min} igual 2 (*sub judice*) e 1 (imediato judice), já para a coleção sobre as Eleições Presidenciais no Brasil o $|\mathcal{S}|$ foi 500 e δ_{min} 1,2 e 1,8. A variação em $|\mathcal{S}|$ se deve a velocidade em que mudança ocorrem no fluxo, é esperado que em um jogo de futebol mudanças ocorram mais rapidamente do em uma eleição, uma vez que uma aplicação poderia ser o acompanhamento minuto a minuto de um jogo enquanto que em uma eleição este acompanhamento seria diário, então o parâmetro $|\mathcal{S}|$ pode ser configurado com uma caracterização prévia do evento.

Na Tabela 3.3 é apresentado o MSE alcançado pela versão estática de 5 classificadores. Estes são contrastados com o MSE alcançado pelo o algoritmo de auto treinamento. O algoritmo de auto treinamento alcançou melhoras significativas

em relação aos algoritmos estáticos, estes ganhos variam de 14,84%, na coleção sobre as Eleições Presidenciais no Brasil, a 41,38% na coleção a respeito da Copa do Mundo (pt). Calculamos o ganho como a porcentagem do melhor algoritmo estático em relação ao algoritmo de auto treinamento menos 1.

Tabela 3.3. MSE alcançado pelo algoritmo de auto treinamento e versões estáticas de diferentes algoritmos

Coleção de Dados	Modelos Estáticos (ME)						Auto Treinamento		Ganhos (%)	
	kNN	Árvore	NB	SVM	SVR	LAC	Julgamento Imediato (JI)	*Sub Judice* (SJ)	SJ vs. ME	SJ vs. JI
Copa do Mundo de Futebol (Português)	0.2066	0.373	0.4999	0.1150	0.1732	0.1522	0.0814	0.08134	41.38	0.07
Copa do Mundo de Futebol (Inglês)	0.3572	0.441	0.5	0.4819	0.2957	0.3315	0.2384	0.2291	29.07	4.06
Eleições Presidenciais no Brasil	0.3708	0.437	0.4996	0.4303	0.2500	0.2795	0.2708	0.2177	14.84	24.39
Personalidade do Ano	0.5992	0.572	0.6581	0.5728	1.658	0.5461	0.5419	0.4487	21.71	20.77

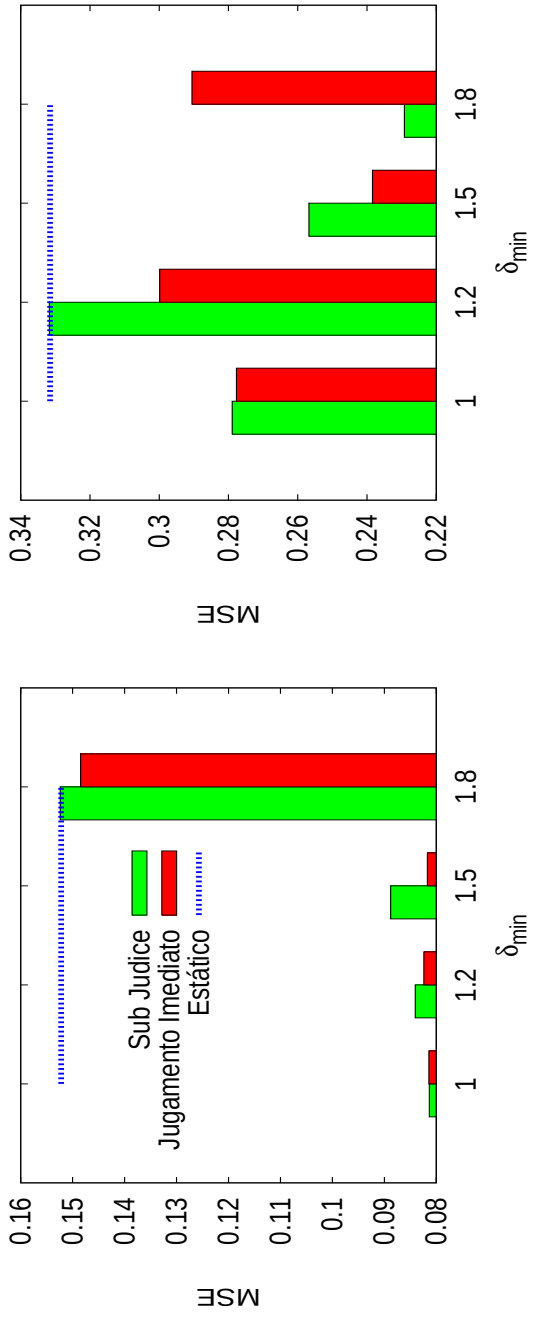
A partir da Figura 3.5 analisamos a sensibilidade do algoritmo em relação ao parâmetro δ_{min}, contrastamos a estratégia de *sub judice* com o julgamento imediato (i.e., quando não é permitido o algoritmo manter mensagens *sub judice*) e mostramos também o MSE do algoritmo estático.

Podemos ver que em todas as coleções o algoritmo de auto treinamento, quando aplicado juntamente com a estratégia de *sub judice*, nunca é significativamente pior do que o algoritmo estático. Além disto, se analisarmos as melhores configurações com o *sub judice* e com o *julgamento imediato* observamos um ganho de até 24%, com a utilização do *sub judice* (Tabela 3.3). Se compararmos o *sub judice* com o algoritmo na sua forma estática apresentamos ganhos de 21 a 87%.

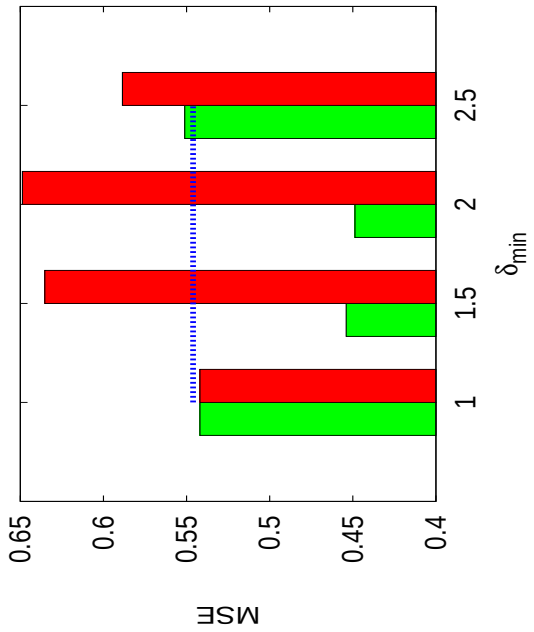
Na Figura 3.6 são exibidos vários resultados que contribuem para um melhor entendimento em relação ao desempenho do algoritmo de auto treinamento na coleção Copa do Mundo de Futebol (Português). A partir das evidências apresentadas argumentamos que o maior responsável pela rápida adaptação à súbita mudança nos dados (i.e., súbito *sentiment drift*) foi a projeção de dados de treinamento sob demanda. Na Figura 3.6(a) regiões claras são regiões onde existem uma concentração maior de mensagens na projeção ($\mathcal{D}_x$) e podemos ver que existe uma localidade temporal nas mensagens que são projetadas para treinamento. Uma mensagem exibe uma localidade temporal se ela somente é acessada em um futuro próximo, isto é, a mensagem $d \in \mathcal{D}$ se torna mais provável de estar em $\mathcal{D}_{t_i}$ e em $\mathcal{D}_{t_j}$ se $t_i in \mathcal{T}$ está próxima no tempo com $t_j \in \mathcal{T}$. Nessa figura mensagens localizadas nas regiões claras são aquelas que aparecem nos dados de treinamento projetados da mensagem correspondente no eixo x por outro

lado regiões escuras são mensagens que não aparecem nos dados projetados para a mensagem no eixo x.

Uma vez que mensagens no eixo x e y estão cronologicamente ordenadas, podemos entender o quão frequente estas mensagens são utilizadas com o passar do tempo. É possível perceber que as mensagens são gradualmente menos e menos utilizadas com o passar do fluxo e mensagens futuras tendem a demandar mensagens que foram inclusas recentemente em $\mathcal{D}$.

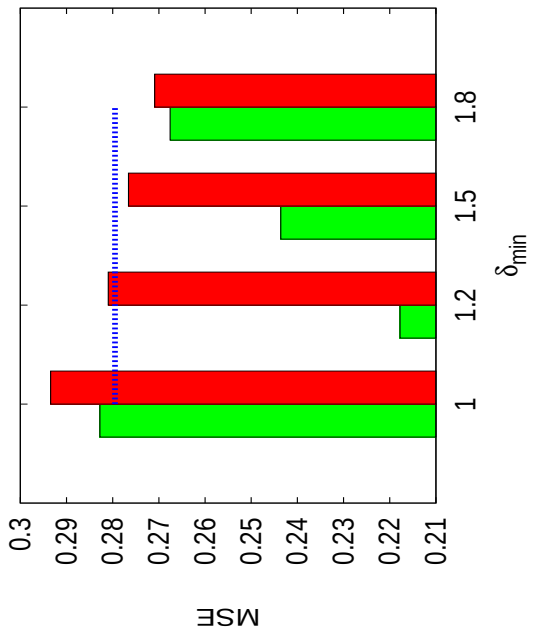


(a) Copa do Mundo (Português)



(b) Copa do Mundo (Inglês)

(c) Eleições Presidenciais no Brasil



(d) Personalidade do Ano

Figura 3.5. Sensibilidade do paramêtro δ_{min}.

As mensagens também possuem um tempo de expiração, após este ela se torna inútil para realizar a predição. Por exemplo, a primeira mensagem a aparecer no fluxo se torna inexpressiva após aproximadamente 2.000 mensagens processadas pelo classificador, outro exemplo é que aquelas mensagens que chegaram antes do *sentiment drift* (i.e., antes da mensagem 1.000), deixam rapidamente de ser projetadas. Com a projeção de dados de treinamento o classificador deixa de utilizar mensagens fora do conceito e distribuição corrente mantendo o classificador concentrado nos últimos acontecimentos do evento.

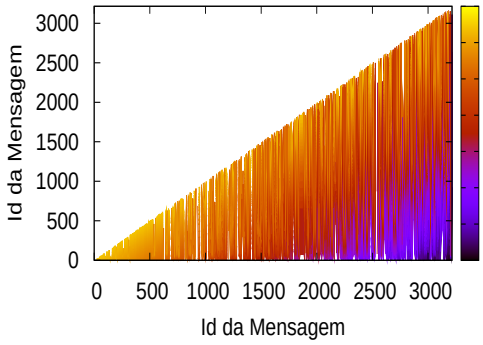
Outra evidência da nossa argumentação é apresentada na Figura 3.6(b), a qual apresenta o MSE ao longo do tempo. Nesta Figura apresentamos duas configurações do SVM, na primeira selecionamos o conjunto de parâmetros (Tabela 3.2) que trouxe o melhor resultado, na segunda apresentamos as configurações do SVR que alcançaram melhor eficácia antes do *sentiment drift*, aproximadamente mensagem 600, (C=30 ϵ-SVR).

É possível observar que no começo do evento o MSE para o SVR, na sua segunda configuração, é melhor do que o da nossa abordagem, contudo após a ocorrência do *sentiment drift* o MSE do SVR enquanto a solução proposta se mantem estável. O SVM utilizando os melhores parâmetros também se mantém estável, porém com um MSE pior do que o do auto treinamento durante todo o período, para os parâmetros δ_{min} 1, 1.2, e 1,5 o parâmetro 1.8 foi bastante próximo ao modelo estático.

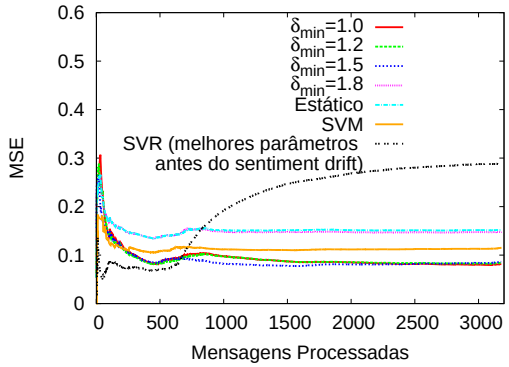
A Figura 3.6(c) permite compreender porque o algoritmo de auto treinamento apresentou ganhos significativos, podemos ver que foi inserida uma proporção grande de mensagens corretamente rotuladas nos dados de treinamento ($\mathcal{D}$). A Figura 3.6(c) explica também porque o $\delta_{min=}1$ teve o melhor desempenho em MSE, uma vez que com esse parâmetro foi inserido 93% de mensagens nos dados de treinamento corretamente, a maior porcentagem entre os δ_{min} apresentados.

Finalmente, na Figura 3.6(d), mostramos uma aplicação para nosso algoritmo e sua qualidade, no sentido de uma boa apresentação. A aplicação é o acompanhamento dos pulsos de sentimentos durante um evento. Na linha de azul verifca-se a porcentagem de mensagens positivas em cada minuto do jogo. Propomos medir os pulsos de sentimentos calculando a média da pontuação ($\hat{p}(s_i|t)$) do sentimento positivo das mensagens que chegam em um mesmo intervalo de tempo ($\mathcal{T}_i$) (i.e, $\frac{\sum_{t \in \mathcal{T}_i} \hat{p}(s=+|t)}{|\mathcal{T}_i|}$), que neste caso $\mathcal{T}_i$ foi definido com um minuto.

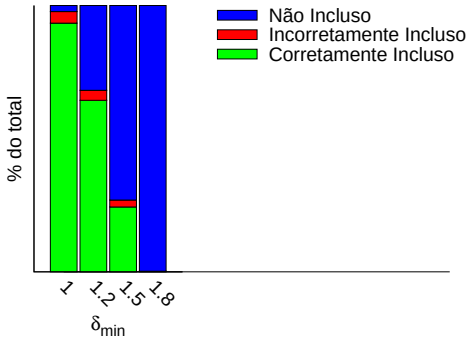
É possível verificar que quando utilizamos o algoritmo na sua versão estática a curva do algoritmo tem a mesma tendência da curva dos dados, porém com uma distância considerável, fazendo com que não seja possível capturar bem os pulsos de sentimentos. Contudo, quando utilizamos o algoritmo com auto treinamento



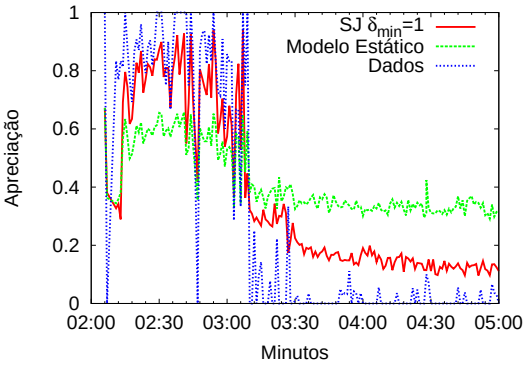
(a) Projeção de Treinamento. Dados de Treinamento Projetados para cada mensagem em $\mathcal{T}$. Dada uma mensagem x no eixo x, o gráfico apresenta o $\mathcal{D}_x$ correspondente no eixo y. Regiões com cores mais claras indicam a presença da mensagem correspondente em $\mathcal{D}_x$.



(b) MSE ao longo do tempo para diferentes valores de δ_{min}, quando o *sub judice* é permitido.



(c) Porcentagem de mensagens em $\mathcal{T}$ que são corretamente, erroneamente e não inclusas em $\mathcal{D}$, para diferentes valores de δ_{min} (permitindo o *sub judice*).



(d) Acompanhamento do sentimento de apreciação ao longo da partida.

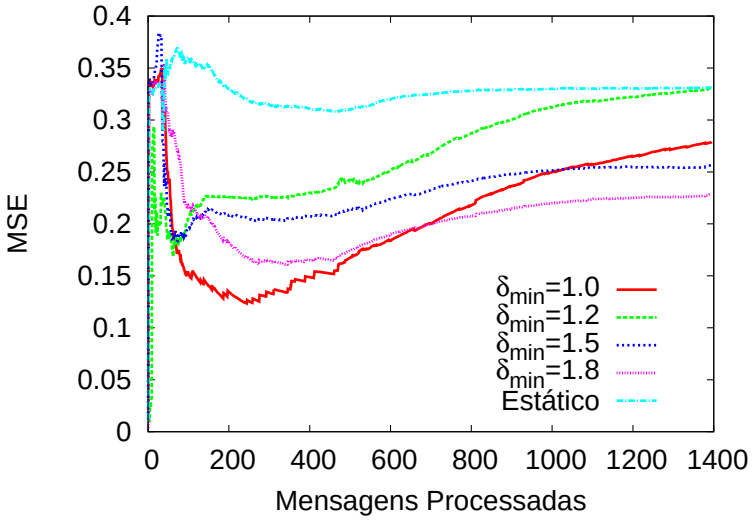
Figura 3.6. Copa do Mundo de Futebol - Derrota do Brasil (Português)

percebemos claramente que os pulsos de sentimentos são capturados de uma forma muito próxima do que aconteceu na realidade.

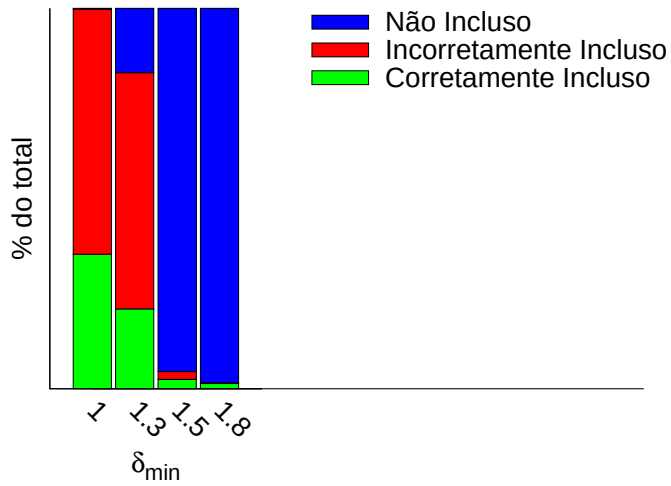
Na Figura 3.7(a) é apresentado o MSE ao longo do tempo na coleção sobre a Copa do Mundo em Inglês. É possível observar que o δ_{min} se mantém mais estável do que os demais parâmetros. Na Figura 3.7(b) vemos a porcentagem de treino inserido corretamente, incorretamente e não inserido em $\mathcal{D}$. É possível verificar que para os δ_{min} 1 e 1.3 foi inserida uma porcentagem considerável de mensagens incorretas em $\mathcal{D}$. Contudo nos parâmetros 1,5 e 1,8 essa porcentagem diminui e aumenta a porcentagem de mensagens inseridas corretamente, para δ_{min} igual a 1,8 foi inserido 1,38% de mensagens corretamente contra 0,18% de mensagens incorretas e para δ_{min} 1,5 foi inserido 2,46% de mensagens corretas contra 2,05% de mensagens incorretas. A porcentagem de mensagens inseridas em $\mathcal{D}$ também diminuiu porque as mensagens que são inseridas com um limiar muito baixo não deveriam ser inseridas e isso é corrigido com o aumento do valor de δ_{min}.

Analisamos os resultados da coleção sobre as Eleições Presidenciais no Brasil a partir da Figura 3.8. Na Figura 3.8(a) podemos ver a localidade temporal das mensagens desta coleção. Foi possível observar que a partir da mensagem trinta mil, as primeiras mensagens que chegaram no fluxo são praticamente inúteis para classificação daquelas que estão chegando. Logo, com o passar do fluxo menos mensagens do inicio do evento são projetadas, em $\mathcal{D}_x$, para treinamento. Na Figura 3.8(b) verifica-se que o δ_{min} 1.2 teve melhor desempenho, em MSE, porque com este foi inserida uma quantidade considerável de mensagens com o rótulo correto e uma porção pequena de mensagens com o rótulo incorreto.

A Figura 3.8(c) apresenta o MSE da abordagem de auto treinamento para diferentes valores de δ_{min}, podemos perceber que o MSE se mantém estável durante todo o evento, isto é uma evidência de que o modelo continua confiável mesmo após um longo período. Na Figura 3.8(d) é possível ver que mesmo aumentando a semente de treino inicial, até 1.000 mensagens do início do evento, nossa abordagem continua tendo um desempenho em MSE superior aos algoritmo estáticos para pelo menos um parâmetro.



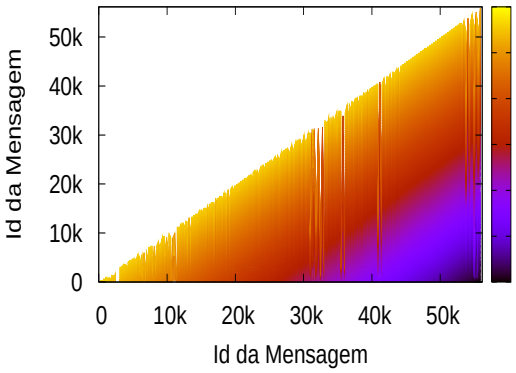
(a) MSE ao longo do tempo para diferentes valores de δ_{min}



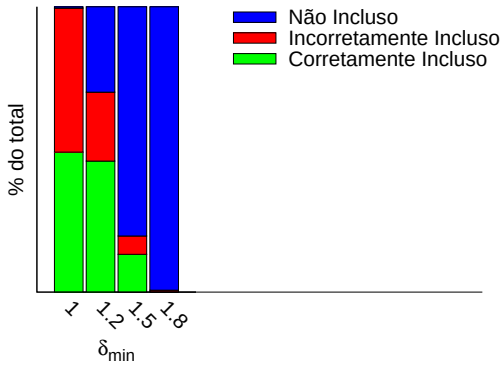
(b) Porcentagem de mensagens em $\mathcal{T}$ que são corretamente, erroneamente e não inclusas em $\mathcal{D}$, para diferentes valores de δ_{min} (permitindo o *sub judice*).

Figura 3.7. Copa do Mundo de Futebol - Derrota do Brasil (Inglês)

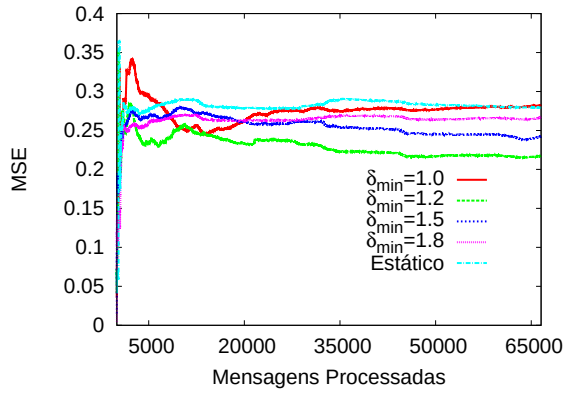
Em relação ao desempenho do algoritmo para a coleção Personalidade do Ano, conforme demonstrado na Figura 3.9(a), é possível verificar como *sub judice* deslocou a mensagem do momento que ela deveria ser classificada. Houve uma maior concentração de mensagens em *sub judice* no intervalo que existe uma maior variação de sentimentos (como visto na Seção 3.2.2). Após este período as mensagens são classificadas próximo ao momento que chegam. Na Figura 3.9(b) que a efetividade do parâmetro δ_{min} igual a 1.5 e 2 é devida ao menor número de mensagens inserida em $\mathcal{D}$ incorretamente,



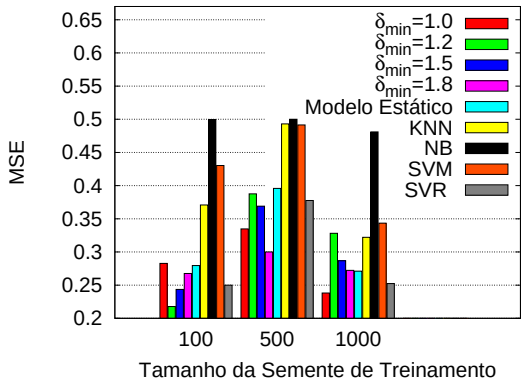
(a) Projeção de Treinamento. Dados de Treinamento Projetados para cada mensagem em $\mathcal{T}$. Dada uma mensagem x no eixo x, o gráfico apresenta o $\mathcal{D}_x$ correspondente no eixo y. Regiões com cores mais claras indicam a presença da mensagem correspondente em $\mathcal{D}_x$.



(b) Porcentagem de mensagens em $\mathcal{T}$ que são corretamente, erroneamente e não inclusas em $\mathcal{D}$, para diferentes valores de δ_{min} (permitindo o *sub judice*).



(c) MSE ao longo do tempo.

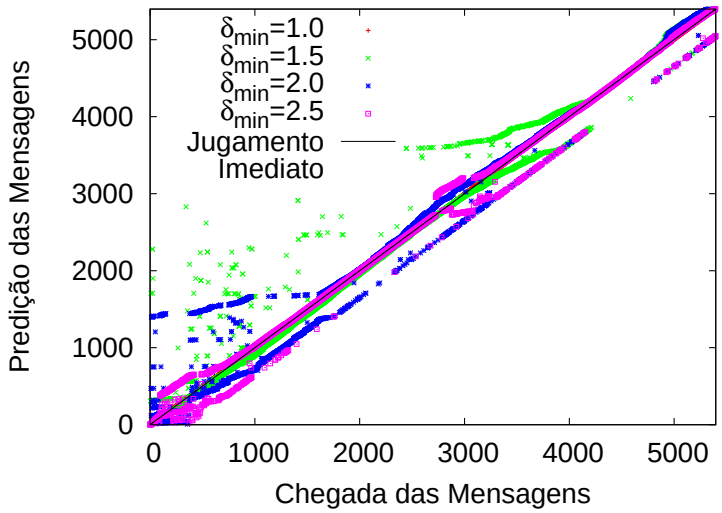


(d) MSE para diferentes valores de δ_{min} (*sub judice*).

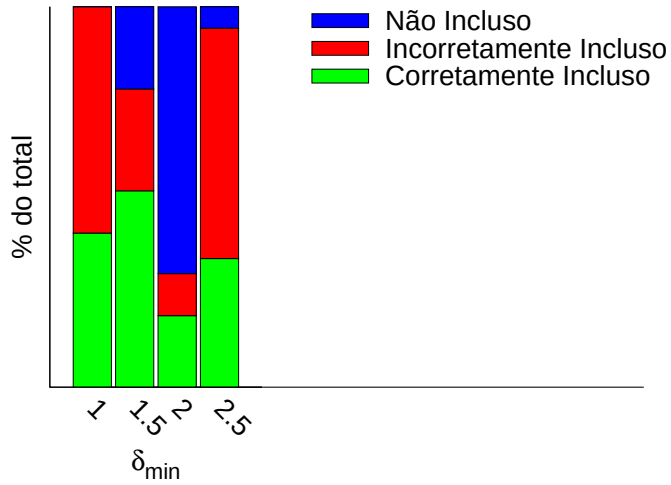
Figura 3.8. Eleições Presidenciais no Brasil

acompanhado de uma quantidade considerável de mensagens inclusas de maneira correta.

A partir da Figura 3.10 verificamos o comportamento do algoritmo ao aumentar a quantidade de mensagens na semente de treinamento, neste experimento treinamentos o algoritmo com o número de mensagens exibido no eixo x e o testamos com o restante das mensagens das coleções, utilizamos o δ_{min} igual a 1,5, pelo fato de termos verificado que este é o parâmetro mais estável. Comparamos com o algoritmo de auto treinamento com o modelo do LAC estático.



(a) Eixo X corresponde a ordem de chegadas das mensagens e eixo Y a ordem de predição, para diferentes valores de δ_{min}.



(b) Porcentagem de mensagens em $\mathcal{T}$ que são corretamente, erroneamente e não inclusas em $\mathcal{D}$, para diferentes valores de δ_{min} (permitindo o *sub judice*)

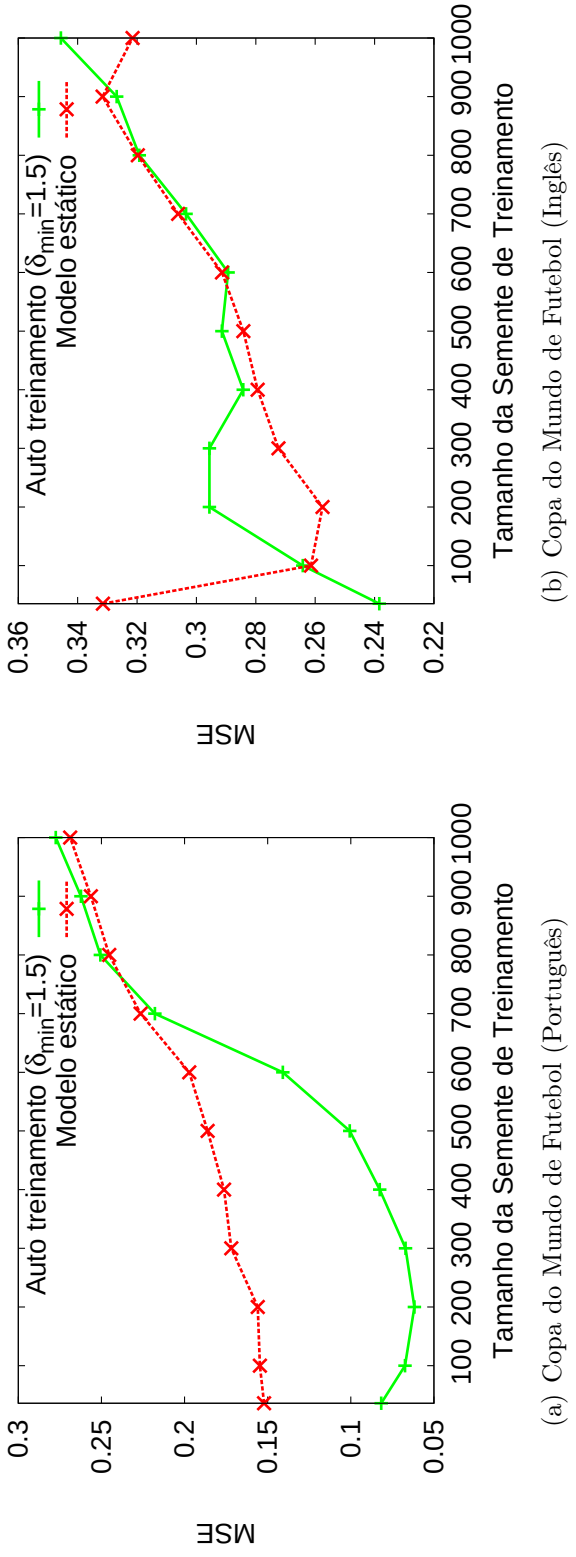
Figura 3.9. Personalidade do Ano (Inglês).

Na coleção sobre a Copado do Mundo de Futebol (Figuras 3.10(a) e 3.10(b)) , de forma contrária do que era esperado o MSE aumenta com o aumento do tamanho da semente de treinamento ao invés de diminuir, isto porque as mensagens de treinamento atribuem um viés para o sentimento positivo, uma vez que estas são de um período que a maioria das mensagens eram positivas. Contudo, quando ocorre o *sentiment drift*, após a mensagem 1.000, a grande maioria as mensagens que chegam é negativa.

Na coleção sobre a Copa do Mundo em Português (Figura 3.10(a)) o algoritmo

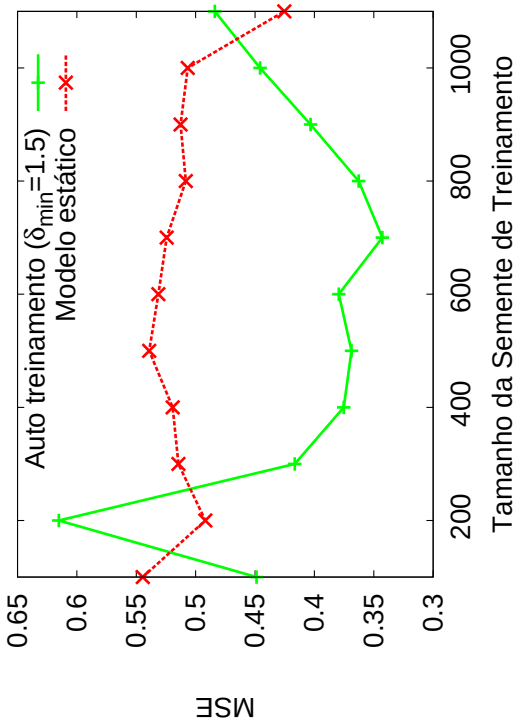
de auto treinamento possui um MSE abaixo do modelo estático até uma semente de tamanho 700, a partir deste ponto o MSE se iguala pelo motivo do algoritmo já possuir dados de treinamento suficiente. Na coleção sobre a Copa do Mundo em Inglês (Figuras 3.10(b)), o MSE do algoritmo de auto treinamento se equipara ao MSE do modelo estático com uma semente de 200 exemplos, com uma semente entre 200 e 500 mensagens o algoritmo estático é melhor que o algoritmo de auto treinamento, isto porque o treino fornecido ao algoritmo foi de um período onde os dados continua uma alta entropia (Figura 3.4(b)), com um semente maior que 500 o algoritmo de auto treinamento possuí um MSE abaixo do modelo estático, mostrando uma convergência nos resultados pelo fato de mais de 34% dos dados terem sido fornecidos como treinamento.

Na Figura 3.10(c), vemos que o MSE do algoritmo de auto treinamento se mantem abaixo do algoritmo estático mesmo com o aumento dos exemplos na semente de treinamento até 1.000 mensagens, ao fornecermos 1.100 mensagens com semente inicial o algoritmo estático apresenta um MSE menor que o algoritmo de auto treinamento, porém neste momento já foi fornecido 19,6% das mensagens da coleção para treinamento. O único ponto que o MSE do algorimto de auto treinamento é acima do algoritmmo estático é com uma semente de 200 mensagens, isto porque neste conjunto de treinamento existe um viés alto para os sentimentos de revolta de reprovação (38,5% e 31%) porém no decorrer das mensagens a distribuição muda (Figura 3.2(d)) e 73% das mensagens que chegam contém o sentimento de sarcasmo.



(a) Copa do Mundo de Futebol (Português)

(b) Copa do Mundo de Futebol (Inglês)



(c) Personalidade do Ano

Figura 3.10. Aumento do Tamanho da Semente de Treinamento

Tabela 3.4. Tempo de Execução

Coleção	Tempo de Execução (segundos)	Mensagem / segundo	
		Taxa de Processamento	Taxa de Chegada
Copa do Mundo (Português)	12	267.83	1.12
Copa do Mundo (Inglês)	5	286.4	1.12
Eleições Presidenciais no Brasil	289752	0.23	0.02
Personalidade do Ano	636	8.83	0.2

Na Tabela 3.4 é apresentada uma comparação do tempo de execução da solução proposta, utilizando os melhores parâmetros, com a velocidade do fluxo (i.e., taxa de chegada de mensagens por segundo). É possível verificar que a velocidade da solução proposta é maior que a taxa de chegada de mensagens, isto faz com que nossa abordagem seja aplicável em cenários reais. Para as coleções sobre as Eleições Presidenciais no Brasil e a *Personalidade do Ano*, o tempo de execução foi alto devido a fila de *sub judice* de tamanho 500 e 350 respectivamente, porém ainda assim a taxa de mensagens processadas por segundo é maior que a taxa de chegada de mensagens.

Capítulo 4

Aproximação Experimental do Limite Superior para Análise de Fluxos de Sentimento

O melhor caso para a abordagem de auto treinamento pode ser alcançado se relaxarmos a restrição de recursos limitados para treinamento do algoritmo e considerarmos que após o seu processamento todas as mensagens estão disponíveis, com o seu rótulo correto, para a atualização do modelo. Esse processo de relaxamento das restrições considerando todos os dados rotulados é equivalente a metodologia de avaliação de algoritmos em cenários de fluxos dados, conhecida como *Interleaved Test-Then-Train* (ou *Prequential*) [Dawid, 1984], utilizada em muitos trabalhos como [Bifet et al., 2009; Gama et al., 2009].

Nesta metodologia o exemplo é utilizado para testar o classificador e logo em seguida para treina-lo. Desta forma é possível verificar o comportamento do modelo que está sendo avaliado em exemplos que ele ainda desconhece. A vantagem em utilizar o *Interleaved Test-Then-Train*, na avaliação de algoritmos em fluxos de dados, consiste no fato dessa abordagem não exigir um conjunto de reservado para o teste, em outras palavras, ele utiliza ao máximo os dados disponíveis [Bifet et al., 2009]. Essa metodologia é exemplificada no Algoritmo 4.

Como o classificador conhece o rótulo de todos os dados anteriores à mensagem que está sendo analisada, consideramos esta configuração como um limite superior para nossa abordagem de auto treinamento. No entanto, mesmo o classificador contendo essa informação sobre os rótulos dos dados, o *sentiment drift* impõem um prazo de validade para as mensagens de treinamento, como visto na Seção 3.2.1. Com o intuito de contornar os efeitos causados na eficácia do classificador pelo *sentiment drift*, diferentes

Algoritmo 4 Interleaved Test-Then-Train

1: **for all** $t \in \mathcal{T}$ **do**
2: predição(t) = mensurar o sentimento de t utilizando $\mathcal{R}$
3: avaliar predição(t)
4: incrementar $\mathcal{R}$ com $< t, s_c >$, onde s_c é sempre o sentimento correto
5: **end for**

estratégias, denominadas "estratégias de esquecimento", foram avaliadas para que fosse possível tratar o *sentiment drift* presente nos dados, e posteriormente, uma abordagem para esquecimento em fluxo de dados foi proposta para que fosse possível alcançar a melhor solução possível com o algoritmo de auto treinamento apresentado nesse trabalho.

4.1 Esquecimento em Fluxos de Dados

No contexto de análise de fluxo de sentimento, o *sentiment drift* consiste em um dos maiores desafios a serem tratados pelos algoritmos de aprendizado de máquina [Hulten et al., 2001; Zhu et al., 2010; Bifet, 2010]. Isso porque, uma vez que o fluxo representa a opinião das pessoas *online*, sobre um determinado assunto, essa opinião pode mudar de maneira inesperada em função de acontecimentos imprevisíveis.

Para que o conjunto de treino possa representar o fluxo de sentimento atual, de forma significativa, é necessário considerar os efeitos do *sentiment drift*, não apenas assimilando novos exemplos, mas também identificando e efetivamente removendo mensagens de treinamento que já não descrevem o fluxo corrente. Em outras palavras, em análise de fluxo de sentimento, aprender sobre o fluxo é tão relevante quanto esquecer o que já não o descreve. Sabendo dos efeitos causados pelo *sentiment drift*, analisamos como o esquecimento afeta a efetividade do classificador proposto utilizando uma janela deslizante de treinamento (com o tamanho w pré-definido) e propomos uma nova abordagem de esquecimento chamada de Janela Deslizante Ativa.

4.1.1 Janela Deslizante de Tamanho Fixo

A utilização de uma janela deslizante é uma abordagem muito utilizada na descoberta de conhecimento em fluxos de dados. Está é uma simples maneira de lidar com o *concept drift*. A ideia é utilizar as w últimas mensagens na análise ao invés de todas as mensagens vistas até o momento atual. É possível dizer que a janela é de tamanho w uma vez que as mensagens que chegaram antes do *momento atual* - w não serão

consideradas na análise [Bifet, 2010].

O tamanho da janela w é um importante parâmetro a ser apreciado, o qual, a princípio, deve ser configurado previamente por um usuário. Esta abordagem pode funcionar quando a taxa de mudança dos dados é conhecida, porém em um ambiente de fluxos de dados, a taxa de mudanças é raramente conhecida. Dessa forma um w pequeno faz com que a janela reflita bem a distribuição corrente, porém os dados podem não ser suficientes para que o classificador alcance uma eficácia esperada. Por outro lado um w grande pode fazer com que o classificador atinja bom desempenho nos períodos de estabilidade do fluxo, mas na ocorrência de um *sentiment drift* a recuperação pode ser lenta [Bifet, 2010]. Além disso, consecutivos *sentiment drift* podem fazer com que o modelo esteja deteriorado por um longo período de tempo.

Visando endereçar as deficiências apontadas pelo uso de uma janela deslizante de tamanho fixo, propomos a Janela Deslizante Ativa que é apresentada na próxima seção.

4.1.2 Janela treino Deslizante Ativa

A Janela de treino Deslizante Ativa (JDA) consiste em uma solução, fundamentada na teoria do aprendizado ativo, porém ao invés de escolher quais exemplos serão selecionados para entrar no conjunto de treinamento, $\mathcal{D}$, o objetivo é permitir que o classificador escolha quais exemplos esquecer (i.e., remover de $\mathcal{D}$), uma vez que já não representam o fluxo atual. Dessa forma, com essa abordagem, espera-se tratar os efeitos do *sentiment drift* no fluxo de sentimento, não apenas com o aumento do treino, mas também com a remoção de exemplos afetados pelo *sentiment drift* e a partir deste processo aumentar a eficácia do classificador.

Aprendizado ativo consiste em uma técnica de amostragem de dados onde, ao invés do conjunto de treinamento ser composto de exemplos aleatórios, são selecionados exemplos que provem maior ganho de informação. Enquanto um aprendiz passivo obtém todos os dados rotulados de uma única vez, um aprendiz ativo seleciona quais exemplos ele gostaria de saber o rótulo [Zhu et al., 2007]. Esta abordagem, quando executada adequadamente, pode reduzir exponencialmente a quantidade de exemplos de treinamento necessária para o aprendizado [Kivinen & Mannila, 1994].

Como estratégia de esquecimento e com o objetivo de descrever melhor o fluxo de sentimento atual, a JDA mantém um conjunto de treino que provê ao classificador um maior ganho de informação com um viés temporal. Através dessa estratégia, espera-se alcançar uma janela de tamanho ótimo (i.e., uma janela cujo tamanho seja pequeno suficiente para não sofrer com os efeitos do *sentiment drift* e grande suficiente para que o

algoritmo possa aprender com eficácia). Tal resultado pode ser alcançado, uma vez que a teoria do aprendizado ativo sustenta a possibilidade de diminuir exponencialmente o conjunto de treino necessário para que o classificador aprenda [Kivinen & Mannila, 1994].

Com o intuito de alcançarmos o objetivo anteriormente descrito utilizamos uma função de *rank* que elege potenciais candidatos a serem removidos da janela atual, a cada nova mensagem rotulada (t) inserida em $\mathcal{D}$, esta função é descrita na Equação 4.1. Para que a operação seja realizada com eficiência computacional o treino é projetado, de acordo com os termos em t [Veloso & Meira Jr., 2011].

$$r(t_j) = \frac{s(t_i, t_j) + (1 - \frac{m(t_j)}{m(t_i)})}{2} \tag{4.1}$$

Na função de *rank* (Equação 4.1) $t_i \in \mathcal{T}$ é uma mensagem a ser inserida em $\mathcal{D}$, t_j é uma mensagem $\in \mathcal{D}$ e $t_j \neq t_i$, $m(t)$ denota o momento que o exemplo foi inserido no treino (uma vez que os exemplos são inseridos sequencialmente), e finalmente, $s(t_i, t_j)$ indica a similaridade entre dois exemplos. Nesse caso, quanto maior a similaridade entre os exemplos e a idade do exemplo do treino, maior é o *rank*. A função de similaridade utilizada no trabalho consiste no coeficiente de *Jaccard*, esta que é apresentada na Equação 4.2.

$$s(t_i, t_j) = \frac{|t_i \cap t_j|}{|t_i \cup t_j|} \tag{4.2}$$

Calculado o *rank* dos exemplos em $\mathcal{D}$, aquele com maior pontuação é removido da janela de treino. Para definir se a pontuação do exemplo selecionado é suficiente, utilizamos um limiar que deve ser arbitrariamente pequeno. Se a pontuação for menor ou igual ao limiar o exemplo será mantido em $\mathcal{D}$. O limiar foi definido experimentalmente como 0.1. Dessa forma o limiar garante que exemplos significativos sejam mantidos no conjunto de treinamento na inserção de exemplos com informações novas.

Através dessa solução pretende-se manter na janela os exemplos que fornecem maior ganho de informação, uma vez que são removidas as mensagens com maior similaridade ao exemplo que está sendo inserido no treino. Além disto, espera-se manter uma amostra de treino com um viés temporal, uma vez que as mensagens mais recentes terão menor probabilidade de serem removidas da janela de treino. Tal estratégia pode impactar significativamente na redução do tamanho da janela e minimizar a chance da mesma contemplar exemplos afetados pelo *sentiment drift*.

4.2 Avaliação Experimental

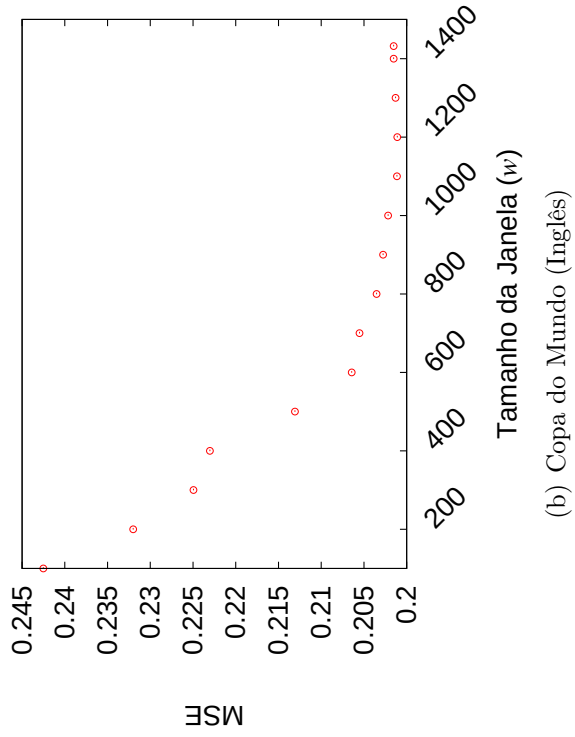
Nesta seção apresentamos e discutimos os resultados obtidos na definição do limite superior, com a utilização da JDA e uma janela de tamanho fixo (w), e contrastamos estes resultados com o desempenho da solução de auto treinamento. Em nossa avaliação utilizamos como métrica o (MSE), onde, quanto menor o valor alcançado, melhor o resultado da abordagem.

Na Figura 4.1 é possível verificar, para cada coleção analisada, o MSE alcançado pelo algoritmo com a variação do tamanho da janela deslizante de treinamento. Podemos observar que em cada coleção o algoritmo apresentou um comportamento diferente com o aumento do tamanho da janela. Por exemplo, nas coleções a respeito da copa do mundo (Figuras 4.1(a) e 4.1(b)), enquanto o MSE aumenta com o aumento do tamanho da janela na coleção em Português (Figura 4.1(a)), na coleção em Inglês (Figura 4.1(b)) o MSE diminui.

Na coleção sobre as Eleições no Brasil (Figura 4.1(c)) o MSE apresenta um leve declínio até o tamanho da janela 1000, mas após este tamanho, o MSE apresenta um rápido e contínuo aumento. Já na coleção Personalidade do Ano (Figura 4.1(d)) o MSE apresenta um crescimento com o aumento do tamanho da janela porém, próximo ao tamanho 3000 ocorre um leve declínio no MSE e posteriormente ele se estabiliza.

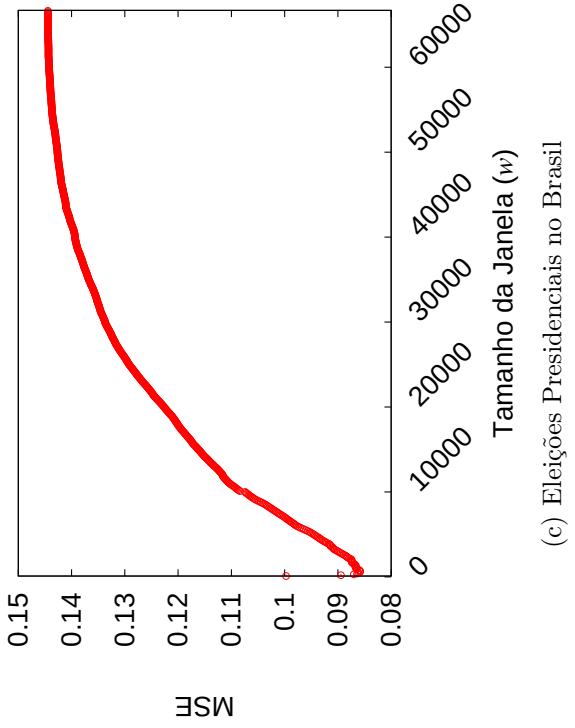
Estas diferenças ocorrem principalmente pelo tipo de *sentiment drift* contido nas coleções. A coleção sobre a Copa do Mundo em Português contém um súbito *sentiment drift*, então um w pequeno é melhor para que a recuperação seja rápida após a ocorrência do *sentiment drift*, neste caso o melhor w foi 200. Já na coleção sobre a Copa do Mundo em Inglês a distribuição é mais ruidosa, uma vez que os sentimentos não apresentam um padrão de acordo com os acontecimentos. Sendo assim, um volume maior de dados provê mais robustez ao ruído, logo, nessa coleção o melhor resultado foi alcançado utilizando praticamente todos os dados vistos (w=1200) para treinamento.

Na coleção sobre as Eleições Presidenciais no Brasil a distribuição muda de uma forma incremental e lenta, então o w muito grande ou muito pequeno não é a configuração mais adequada, neste caso a melhor janela foi de tamanho 600, o que pode ser considerado um tamanho médio. Na coleção Personalidade do Ano o melhor w foi 100.

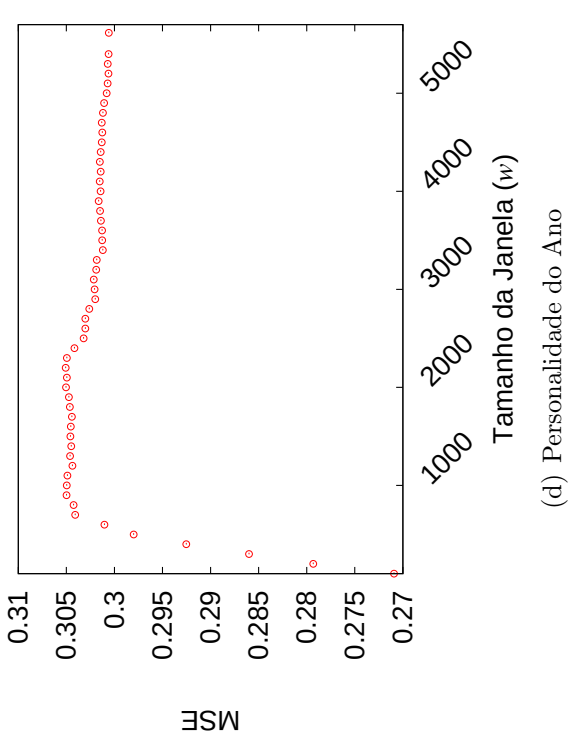


(a) Copa do Mundo (Português)

(b) Copa do Mundo (Inglês)



(c) Eleições Presidenciais no Brasil



(d) Personalidade do Ano

Figura 4.1. MSE com a variação do tamanho da janela deslizante de treinamento aumentando-a de 100 em 100

Através dos resultados descritos na Tabela 4.1 é possível realizar uma comparação entre a JDA e o melhor tamanho de janela encontrada nos experimento. A JDA alcançou resultados equivalentes ou melhores que a utilização de uma janela de tamanho fixo. Contudo, enquanto a JDA não exigiu a configuração prévia do parâmetro de tamanho de janela (w), os resultados com a janela de tamanho fixo foram alcançados a partir de uma busca exaustiva pelo melhor tamanho (ver Figura 4.1). A dificuldade de configurar o tamanho da janela é evidenciada ao verificar que o valor que alcançou melhor eficácia é diferente para cada coleção de dados, além disso, o comportamento do algoritmo com a variação do tamanho da janela de treinamento é diferente em cada coleção.

Tabela 4.1. MSE alcançado com o limite superior e utilizando o algoritmo de auto treinamento.

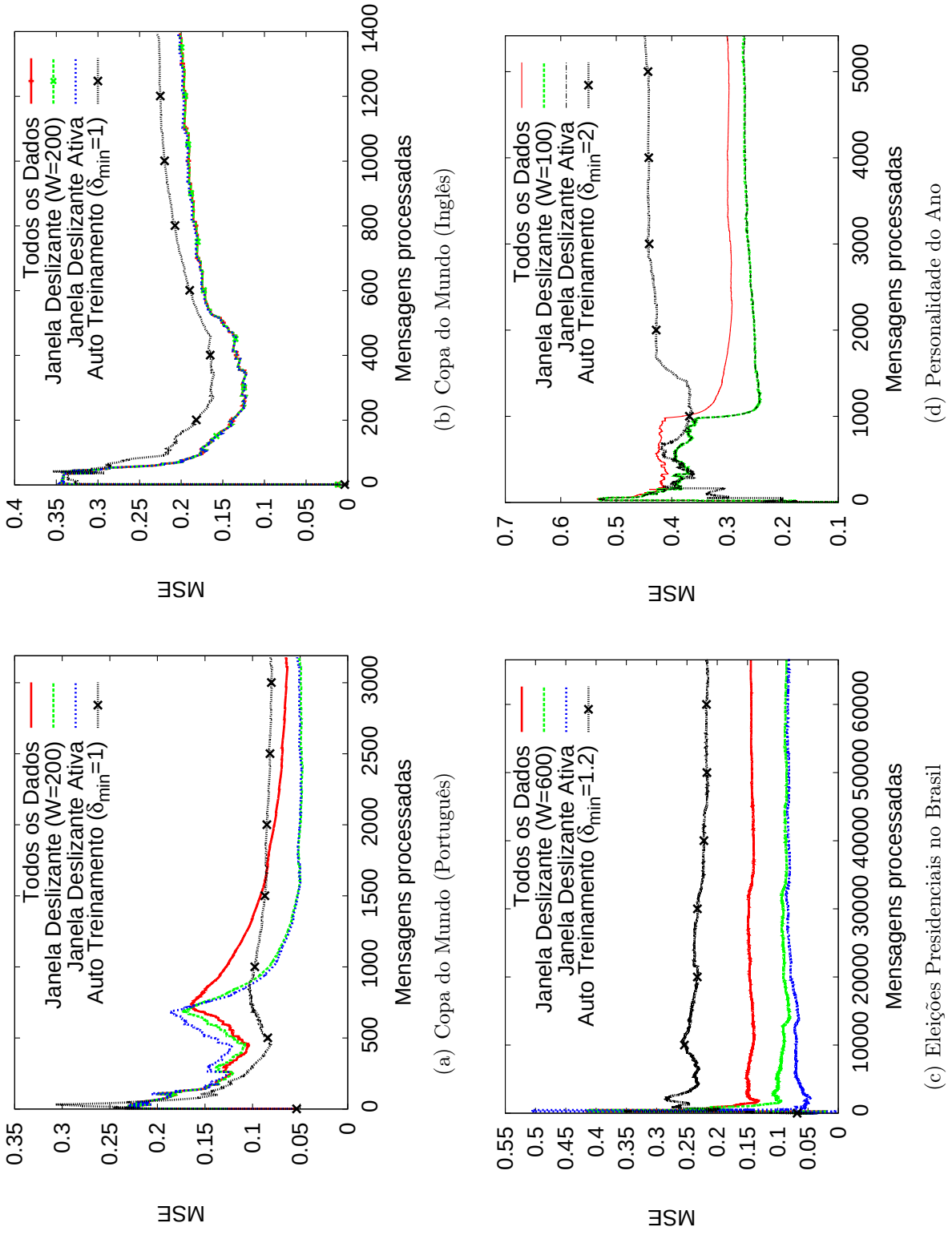
Coleção de Dados	Limite Superior		Auto Treinamento
	Janela Deslizante de Tamanho Fixo (melhor w)	Janela Deslizante Ativa (JDA)	
Copa do Mundo de Futebol (Português)	0.05133 (w=200)	0.0531 (w *médio*=73)	0.08134
Copa do Mundo de Futebol (Inglês)	0.2015 (w=1200)	0.2030 (w *médio*=69)	0.2291
Eleições Presidenciais no Brasil	0.0858 (w=600)	0.0820 (w *médio*=526)	0.2177
Personalidade do Ano	0.271 (w=100)	0.2690 (w *médio*=104)	0.4487

A Tabela 4.1 também descreve os resultados obtidos a partir da abordagem de auto treinamento. Esses resultados foram comparados com o limite superior para o algoritmo proposto. No contexto desse trabalho, o limite superior consiste nos resultados alcançados pelas abordagens que consideram todas as mensagens rotuladas. É possível observar que a abordagem de auto treinamento se aproxima do limite superior nas coleções sobre a Copa do Mundo, com uma distância de 0.02 em MSE. Isto mostra a capacidade do algoritmo proposto de se adaptar a mudanças repentinas e profundas.

Nas coleções sobre as Eleições Presidenciais no Brasil e Personalidade do Ano existe uma diferença de 0.1319 e 0.1797 em MSE, respectivamente, este resultados mostram que nossa abordagem é mais sensível em cenários que o *sentimento* muda lentamente e de forma contínua. Contudo devemos destacar que foi fornecida uma semente de treinamento extremamente pequena para o algoritmo de auto treinamento, o que viabiliza a realização da análise de fluxos de sentimentos no mundo real, enquanto que o nosso limite superior é uma abordagem teórica que não é viável na prática.

Com o intuito de verificar o comportamento do algoritmo no momento que o *sentiment drift* ocorre, na Figura 4.2 são apresentados os resultados ao longo do período analisado. Nesta figura apresentamos o MSE obtido por cada abordagem a medida

que as mensagens são processadas. Em relação a JDA é possível observar que o MSE nunca é significativamente pior do que o melhor tamanho (fixo) da janela deslizante. Na coleção das Eleições Presidenciais a JDA apresenta MSE melhor que as demais abordagens até próximo à mensagem 15.000.



(a) Copa do Mundo (Português)

(b) Copa do Mundo (Inglês)

(c) Eleições Presidenciais no Brasil

(d) Personalidade do Ano

Figura 4.2. MSE ao longo do tempo. As abordagens *Todo o Treinamento*, *Janela Deslizante* e *Janela Delizante Ativa* consideram todas as mensagens para treinamento com o rótulo correto logo após seu processamento.

Na Figura 4.2(a) vemos que a abordagem de auto treinamento tem um MSE menor que as demais abordagens até próximo a mensagem 1000, isto ocorre devido a inclusão seletiva de mensagens em $\mathcal{D}$ evitando um alto viés para o sentimento positivo. Contudo após o *sentiment drift*, com o processamento de novas mensagens, o limite superior alcança MSE melhor do que o auto treinamento, devido principalmente a queda da entropia nos dados (Figura 3.4). Além disso, uma vez que o modelo é sempre atualizado com mensagens corretamente rotuladas e os dados afetados pelo o *sentiment drift* são removidos, pelo uso de uma janela deslizante, a predição dos sentimentos neste período se torna trivial.

Nas demais coleções o algoritmo de auto treinamento mantem um distância constante do limite superior. É possível observar que, se comparada ao limite superior, a eficácia do algoritmo de auto treinamento não cai de maneira inesperada durante o processamento dos fluxos de sentimento, i.e., os algoritmos apresentam o mesmo comportamento. Novamente está é uma evidência da capacidade da solução proposta para tratar o *sentiment drift*.

Capítulo 5

Conclusão e Trabalhos Futuros

Este trabalho concentrou-se no importante problema de análise de sentimento em ambientes de fluxo de dados. Motivados pelos desafios impostos pela análise de sentimento em fluxo de dados, bem como pela carência de técnicas para resolver os problemas existentes nesse cenário, o objetivo deste trabalho consistiu em analisar e propor soluções tanto para o problema relacionado a falta de dados de treinamento para o classificador, quanto para minimizar os desafios em relação a identificação, caracterização e tratamento do *sentiment drift*.

A partir de uma abordagem que faz uso de aprendizado semi-supervisionado, foi proposto um algoritmo de auto treinamento, que mantém o classificador atualizado automaticamente. O algoritmo proposto auto expande o conjunto de treino, a partir de novas mensagens do fluxo e uma pequena semente de treinamento inicial. Além disso, as regras de sentimento são extraídas a partir dos dados de treinamento sob demanda, projetando o espaço de busca para extração das regras sentimento de acordo com as informações contidas na mensagem analisada, permitindo uma extração eficiente.

Através da projeção dos dados de treinamento, o classificador elimina informações irrelevantes e desatualizadas. Isto acontece porque as mensagens que vêm no fluxo apresentam uma localidade temporal, de forma que mensagens antigas são pouco susceptíveis a serem demandadas para as mensagens mais recente que passam através do fluxo. Para melhorar o desempenho de predição do classificador, foi introduzida uma estratégia inovadora denominada *sub judice*. Essa estratégia faz com que os classificadores sejam capazes de abster-se de predições duvidosas e bloquear temporariamente estas previsões até que mais evidências sejam recolhidas com a inclusão de novas mensagens confiáveis.

O algoritmo proposto foi avaliado a partir da análise do fluxo de sentimento expresso sobre importantes eventos no ano de 2010: (1) Copa do Mundo de Futebol;

(2) Eleições Presidenciais do Brasil; e (3) Escolha da personalidade do ano pela revista TIME. Nessa avaliação a eficácia da solução proposta foi comparada com algoritmos estado da arte e os resultados monstraram que as estratégias fornecem ganhos de até 41%.

Posteriormente, nossa abordagem foi contrastada com um limite superior definido experimentalmente. O limite superior foi definido a partir do relaxamento de uma restrição do problema em análise. Nesse caso, o algoritmo de aprendizado considera que para cada mensagem processada, o rótulo de todas as mensagens anteriormente recebidas é conhecido. Além disso, esse classificador faz uso de técnicas para o esquecimento de mensagens desatualizadas presentes no conjunto de treino. Nesses cenários propomos uma técnica de esquecimento, denominada Janela Deslizante Ativa, a qual é capaz de atingir o mesmo (ou superior) desempenho do melhor tamanho de *janela deslizante de tamanho fixo* sem que seja necessário a configuração prévia deste parâmetro. Dessa forma, ao contrastarmos nossa solução proposta com o limite superior, foi possível verificar que o algoritmo de auto treinamento alcança até 87% desse limite superior.

Apesar dos resultados favoráveis a solução proposta apresentados devemos citar que a esta possui limitações, como a sensibilidade do parâmetro δ_{min} e $|\mathcal{S}|$. Contudo, os experimentos realizados o δ_{min}=1.5, juntamente com o *sub judice*, apresentam resultados superiores ao uso de um modelo estático em todas as coleções. Em relação ao $|\mathcal{S}|$, tamanho da fila do *sub judice*, argumentamos que a partir de uma caracterização do tipo de evento este parâmetro pode ser configurado com um nível de segurança.

Outra limitação é o caso de não garantirmos que o modelo continue consistente durante toda a análise, uma vez que a inserção de mensagens com o rótulo incorreto pode deteriorar o modelo. Entretanto, foram impostas restrições severas de volume de dados rotulados para treinamento, e ainda assim o modelo continuou consistente e com desempenho em predição superior aos algoritmos no estado da arte. Além disso, formas de verificar se o modelo continua consistente podem ser implementadas facilmente. Por exemplo, com a avaliação periódica do modelo com novas sementes de dados rotulados ao decorrer do fluxo.

Como trabalhos futuros pretende-se minimizar a dependência do parâmetro do tamanho da fila do *sub judice*. Esta dependência pode ser diminuída a partir de um tamanho de fila adaptativo. Este tamanho pode ser adaptado a partir de detectores de mudanças do fluxo. Além disso, espera-se integrar a técnica de esquecimento, Janela Deslizante Ativa, propostas neste trabalho ao algoritmo de auto treinamento.

Referências Bibliográficas

Abdulsalam, H.; Skillicorn, D. B. & Martin, P. (2008). Classifying evolving data streams using dynamic streaming random forests. Em *Proceedings of the 19th international conference on Database and Expert Systems Applications*, DEXA '08, pp. 643--651, Berlin, Heidelberg. Springer-Verlag.

Aggarwal, C. C. (2006). On biased reservoir sampling in the presence of stream evolution. Em *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pp. 607--618. VLDB Endowment.

Agrawal, R.; Imielinski, T. & Swami, A. (1993). Mining association rules between sets of items in large databases. Em *SIGMOD*, pp. 207--216. ACM.

Aha, D. W.; Kibler, D. & Albert, M. K. (1991). Instance-based learning algorithms. *Mach. Learn.*, 6:37--66.

Al-Kateb, M.; Lee, B. S. & Wang, X. S. (2007). Adaptive-size reservoir sampling over data streams. Em *Proceedings of the 19th International Conference on Scientific and Statistical Database Management*, SSDBM '07, pp. 22--, Washington, DC, USA. IEEE Computer Society.

Balcan, M.-f. & Blum, A. (2005). A pac-style model for learning from labeled and unlabeled data. Em *In Proceedings of the 18th Annual Conference on Computational Learning Theory (COLT*, pp. 111--126. COLT.

Bayardo, R.; Goethals, B. & Zaki, M., editores (2004). *Workshop on Frequent Itemset Mining Implementations*, volume 126.

Ben-Haim, Z. & Eldar, Y. (2009). A lower bound on the bayesian mse based on the optimal bias function. *Information Theory, IEEE Transactions on*, 55(11):5179 – 5196.

Bermingham, A. & Smeaton, A. F. (2010). Crowdsourced real-world sensing: sentiment analysis and the real-time web. Em *n AICS 2010 - Sentiment Analysis Workshop at Artificial Intelligence and Cognitive Science*.

Bifet, A. (2010). Adaptive stream mining: Pattern learning and mining from evolving data streams. Em *Proceeding of the 2010 conference on Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams*, pp. 1--212, Amsterdam, The Netherlands, The Netherlands. IOS Press.

Bifet, A. & Frank, E. (2010). Sentiment knowledge discovery in twitter streaming data. Em *Discovery Science*, pp. 1–15.

Bifet, A. & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. Em *SIAM ICDM*.

Bifet, A.; Holmes, G.; Pfahringer, B.; Kirkby, R. & Gavaldà, R. (2009). New ensemble methods for evolving data streams. Em *Proc SIGKDD*, KDD '09, pp. 139--148.

Blum, A. & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. Em *Proceedings of the eleventh annual conference on Computational learning theory*, COLT' 98, pp. 92--100, New York, NY, USA. ACM.

Breiman, L.; Friedman, J.; Olshen, R. & Stone, C. (1984). Classification and regression trees. *Wadsworth Intl.*

Chang, C. C. & Lin, C. J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2:27:1--27:27.

Chapelle, O.; Schölkopf, B. & Zien, A. (2006). *Semi-Supervised Learning*. MIT Press.

Chew, C. & Eysenbach, G. (2010). Pandemics in the age of twitter: Content analysis of tweets during the 2009 h1n1 outbreak. *PLoS ONE*, 5(11):13.

Chu, F.; Wang, Y. & Zaniolo, C. (2004). Mining noisy data streams via a discriminative model. Em *Discovery Science*, pp. 47–59.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Dawid, A. P. (1984). Present Position and Potential Developments: Some Personal Views: Statistical Theory: The Prequential Approach. *Journal of the Royal Statistical Society. Series A (General)*, 147(2):278--292.

Diakopoulos, N. A. & Shamma, D. A. (2010). Characterizing debate performance via aggregated twitter sentiment. Em *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pp. 1195--1198, New York, NY, USA. ACM.

Easley, D. & Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, New York, NY, USA.

Forman, G. (2006). Tackling concept drift by temporal inductive transfer. Em *SIGIR*, pp. 252–259.

Fukuhara, T.; Nakagawa, H. & Nishida, T. (2007). Understanding sentiment of people from news articles: Temporal sentiment analysis of social events. Em *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

Gama, J. & Mohamed, M. G. (2007). *Learning from Data Streams – Processing techniques in Sensor Networks*. Springer.

Gama, J. a.; Rodrigues, P. P. & Sebastião, R. (2009). Evaluating algorithms that learn from data streams. Em *Proceedings of the 2009 ACM symposium on Applied Computing*, SAC '09, pp. 1496--1500, New York, NY, USA. ACM.

Han, J.; Pei, J.; Yin, Y. & Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining Knowledge Discovery*, 8(1):53--87.

Hulten, G.; Spencer, L. & Domingos, P. (2001). Mining time-changing data streams. Em *Proc SIGKDD*, pp. 97--106.

Jansen, B. J.; Zhang, M.; Sobel, K. & Chowdury, A. (2009). Micro-blogging as online word of mouth branding. Em *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI EA '09, pp. 3859--3864, New York, NY, USA. ACM.

John, G. H. & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. Em *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, UAI'95, pp. 338--345, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Kivinen, J. & Mannila, H. (1994). The power of sampling in knowledge discovery. Em *Proc SIGACT-SIGMOD-SIGART*, pp. 77--85. ACM.

Li, P.-P.; Wu, X. & Hu, X. (2010). Learning from concept drifting data streams with unlabeled data. Em *AAAI*.

Li, W.; Jin, X. & Ye, X. (2007). Detecting change in data stream: Using sampling technique. Em *Proceedings of the Third International Conference on Natural Computation - Volume 01*, ICNC '07, pp. 130--134, Washington, DC, USA. IEEE Computer Society.

Masud, M. M.; Gao, J.; Khan, L.; Han, J. & Thuraisingham, B. (2008). A practical approach to classify evolving data streams: Training with limited amount of labeled data. Em *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pp. 929--934, Washington, DC, USA. IEEE Computer Society.

Pang, B.; Lee, L.; & Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. Em *EMNLP*, pp. 79–86.

Pang, B. & Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2:1--135.

Platt, J. C. (1999). *Fast training of support vector machines using sequential minimal optimization*, pp. 185--208. MIT Press, Cambridge, MA, USA.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Seidl, T.; Assent, I.; Kranen, P.; Krieger, R. & Herrmann, J. (2009). Indexing density models for incremental learning and anytime classification on data streams. Em *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, pp. 311--322, New York, NY, USA. ACM.

Settles, B. (2009). Active Learning Literature Survey. Relatório técnico 1648, University of Wisconsin–Madison.

Shannon, C. E. (2001). A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5:3--55.

Silva, I. S.; Barbosa; Veloso, A.; Jr., W. M. & Ferreira, R. (2011a). Análise adaptativa de fluxo de sentimento baseada em janela deslizante ativa. Em *Simpósio Brasileiro de Banco de Dados (SBBD)*.

Silva, I. S.; Gomide, J.; Barbosa, G.; Veloso, A.; Santos, W.; Ferreira, R. & Jr., W. M. (2011b). Observatório da dengue: Surveillance based on twitter sentiment stream analysis. Em *Simpósio Brasileiro de Banco de Dados (SBBD)*.

Silva, I. S.; Gomide, J.; Veloso, A.; Meira, Jr., W. & Ferreira, R. (2011c). Effective sentiment stream analysis with self-augmenting training and demand-driven projection. Em *Proc SIGIR*, pp. 475--484.

Tan, P.; Kumar, V. & Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. Em *SIGKDD*, pp. 32–41.

Veloso, A. & Meira Jr., W. (2011). *Demand-Driven Associative Classification*. SpringerBriefs Computer Science.

Veloso, A.; Meira Jr., W. & Zaki, M. J. (2006). Lazy associative classification. Em *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pp. 645--654, Washington, DC, USA. IEEE Computer Society.

Vorburger, P. & Bernstein, A. (2006). Entropy-based concept shift detection. Em *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pp. 1113--1118, Washington, DC, USA. IEEE Computer Society.

Widmer, G. & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101.

Wu, S.; Yang, C. & Zhou, J. (2006). Clustering-training for data stream mining. Em *Proc ICDM*, pp. 653–656.

Zaki, M.; Parthasarathy, S.; Ogihara, M. & Li, W. (1997). New algorithms for fast discovery of association rules. Em *SIGKDD*, pp. 283–286.

Zhang, P.; Zhu, X. & Shi, Y. (2008). Categorizing and mining concept drifting data streams. Em *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pp. 812--820, New York, NY, USA. ACM.

Zhang, Y.-s.; Zhang, J.-p.; Yang, J. & Yin, Z.-W. (2009). Svms' cooperative learning strategy based on mas to data streams mining. Em *Proc ICICSE*, pp. 156–159, Washington, DC, USA. IEEE Computer Society.

Zhu, X.; Zhang, P.; Lin, X. & Shi, Y. (2007). Active learning from data streams. Em *Proc ICDM*, pp. 757–762.

Zhu, X.; Zhang, P.; Lin, X. & Shi, Y. (2010). Active learning from stream data using optimal weight classifier ensemble. *Trans. Sys. Man Cyber. Part B*, 40:1607--1621.