SISTEMA AUTONÔMICO DE RASTREAMENTO DE TÓPICOS

Thalles Rodrigues de Sá Moraes

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Geraldo Bonorino Xexéo

Rio de Janeiro
Setembro de 2016

SISTEMA AUTONÔMICO DE RASTREAMENTO DE TÓPICOS

Thalles Rodrigues de Sá Moraes

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

_____

Prof. Geraldo Bonorino Xexéo, D.Sc.

_____

Prof. Jano Moreira de Souza, Ph.D.

_____

Prof.ª Adriana Santarosa Vivacqua, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO DE 2016

Moraes, Thalles Rodrigues de Sá

Sistema Autonômico de Rastreamento de Tópicos/ Thalles Rodrigues de Sá Moraes. – Rio de Janeiro: UFRJ/COPPE, 2016.

XI, 68 p.: il.; 29,7 cm.

Orientador: Geraldo Bonorino Xexéo

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2016.

Referências Bibliográficas: p.58-68.

1. Rastreamento De Tópicos. 2. Sistemas Autonômicos. 3. Construção de Consultas. I. Xexéo, Geraldo Bonorino. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Agradecimentos

Gostaria de agradecer a minha família que sempre me apoiou e motivou.

A minha namorada Alyne pela motivação e compreensão.

Aos meus amigos por ajudar a manter meu interesse em computação sempre em alta.

Ao meu orientador Geraldo Bonorino Xexéo, pela paciência e dedicação.

Ao meu coorientador Bruno Adam Osiek pela dedicação e motivação.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SISTEMA AUTONÔMICO DE RASTREAMENTO DE TÓPICOS

Thalles Rodrigues de Sá Moraes

Setembro/2016

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

Uma grande parte da informação disponível na web está escondida e somente pode ser acessada através de consultas. Para obter informações sobre um determinado assunto de interesse devem ser feitas consultas pertinentes. Com o passar do tempo novas informações são adicionadas ao assunto, e a consulta deve sofrer mudanças para continuar obtendo o conteúdo desejado. Neste trabalho desenvolvemos um sistema que faz consultas para recuperar documentos de determinado tópico e realiza mudanças para que essas continuem recuperando documentos relevantes quando novas informações são adicionadas ao tópico.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

AUTONOMIC TOPIC TRACKING SYSTEM

Thalles Rodrigues de Sá Moraes

September/2016

Advisor: Geraldo Bonorino Xexéo

Department: Computer and Systems Engineering

A large part of the available information on the web is hidden and can only be accessed through queries. To obtain information on a particular subject of interest relevant queries should be made. As time goes by new information is added to the subject and the query must undergo changes to keep retrieving the desired content. In this work we propose a system that creates queries to retrieve documents of a particular topic and makes changes to keep retrieving relevant documents even when new information is added to the topic.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Equações

1 Introdução

Este capítulo descreve a motivação para o presente trabalho, quais são os objetivos e os problemas de pesquisa associados, uma discussão sobre a metodologia, o detalhamento da estrutura desta dissertação e as contribuições do trabalho.

1.1 Motivação

Uma grande parte da informação disponível na internet está disponível somente na *deep web*, esse termo designa as informações que não podem ser obtidas de forma tradicional impossibilitando o funcionamento dos rastreadores que seguem ligações.

As informações da *deep web* são acessíveis somente através de consultas feitas em interfaces com o usuário ou interfaces de programação de aplicação (*API, do inglês Application programming interface*). O conteúdo criado por usuários de algumas redes sociais pode ser considerado integrante da *deep web*, esse conteúdo é de extremo valor para obtenção de fatos e opiniões pois qualquer pessoa pode expressar a sua opinião nas redes. Adicionalmente a velocidade de divulgação é enorme. Para obter essas informações de forma automática é necessário que consultas pertinentes sejam executadas, entretanto essa tarefa não é fácil pois uma consulta abrangente tem muitas respostas irrelevantes, e uma muito específica tem uma cobertura incompleta. Outro desafio é a evolução dos fatos de interesse, que degrada a eficácia das consultas.

1.2 Objetivo

O objetivo do trabalho é projetar um sistema capaz de coletar informações sobre um determinado evento dentro de um fluxo de mensagens, se estas forem acessíveis somente

através de consultas, considerando que há mudanças nas palavras chaves que caracterizam o evento ao longo do tempo.

1.3 Os Problemas de Pesquisa

Os problemas de pesquisa abordados no trabalho são:

- Os tópicos textuais evoluem?
- Existe uma abordagem para o problema de realizar consultas que recuperem documentos em tópicos dinâmicos?
- O arcabouço autonômico é suficiente para organizar o sistema de forma a almejar o funcionamento contínuo a longo prazo?
- Encontrar uma forma de auto avaliação para um sistema autonômico que não tem resposta constante externa.

1.4 Escopo e Definições

Neste trabalho serão discutidas as áreas que compõem a base teórica para o sistema desejado, serão discutidas as propriedades autonômicas desejadas para tal sistema e apresentados os resultados de uma prova de conceito que implementa a estrutura autonômica e propriedade de autoconfiguração para geração das consultas.

1.5 Metodologia

Os problemas de pesquisa relacionados a consulta induziram uma busca na literatura, e essa levou ao estudo mais profundo de duas áreas para obtenção de uma base para

tratamento do problema e concepção de uma solução. Estas áreas são: Detecção e Rastreamento de Tópicos e *Deep Web Crawling.*

A solução proposta está sujeita a perda de eficácia no decorrer do tempo, para tratar tal problema o arcabouço de sistemas autonômicos é utilizado para enunciar propriedades necessárias para manter o bom funcionamento. Nesse arcabouço a elaboração da consulta se torna a propriedade de autoconfiguração. Para avaliar a estratégia de autoconfiguração utilizada e a estrutura autonômica foi criada uma prova de conceito que tem os resultados avaliados e discutidos.

1.6 Organização da dissertação

O capítulo 2 trata de forma resumida sobre os conceitos necessários para compreensão e desenvolvimento do problema de rastreamento de tópicos e acesso a informação através de consultas na *Deep Web* e sobre Informação nas Redes Sociais.

O capítulo 3 faz uma brevíssima introdução a área de aprendizado de máquina, que é utilizada na estratégia de geração de consultas e no sistema autonômico proposto

O capítulo 4 aborda sistemas autonômicos e uma proposta para a forma de auto avaliação do sistema.

O capítulo 5 discute o sistema proposto e o capítulo 6 avalia o resultado da estrutura autonômica com a propriedade de autoconfiguração implementada.

O capítulo 7 traz conclusões e indica trabalhos futuros.

1.7 Contribuições

Este trabalho tem duas contribuições, são elas:

- Uma proposta de sistema autonômico de rastreamento de tópicos.

- Uma abordagem para o problema de gerar consultas para obter informações sobre tópicos dinâmicos.

2 Informação

O objetivo desta seção é dar uma visão superficial da área de busca e recuperação, para possibilitar a compreensão da função do sistema proposto.

2.1 Introdução

Informação é um conceito peculiar, que embora facilmente entendido por todas as pessoas, não possui uma definição qualitativa amplamente aceita e nem uma teoria quantitativa única que cubra todos os seus usos (VAN BENTHEM e VAN ROOY, 2003;PIETER, 2013).

Neste trabalho será assumido o significado de objeto de interesse, aquele que se deseja obter, podendo assim assumir a forma de notícias, livros, arquivos multimídia, e-mails, páginas da internet, trechos de vídeos, fotos de um local específico ou evento, ou o significado do que se é consumido ao explorar um desses objetos.

Para entender melhor a última definição, toma-se emprestado um modelo utilizado em BELKIN e CROFT, 1992. Uma pessoa com um objetivo percebe que seus recursos e conhecimentos atuais são inadequados para cumprir o mesmo. Este estado pessoal pode ser chamado de estado anômalo de conhecimento (BELKIN e CROFT, 1987), ou necessidade de informação. Esse estado leva a pessoa a assumir um comportamento de procura de conteúdo, e ao obter esse ela mudará seu estado consumindo um objeto encontrado. Nesse modelo o objeto é somente o meio, não existindo um específico a ser encontrado.

A necessidade de lembrar como é o Coliseu, o levará a busca por fotos do mesmo, e cada foto encontrada atenderá de forma diferente esta necessidade. Esta diferença de satisfação é chamada de relevância.

Na primeira definição de informação, os objetos encontrados nessa busca têm relevância binária, os objetos encontrados são exatamente os que eram desejados ou não, enquanto na seguinte existe apenas uma ordem entre os objetos encontrado, uns satisfazem a necessidade melhor que outros.

A busca por informação auxiliada pelo computador é o assunto do campo de pesquisa cunhado de Recuperação de Informação, que pode ser definido como SALTON, 1983:

"Recuperação da Informação trata da representação, armazenamento, organização e acesso a itens de informação, a princípio nenhuma restrição é posta no tipo de item tratado na recuperação da informação"

A definição acima leva em consideração somente a visão centrada no lado da máquina, existe ainda uma outra que leva em consideração o lado do usuário, onde o seu comportamento é estudado, suas necessidades e também o quanto esses dados podem afetar a organização e a operação do sistema de recuperação da informação (BAEZA-YATES e BERTHIER, 1999), entretanto essa visão não será abordada nesse trabalho. Para este tópico consultar INGWERSEN, 2002.

Neste trabalho somente serão abordados os problemas referentes a informação na forma textual, ignorando outros tipos de media, pois essas fogem do escopo do trabalho. Entretanto algumas seções deste trabalho são aplicáveis ao campo amplo de recuperação da informação.

As atividades descritas na definição podem ser divididas em atividades relativas ao processo de armazenar e ao processo de acessar.

2.2 Armazenar

Uma representação adequada dos objetos que serão armazenados é o primeiro passo para o bom funcionamento de um sistema de RI, ela deve ser tal que exponha as características que serão procuradas pelos usuários quando a recuperação for necessária. Existe uma ligação inerente entre o modelo utilizado para representar os documentos e o modelo utilizado para as representar consultas que desejam recuperar estes documentos. Sendo assim o desenvolvimento de ambos se dá de forma combinada, com o objetivo de melhorar sempre a eficácia de uma busca, ou seja, resolver da melhor forma o problema de RI, que pode ser definido como:

"Recuperar todos os documentos que são relevantes para uma consulta do usuário, recuperando a menor quantidade possível de documentos não relevantes". (BAEZA-YATES e BERTHIER, 1999)

Para entender a escolha pelo modelo utilizado atualmente, é necessário entender que os modelos evoluíram historicamente com o desenvolvimento das máquinas mecânicas e eletromecânicas e a adaptação dos sistemas de catálogo de bibliotecas (SANDERSON e CROFT, 2012). No início os documentos eram armazenados em categorias, cada documento era associado uma categoria, e subcategorias no que é conhecido como *Dewey Decimal System*, utilizado nas bibliotecas, esse sistema tem como objetivo classificar todo o conhecimento humano. Realizar uma busca nesse sistema é restringir categoria e subcategorias sucessivamente até a granularidade mínima do sistema.

O próximo passo foi a inclusão de palavras chave na busca, estas são palavras relevantes de um documento, sugerida em TAUBE et al., 1952. Algo bem próximo do que temos hoje, onde todo o conteúdo do documento é usado, exceto *stop-words*, com algumas transformações para reduzir palavras semelhantes a uma só representação. Os

documentos são analisados como pontos em um espaço vetorial constituído por palavras únicas, sendo assim os documentos são reduzidos a combinações lineares das palavras (SWITZER, 1964). Essa análise é muito utilizada. Entretanto não é prático olhar todo o conteúdo dos documentos para encontrar por palavras procuradas, e por isso utilizam-se índices para estas palavras. Assim como podemos imaginar como índices as abordagens anteriores, inclusive o *Dewel Decimal System*, que possibilitava agrupar documentos de subcategorias próximas e assim dispensar a necessidade de procurar em todos as subcategorias para se encontrar um objeto.

Índices invertidos são os mais utilizados para esse caso, cada palavra no universo de todas as palavras indexadas indica um conjunto com todos os documentos que contém a mesma. Mas existe perda de informação nessa representação. Observando por outro aspecto os documentos textuais podem ser representados como um conjunto ordenado de palavras, nesse caso não há nenhuma perda de informação, entretanto essa representação não se mostrou propícia para o desenvolvimento do IR.

Uma redução deste modelo é a representação por *N-GRAMS*, onde cada palavra tem a informação de ordem das palavras que estão distantes dela até N posições. Esse modelo é utilizado para descobrir diversas informações, como termos para enriquecimento das consultas e adição de palavras relacionadas para ampliar os resultados (MANNING e SCHÜTZE, 1999). Mesmo com grande perda de informações o modelo que mais se desenvolveu, por seus menores requisitos computacionais, maior simplicidade e bons resultados foi o *BAG-OF-WORDS*, utilizado na representação vetorial, onde toda a informação de ordem é perdida, e só a quantidade de ocorrências da cada palavra é computada.

Entretanto várias estudos para evolução do IR utilizam cada vez mais informação sobre proximidade e ordem (MISHNE e RIJKE, 2005).

É possível que os sistemas evoluam para um modelo que considere outros fatores, além desses para representar melhor os documentos, incluindo funções semânticas e outros (LI e XU, 2012).

O modelo que proporcionou os resultados que temos hoje em dia e o utilizado no sistema proposto utiliza somente informação sobre quais palavras e a quantidade de cada, este permite representar os documentos como vetores em um espaço n-dimensional, onde cada palavra única representa uma dimensão e o módulo da componente do vetor naquela dimensão a quantidade de repetições da palavra, ou uma métrica baseada nesta,

2.3 Acessar

A tarefa do usuário é traduzir a sua necessidade de informação em uma linguagem de consulta específica para o sistema (BAEZA-YATES e BERTHIER, 1999). Um sistema deve facilitar essa tarefa, utilizando uma linguagem de consulta mais próxima da intenção do usuário, ou guia-lo de forma a facilitar essa tradução.

Os livros são indexados pelo *Dewel Decimal Systems*, uma consulta é descobrir o código da subcategoria desejado e procurar nas seções indexadas pelo código.

Quando termos passam a ser utilizados precisamos separar seções não disjuntas referentes a cada termo, e restringir esse conjunto de documentos com cortes sucessivos das áreas de interesse na interseção ou disjunção dos termos. Esse é o procedimento chamado de Recuperação Booleana. São retornados os documentos que possuem o conteúdo de busca de acordo com a inclusão ou exclusão de termos. O modelo de espaço vetorial pode ser usado para organizar a relevância dos documentos, para isso será calculada a distância dos termos de pesquisa para cada documento, e estes serão ordenados para mostrar os mais próximos primeiro.

2.4 Information Filtering

Information filtering trata o desafio de destinar informação relevante para o usuário. Um perfil é definido, e então informações que se encaixem nesse perfil são destinadas a esse. A atividade é exercida geralmente num fluxo de dados, e os perfis de interesse podem ser dinâmicos, modificados pelos usuários (HANANI, et al., 2001). Uma diferença para recuperação de informação é a ausência da preocupação com a interação com o usuário, os perfis de usuários são considerados especificações corretas das necessidades de informação (BELKIN e CROFT, 1992). *Information filtering* pode ser visto como uma outra face de Recuperação de Informação, utilizando as mesmas técnicas ou técnicas semelhantes a essa, que serão discutidas neste trabalho na parte relacionada a TDT. Estratégias de criação do perfil do usuário são também estudadas na área mais recente chamada de sistemas de recomendação.

2.5 Avaliação

As métricas utilizadas mais frequentemente em Recuperação da informação são precisão e cobertura (BAEZA-YATES e BERTHIER, 1999), e o uso dessas se espalhou para outras áreas como aprendizado de máquina.

Para problemas de RI temos itens alvo (relevantes) contidos em uma coleção maior, e o objetivo é selecionar dentro dessa coleção completa o subconjunto que contém o máximo de itens relevantes e mínimo de irrelevantes. Esse objetivo é expressado na métrica precisão.

Precisão = número de relevantes selecionados / todos os selecionados

Equação 1 Precisão

A quantidade de elementos relevantes não selecionados no subconjunto é uma informação importante que não é informada na métrica precisão. Para essa informação existe a métrica cobertura que é a razão entre os relevantes selecionados e o total de relevantes.

Cobertura = número de relevantes selecionados / todos os relevantes

Equação 2 Cobertura

Existe uma relação entre as duas métricas, é possível aumentar artificialmente uma, diminuindo a outra. Por isso ambas devem ser levadas em consideração em uma avaliação.

Por causa dessa relação pode ser conveniente combinar acurácia e cobertura em uma única medida (MANNING e SCHÜTZE, 1999), um forma de combinação é a *F-measure* que foi concebida por RIJSBERGEN, 1979. α determina o peso dado entre precisão e cobertura, usualmente 0,5 para pesos idênticos.

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

Equação 3 F-Measure

2.6 Rastreio e Detecção de Tópico

O programa *TDT* (*do inglês, Topic Detection and Tracking*) começou em 1997 com um estudo piloto envolvendo um número pequeno de pesquisadores. Durante o estudo piloto e os anos seguintes foram definidas as tarefas que formaram o *TDT* e realizadas competições abertas para avaliar os avanços da área.

TDT é uma área de pesquisa criada com o objetivo de tratar notícias de diversas mídias em tempo real, organizando-as em eventos. A motivação para criação da área é a

construção de um sistema que fosse capaz de monitorar notícias e alertar analistas para eventos novos de interesse destes, acontecendo no mundo (ALLAN, 2002).

A ideia da área é similar à de áreas anteriores como Filtragem da Informação (*Information Filtering)*, e Recuperação da Informação (*Information Retrieval*), entretanto a noção vaga de "sobre"(*aboutness*), que indica a pertinência a um assunto, é substituída pela relação de notícias sobre um acontecimento no mundo real (ALLAN, 2002).

Outra diferença é encontrada na relação com o tempo. Uma notícia ligada a um evento está relacionada com a data de ocorrência deste, e no decorrer do tempo o evento pode evoluir, por exemplo ao incluir várias hipóteses para explicar um acontecimento. A noção de relevância entre uma notícia e um evento pode inclusive mudar através do tempo, na medida em que hipóteses são descartadas, por exemplo (ALLAN, 2002).

2.6.1 Método

A pesquisa em TDT foi dividida entre duas abordagens, *NED* e *RED* (ALLAN, *et al.*, 1998), NED (do inglês, *New Event Detection*) aborda a construção de um sistema que recebe textos em ordem cronológica e para cada texto recebido faz sua classificação em um novo tópico ou o associa a um tópico existente sem utilizar informações de textos futuros. RED (do inglês, *Retrospective Event Detection*) foca em dividir em tópicos os documentos possuindo o conhecimento de todo o conjunto de documentos. É esperado que esta abordagem tenha melhor resultados vistos que seu ambiente é completamente observável enquanto a anterior é somente parcialmente observável (RUSSELL e NORVIG, 2003).

Existe outra divisão entre os tipos de eventos detectados e rastreados, essa separa os algoritmos que tem conhecimento dos eventos que procuram e os que não têm, dividindo

estes entre eventos especificados e eventos não especificados (ATEFEH e KHREICH, 2015).

Com a categorização do problema bem definida, esse foi dividido em cinco problemas menores para facilitar sua solução (ALLAN, 2002):

1. Segmentação de histórias, considerando o texto sem delimitação de notícias essa tarefa tem como objetivo indicar o fim de uma notícia e início de outra.
2. Detecção de agrupamento, essa tarefa tem como objetivo agrupar as notícias sobre um mesmo evento.
3. Detecção de primeira história, deve identificar em uma sequência de notícias a primeira notícia sobre um tópico ou evento.
4. Monitoramento. Adicionar histórias novas a agrupamentos já existentes criados com amostras de evento.
5. Detecção de ligação entre histórias. Detectar se duas histórias são sobre o mesmo evento.

O foco deste trabalho é a tarefa 4. As técnicas também podem ser divididas em pivoteamento de documentos, quando são utilizadas somente informações textuais contidas nos mesmos, e pivoteamento de atributos, quando são utilizadas também outras informações, como o tempo (ATEFEH e KHREICH, 2015).

2.6.2 Segmentação De História

Um algoritmo de segmentação de histórias deve ser capaz de dividir um corpus composto de texto não demarcado, como por exemplo a transcrição de um programa de notícias, em textos menores onde cada um representa um trecho contínuo sobre uma notícia. Este também deve ser capaz de processar textos com ruído, provenientes de um programa de

reconhecimento de voz (FRANZ, *et al.*, 1999). A tarefa de segmentação pode ser transformada em um problema de aprendizado de máquina de seguinte formulação:

Aprender a colocar delimitadores em um texto não demarcado, observando um conjunto de exemplos rotulados (BEEFERMAN, et al., 1999).

Muitas abordagens utilizam uma medida de diferença entre a utilização de palavras nos dois lados de um limite potencial. HEARST, 1994 utiliza uma abordagem baseada em atribuir um valor para cada espaço candidato a separador, esse valor é a similaridade dos cossenos entre o trecho anterior e o seguinte ao espaço, depois disso a separação é ajustada para coincidir com os parágrafos. REYNAR, 1994 usa uma medida que calcula a repetição de palavras para separar trechos de alta coesão. Outros trabalhos, PASSONNEAU e LITMAN, 1997 e BEEFERMAN, et al., 1999 usaram árvores de decisão combinando atributos de discurso, como tempo de pausa com atributos léxicos como presença de certas palavras perto de limites, e atributos semânticos como referências entre as duas frases. YAMRON, 1998, usou uma abordagem que trata uma história como uma instância de um tópico escondido e modela um texto sem marcações como uma sequência sem rótulo de tópicos. Utilizando Cadeias Escondidas de Markov, encontrar divisões é equivalente a achar transições de tópicos.

As próximas tarefas compartilham grande parte da estratégia de solução, tendo como principal diferença o modo como o problema é submetido a avaliação.

2.6.3 Detecção de Agrupamentos

A tarefa de detecção de Agrupamentos é dividida em três fases (YANG, et al., 1998, 2002):

- Pré-Processamento de dados:

  O texto é dividido em um conjunto de sentenças, e essas são divididas em *tokens*.

São excluídas as *stop-words*, palavras que não agregam informação ao texto como artigos, preposições e pronomes, entretanto alguns trabalhos não realizam essa retirada (CATALDI, et al., 2010). Adicionalmente alguns trabalhos utilizam o processo de *stemming* (ALLAN, *et al.*, 2000), que reduz palavras a um radical comum, diminuindo a quantidade de termos únicos. Todo esse processo é semelhante ao processo que ocorre durante a etapa de indexação de documentos em ferramentas de busca e recuperação de documentos.

- Representação dos dados

A representação é feita utilizando vetores cujas entradas representam termos, e os valores destas estimam a importância do termo. Esse modelo que não utiliza informações de posicionamento de palavras é o mais utilizado. Para estimar a importância dos termos geralmente são utilizados valores *tf-idf* (SALTON, 1989) normalizados ou não.

Um vetor contendo entidades nomeadas é uma representação alternativa (KUMARAN e ALLAN, 2004), que tem como objetivo extrair informações para responder as perguntas: Quem? O quê? Quando? Onde? (MOHD, 2007). Essa foi também utilizada em abordagens híbridas (KUMARAN e ALLAN, 2004).

Representações estocásticas utilizadas incluem Modelos de Linguagem (LEEK, et al., 2002) e uma combinação de conteúdo, tempo e outros atributos (LI, *et al.*, 2005).

- Organização dos dados:

A organização é geralmente executada através de um algoritmo de "clusterização" (*clustering*) de passagem única. Em uma abordagem simples a similaridade entre os documentos é calculada e então os que possuem maior semelhança são agrupados iterativamente, até que não exista documento ou grupo com similaridade maior do que uma semelhança mínima, constante escolhida, não pertencentes ao mesmo tópico.

Se essa for menor do que a semelhança mínima, esse é considerado um novo evento.

A similaridade entre documentos utiliza métricas mais tradicionais como a distância euclidiana, coeficiente de Pearson e a similaridade de cossenos e outras como a distância de Hellinger (BRANTS, et al., 2003) e o *clustering index* (JO e LEE, 2007).

As etapas de pré-processamento e representação são etapas comuns as próximas tarefas de TDT apresentadas a seguir.

2.6.4 Detecção da Primeira História

Um algoritmo para realizar essa tarefa deve ser capaz de monitorar um fluxo de notícias e indicar ao encontrar a primeira notícia discutindo um evento.

Uma abordagem simples, porém muito utilizada, consiste em comparar uma história nova com todas as histórias anteriores, se não houver história com semelhança que exceda um determinado limiar, esta será considerada pertencente a tópico novo. Em (ALLAN, *et al.*, 1998) cada notícia é representada usando um vetor no espaço de termos com os pesos dados por medidas como Okapi's tfi-df ou sua versão mais simples tf-idf.

Uma abordagem simplificada compara somente com as notícias já declaradas como primeira história.

2.6.5 Monitoramento

No Monitoramento o sistema recebe um pequeno número de histórias sobre um mesmo evento e então monitora o fluxo de notícias subsequentes para indicar outras que tenham como assunto o mesmo evento. O primeiro passo é a representação do tópico e depois a filtragem do fluxo com as decisões sendo tomadas na medida em que os documentos chegam. Pontuações são atribuídas as notícias e essa comparada com um limite mínimo para a atribuição ao tópico. As abordagens mais simples dessa tarefa utilizam a

representação de um tópico como um vetor no espaço de termos, com as entradas correspondentes a pesos das palavras. Os valores dos tópicos são calculados como o centroide das notícias correspondentes ao tópico e então cada história nova é comparada com todos os centroides, para ser associada a um deles. Para comparação são utilizadas métricas descritas na tarefa de organização.

2.6.6 Detecção de Ligação Entre Histórias

A tarefa de detecção de ligação entre duas histórias é definida no TDT com o objetivo de indicar se duas histórias têm como assunto o mesmo evento. Assim como as demais tarefas a abordagem de modelar a história como um vetor de termos e utilizar a semelhança dos cossenos é bem utilizada por sua simplicidade, independência de contexto e idioma.

2.7 Informação na Deep Web

Deep Web é o nome dado a parte da web que não está na superfície, isto é, não é indexada diretamente pelos buscadores tradicionais. Esse conteúdo geralmente se encontra em bancos de dados que são usados para gerar as páginas durante a interação com o usuário ou programa e essas não permanecerão disponíveis através de links na parte que está na superfície dessa entidade.

Se a web da superfície é valiosa e repleta de informações, é de se imaginar que a Deep Web também o seja, e alguns estudos fortalecem essa suposição afirmando que ela possui informações de alta qualidade e seu tamanho estimado é de centenas de vezes o tamanho da web de superfície (BERGMAN, 2001;HE, *et al.*, 2007;MADHAVAN, *et al.*, 2007).

A dificuldade de se obter informações da *deep web* causa uma falha significante de cobertura nos mecanismos de busca (MADHAVAN, *et al.*, 2008). Os métodos utilizados para obter acesso ao conteúdo da *deep web* são classificados em dois tipos (BERGMAN, 2001):

- Integração Virtual, ou Descoberta e Redirecionamento (HE, *et al.*, 2007).
- *Emerge*, *Surfacing* ou "buscar e indexar" (*Crawl-and-index*), (HE, *et al.*, 2007; MADHAVEN, *et al.*, 2009).

Integração Virtual é a estratégia de redirecionar o usuário até que o mesmo esteja em uma entidade capaz de lhe fornecer as informações que procura (SHOKOUHI e SI, 2011). Após a consulta do usuário, o buscador descobre qual entidade da *deep web* tem mais relevância para sua pesquisa e direciona o usuário para refazer sua consulta naquela entidade. Alguns trabalhos nessa área são HE, *et al.*, 2003; CHANG, et al. 2005; SARMA, et al., 2008.

Emergir é solução que tem como objetivo pesquisar a *deep web*, simulando a interação com usuários em campos de pesquisa, recuperando e indexando todos os resultados.

O desafio na abordagem de emergir é definir uma estratégia eficiente para gerar as consultas com o objetivo de recuperar o máximo de conteúdo. Essa é a abordagem adotada no trabalho.

Considerando os custos de obtenção desses registros adicionamos mais uma dimensão ao problema (BARBOSA e FREIRE, 2004) (WU, *et al.*, 2006). Os métodos de criação de consultas são classificados de acordo com o uso o não de conhecimento sobre o domínio que efetuarão a busca em:

- Métodos com conhecimento Prévio
- Métodos sem Conhecimento Prévio

Trataremos somente dos métodos sem conhecimento prévio. Esses métodos criam consultas candidatas analisando os resultados já obtidos nas consultas. (BARBOSA e FREIRE, 2004) foi pioneiro nesses métodos, e apresentou um método de criação de consultas candidatas que utilizava as palavras mais frequentes dos resultados anteriores. Entretanto isso não garante que mais registros novos serão recuperados. NTOULAS, et al.,2005 Propuseram um método baseado na taxa de retorno esperado. Nesse método as consultas candidatas são criadas a partir dos registros obtidos e a expectativa de retorno de cada termo é calculada, o termo com maior expectativa é selecionado. WU, *et al.*, 2006 Modelaram cada entidade da *Deep web* como um grafo de atributos e valores utilizados nas suas consultas, e utilizando esse framework teórico o problema de encontrar a melhor sequência de consultas foi aproximado ao problema de encontrar o conjunto mínimo dominante de vértices com pesos. Essa estratégia é a utilizada neste trabalho de forma adaptada.

2.7.1 Grafo Atributo-Valor

É possível modelar uma entidade da deep web como um banco de dados DB, onde cada entrada t_i representa um item a ser retornado em uma busca, temos:

$DB=\{t_1,t_2,...,t_n\}$.

Onde DB tem sua estrutura definida por k atributos, formando o conjunto de atributos:

$AS=\{attr_1, attr_2, ..., attr_k\}$

Definimos então o conjunto de atributos distintos DAV, formado por todos os pares de atributo e valor distintos encontrados em DB, o par (x,y) pertence a DAV se existe

atributo x em AS e existe t em DB que possui y como valor no atributo correspondente x em alguma instância t_i em DB.

Estamos prontos para definir o grafo Atributo-Valor AVG:

G(V,E) para o banco DB é um grafo não-direcionado que é construído da seguinte forma: Para cada par (x,y) em DAV existe um único vértice v_i pertencente a V que representa esse.

Uma aresta (v_i,v_j) pertence a E se os pares que estes vértices representam coexistem em um documento, ou seja, existe um item t que contém o par atributo x_i e valor y_i e também contém o par atributo x_j e o valor y_j.. Um exemplo é exibido na tabela 1 e seu AVG correspondente na figura 1

	Item 1	**Item 2**	**Item 3**
Atributo 1	A1	A2	A2
Atributo 2	B1	B1	B3
Atributo 3	C1	C2	C2

Tabela 1 Exemplo DB com atributos e valores

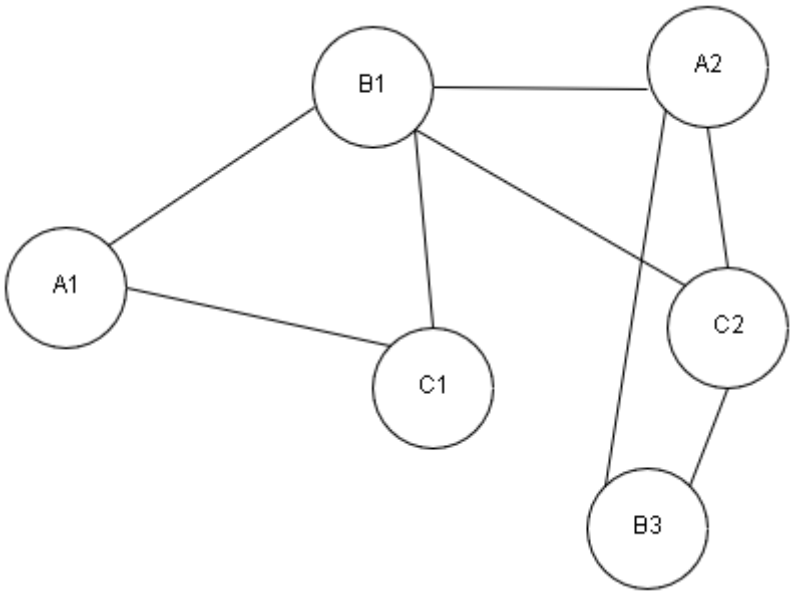

Figura 1 Exemplo grafo AVG

A cada passo dado, uma busca pelo par atributo valor de um vértice, novos vértices e arestas são descobertos, essas ligadas ao vértice alvo e também novas arestas para o grafo já descoberto anteriormente.

Este grafo transforma o problema de obter itens de uma entidade no problema de percorrer um grafo assim como é feito na web tradicional, começando com um conjunto de vértices e em cada etapa escolhendo um vértice para visitar formulando a pesquisa.

2.8 Informação nas Redes Sociais

A última década foi marcada pela explosão das redes sociais, e com essa surgiu um interesse muito grande em pesquisas relacionadas as redes. A possibilidade de detectar movimentos reais através de sua expressão em redes sociais é uma das mais exploradas.

2.8.1 Twitter

O Twitter é uma das principais redes sociais do mundo com mais de 250 milhões de usuários ativos por mês, essa popularidade o transformou em um meio rápido para propagação de notícias de última hora (AMER-YAHIA, *et al.*, 2012; PHUVIPADAWAT e MURATA, 2010; SANKARANARAYANAN, *et al.*, 2009).

2.8.2 Dificuldades Redes Sociais

Dados de redes sociais são em geral difíceis de se obter, a maior parte das redes sociais restringe o acesso aos dados. O Twitter adotou o caminho contrário, disponibilizando uma *API*(do inglês, *Application Programming Interface*) chamada *Twitter Streaming API* que possibilita o acesso de qualquer pessoa a uma amostra de 1% de todo o fluxo de *tweets* gratuitamente. Se os parâmetros da consulta retornarem resultados que ultrapassem esse

limite, o Twitter faz uma amostragem dos dados. A API aceita três parâmetros, palavras, limites geográficos e ids de usuários (MORSTATTER, *et al.*, 2013). Graças a essa facilidade de obtenção dos dados o Twitter passou a ser estudado de diversas formas. O Twitter também oferece outros modos pagos de acesso aos dados, inclusive uma versão integral, comparações entre a amostra e a versão integral foram feitas em MORSTATTER, *et al.*, 2013.

2.8.3 Detecção e Monitoramento no Twitter

Em diversos acontecimentos o Twitter se mostrou uma fonte de informações rica e veloz, noticiando em alguns casos em primeira mão através de usuários comuns acontecimentos, por exemplo:

- Explosões de bombas em Mumbai em novembro de 2008(OH, et al., 2010).
- A colisão de um avião no rio Hudson em janeiro de 2009
- A primavera Árabe (KHAN, 2012).

Diante desses acontecimentos, é inegável que o monitoramento do Twitter pode trazer benefícios, estudos utilizam o mesmo para detecção de tragédias (SAKAKI, et al., 2010), predição de crimes (WANG, et al., 2012) e monitoramento de atividades terrorista (COOK, *et al.*, 2014).

A detecção de tópicos é feita geralmente de modo não supervisionado, podendo ser classificada também em TDT como evento não especificado, como é observado em

CATALDI, et al., 2010 e SAYYADI, et al., 2009 nesse modo nenhuma consulta é submetida ao Twitter.

Em outros trabalhos com eventos especificados, os tweets são coletados utilizando consultas simples sobre o objeto de interesse, como POPESCU e PENNACCHIOTTI, 2010 e SAKAKI, et al., 2010, ou selecionando manualmente as palavras para pesquisar sobre o tópico (GU, *et al.*, 2011). Existem ainda os trabalhos que utilizam a consulta geográfica somente (BECKER, *et al.*, 2012).

3 Aprendizado de Máquina

Computadores que aprendem tarefas genéricas tão bem quanto pessoas são objetivos de pesquisas há muitas décadas, ainda que não tenhamos descoberto como fazê-los enfrentar qualquer problema, já existe um grande conjunto de problemas onde as máquinas que aprendem já tem uma performance que possibilita seu uso, e existem estudos de desempenho superior como reconhecimento de faces (LU e TANG) ,diagnóstico de câncer (CHA, 2015) , e prognóstico de câncer no pulmão (YU, *et al.*, 2016).

Podemos definir um programa que aprende como um programa que melhora a sua performance relativa a uma tarefa com experiência (MITCHEL, 1997). A tarefa de aprendizado pode ser classificada como supervisionada ou não-supervisionada. Quando a tarefa é não-supervisionada o objetivo é encontrar relações entre os dados, sem um rótulo especificado, encontrando agrupamentos.

Quando a tarefa é supervisionada precisamos dos exemplos de treino, esses são instâncias do problema que queremos resolver, junto com o parecer correto para cada caso. O objetivo é uma representação adequada para objeto dos exemplos e uma função F:(D) → I que recebendo esta representação retorne o resultado correto. Esta função classifica o problema em Regressão, quando a imagem é o conjunto dos Número Reais, e classificação quando a saída é binária.

A representação do objeto deve ser capaz de expor suas características necessárias para a resolução do problema, se essa não for adequada o desempenho será limitado, pois um sistema nunca poderá aprender algo que não consegue representar adequadamente (MITCHEL, 1997). Sendo assim a representação impõe um limite inicial no problema, quanto melhor for a representação maior a chance de se conseguir um bom resultado.

A função pode assumir diversas formas, alguns exemplos são:

- Um polinômio,
- Um conjunto de regras
- Uma distribuição de probabilidades

Os parâmetros ideais dessas serão calculados de forma a possuir um bom desempenho no conjunto de treino. Nem sempre os parâmetros que tem o melhor resultado são os buscados, pois é necessário que a função não seja precisa em excesso nos dados de treino de forma a não ser genérica o suficiente para ter bom desempenho nas instâncias fora desse conjunto. Outra premissa importante é que os dados dos conjuntos de teste sejam tão próximos quanto for possível das instâncias que o programa exercerá sua atividade, se isto não acontecer esse apresentará desempenho bom no conjunto de treino, porém não será útil ao enfrentar as instâncias reais do problema.

3.1 Estimando Parâmetros

De acordo com a forma escolhida para a função, determina-se o método para estimar os parâmetros dessa, no caso de um polinômio de parâmetro W, temos:

$$\mathbf{Y}(\mathrm{x}, \mathbf{w}) = \mathrm{w}_0 + \mathrm{w}_1\mathrm{x}_1 + \mathrm{w}_2\mathrm{x}_2^2 + \cdots + \mathrm{w}_\mathrm{m}\mathrm{x}_\mathrm{n}^\mathrm{n}$$

Equação 4 polinômio de parâmetro W

Transformamos o problema em um problema de minimização com uma função objetivo que represente o erro da função atual comparada aos valores das instâncias de treino. Uma função objetivo muito utilizada para isto é o erro médio quadrático (BISHOP, 2006).

$$E(w) = \frac{1}{\mathrm{N}}\sum_{k=1}^{n}\{\mathbf{Y}(x_n, \boldsymbol{w}) - t_n\}^2$$

Equação 5 Erro médio quadrático

Onde $\mathbf{Y}(\mathrm{x}_\mathrm{n},\mathbf{w})$ representa o valor da instância x calculada com a versão atual da aproximação, representada pelo vetor de parâmetros w e t_n representa o valor da instância

x_n no conjunto de treino. Conceitualmente o vetor W delimita um hiperplano que faz a divisão no espaço do vetor de atributos.

Nas próximas seções serão introduzidos de forma superficial os classificadores utilizados no trabalho, cada um deles possui sua função e forma particular de representação.

3.2 Naive Bayes

É um modelo de probabilidade condicional, de forma abstrata podemos dizer que ele atribui a cada valor de um atributo uma probabilidade de se manifestar em um evento probabilístico de interesse, é chamado de ingênuo(*naive*) porque despreza toda a informação de correlação entre os atributos para simplificação computacional (NIGAN, *et al.*, 1998).

Aplicado a classificação de textos ignora a ordem da palavra no documento e também a ocorrência de *N-GRAMS*, assim como o modelo de representação simples utilizado em Busca e Recuperação da Informação, embora teoricamente possa utilizar essas informações, tecnicamente surgem problemas de ordem computacional com o crescimento do número de atributos. Ao avaliar se um documento pertence ao tópico ele calcula a probabilidade conjunta de pertencimento dos termos ao tópico e compara com a de não pertencimento.

3.3 SVM

É um modelo algébrico que transforma um vetor de atributos de entrada de dimensão n, que pode não ser linearmente separável, em um vetor em uma dimensão muito maior, onde é construída uma função para classificá-lo (CORTES e VAPNIK, 1995).

JOACHIMS, 1998 mostrou que o SVM é capaz de superar os classificadores até então utilizados para o problema de categorização de texto.

3.4 SLDA

É um modelo probabilístico, que surgiu de modificações no LDA (BLEI, et al., 2003). O LDA é um modelo não-supervisionado que encontra tópicos nos documentos. Os tópicos encontrados não correspondem necessariamente aos esperados de acordo com alguma organização humana. Cada documento é uma mistura desses tópicos, e esses tópicos podem ser considerados como atributos para uma tarefa de classificação, essa combinação dá origem ao SLDA (BLEI e MCAULIFFE, 2007). O SLDA não é restrito a classificação em tópicos, podendo ser utilizado para qualquer tarefa de classificação de textos.

3.5 Avaliando Desempenho

Para avaliar a qualidade dos resultados quando o problema é de classificação, é comum usar as métricas: acurácia, precisão e derivadas desta, essas que foram tratadas no capítulo II.

Entretanto para problemas de busca e recuperação não é importante calcular essas relações para os não relevantes, para problemas de aprendizado de máquina é importante avaliar essas métricas para ambas as classes, dado que as ambas são relevantes para a tarefa de classificação.

Quando o problema é de regressão, o erro quadrático médio e variância do mesmo são bastante utilizados para a tarefa.

4 Sistemas Autonômicos

Os sistemas de computação atingiram um nível onde configurar e manter o funcionamento é uma tarefa complexa. Com o objetivo de tornar os sistemas autogeridos, surgiu à área de pesquisa de sistemas autonômicos (HORN, 2001), proposta pela IBM em 2001, este nome foi criado para fazer alusão ao sistema nervoso autônomo, que controla nossas atividades básicas. A nova área surge relacionada com pesquisas anteriores em sistemas multi-agentes (MARKUS, et al., 2008;WOOLDRIDGE e JENNINGS, 1995) e teoria de controle (DIAO, *et al.*, 2006).

Um sistema autonômico tem como principal objetivo se adaptar a mudanças no ambiente, com a menor intervenção humana possível. O primeiro projeto notável foi realizado pela *DARPA*(do inglês, *Defense Advanced Research Projects Agency*) para fins militares em 1997. Esse projeto surgiu antes da criação da área de pesquisa, mas seus objetivos e técnicas são de um sistema autonômico (MARKUS, et al., 2008).

O SAS(do inglês, *Situation Awareness System*) tinha como objetivo possibilitar a comunicação e localização entre soldados em combate. Um dos ajustes necessárias era o ajuste da frequência de transmissão, quanto mais alta a frequência melhor era a taxa de transferência e menor era o alcance.

Nesse sistema também é empregado um sistema de autoproteção, pois o sistema seria atacado com interceptações e interferências.

Também criado pela *DARPA* em 2000 o projeto *DASADA*(do inglês, *Dynamic Assembly for System Adaptability, Dependability, and Assurance*) tem um objetivo muito semelhante ao da área de pesquisa criada pela IBM, esse é de pesquisar e desenvolver tecnologias que possibilitem que sistemas de missão crítica possuam alta confiança, disponibilidade e adaptação (HUEBSCHER e MCCANN, 2008).

As propriedades desejadas de um sistema autonômico foram listadas pela IBM (HORN, 2001) como: autoconfiguração, auto-otimização, autocura e autoproteção.

4.1 Autoconfiguração

Um sistema autonômico se configura de forma a cumprir seu objetivo de alto nível, modificando todos os parâmetros que possui e com ajuda da arquitetura autonômica avaliando e fazendo novas mudanças quando essas são necessárias.

4.2 Autocura

Um sistema autonômico detecta e diagnostica problemas em seu funcionamento, utilizando a arquitetura autonômica para efetuar correções e avaliar se essas foram efetivas, caso não efetivas outras medidas são tomadas até que os problemas sejam corrigidos.

4.3 Autoproteção

O sistema deve realizar constantemente estratégias de defesa para se manter funcionando imune a tentativas propositais e acidentais de interromper o seu objetivo principal.

4.4 Auto-Otimização

Um sistema autonômico otimiza o uso de seus recursos por ação espontânea, para buscar melhor atender seu objetivo. Essa forma pode ser mais econômica ou não dependendo da forma que seu objetivo principal está definido.

4.5 Organização

Proposto pela IBM (IBM, 2005) o modelo MAPE-K (Monitorar, Analisar, Planejar, Executar e Conhecimento) é um modelo de referência para arquiteturas autonômicas, provavelmente inspirado pelo modelo de agente genérico proposto em RUSSELL e NORVIG, 2003 (MARKUS, et al., 2008) que mostra como um agente inteligente observa o ambiente através de sensores e usa as observações para planejar e determinar ações para realizar no ambiente.

No modelo o elemento gerenciado representa o software ou hardware que recebe comportamento autonômico através do acoplamento com um gerente autonômico.

Através de sensores o gerente obtém os dados do gerenciado, com esses dados ele monitora e analisa o comportamento deste, utilizando o seu conhecimento acrescido dos dados obtidos, ele planeja ações e as executa através de atuadores.

Formas simples de representar o conhecimento são[40] (KEPHART e WALSH, 2004):

- Regras evento-ação
- Políticas de Objetivo
- Funções de Utilidade

Regras evento-ação são regras simples da forma “se então”, o problema com essa forma, é que o número de regras pode crescer além da compreensão dos administradores e causar conflitos entre as regras que são difíceis de detectar, conflitos estes que podem compreender múltiplas camadas com diversos agentes.

Política de Objetivo é uma forma de representação onde os possíveis estados do sistema desejáveis são indicados, e o sistema deve fazer ajustes para que estes sejam obtidos. Um problema desse modo aparece quando o sistema não consegue obter um dos estados desejáveis e não sabe avaliar qual dos outros estados é preferível.

Funções de utilidade são funções que atribuem para cada possível estado uma pontuação de qualidade, dessa forma o sistema consegue sempre comparar estados. Funções de utilidade podem ser extremamente difíceis de definir.

4.6 Classificação

A IBM propôs uma divisão em níveis de adoção ao modelo de computação autonômica que dividia os sistemas nos seguintes níveis:

- Básico, nesse nível os sistemas são acompanhados por equipes altamente treinadas que utilizam ferramentas para monitorar e realizar mudanças manualmente.
- Gerenciado, nesse nível ferramentas de monitoramento do próprio sistema agregam informações de forma inteligente diminuindo assim a quantidade de informações que devem ser observadas pela equipe.
- Preditiva, nesse nível o sistema é dotado de inteligência suficiente para sugerir ações baseadas em seu próprio monitoramento, descobrindo seus padrões de funcionamento. Estas ações são realizadas pela equipe.
- Adaptativo, nesse nível o sistema é capaz de executar mudanças em resposta a comportamentos detectados, a necessidade de intervenção humana é minimizada. É esperado que o sistema se ajuste de forma a atender o nível de serviço desejado.
- Completamente autonômico, nesse nível o funcionamento do sistema é totalmente auto ajustado para cumprir regras de negócio e políticas.

Essa classificação é muito focada em sistemas tradicionais (MARKUS, et al., 2008), não ajuda a classificar os trabalhos na área, outra melhor para este fim leva em consideração

os elementos autônomos que são o foco do trabalho e o nível arquitetural que estes foram aplicados. Essa os classifica em:

- Suporte, estes trabalhos têm foco em um único aspecto ou componente da arquitetura para auxiliar no desempenho do sistema.
- Núcleo: O elemento autônomo é o núcleo da aplicação, sendo assim seu item principal se ajusta a variações do ambiente, um exemplo é uma aplicação de transmissão de vídeo que ajusta a qualidade em resposta a velocidade de transmissão. Esse nível não leva em consideração objetivos definidos em alto nível ou regras de negócio.
- Autônomo: Trabalhos que utilizam técnicas emergentes de inteligência e agentes, A solução é completamente autonômica, atuando em um ambiente hostil que pode causar falhas, a aplicação desempenha múltiplas tarefas. Um exemplo é o *Curiosity Rover*, um robô que explora a superfície de marte.
- Autonômico: Nesse nível o trabalho tem como foco a arquitetura completa, descrevendo o trabalho em sua forma arquitetônica, com interesse em objetivos em alto nível, regras de negócio, acordos de nível de serviço. O sistema percebe sua própria performance e se adapta.

Existem argumentos para uma outra classificação, chamada de loop fechado, que inclui sistemas que são autonômicos, e que possibilitem que a inteligência amplie e refine o autogerenciamento. Entretanto não existem muitos trabalhos que se enquadrem nessa área.

4.7 Estimando Desempenho

Em um sistema autonômico é necessário que o sistema estime a qualidade de seu funcionamento, esse precisa ser capaz de avaliar se deve ou não realizar modificações em seu comportamento.

Como o sistema proposto atuará em um ambiente não supervisionado, ele não terá conhecimento sobre o pertencimento ou não de um documento ao tópico. Existem métodos para estimar a precisão de um classificador, entretanto a tarefa se torna mais difícil quando os erros não são independentes, ou seja, o fato de um classificador errar não afeta a probabilidade de acerto de outro, situação que não ocorre na prática (PLATANIOS, et al, 2014). Para este problema foi utilizada a abordagem descrita em (PLATANIOS, et al., 2014), entretanto essa abordagem não obteve sucesso neste trabalho, a hipótese para este problema consiste na suposição de que os erros dos classificadores em uso neste trabalho são muito dependentes, uma vez que utilizam os mesmos atributos para classificação, o que não acontece no trabalho original. Consideremos as Instâncias de entrada **X**, e os respectivos **R**ótulos **Y.**

Consideramos as instâncias de entrada como uma distribuição D genérica.

$$\mathbb{P}\,(\mathbf{X})=D$$

Definimos $\mathbf{E_A}$ como o conjunto composto pelos elementos classificados erroneamente por todos os classificadores do conjunto A.

$$\mathbf{E_A} = \bigcap_{i \in A} \left[\hat{f}_i(X) \neq Y\right]$$

$$e_A = \mathbb{P}_D(E_A)$$

e_A é a probabilidade de selecionar um item pertencente a $\mathbf{E_A}$ no conjunto das instâncias.

$$a_A = \mathbb{P}_D\left(\{\hat{f}_i(X) = \hat{f}_j(X)\ ; \forall i, j \in A : i \neq j\}\right)$$

a_A é a probabilidade de concordância entre os todos classificadores do conjunto A.

Esta quantidade pode ser definida em termos das probabilidades de erros das funções em A, e também pode ser calculada de forma numérica com as classificações realizadas. Para simplificar o desenvolvimento consideraremos A composto por dois classificadores e depois enunciaremos o caso genérico. A probabilidade de duas funções F_i e F_j concordarem é igual a probabilidade de ambas errarem mais a probabilidade de nenhuma errar.

$$a_{i,j} = \mathbb{P}_D(E_{\{i\}} \cap E_{\{j\}}) + \mathbb{P}_D(\bar{E}_{\{i\}} \cap \bar{E}_{\{j\}})$$

Reescrevendo o primeiro termo na forma anterior, e utilizando a lei de De Morgan no segundo termo temos.

$$a_{i,j} = e_{\{i,j\}} + \mathbb{P}_D(\overline{E_{\{i\}} \cup E_{\{j\}}})$$

Utilizando a propriedade do evento complementar e o princípio da Inclusão-Exclusão.

$$a_{i,j} = e_{\{i,j\}} + (1 - ((e_{\{i\}} + e_{\{j\}}) - e_{\{i,j\}}))$$

$$a_{i,j} = 1 - e_{\{i\}} - e_{\{j\}} + 2e_{\{i,j\}}$$

Equação 6 Taxa de concordância para 2 classificadores

Para o caso genérico temos:

$$a_A = p_D(\bigcap_{i \in A} E_i) + p_D(\bigcap_{i \in A} \bar{E}_i)$$

E novamente pela lei de De Morgan e princípio da inclusão e exclusão temos:

$$a_A = e_A + 1 + \sum_{k=1}^{|A|} [(-1)^k \sum_{\substack{I \subseteq A \\ |I|=k}} e_I]$$

Equação 7 Taxa de concordância entre conjunto A de classificadores

Calculando a taxa de concordância para cada um dos possíveis subconjuntos, teremos um total de $2^N - N - 1$ equações com $2^N - 1$ variáveis. Formando assim um sistema indeterminado, e por esse motivo o problema é formulado como um problema de otimização. Como funções para serem minimizada foram utilizadas:

- O módulo do vetor composto pelos erros, essa função levaria a estimativa mais otimista para o desempenho dos classificadores.
- O simétrico do módulo do vetor composto pelos erros, essa função levaria a estimativa mais pessimista para o desempenho dos classificadores.

Enunciado dessa forma o problema é tratado como problema de otimização quadrática, que pode ser resolvido computacionalmente. As estimativas resultantes desse método aplicado ao problema de classificação de tópicos deste trabalho, utilizando como classificadores SVM, SLDA e Naive Bayes não se aproximaram dos valores corretos.

5 Solução Proposta

5.1 Introdução

O entrelaçamento de aspectos das áreas citadas anteriormente foi necessário devido as particularidades do problema. A sua diferença para um sistema tradicional de Rastreamento e Detecção de Tópicos (*TDT, do Inglês Topic Detection and Tracking*) é que ao contrário desse o sistema desenvolvido acessa a informação através de consultas. Essa diferença é o que o aproxima de um rastreador da *Deep Web*, que é um sistema que faz consultas para obter todo o conteúdo de um determinado repositório de informações, entretanto existe uma diferença, como o objeto de interesse é um tópico, e não todo o conteúdo, é necessário projetar uma forma de obter o maior número de itens do tópico e menor número de itens que não pertencem a este.

5.2 Por que Consultas?

Existe outra diferença fundamental entre o sistema desenvolvido e os sistemas encontrados na literatura, inclusive na tarefa do *TDT* equivalente, todos os trabalhos encontrados para monitoramento de tópicos consideram disponível todo o fluxo de dados de forma a efetuar operações no conjunto com todos os documentos, essa não é a abordagem deste trabalho.

O fluxo completo quando disponível, é muito custoso computacionalmente, financeiramente, ou ambos, por esse motivo o propósito desse sistema é preencher essa lacuna, visando diminuir os custos e na maioria dos casos possibilitar o monitoramento.

Alguns Exemplos de Fluxo e suas respectivas disponibilidades são mencionados na tabela 2.

Objetos	Disponibilidade Direta
Páginas de Internet	Parcial
Tweets	Sim (Financeiramente Custosa)
Facebook	Não

Tabela 2 Disponibilidade total de documentos

Para acompanhar um tópico considerando todo o universo das páginas de internet é necessário construir um sistema de proporções computacionais extraordinárias, que já existe, porém não para este único fim.

Podemos restringir o universo de páginas a um universo tratável dentro das limitações financeiras desejadas. O sistema proposto neste trabalho pode ser utilizado como uma alternativa que consegue utilizar os buscadores existentes para explorar todo o universo já obtido por esses. A disponibilidade das páginas da internet foi caracterizada como parcial pois mesmo com todas as páginas que podem ser obtidas seguindo hiperlinks, ainda existe uma porção gigantesca na *deep web*, como é discutido no capítulo 2. Para acompanhar tópicos em redes sociais somos submetidos as particularidades de cada uma delas, o *Twitter* por exemplo disponibiliza todo o fluxo de tweets com um custo financeiro, e também disponibiliza outras formas restritas e gratuitas.

Já o *Facebook* não disponibiliza nenhuma amostra do fluxo, a forma de coletar informação não ligadas a um determinado usuário ou página é através de consultas, um exemplo desta utilização é CVIJIKJ, et al., 2011 que utilizou 26 consultas de a a z para obter uma amostra do fluxo, o que reafirma a necessidade do sistema proposto neste trabalho.

5.3 Solução Adotada

A solução proposta por esse trabalho utiliza o arcabouço de sistemas autonômicos, utilizando as propriedades autonômicas definidas como alicerce para a construção de um sistema de monitoramento de tópicos, projetando um sistema robusto e preparado para o monitoramento contínuo de longo prazo, tal arcabouço enriquece a organização do sistema e separa as etapas em módulos, que podem ser modificados e avaliados isoladamente.

5.4 Propriedades autonômicas

O principal objetivo do sistema é obter os documentos relevantes, fazendo consultas. Todas as propriedades autonômicas foram moldadas de forma a fazer modificações com este objetivo em foco. Somente a propriedade de autoconfiguração foi implementada na prova de conceito.

5.4.1 Autoconfiguração

Existem múltiplos parâmetros em um sistema que faz consultas e podem ser administrados de forma autonômica, alguns exemplos são:

- Tempo entre consultas, que não precisa ser constante.
- A consulta.
- A quantidade de documentos retornados por consulta, que não precisa ser constante e que pode ser necessária se o número de total de documentos por intervalor for limitado pelo repositório e o mesmo permitir esse ajuste.

- Se o resultado é retornado em páginas, podemos ajustar o número de resultados por página, se o repositório permitir, e também pode-se desistir de continuar recuperando as páginas, se o resultado não for o esperado. Na prova de conceito enfrentamos um caso onde a quantidade de resultados tem uma cota por intervalo e o resultado é dividido em páginas.
- A consulta por sua vez pode conter cláusulas AND e OR intercaladas, e demais ajustes de acordo com a forma aceita pelo repositório.

5.4.2 Autocura

A autocura no sistema proposto pode se referir a reparar erros na consulta, ou erros nos documentos que foram atribuídos as classes de tópico e não tópico.

Reparos na consulta podem ser necessários por inserção de termos errados, ou na ausência de termos que seriam necessários para melhor cobertura do tópico.

Recuperar os documentos perdidos por erros no sistema.

5.4.3 Auto-otimização

A tarefa de otimização está ligada a uma função de custo, e para melhorar essa meios podem ser definidos, alguns exemplos são:

Distribuição do processamento, se a função custo for relacionada ao tempo de obtenção dos documentos.

Monitorar o fluxo por movimentos incomuns, sugerindo tópicos a serem seguidos, indicando possíveis relações entre os seguidos pelo sistema, se a função de custo for relacionada com a quantidade de documentos perdidos.

5.4.4 Autoproteção

A proteção ao sistema pode ser interpretada como proteção a desvios maliciosos de tópico, causados por entidades como os robôs sociais, que são algoritmos que produzem conteúdo e até interagem com humanos, para causar alterações na percepção geral sobre determinado tópico (FERRARA, *et al.*, 2016).

5.5 Avaliação de Desempenho

A definição de tópico possui duas vertentes, tópico pode ser definido amplamente como informações que estão relacionadas entre si, modo utilizado em Busca e Recuperação, ou de forma mais restrita, como em *TDT*, onde são considerados do mesmo tópico informações sobre o mesmo evento no mundo real. A primeira necessita de julgamento humano mais opinativo, a segunda é mais rígida e não admite muita divergência de julgamento. Na prova de conceito deste trabalho utilizamos para avaliar as aproximações a forma mais restrita, enquanto essa certamente confere uma avaliação de grau mais baixo ao desempenho do classificador, é intuitivo que se esta avaliação mostrar-se satisfatória o método também seria satisfatório pela a outra, que possui um julgamento de tópico que inclui o mais restrito. Outro motivo para a escolha dessa forma é a aproximação boa que é possível sem um julgador humano, na prova de conceito usando *Hashtags* como indicador de tópico, essa aproximação é crucial para a avaliação do desempenho feita por um sistema autonômico. As *hashtags* são utilizadas pelos usuários para indicar um tópico de forma mais abrangente, entretanto nos casos escolhidos as *hashtags* tratam de um evento, um ataque terrorista nos EUA e a votação que legalizou o casamento entre pessoas do mesmo sexo também nos EUA. Os trabalhos de detecção e monitoramento de tópicos

utilizam a definição mais abrangente, então podemos inferir que este trabalho está se subavaliando em suas métricas se comparado a trabalhos clássicos de TDT.

Com o desejo de utilizar a definição mais abrangente, ocorreu a tentativa de estimar a precisão dos classificadores sem possuir o rótulo correto, tal estimativa permitiria o funcionamento de um sistema autonômico, que precisa se auto avaliar durante o funcionamento. Para tal tentativa três classificadores foram utilizados em conjunto para permitir a auto avaliação conforme o capítulo 4, O método desenvolvido permite estimar a precisão de cada classificador mesmo se estes não possuírem erros independentes, situação que ocorre na prática. Entretanto por utilizarem os mesmos atributos, é possível que a dependência dos erros foi grande o suficiente de forma a impossibilitar que o método funcione. Nenhuma das formulações da função objetivo foi capaz de dar estimativas consideráveis.

5.5.1 Consulta

A estratégia de configuração da consulta no sistema tem como objetivo desenvolver um caminho mais equilibrado considerando as duas possíveis abordagens triviais. A não evolução é uma das formas triviais de configuração, a mesma consulta mantida ao longo de todo o funcionamento, essa abordagem será mais precisa, entretanto essa abordagem não será capaz de recuperar boa parte dos documentos, pois não acompanhará nenhum termo que se torne relevante ao longo do tempo. Na prova de conceito que utiliza dados do *Twitter* essa abordagem mantém a *hashtag* original durante todo o tempo, como este é o indicador de tópico no experimento essa abordagem terá precisão total, entretanto isso não ocorre ao utilizar o sistema com outros dados.

A segunda opção trivial é incluir todos os termos que aparecem nos documentos recuperados. Essa abordagem é inviável, pois é esperado que após poucas iterações uma

fração bem grande de todos os documentos seja recuperada, fugindo do tópico. Entretanto no Twitter podemos utilizar de um critério ainda trivial, porém mais sutil, a inclusão de todas as *hashtags*.

A alternativa proposta nesse trabalho é utilizar regras para escolher os termos que serão adicionados, porém só aplicar essas regras nos documentos considerados relevantes ao tópico, essa separação é feita com um classificador. Os novos documentos obtidos por essa nova consulta são também classificados, e passam a integrar esse banco. Estratégia semelhante a RUNGSAWANG e ANGKAWATTANAWIT, 2005. A arquitetura proposta é descrita na figura 2.

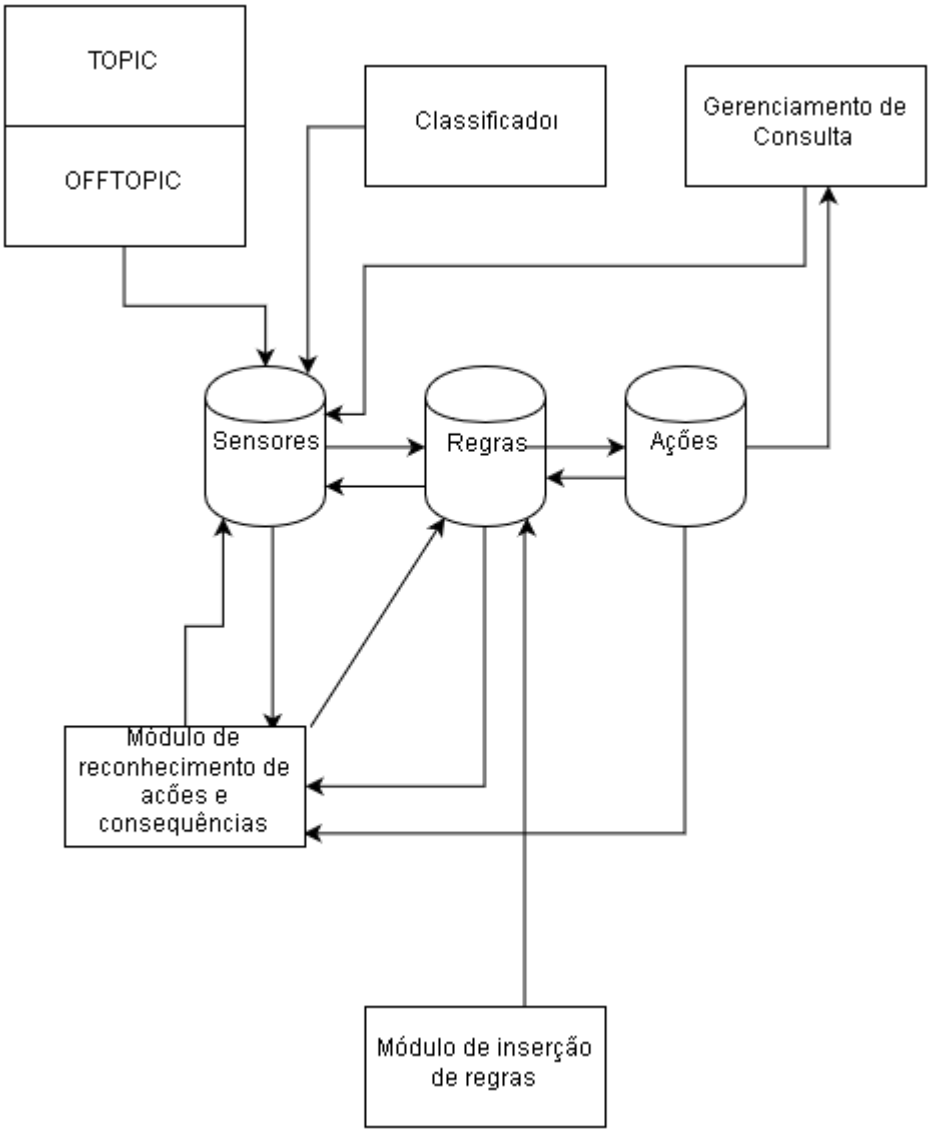

Figura 2 Arquitetura Auto-Configuração Proposta

A primeira regra implementada, é baseada no modelo descrito no capítulo 2.

O objetivo em cada consulta no sistema proposto é obter o maior número de documentos relevantes a cada consulta, podemos utilizar o AVG_{local} como aproximação do grafo AVG oculto, sendo assim desprezando a relação de dependência entre os termos, podemos estimar que o termo que tem maior frequência no AVG_{local} possuirá a maior estimativa do total de documentos a retornar. Para essa estimativa usa-se o fato de que a distribuição dos termos na linguagem segue uma distribuição de potência (MANNING e SCHÜTZE, 1999). É importante lembrar que essa estratégia não usa a *hashtag* inicial como termo de pesquisa.

O diagrama de sequência a seguir na figura 3 mostra o início da abordagem do sistema.

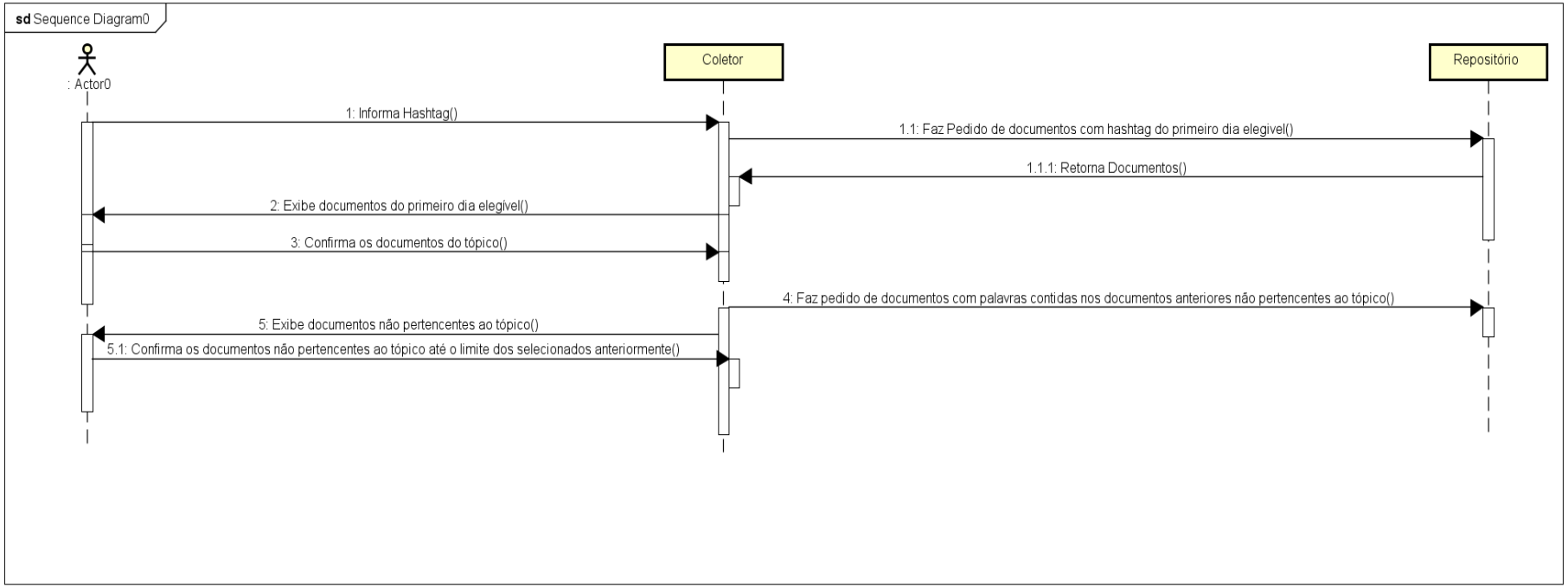

Figura 3 Diagrama de sequência

Após esse início o sistema passa a repetir iterativamente a etapa busca, seguida da classificação, todas as outras propriedades autonômicas acontecem nesse loop, embora não implementadas nessa prova de conceito. O diagrama a seguir, na figura 4, detalha essa repetição.

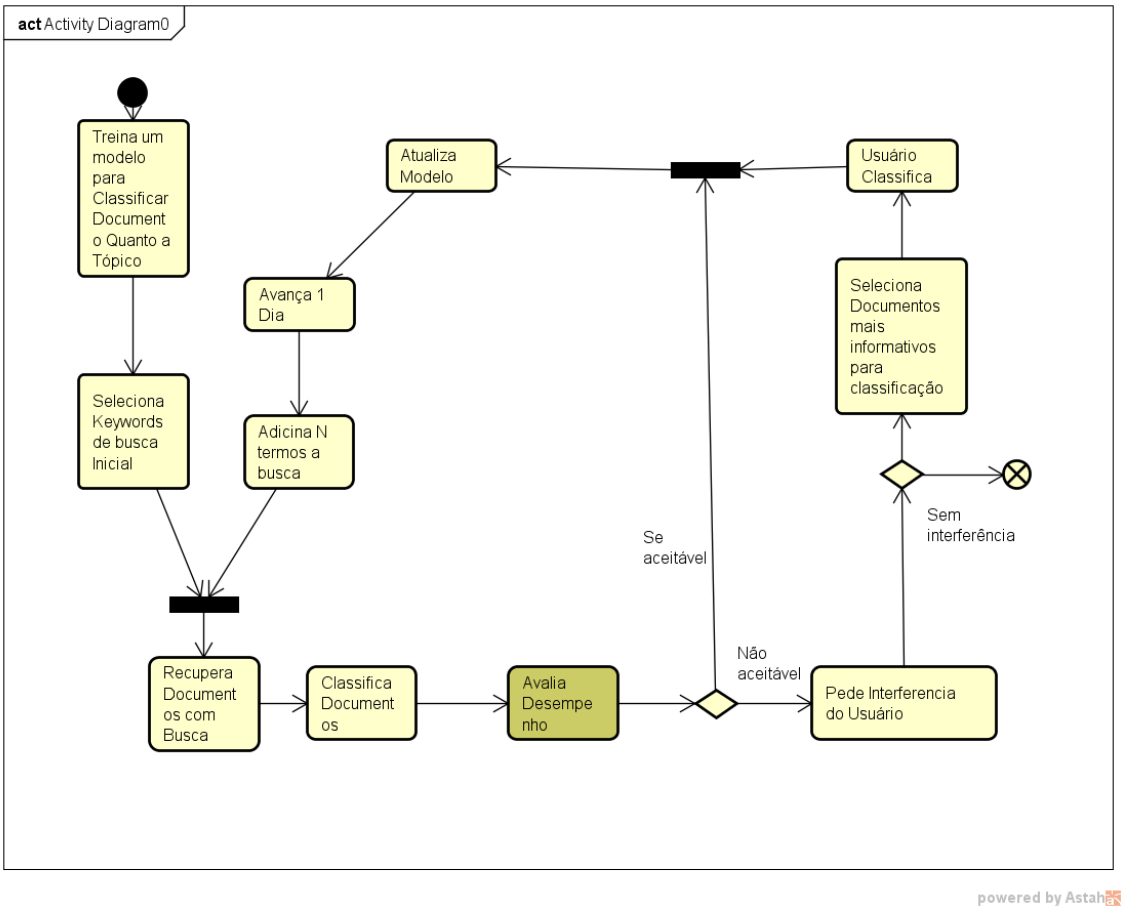

Figura 4 Diagrama de atividade

É previsto nesse diagrama o encerramento do sistema caso a qualidade dos resultados torne-se ruim e não seja possível a intervenção do usuário.

6 Experimento

O experimento tem como objetivo avaliar a qualidade da tarefa final, que é a recuperação de documentos e a qualidade da tarefa auxiliar, que é a classificação de documentos pertencentes ao tópico ou não. A consulta inicial possui os dez termos mais frequentes no conjunto de documentos selecionados pelo usuário retirando as *stop-words*, e a cada iteração é adicionado mais um termo. O refinamento dessas constantes não foi realizado, foram utilizados dois exemplos com os mesmos parâmetros para mostrar como esses podem ser modificados para melhorar o desempenho em cada caso.

O termo da *hashtag* não é incluído na consulta da abordagem proposta, para que se possa comparar a cobertura sobre o tópico atingida pelos termos escolhidos, pois a *hashtag* é o indicador de tópico utilizado e incluí-lo inviabilizaria essa comparação. O classificador utilizado foi somente o *naive bayes*, pois se mostrou mais robusto a documentos erroneamente classificados sendo utilizados como treino. O experimento também não leva em consideração a interação com o usuário para classificar exemplos, pois a mesma não teve resultados positivos, ao forçar o classificador a separar documentos sobre o tópico, mas que não possuem a *hashtag* o mesmo piorou sua eficácia.

6.1 Base de dados

A base de dados utilizada no o trabalho foi coletada utilizando a *API* do *Twitter* que disponibiliza 1% dos fluxo total do mesmo durante os dias 12 de junho e 7 de julho de 2015. Em Seguida foram selecionados dois assuntos que tiveram maior repercussão durante esse período de tempo.

Os eventos escolhidos são representados pelas *hashtags* “#charleston” e “#lovewins” e são referentes a:

- O assassinato de nove pessoas em uma igreja no estado de Carolina do Sul, nos Estados Unidos. O crime teve motivação racial, todos os mortos eram negros.
- A legalização do casamento homossexual em todos os 50 estados dos EUA pela corte suprema.

6.2 Resultados e Discussão

A primeira comparação é sobre a capacidade de recuperação da estratégia de geração de consulta proposta, frente as duas opções triviais, no exemplo do ataque em Charleston. O resultado esperado é que a estratégia proposta tenha um número de documentos recuperados entre as duas abordagens triviais, indicando uma cobertura maior do que a consulta somente pela *hashtag* e uma precisão melhor do que a estratégia de expansão sem critério. Essa comparação é visualizada nas figuras 5 e 6, a primeira em números absolutos e a segunda em escala logarítmica, as barras representam a quantidade de documentos recuperados em cada dia. Como era esperado, a estratégia de adicionar todas as *hashtags* encontradas resulta rapidamente na recuperação de quase todos os tweets disponíveis, enquanto a estratégia de não expansão mantém previsivelmente poucos resultados, que não são visíveis em escala absoluta.

Figura 5 Comparação Recuperação Valores Absolutos

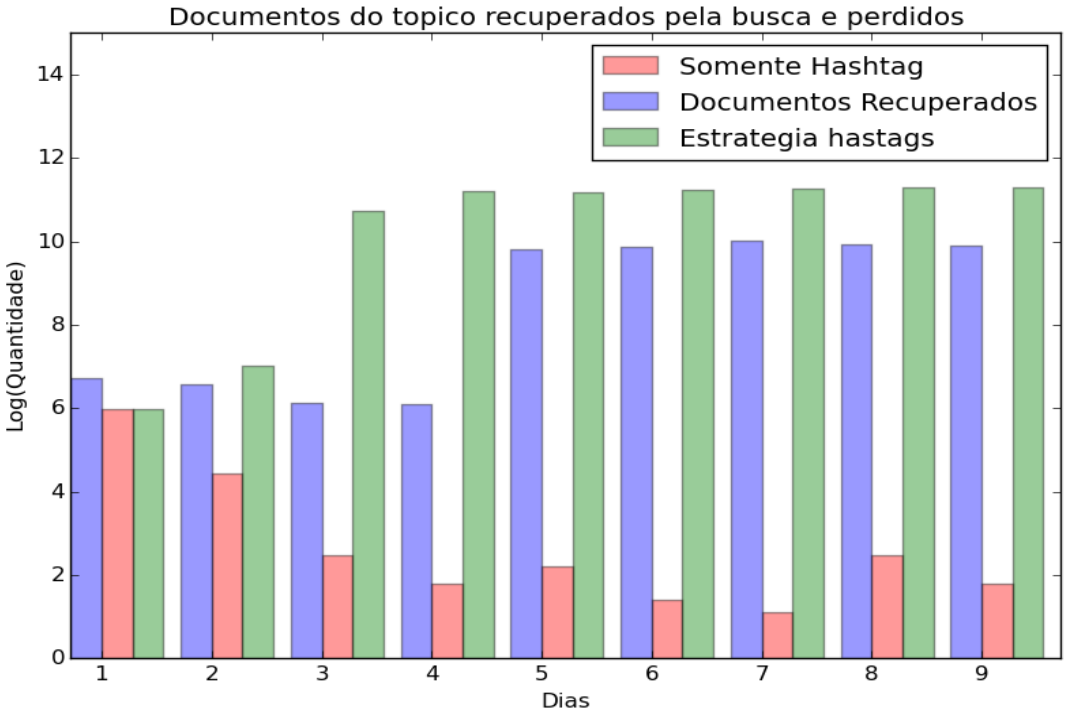

Figura 6 Comparação Recuperação Escala Logarítmica

A estratégia proposta apresentou um resultado condizente com o esperado, conseguiu recuperar uma quantidade de documentos intermediária entre os dois modos triviais e apresentou um quantidade estável ao longo dos dias.

A limitação imposta ao sistema autonômico de utilizar somente a *hashtag* como forma de se auto avaliar impõe uma forma de avaliação da classificação através do mesmo critério. Na figura 7 e na figura 8 é exibido o desempenho do classificador em relação aos documentos que foram recuperados utilizando os dados do exemplo do ataque a igreja.

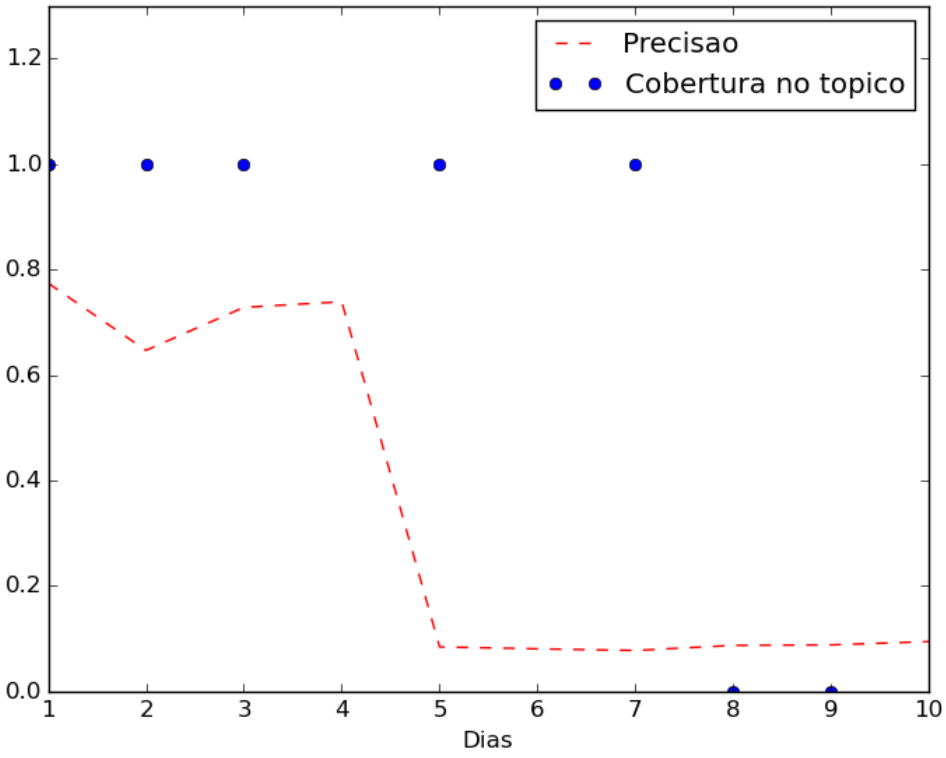

Figura 7- Avaliação da classificação 1º exemplo

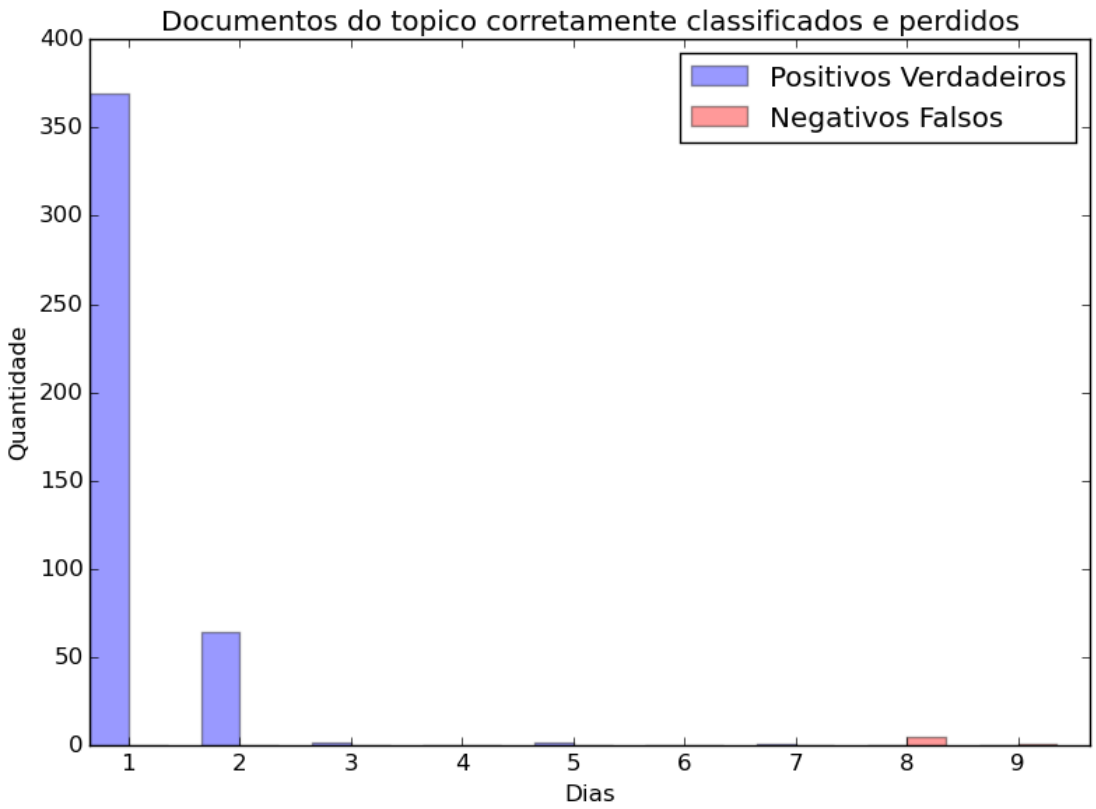

Figura 8 Avaliação da Classificação 2 1º Exemplo

Esses dois gráficos permitem algumas observações. Enquanto a quantidade de documentos recuperados do tópico é alta, nos dois primeiros dias, a cobertura do classificador é máxima, o que mostra que o mesmo não classificou como não relevante nenhum documento relevante. A precisão do classificador cai muito com o passar do tempo, isso pode ser explicado por 2 fatores. O primeiro diz respeito a rigidez do critério utilizado para indicar se o documento pertence ao tópico, são muitos os documentos que pertencem ao tópico, mas não possuem a *hashtag* apropriada. O segundo diz respeito ao volume baixo de documentos do tópico, que faz somente os erros de classificação serem adicionados à base de conhecimento como tópico, criando um efeito cascata para as próximas iterações.

Na figura 9 é possível avaliar o resultado final da recuperação de documentos, exibindo a quantidade de documentos recuperados e sua relação com o total.

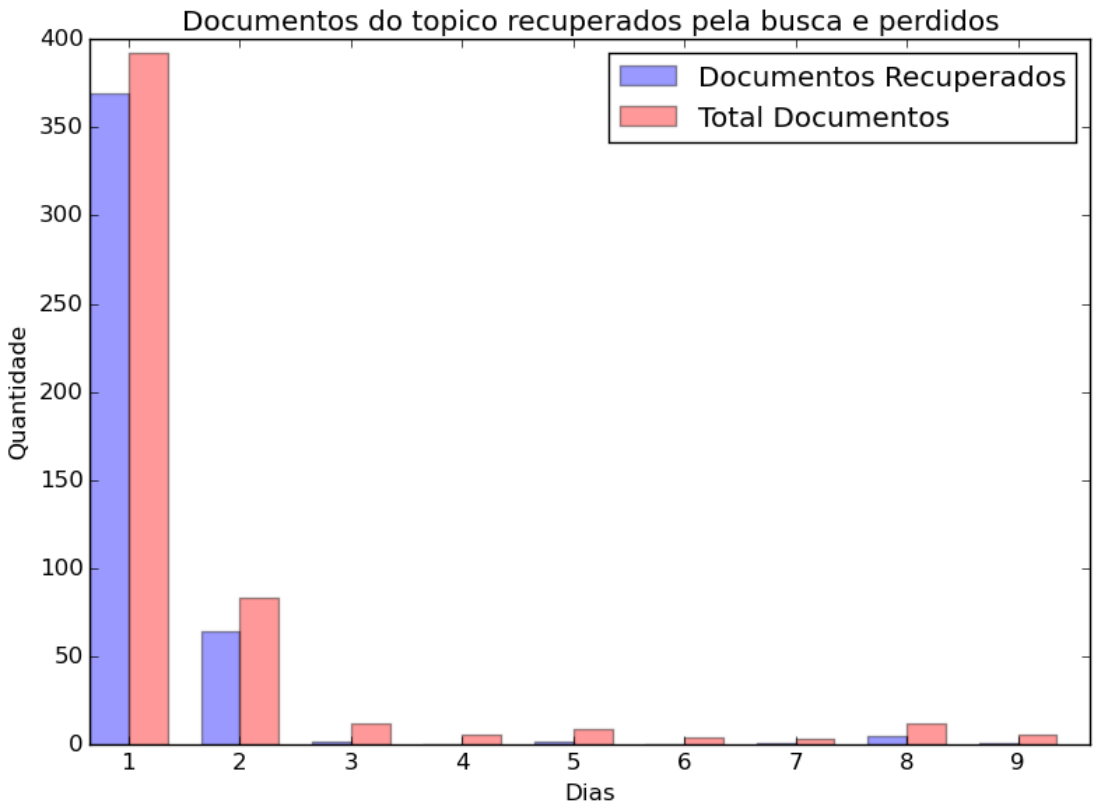

Figura 9- Avaliação De Recuperação 1º Exemplo

Esse gráfico mostra que a recuperação ainda não tem um desempenho adequado com o passar dos dias e um volume mais baixo embora tenha conseguido uma cobertura alta nos primeiros dias. Esse resultado remete a hipótese de que a evolução da consulta ainda não consegue acompanhar as mudanças do tópico. Ao aumentar o número de termos adicionados por iteração a situação não melhorou, pois o sistema adicionou termos de documentos erroneamente classificados como tópico e isto piorou seu desempenho.

Avaliaremos o exemplo sobre a legalização do casamento homossexual. Nas figuras 10 e 11, podemos notar que o sistema manteve precisão e cobertura em bons níveis mesmo após queda significativa no volume de documentos no tópico, o que pode ser explicado pela maior quantidade de documentos de treino, em relação ao exemplo anterior, nesse exemplo foram 754 contra apenas 32 no exemplo anterior. Esse número é determinado com base nos critérios para início do sistema.

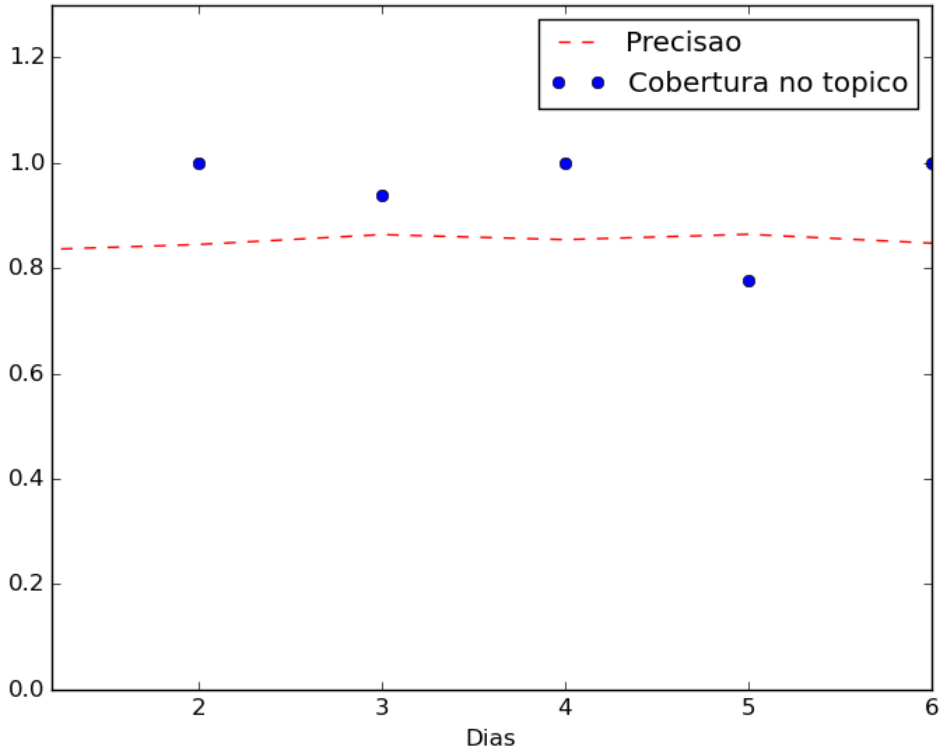

Figura 10 - Avaliação da classificação 2º exemplo

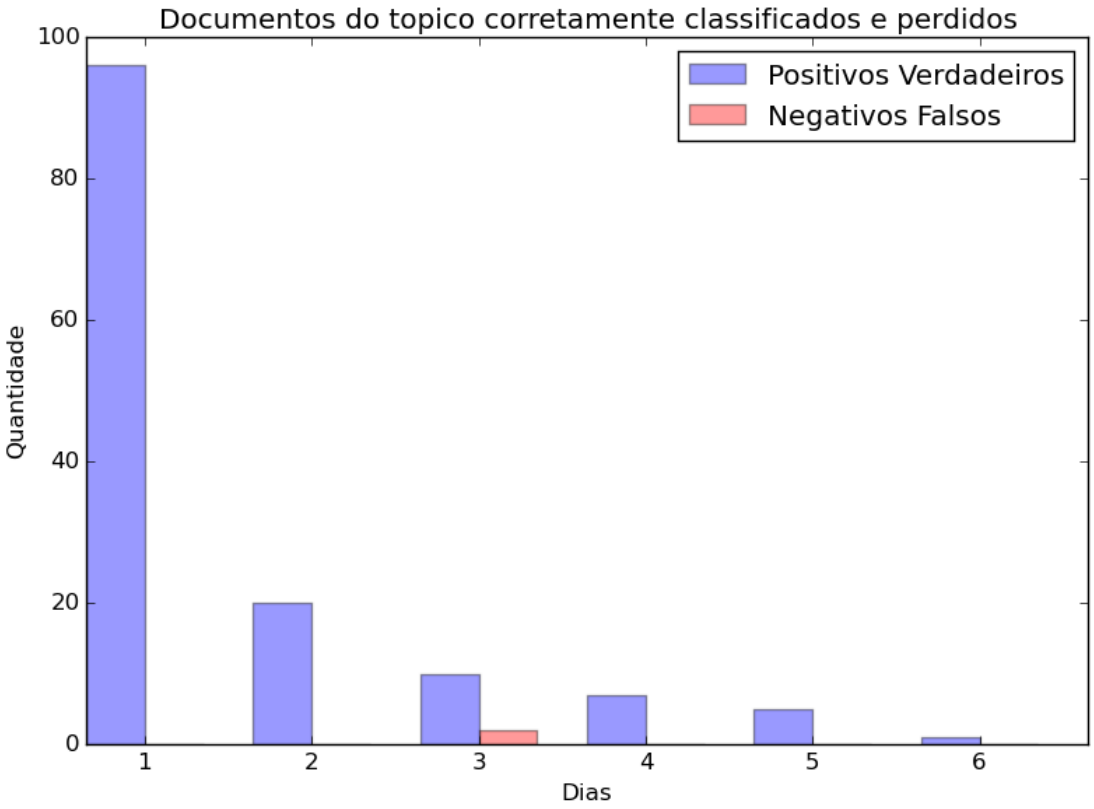

Figura 11 - Avaliação da classificação 2 2º exemplo

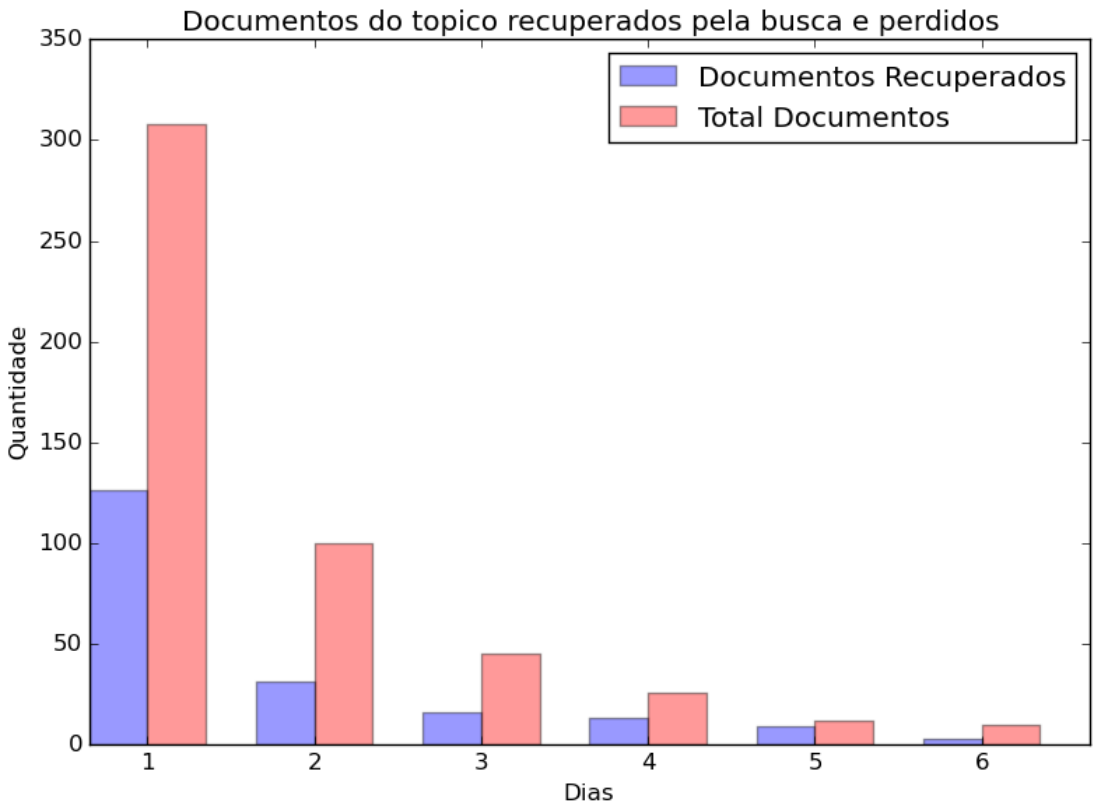

Figura 12 - Avaliação De Recuperação 2º Exemplo

Na figura 12 notamos que a cobertura nos primeiros dias não foi suficiente, sugerindo a necessidade de um aumento no número de termos iniciais, entretanto a partir do 4 dia a cobertura já apresenta uma melhora. Essa melhora é evidenciada na figura 13, onde é exibida a quantidade de documentos recuperados pelo termos adicionados.

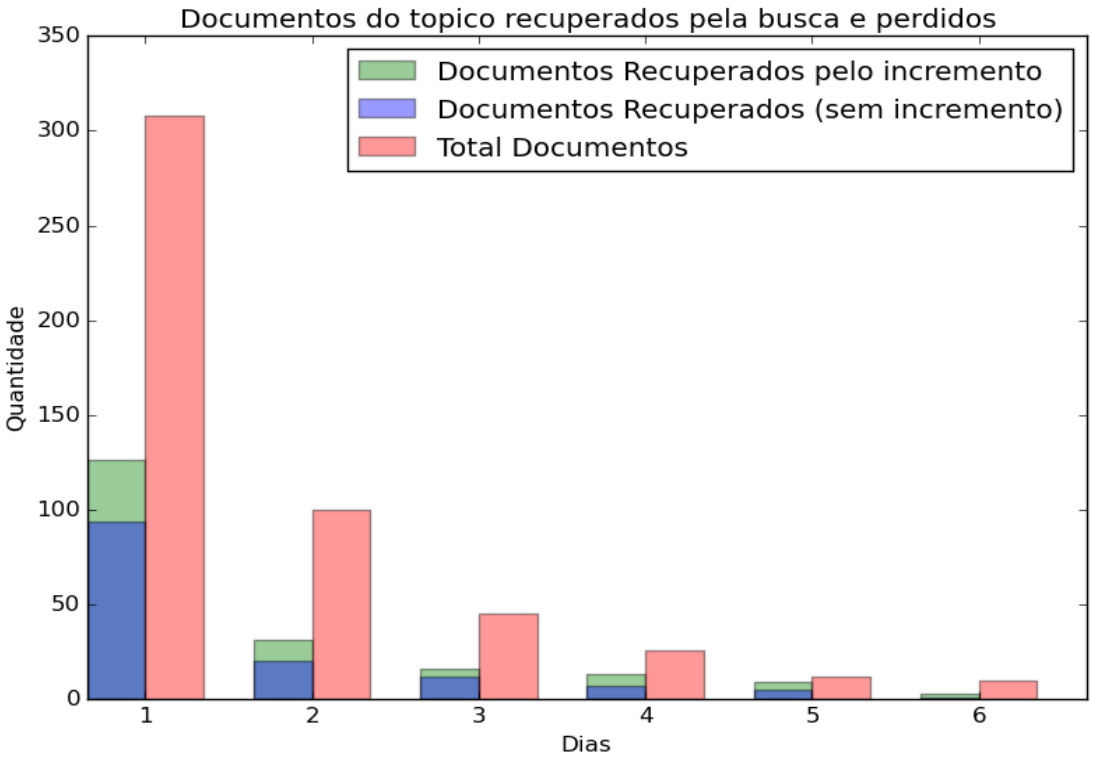

Figura 13 - Recuperação com e sem incremento

Na figura 13 é exibido o resultado anterior com a consulta inicial incrementada para 20 termos.

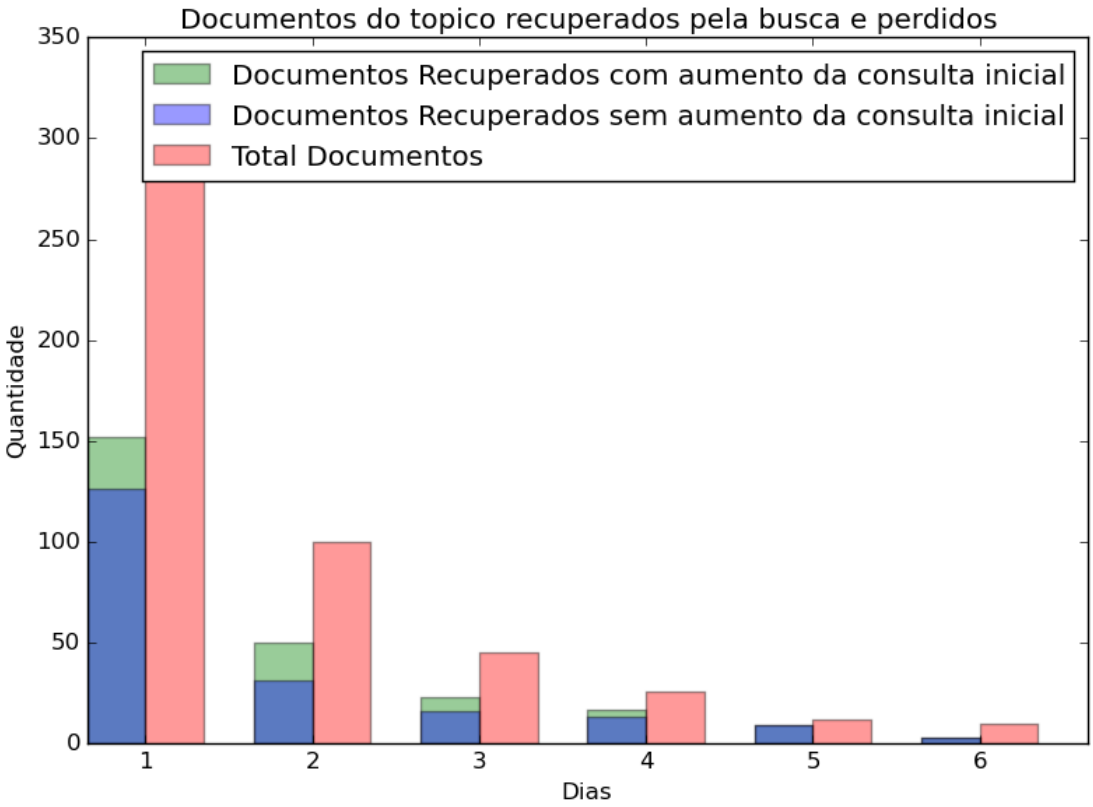

Figura 14 - Avaliação Recuperação com Consulta Inicial Aumentada

O aumento do número de termos não obteve o resultado esperado, mostrando que o vocabulário desse exemplo é extenso, dificultando o problema.

A autoconfiguração proposta é uma alternativa viável frente as abordagens triviais, ambas com seus problemas, a primeira incapaz de generalizar, e que só obtém resultados aceitáveis no ambiente do *Twitter* e a segunda que não é capaz de direcionar a busca. A abordagem utilizada é complementar a abordagem de manter a consulta original, essa só foi separada para comparação, se retirarmos a limitação de não usar a *hashtag* na autoconfiguração temos cobertura máxima nos exemplos e precisão maior do que a segunda abordagem trivial.

Muitos ajustes são possíveis e necessários, principalmente para as escolhas da quantidade de termos iniciais e incremento de termos, pois os resultados finais do sistema dependem muito da escolha desses, além disso cada tópico tem suas particularidades e demanda ajustes diferentes para obter o melhor resultado.

7 Conclusão e Trabalhos Futuros

Não foram encontrados trabalhos que tratem a tarefa de gerar consultas que recuperem documentos de tópicos. A motivação desse problema é exemplificada neste trabalho e também surge em outros trabalhos que utilizam a consulta por somente uma palavra para obter documentos de um tópico. Para tratar esse problema foi necessária uma mistura de duas áreas de pesquisa, *TDT* e *Deep Web Crawling*, pois o modo como o problema é definido não se enquadra completamente em nenhuma das áreas anteriores.

Os tópicos textuais evoluem e existem estudos sobre como projetar modelos textuais que se adaptem a essa evolução (GOHR, *et al.*, 2009), inclusive com foco em redes sociais (ANKAN e SINDHWANI, 2012), existe também uma área de pesquisa em computação

gráfica que estuda a melhor forma de visualização da interação entre tópicos e evolução destes, muito bem exemplificada em CUI, *et al.*, 2011. A evolução é tratada no trabalho de forma iterativa, avaliando o conhecimento obtido sobre o tópico e incrementando a busca pelo mesmo em uma direção, essa direção pode levar a erros e por isso foi adotado o arcabouço de sistemas autonômicos para enumerar as propriedades desejáveis para contornar os problemas causados pelo desconhecimento de uma avalição perfeita sobre o tópico. O objetivo de um sistema autonômico é se adaptar a mudanças em seu ambiente para executar suas tarefas da melhor forma possível, esse objetivo é semelhante ao desejado para uma consulta, que deve se adaptar a mudanças no tópico que deve representar. Nesse arcabouço a geração da consulta é vista como um problema de autoconfiguração, essa propriedade ainda precisa abordar outros fatores não tratados nesse trabalho. O experimento mostrou que ainda existe um longo caminho a percorrer, mas a abordagem adotada tem um desempenho que a torna uma opção com pontos positivos em relação as possíveis abordagens triviais.

A estratégia de auto avaliação abordada no capítulo 4 deve ser retomada para uma melhor avaliação dos resultados e consequentemente da configuração da consulta, para isso é necessário que outros atributos sobre o texto sejam utilizados.

A modelagem dos termos no grafo AVG também permite que critérios de detecção de tópico em grafos sejam utilizados para selecionar melhor os termos que devem ser utilizados (SAYYADI e L., 2013).

A implementação das outras propriedades autonômicas enumeradas é essencial para o funcionamento por tempo prolongado do sistema sem a intervenção humana constante.

8 Bibliografia

ALLAN, J. **Topic Detection And Tracking**. [S.l.]: Springer, 2002.

ALLAN, J. et al. **Inquery and TREC-7**. The Seventh Text Retrieval Conference (TREC-7). Maryland, USA: [s.n.]. 1998. p. 201-216.

ALLAN, J. et al. **Topic Detection and Tracking pilot study final report**. Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. Lansdowne, VA: [s.n.]. 1998. p. 194-218.

ALLAN, J. et al. **Detections, Bounds, and Timelines:** UMass and TDT-3. Proceedings of Topic Detection and Tracking Workshop. [S.l.]: [s.n.]. 2000. p. 167-174.

ALLAN, J.; LAVRENKO, V.; JIM, H. J. **First Story Detection in TDT is hard**. CIKM '00 Proceedings of the ninth international conference on Information and knowledge management. New York, NY, USA: [s.n.]. 2000. p. 374-381.

ALVAREZ, M. et al. **DeepBot:** A Focused Crawler for Acessing Hidden Web Content. Proceedings of DEECS2007. San Diego, CA: [s.n.]. 2007. p. 18-25.

AMER-YAHIA, S. et al. **MAQSA:** a system for social analytics on news. In proceeding of the 2012 ACM SIGMOD International Conference on Management of Data. Scottsdale, AZ, USA: ACM, New York, NY. 2012. p. 653-656.

ANKAN, S.; SINDHWANI, V. **Learning evolving and emerging topics in social media:** a dynamic nmf approach with temporal regularization. Proceedings of the fifth ACM international conference on Web search and data mining. Seattle, Washington, USA: ACM. 2012. p. 693-702.

ATEFEH, F.; KHREICH, W. A Survey of Techniques for Event Detection in Twitter. **Journal Computational Intelligence**, 1, fev. 2015. 132-164.

BAEZA-YATES, R.; BERTHIER, R.-N. **Modern Information Retrieval**. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1999.

BARBOSA, L.; FREIRE, J. **Siphoning Hidden-Web Data through Keyword-Based Interfaces**. Proceeding of SBBD2004. Brasilia, Brasil: [s.n.]. 2004. p. 309-321.

BARBOSA, L.; FREIRE, J. **Searching for Hidden Web Databases**. Proceedings of WEBDB2005. Baltimore MD: [s.n.]. 2005. p. 1-6.

BECKER, H. et al. **Identifying content for planned events across social media sites**. WSDM '12 Proceedings of the fifth ACM international conference on Web search and data mining. ACM New York, NY, USA: [s.n.]. 12. p. 533-542.

BEEFERMAN, D.; BERGER, A.; LAFFERTY, J. Statistical Models for Text Segmentation. **Journal Machine Learning - Special issue on natural language learning**, n. 1-3, fev. 1999. 177-210.

BELKIN, N.; CROFT, W. Retrieval Techniques. In: CUADRA, C. A. **Annual Review of Information Science**. [S.l.]: Elsevier Science Inc, v. 22, 1987. p. 109-146.

BELKIN, N.; CROFT, W. Information filtering and information retrieval: two sides of the same coin? **Communications of the ACM - Special issue on information filtering**, v. 12, n. 35, p. 29-38, dez. 1992.

BERGMAN, M. K. The deep web: surfacing hidden value. **The Journal of Electronic Publishing 7**, 2001. 3-21.

BISHOP, C. **Pattern Recognition and Machine Learning**. [S.l.]: Springer-Verlag New York, Inc, 2006.

BLEI, D.; A., N.; JORDAN, M. Latent dirichlet allocation. **Journal of machine Learning research**, v. 3, p. 993-1022, jan. 2003.

BLEI, D.; MCAULIFFE, J. **Supervised topic models**. Advances in Neural Information Processing Systems. Vancouver, in Vancouver, B.C., Canada: Neural Information Processing Systems Foundation, 2009. 2007.

BRANTS, T.; CHEN, F.; FARAHAT, A. **A system for new event detection**. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR '03). New York, NY, USA: [s.n.]. 2003. p. 106–113.

BRANTS, T.; CHEN, F.; TSOCHANTARDIS, I. **Topic-based document segmentation with probabilistic latent semantic analysis**. CIKM '02 Proceedings of the eleventh international conference on Information and knowledge management. New York, NY, USA: [s.n.]. 2002. p. 211-218.

CATALDI, M.; DI CARO, L.; SCHIFANELLA, C. **Emerging Topic Detection on Twitter based on Temporal and Social Terms Evaluation.** MDMKDD '10 Proceedings of the Tenth International Workshop on Multimedia Data. New York, NY, USA: [s.n.]. 2010. p. (4).

CHA, A. E. http://www.washingtonpost.com/sf/national/2015/06/27/watsons-next-feat-taking-on-cancer/. **Washington Post**, 2015. Disponivel em: <http://www.washingtonpost.com/sf/national/2015/06/27/watsons-next-feat-taking-on-cancer/>. Acesso em: 30 jul. 2016.

CHANG, K. C. C.; HE, B.; ZHANG, Z. **Towards Large Scale Integration:** Building a Metaquerier over Databases on the Web. Proceedings of CIDR. Asilomar, CA, USA, 2005: [s.n.]. 2005.

CHEN, F.; FARAHAT, A.; BRANTS, T. **Multiple Similarity Measures and Source-Pair Information in Story Link Detection**. Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics. Stroudsburg, PA, USA: [s.n.]. 2004. p. 313-320.

COOK, D. M. et al. Twitter Deception and Influence: Issues of Identity, Slacktivism, and Puppetry. **Journal of Information Warfare 13.1**, 2014. 58-71.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, fev. 1995.

CUI, W. et al. Textflow: Towards better understanding of evolving topics in text. **IEEE transactions on visualization and computer graphics**, 17, n. 12, 2011. 2412-2421.

CVIJIKJ, I. P.; PLETIKOSA, I.; MICHAHELLES, F. **Monitoring trends on facebook**. IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing. Sydney, Australia: IEEE. 2011. p. 895-902.

DIAO, Y. et al. A control theory foundation for self-managing computing systems. **IEEE Journal on Selected Areas in Communications**, Piscataway, NJ, USA, 12, set. 2006. 2213-2222.

FERRARA, E. et al. The rise of social bots. **Communications of the ACM**, v. 7, n. 59, p. 96-104 , jul. 2016.

FRANZ, M. et al. **Segmentation and Detection at IBM:** Hybrid Statistical Models and Two-tiered Clustering. Procedings of TDT-3 Workshop. [S.l.]: [s.n.]. 1999.

GOHR, A. et al. **Topic Evolution in a Stream of Documents**. Proceedings of the SIAM International Conference on Data Mining. Sparks, Nevada, USA: [s.n.]. 2009. p. 859-872.

GU, H. et al. **ETree:** Effective and Efficient Event Modeling for Real-Time Online Social Media Networks. WI-IAT '11 Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology. Washington, DC, USA: [s.n.]. 2011. p. 300-307.

HANANI, U.; SHAPIRA, B.; SHOVAL, P. Information Filtering: Overview of Issues, Research and Systems. **User Modeling and User-Adapted Interaction**, 11, ago. 2001. 203-259.

HE, B. et al. Acessing the deep web: A survey. **Communications of the ACM 50**, 5, 2007. 95.101.

HE, B. et al. Acessing the deep web: A survey. **Communications of the ACM**, v. 5, n. 50, p. 95-101, 2007.

HE, H. et al. **Wise-integrator:** an Automatic Integrator of Web Search Interfaces for E-commerce. Proceedings of VLDB2003. Berlim, Germany: [s.n.]. 2003. p. 357-368.

HEARST, M. A. **Multi-paragraph segmentation of expository text**. ACL '94 Proceedings of the 32nd annual meeting on Association for Computational Linguistics. Stroudsburg, PA, USA: [s.n.]. 1994. p. 9-16.

HORN, P. **Autonomic computing: IBM's Perspective on the state of information Technology**. IBM. [S.l.]. 2001.

HUEBSCHER, M.; MCCANN, J. A survey of autonomic computing-degrees, models, and applications. **ACM Computing Surveys (CSUR)**, New York, NY, USA, p. 7º, ago. 2008.

IBM. **An Architectural Blueprint for Autonomic Computing**. IBM. [S.l.]. 2005.

INGWERSEN, P. **Information Retrieval Interaction**. [S.l.]: [s.n.], 2002.

IPEIROTICS, P. G.; GRAVANO, L. Classification-aware hidden web text database selection. **ACM Transactions on Information Systems**, v. 2, n. 26, p. 1-66, 2008.

JIANG, L. et al. **Learning Deep Web Crawling with Diverse Features.** Proceedings of IEEE/WIC/ACM Web Intelligence. Milan, Italy: [s.n.]. 2009. p. 572-575.

JO, T.; LEE, M. **The evaluation measure of text clustering for the variable number of clusters**. In Proceedings of the 4th International Symposium on Neural Networks: Part II Advances in Neural Networks, ISNN ’07. Berlin, Heidelberg,: Springer-Verlag. 2007. p. 871–879.

JOACHIMS, T. **Text categorization with support vector machines:** Learning with many relevant features". European conference on machine learning. Chemnitz, Germany: Springer, Berlin Heidelberg. 1998. p. 137-142.

KEPHART, J. O.; WALSH, W. E. **An Artificial Intelligence Perspective on Autonomic Computing Policies**. olicies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on. IEEE, 2004. [S.l.]: [s.n.]. 2004. p. 3-12.

KHAN, A. A. The role social of media and modern technology in arabs spring. **Far East Jounal of Psychology and Business**, 1, 2012. 56-63.

KHONDKER, H. H. Role of new media in the Arab Spring. Globalizations. **Globalizations**. 675-679.

KOZIMA, H. **Text segmentation based on similarity between words**. ACL '93 Proceedings of the 31st annual meeting on Association for Computational Linguistics. Stroudsburg, PA, USA: [s.n.]. 93. p. 286-288.

KUMARAN, G.; ALLAN, J. **Text Classification and named entitites for new event detection**. SIGIR '04 Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: [s.n.]. 2004. p. 297-304.

LEEK, T.; SCHWARTZ, R.; SISTA, S. Probabilistic approaches to topic detection and tracking. In: ALLAN, J. **Topic Detection and Tracking**. Norwell, MA: Kluwer Academic Publishers, 2002. p. 67-83.

LI, H.; XU, J. **Beyond Bags of Words:** Modeling Implicit User Preferences in Information Retrieval. SIGIR '12 Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. Partland, Oregon USA: ACM New York, NY, USA ©2012. 2012. p. Pages 1177-1177.

LI, Z. et al. **A probabilistic model for retrospective news event detection.** In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’05. New York, NY, USA: [s.n.]. 2005. p. 106–113.

LIU, J. et al. **Crawling deep web content through query forms**. Proceedings of WEBIST2009. Lisbon, Portugal: [s.n.]. 2009. p. 634-642.

LIU, J. et al. Deep Web adaptive crawling based on minimum executable pattern. **Journal of Intelligent Information Systems**, 2, 2011. 197-215.

LU, C.; TANG, X. Surpassing human-level face verification performance on LFW with GaussianFace. **http:** //arxiv.org/. Disponivel em: <http://arxiv.org/abs/1404.3840>.

MADHAVAN, J. et al. Web-scale Data Integration: You Can Only Afford to Pay As You Go. **In Proceedings of CIDR2007**, 2007. 342-350.

MADHAVAN, J. et al. **Google's Deep-Web Crawl**. Proceedings of VLDB2008. Auckland, New Zealand: [s.n.]. 2008. p. 1241-1252.

MADHAVEN, J. et al. **Harnessing the Deep Web:** Present and Future. Proceedings of CIDR. Asilomar, CA, USA: [s.n.]. 2009.

MANNING, C.; SCHÜTZE, H. **Foundations of Statistical Natural Language Processing**. [S.l.]: The MIT Press, 1999.

MARKUS, C. H.; MCCANN; J.A. A survey of Autonomic Computing - Degrees, Models, and Applications. **ACM Computing Surveys (CSUR)**, New York, NY, USA, 3, 2008. (7).

MISHNE, G.; RIJKE, M. **Boosting web retrieval through query operations**. ECIR'05 Proceedings of the 27th European conference on Advances in Information Retrieval Research. Santiago de Compostela, Chile: Springer-Verlag Berlin, Heidelberg. 2005. p. 502-516.

MITCHEL, T. **Machine Learning**. [S.l.]: [s.n.], 1997.

MOHD, M. **Named entity patterns across news domains**. Proceedings of the 1st BCS IRSG conference on Future Directions in Information Access, FDIA'07. Swinton, UK: [s.n.]. 2007. p. 5-5.

MORSTATTER, F. et al. **Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose**. Proceedings of ICWSM (2013). Cambridge, MA, USA: [s.n.]. 2013.

NIGAN, K. et al. **Learning to Classify Text from Labeled and Unlabeled Documents**. Fifteenth National Conference on Artificial Intelligence. Madison, Wisconsin, EUA: [s.n.]. 1998.

NTOULAS, A.; ZERFOS, P.; CHO, J. **Downloading Textual Hidden Web Content through Keyword Queries**. Proceedings of JCDL2005. Denver, USA: [s.n.]. 2005. p. 100-109.

NTOULAS, A.; ZERFOS, P.; CHO, J. **Downloading Textual Hidden Web Content through Keyword Queries**. Proceedings of JCDL2005. Rome, Italy: [s.n.]. 2005. p. 100-109.

OH, O.; AGRAWAL, M.; RAGHAV, H. R. Information control and terrorism: Tracking the Mumbai terrorist attack through twitter. **Information Systems Frontiers**, 1, set. 2010. 33-43.

PASSONNEAU, R. J.; LITMAN, D. J. Discourse segmentation by human and automated means. **Journal Computational Linguistics**, 1, n. 23, mar. 1997. 103-139.

PHUVIPADAWAT, S.; MURATA, T. **Breaking news detection and tracking in Twitter.** In IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). Toronto, ON, Canada: IEEE Computer Society Washington, DC, USA. 2010. p. 120-123.

PIETER, A. Information. **The Stanford Encyclopedia of Philosophy**, 2013. Disponivel em: <http://plato.stanford.edu/archives/fall2013/entries/information/>. Acesso em: Fall 2013.

PLATANIOS, E.; BLUM, A.; MITCHELL, T. **Estimating Accuracy from Unlabeled Data**. Conference on Uncertainty in Artificial Intelligence. Quebec, Canada: [s.n.]. 2014. p. 1-10.

POPESCU, A.-M.; PENNACCHIOTTI, M. **Detecting controversial events from twitter**. CIKM '10 Proceedings of the 19th ACM international conference on Information and knowledge management. New York, NY, USA: [s.n.]. 2010. p. 1873-1876.

RAGHAVAN, S.; GARCIA-MOLINA, H. **Crawling the Hidden Web**. Proceedings of VLDB2001. Rome, Italy: [s.n.]. 2001. p. 129-138.

REYNAR, J. C. **An automatic method of finding topic boundaries**. ACL '94 Proceedings of the 32nd annual meeting on Association for Computational Linguistics. Stroudsburg, PA, USA: [s.n.]. 1994. p. 331-333.

RIJSBERGEN, C. **Information Retrieval**. London: Butterworths, 1979.

RUNGSAWANG, A.; ANGKAWATTANAWIT, N. Learnable topic-specific web crawler. **Journal of Network and Computer Applications**, v. 28, n. 2, p. 97–114, abr. 2005.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence:** A Modern Approach. [S.l.]: [s.n.]. 2003.

SAKAKI, T.; OKAZAKI, M.; MATSUO, Y. Earthquake shakes Twitter users: real-time event detection by social sensors. **Proceedings of the 19th international conference on World wide web (WWW '10)**, New York, NY, USA, 2010. 851-860.

SAKAKI, T.; OKAZAKI, M.; MATSUO, Y. **Earthquake shakes Twitter users:** real-time event detection by social sensors. Proceedings of the 19th international conference on World wide web (WWW '10). New York, NY, USA: [s.n.]. 2010. p. 851-860.

SALTON, G. **Introduction to Modern Information Retrieval**. [S.l.]: Mcgraw-Hill College, 1983.

SALTON, G. **Automatic text processing:** the transformation, analysis, and retrieval of information by computer. Boston, MA, USA: [s.n.], 1989.

SANDERSON, M.; CROFT, W. The history of information retrieval research. **Proceedings of the IEEE**, 13, 2012. 1444-1451.

SANKARANARAYANAN, J. et al. **TwitterStand:** news in tweets. GIS '09 Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Seattle, Washington, USA: ACM New York, NY, USA. 2009. p. 42-51.

SARMA, A. D.; DONG, X.; HAVELY, A. **Bootstrapping Pay-As-You-Go Data Integration Systems**. Proceedings of SIGMOD2008. Vancouver, Canada: [s.n.]. 2008. p. 861-874.

SAYYADI, H.; HURST, M.; MAYKOV, A. Event Detection and Story Tracking in Social Streams. **Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2009). AAAI**, San Jose, CA, 2009.

SAYYADI, H.; L., R. A Graph Analytical Approach for Topic Detection. **ACM Transactions on Internet Technology (TOIT)**, dez. 2013. Nº 4.

SHOKOUHI, M.; SI, L. Federated Search. **Foundations and Trends in Information Retrieval**, 1, n. 5, 2011. 1-102.

STOKES, N.; J., C. **Combining semantic and syntactic document classifiers to improve first story detection**. SIGIR '01 Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY,USA: [s.n.]. 2001. p. 424-425.

SWITZER, P. **Vector Images in Document Retrieval**. Statistical association methods for mechanized documentation. [S.l.]: [s.n.]. 1964. p. 163-172.

TAUBE, M.; GULL, C.; WACHTEL, I. Unit terms in coordinate indexing. **American Documentation**, v. 3, n. 4, p. 213-218, 1952.

VAN BENTHEM, J.; VAN ROOY, R. Connecting the Different Faces of Information. **Journal of Logic, Language, and Information,** 4, 2003. 375-379.

VOORHEES, E. M.; HARMAN, D. **Overview of the Seventh Text REtrieval Conference TREC-7**. Proceedings of the Seventh Text REtrieval Conference (TREC-7). [S.l.]: [s.n.]. 1998. p. 1-24.

WANG, X.; GERBER, M. S.; BROWN, D. E. Automatic crime prediction using events extracted from Twitter posts. **Proceedings of the 5th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction, SBP'12**, Berlin, Heidelberg, 2012. 231-238.

WANG, Y. et al. Selecting queries from sample to crawl deep web data sources. **Web Intelligence and Agent Systems**, v. 1, n. 10, p. 75-88, 2010.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent Agents: Theory and Practice. **Knowledge Engineering Review**, 1995. 115-152.

WU, P. et al. **Query Selection Techniques for Efficient Crawling of Structured Web Sources**. Proceedings of ICDE2006. Atlanta, GA: [s.n.]. 2006. p. 47-56.

YAMRON, J. **Topic detection and tracking segmentation task**. Proceedings Broadcast News Transcription andd Understanding Workshop. [S.l.]: [s.n.]. 1998.

YANG, Y. et al. **Topic-conditioned novelty detection**. Proceedings of The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD. New York, NY, USA: [s.n.]. 2002. p. 688-693.

YANG, Y. et al. Learning Approaches for detecting and tracking news events. **IEEE Intelligent Systems**, v. 4, n. 14, p. 32-43.

YANG, Y.; PIERCE, T.; CARBONELL, J. **A study of retrospective and on-line event detection**. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. New York, Ny, USA: [s.n.]. 1998. p. 28-36.

YU, K.-H. et al. Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. **Nature Communications**, 7, jul. 2016. 12474º.