COPPE/UFRJ

DETECÇÃO DE LINHAS DE PESQUISA EM ARTIGOS CIENTÍFICOS

Michele Socorro Silva Machado

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

Outubro de 2009

DETECÇÃO DE LINHAS DE PESQUISA EM ARTIGOS CIENTÍFICOS

Michele Socorro Silva Machado

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.

Aprovada por:

Prof. Geraldo Bonorino Xexéo, D. Sc.

Prof. Jano Moreira de Souza, D. Sc.

Prof. Eduardo Bezerra da Silva, D. Sc.

RIO DE JANEIRO, RJ – BRASIL

OUTUBRO DE 2009

Machado, Michele Socorro Silva

Detecção de Linhas de Pesquisa em Artigos Científicos / Michele Socorro Silva Machado. – Rio de Janeiro: UFRJ/COPPE, 2009.

XII, 66 p.: il.; 29,7 cm.

Orientador: Geraldo Bonorino Xexéo

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2009.

Referencias Bibliográficas: p. 44-50.

1. Detecção de linhas de pesquisa. 2. Agrupamento de documentos. 3. Agrupamento hierárquico. I. Xexéo, Geraldo Bonorino. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Àquele que esteve ao meu lado durante todo o desenvolvimento deste trabalho e não me deixou fraquejar diante das dificuldades encontradas, meu marido Paulo.

E ao meu sobrinho Silvinho (*in memoriam*), por me ensinar o que é ter força de vontade.

Vida!

Charles Chaplin

Já perdoei erros quase imperdoáveis, tentei substituir pessoas insubstituíveis e esquecer pessoas inesquecíveis.

Já fiz coisas por impulso, já me decepcionei com pessoas quando nunca pensei me decepcionar, mas também decepcionei alguém.

Já abracei pra proteger, já dei risada quando não podia, fiz amigos eternos, amei e fui amado, mas também já fui rejeitado, fui amado e não amei.

Já gritei e pulei de tanta felicidade, já vivi de amor e fiz juras eternas, "quebrei a cara muitas vezes"!

Já chorei ouvindo música e vendo fotos, já liguei só para escutar uma voz, me apaixonei por um sorriso, já pensei que fosse morrer de tanta saudade e tive medo de perder alguém especial (e acabei perdendo).

Mas vivi, e ainda vivo!
Não passo pela vida... E você também não deveria passar!

Viva!
Bom mesmo é ir à luta com determinação, abraçar a vida com paixão,
perder com classe e vencer com ousadia, porque o mundo pertence a quem se atreve e a vida é "muito" pra ser insignificante.

Agradecimentos

A Deus, por me guiar em todos os momentos da minha vida e por mostrar que tudo tem um propósito positivo.

Ao meu marido Paulo, por existir e fazer parte da minha vida. Meu maior incentivador.

A minha família, principalmente meus pais, que nunca mediram esforços para que tivéssemos uma educação de qualidade, e meus irmãos que sempre me incentivaram a seguir meus sonhos, por mais difíceis que eles pudessem parecer.

Ao orientador Geraldo Xexéo, por ter aceitado a me orientar, pela paciência e por todos os ensinamentos.

Ao prof. Jano e ao prof. Eduardo, por fazerem parte da banca.

Às amigas, Melissa Paes e Patrícia Machado, por dividirem e entenderem todas as minhas angustias, afinal, estávamos no mesmo barco. E a Patrícia Fiuza, pelo apoio, idéias e revisão do texto.

Aos ex-professores de graduação Carla Delgado, José Antônio Xexéo e Clevi Rapkiewicz, pelo exemplo de profissionais que sempre foram para mim e indicação ao mestrado.

Aos chefes que tive durante esses longos anos, Jorge Narciso e Eduardo Cordts, por entenderem e aceitarem minha ausência.

E a todos aqueles que direta ou indiretamente contribuíram na elaboração deste trabalho.

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

DETECÇÃO DE LINHAS DE PESQUISA EM ARTIGOS CIENTÍFICOS

Michele Socorro Silva Machado

Outubro/2009

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

Este trabalho propõe desenvolver uma ferramenta para identificar linhas de pesquisa em um corpus de publicações científicas, de forma a facilitar pesquisadores, professores e estudantes a realizarem buscas por material para suas pesquisas. A ferramenta deverá ser capaz de processar o corpus, extraindo os dados das publicações de forma automatizada e organizar os artigos dentro das linhas de pesquisa identificadas, mostrando sua evolução no tempo e os autores e publicações mais influentes.

O trabalho foi baseado em técnicas de Detecção e Rastreamento de Tópicos (TDT) e na implementação da ferramenta BuzzTrack. Para realização dos testes, foi criado um corpus de publicações científicas, contendo dez anos de artigos de três congressos da área de banco de dados.

Além da apresentação e análise dos resultados da avaliação, são discutidas as limitações do trabalho e sugeridas algumas melhorias que podem ser futuramente realizadas na ferramenta, bem como são apontados novos horizontes de pesquisa.

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

RESEARCH LINES DETECTION IN SCIENTIFIC PAPERS

Michele Socorro Silva Machado

October/2009

Advisor: Geraldo Bonorino Xexéo

Department: Systems and Computer Engineering

This work proposes the development of a tool to identify research lines on a corpus of scientific publications, in order to facilitate researchers, teachers and students to conduct searches for material to help on their research. The tool should be able to process the corpus, extracting data from publications in an automated manner and organize articles within the research lines identified, showing its evolution over time, the authors and most influential publications.

The work was based on techniques of Topic Detection and Tracking (TDT) and on the implementation of the BuzzTrack tool. For testing, was created a corpus of scientific publications with ten years of articles from three meetings in the database research area.

In addition to the presentation and analysis of the evaluation results, the limitations of the work are discussed, some improvements that can be done in the tool are suggested and new research horizons are pointed out.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1
INTRODUÇÃO

Devido à evolução da tecnologia e ao avanço dos meios de comunicação, a quantidade de informação disponível cresce de forma exponencial (O'LEARY, 1997). Todos os dias são disponibilizadas enormes quantidades de informação, e sem um bom método de organização uma pessoa pode facilmente ficar perdida. A quantidade de informação que esta pessoa pode assimilar é muito menor do que a quantidade produzida dentro do mesmo período de tempo. Este fenômeno é conhecido por sobrecarga de informação (FENG e ALLAN, 2007).

Nos tempos de guerra do século passado, as notícias eram transmitidas através de cartas, que poderiam levar dias até a chegada a seu destino e cuja divulgação era bastante restrita. O desenvolvimento das tecnologias, tais como a criação dos jornais impressos, do rádio, da televisão e da Internet, aumentaram consideravelmente a velocidade de encaminhamento e divulgação dos fatos.

No meio acadêmico, algumas décadas atrás, pesquisadores, professores e alunos possuíam apenas livros, revistas e anais de congressos de forma impressa, que precisavam ser comprados e poderiam levar dias até a sua disponibilização. Atualmente, com a ajuda da Internet esses dias foram diminuídos a segundos.

Embora hoje tudo pareça mais fácil, muitas vezes temos a informação necessária disponível e não conseguimos encontrá-la. Este fato demonstra a necessidade de sistemas que nos auxiliem nessa localização. Esse problema, que é a gestão da informação em grandes volumes de dados multimídia distribuídos, foi apontado como um dos cinco grandes desafios da pesquisa em computação no Brasil para o período de 2006 a 2016 (SBC, 2006).

Nesse contexto, não há como saber o que existe em uma determinada área de estudo ou o que compõe um assunto. Para uma pessoa que quer rapidamente saber sobre os acontecimentos de um determinado período de tempo, é praticamente impossível ler

sobre todos os acontecimentos, assim como gerar buscas com termos de fatos desconhecidos (YANG *et al.*, 1998).

Pesquisadores e estudantes têm o hábito de participar de congressos e ler artigos para se manter atualizados. A quantidade de congressos e artigos produzidos torna inviável para estas pessoas o acesso a tudo que é produzido simultaneamente.

Sendo assim, oferecer uma forma de organizar artigos científicos baseados em linhas de pesquisa mostra-se ser uma tarefa bastante interessante e proveitosa, que pode beneficiar muitas pessoas.

1.1 – Motivação

Em fevereiro de 2007, o Portal ACM atingiu a marca de 1 milhão de publicações, entre livros, jornais, relatórios técnicos e dissertações. Sua taxa de crescimento é de 50 mil itens por ano (ACM, 2007). O Portal ACM é apenas uma das bases de conhecimento utilizadas em pesquisas acadêmicas.

Ao iniciar uma pesquisa, uma etapa fundamental é encontrar o estado da arte no que se refere a um determinado assunto. Nessa etapa é necessário identificar autores, termos e publicações relevantes sobre o assunto. Muitas vezes não temos esses dados no início da pesquisa, o que dificulta as buscas por material.

Essa pesquisa é prejudicada, pois atualmente os sistemas de busca de bibliotecas ou de sites indexadores de publicações são baseados em buscas por palavra-chave, como vemos abaixo nas imagens do site de Periódicos da CAPES, do Portal ACM e do IEEE Xplore:

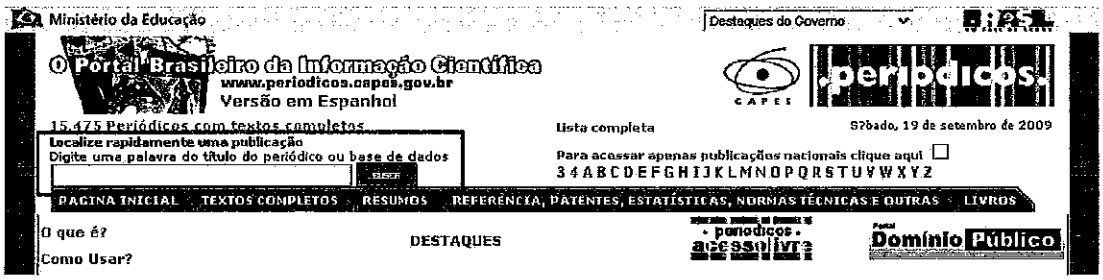

Figura 1 – Portal Periódico (CAPES)
http://www.periodicos.capes.gov.br

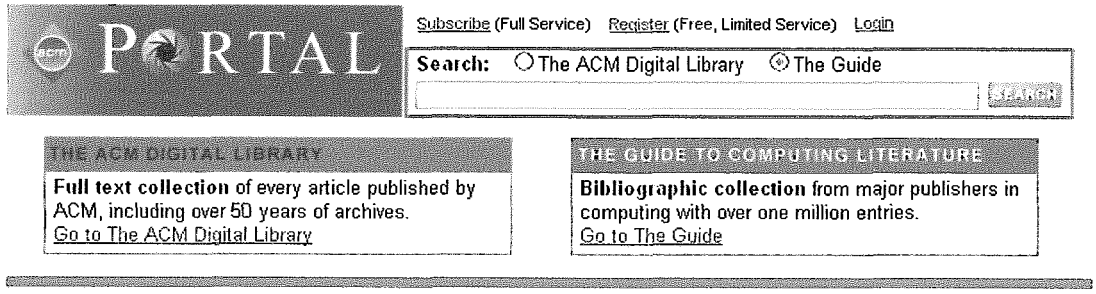

Figura 2 - ACM Portal
http://portal.acm.org

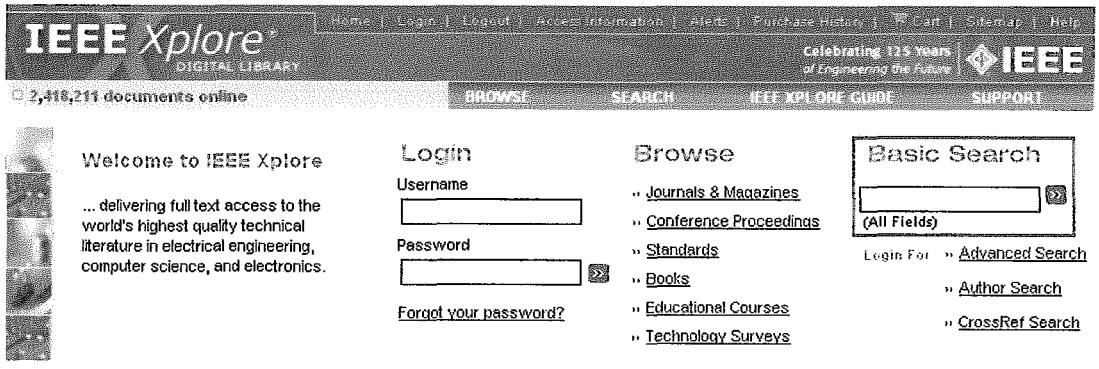

Figura 3 - IEEE Xplore
http://www.ieeexplore.ieee.org/

Desta forma, é necessário dispor de ferramenta computacional capaz de organizar um corpus de publicações científicas em linhas de pesquisa, identificando as linhas de pesquisa que possuem no corpus, mostrando as publicações e autores que a compõem.

Segundo FUKUMOTO e SUZUKI (2007), é cada vez mais difícil de encontrar e organizar o material relevante, e Detecção e Rastreamento de Tópicos (*Topic Detection and Tracking* – TDT) é uma área de investigação que visa resolver este problema.

O estudo em TDT iniciado em 1996 tinha como objetivo principal detectar, agrupar e organizar artigos de jornal que discutam sobre o mesmo evento (JIN *et al.*, 2007). Apesar de o projeto TDT ser voltado para o estudo de técnicas aplicadas a notícias, já é possível encontrar a utilização dessas técnicas em outros domínios, como a ferramenta BuzzTrack (CSELLE *et al.*, 2007), que é uma extensão para o Mozilla Thunderbird 1.5, oferecendo aos usuários a opção de organizar os e-mails baseados em tópicos, e a pesquisa para identificar o relato de um novo erro (*bug*) em um repositório de erros de um sistema (HIEW, 2006).

Durante o desenvolvimento desse trabalho, não foram encontradas pesquisas e ferramentas que utilizem técnicas de TDT para o processamento de publicações científicas com o objetivo de identificar linhas de pesquisa.

1.2 – Objetivos

O objetivo central da presente dissertação é desenvolver uma ferramenta para identificar linhas de pesquisa em um corpus de publicações científicas utilizando técnicas de TDT, de forma a facilitar pesquisadores, professores e estudantes a realizarem buscas por material para suas pesquisas.

Neste trabalho é adotada como definição de linha de pesquisa a definição dada pelo CNPQ, onde: "Linha de pesquisa representa temas aglutinadores de estudos científicos que se fundamentam em tradição investigativa, de onde se originam projetos cujos resultados guardam afinidades entre si." (CNPQ, 2009).

A ferramenta deverá ser capaz de processar o corpus, extraindo os dados das publicações de forma automatizada e organizar os artigos dentro das linhas de pesquisa identificadas, mostrando sua evolução no tempo e os autores e publicações mais influentes.

O programa TDT dividiu o estudo em cinco tarefas, que são: segmentação de histórias, detecção da primeira história, rastreamento de tópico, detecção de tópico e detecção de links. Mas, para identificar as linhas de pesquisa e organizar as publicações, são implementadas apenas as tarefas de detecção e rastreamento de tópicos. No estudo do TDT, os dados usados são histórias e nesse trabalho são artigos científicos. Nesse mesmo paralelo, os tópicos identificados no TDT são as linhas de pesquisa que se deseja identificar.

O trabalho está desenvolvido com base na técnica apresentada por MAKKONEN *et al.*, (2002), onde são propostas quatro classes, que representam lugares, tempo, nomes próprios e por último as demais palavras do documento. Tais classes são trabalhadas com pesos diferentes de forma a se alcançar o discernimento entre notícias semelhantes, como, por exemplo, dois diferentes acidentes de avião ocorridos em um curto espaço de tempo.

A implementação da solução está baseada no desenvolvimento da ferramenta BuzzTrack (CSELLE *et al.*, 2007), onde são usados algoritmos de agrupamento hierárquico, para o agrupamento dos e-mails, em razão destes algoritmos apresentarem

flexibilidade para a análise dos agrupamentos em diferentes níveis, o que naturalmente sugere um refinamento na análise dos padrões neles descritos.

Diante disso, vemos a necessidade de criação de um corpus de publicações científicas e a identificação de classes que caracterizem uma linha de pesquisa e auxiliem na separação dos documentos, para realização dos testes com a ferramenta proposta.

1.3 – Trabalhos relacionados

A identificação automática de tópicos de pesquisa tem despertado o interesse dos pesquisadores de várias áreas. THO *et al.* (2003) utilizam agrupamento de palavras-chave e autores para encontrar especialistas em específicas áreas de pesquisa. Agentes de indexação buscam e recuperam artigos científicos em sites acadêmicos. Destes artigos são extraídas as citações e armazenadas em um banco de dados de citações. As informações sobre os pesquisadores também são salvas em um banco de dados de pesquisadores. Uma técnica de multi-agrupamento (*multi-clustering*) é aplicada sobre estes bancos de dados e determinam os *expertises* em cada área. CHAN *et al.* (2006) utilizam uma técnica semelhante para identificação de grupos de pesquisa na área médica. NEWMAN (2006) utiliza autovetores de matrizes na identificação da estrutura que representa a co-autoria entre pesquisadores em um centro de pesquisa, e LEYDESDORFF e RAFOLS (2009) utilizam técnicas de agrupamento para criar mapas globais que identificam, além de co-autoria, a interdisciplinaridade.

A abordagem proposta neste trabalho difere destes relacionados, pois considera tanto as palavras-chave quanto autores e referências como um conjunto único de características de cada artigo. Além disso, faz uso de algoritmo de agrupamento hierárquico, o que permite níveis diferenciados no agrupamento e na análise dos grupos.

1.4 – Organização dos capítulos

Além deste capítulo de introdução, este trabalho apresenta mais 5 capítulos organizados da seguinte forma: o Capítulo 2 traz a fundamentação teórica sobre a área-chave para esta dissertação, que é Detecção e Rastreamento de Tópicos (TDT), relacionando algumas das abordagens existentes para solucionar cada etapa do TDT. O Capítulo 3 explica agrupamento de documentos e apresenta as principais técnicas

discutidas na literatura. O Capítulo 4 apresenta o passo a passo da implementação deste trabalho. O Capítulo 5 aborda os testes realizados, incluindo a descrição do corpus utilizado e a descrição do método de avaliação dos resultados. Finalmente, no Capítulo 6, são apresentadas conclusões do trabalho e algumas sugestões para trabalhos futuros.

CAPÍTULO 2
DETECÇÃO E RASTREAMENTO DE TÓPICOS

Neste capítulo serão apresentados os conceitos e definições pertinentes à Detecção e Rastreamento de Tópicos ou *Topic Detection and Tracking*, citada apenas como TDT, fornecendo o embasamento teórico necessário para compreensão do trabalho desenvolvido na presente dissertação.

2.1 – Origem

Os estudos em TDT tiveram início em 1996, como parte do programa TIDES (Translingual Information Detection, Extraction and Summarization) financiado pela DARPA (Defense Advanced Research Projects Agency).

O programa TIDES tinha como objetivo desenvolver tecnologias para o processamento de linguagens, de forma a possibilitar falantes da língua inglesa a encontrar e interpretar informações críticas em diversos idiomas, sem requerer conhecimento específico nos mesmos. Foram conduzidas pesquisas para desenvolver eficazes algoritmos de detecção, extração, sumarização e tradução, onde as fontes de dados são grandes volumes de texto ou fala em vários idiomas (DARPA, 2008).

O projeto TDT visa promover o desenvolvimento de técnicas que podem efetivamente organizar, pesquisar e estruturar textos de notícias a partir de uma variedade de fontes e meios de divulgação (NIST, 2008).

Durantes as pesquisas realizadas, foram criados corpus de teste e medidas de avaliação, que foram utilizados nos workshops organizados pelo NIST (*National Institute of Standards and Technology*) entre 1998 e 2004, onde os algoritmos eram submetidos a um mesmo corpus anotado e os resultados avaliados e comparados.

O corpus foi desenvolvido e é mantido pelo LDC (*Linguistic Data Consortium*). Até o momento estão disponíveis cinco versões do corpus, mas o TDT2 e o TDT3 ainda são os mais usados. O TDT2, por exemplo, possui 74.000 notícias de janeiro a junho de 1998, de 6 fontes em inglês e 3 em mandarim, com 100 tópicos anotados. O TDT3

possui notícias dos meses de outubro a dezembro de 1998, somando 45.000 notícias, de 8 fontes em inglês e 3 em mandarim, com 240 tópicos anotados. Todos os corpora possuem notícias em texto e transcrição de áudio de rádio e televisão (LDC, 2008).

2.2 – Objetivo

O objetivo principal do TDT é detectar, agrupar e organizar artigos de jornal que discutam sobre o mesmo evento (JIN *et al.*, 2007).

Segundo FUKUMOTO e SUZUKI (2007), com o crescimento exponencial das informações na internet, é cada vez mais difícil de encontrar e organizar o material relevante, e o TDT é uma área de investigação que visa resolver este problema.

O TDT consiste em quebrar o fluxo de notícias em várias histórias individuais, para acompanhar as notícias de eventos que não tenham sido vistos antes e agrupar as histórias em cada um dos grupos que discutem um único tópico (NALLAPATI, 2003).

O estudo do TDT irá beneficiar muito leitores, pois, além de identificar eventos e apresentar tópicos e palavras-chaves das notícias, permite ao usuário manter-se atualizado sobre o desenvolvimento de um tópico e identificar rapidamente as novidades, permitindo uma organização diferente das notícias, agrupando diversas fontes de dados e mídias, como internet, jornal, rádio e televisão.

O TDT integra a pesquisa de recuperação de informação, gestão da informação e mineração de dados para planejar algoritmos poderosos, amplamente úteis e inteiramente automáticos para determinar a estrutura de tópicos de dados em linguagem humana (LIN e LIANG, 2008). Existem propostas de soluções baseadas em mineração de dados, recuperação de informação, filtragem, agrupamento, classificação e sumarização, mas ainda não existe uma solução ótima para o problema.

2.3 – Definições: História, Evento e Tópico

As definições de história, evento e tópico são muito importantes para o estudo do TDT, pois visam unificar o entendimento das tarefas a serem realizadas.

Em FISCUS e DODDINGTON (2002), "história é um segmento topicamente coeso de notícias que inclui duas ou mais cláusulas declarativas independentes sobre um único evento". A história é a menor unidade da notícia.

ALLAN (2002a) definiu evento como algo não trivial, que aconteceu em um determinado momento e lugar específico.

"Tópico é definido como um evento ou atividade seminal, juntamente com os eventos e atividades diretamente relacionadas" (CIERI *et al.*, 2002).

Por exemplo, quando uma bomba explode em um prédio, este é o evento seminal que desencadeia o tópico. Qualquer história que discuta a explosão, tentativa de resgate, a busca por culpados, prisões, julgamentos, e assim por diante, são todas parte do tópico. Histórias sobre outra bomba que explode no mesmo dia em qualquer outro lugar no mundo não são esperadas que esteja no mesmo tópico.

2.4 – Tarefas

Em 1997, um estudo-piloto lançou as bases essenciais do TDT, realizando três tarefas principais: a segmentação, detecção e acompanhamento de tópicos. Logo depois, o programa TDT definiu as cinco tarefas que dividem o estudo:

1. Segmentação de Histórias: detecta o limite das histórias.
2. Detecção da Primeira História: detecta a primeira história que discute um tópico.
3. Rastreamento de Tópico: detecta histórias que discutem um tópico específico.
4. Detecção de Tópico: detecta grupos de histórias que discutem o mesmo tópico.
5. Detecção de Links: detecta se duas histórias discutem sobre o mesmo tópico.

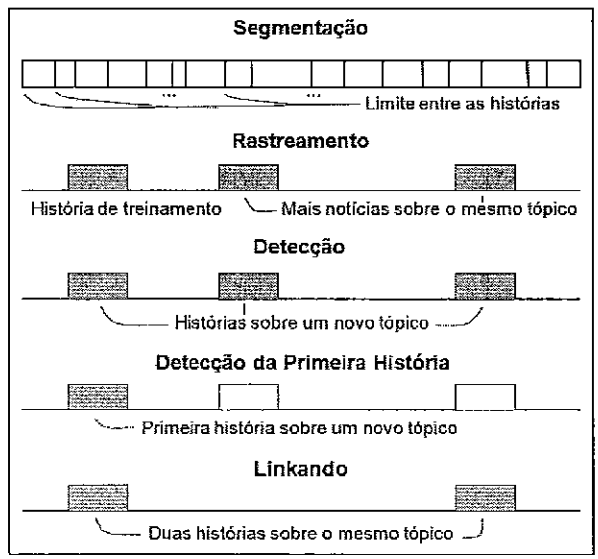

Figura 4 - Tarefas do TDT
(Adaptado de WAYNE, 2000)

Estas cinco tarefas não são comumente estudadas em conjunto. As publicações sobre estes assuntos normalmente abordam uma ou duas tarefas do TDT, e, como todas as tarefas normalmente devem ser desenvolvidas de forma a trabalharem com vários idiomas e tipos de mídia, existem poucos artigos científicos que estudem o conjunto de todas as tarefas, suas correlações e aplicações como uma solução de TDT.

2.4.1 – Segmentação de Histórias

Segmentação de histórias é a tarefa de dividir um fluxo contínuo de notícias em histórias individuais. Como as fontes de notícias em texto são fornecidas de forma segmentada, esta tarefa se aplica somente ao subconjunto de áudio, rádio e televisão. A segmentação pode ser realizada utilizando o próprio sinal de áudio ou a transcrição textual, manual ou automática do sinal de áudio (NIST, 2008).

Fontes de áudio de notícia geralmente transmitem várias histórias e raramente fornecem uma quebra óbvia entre as histórias, embora humanos consigam distinguir uma história de outra. Apesar de anúncios serem aparentemente indicadores naturais do limite de uma história, não é fácil identificar um anúncio no fluxo de notícias, e em alguns casos a transmissão não possui anúncios (ALLAN, 2002a).

Embora o trabalho de transcrição seja parte da tarefa, a maior parte das pesquisas está focada em como fazer a segmentação usando texto transcrito (STOLCKE *et al.*, 1999). As demais tarefas do TDT irão usar apenas texto em seu processamento. Assim, a tarefa de segmentar histórias é um pré-processamento, ou seja, cada uma das outras tarefas espera como entrada texto de histórias individuais. Dessa forma, um sistema de segmentação deve ser capaz de segmentar o fluxo e separar os anúncios, para não interferir no resultado das tarefas seguintes. Segundo ALLAN (2002a), observou-se que sistemas de segmentação com maus resultados apresentaram pequenos efeitos na tarefa de rastreamento, mas forte impacto nas tarefas de detecção.

Como exemplo de solução para essa tarefa, temos FRANZ e XU (2007), onde a segmentação de histórias é tratada como um problema de classificação binária. O modelo é treinado para estimar a probabilidade do limite de uma história ocorrer em um momento no qual a fala é dividida por ocorrências de trechos de silêncio, dado o contexto que a envolve e outros recursos. O classificador é um modelo de entropia máxima, e utiliza recursos léxicos, de similaridade de texto, de prosódia e de posição na aparição.

2.4.2 – Detecção da Primeira História

O objetivo da tarefa de detecção da primeira história é reconhecer quando um novo tópico aparece, que não tenha sido discutido anteriormente. Como o processamento de um sistema de TDT é em linha de tempo, a primeira história é relativa ao que já foi visto. Então, no momento que o sistema é ligado, a primeira história passa a ser um novo tópico (ALLAN, 2002a).

A tarefa de detecção de primeira história seria parte de um sistema de TDT que alerta ao usuário quando um novo evento ocorrer (NIST, 2008).

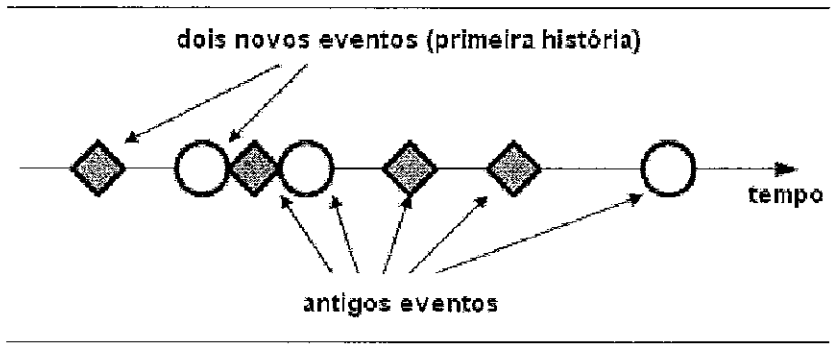

Figura 5 - Detecção da primeira história em um fluxo de notícias. Dois eventos são marcados por diamantes e círculos, assinalado a primeira história de cada caso (Adaptado de BRANTS *et al*, 2003)

KUMARAN *et al.* (2004) explorou a aplicação de técnicas de classificação baseadas em aprendizagem de máquina para a tarefa de detecção de novas histórias. O artigo trata o tema como um problema de classificação booleana, onde cada história precisa ser classificada como nova ou antiga. Foram desenvolvidos novos classificadores baseados no conceito de triangulação de entidades nomeadas dos documentos. Os resultados mostraram uma melhora significativa e consistente sobre o modelo de espaço vetorial em todas as coleções que foram testadas.

2.4.3 – Rastreamento de Tópico

Rastreamento de tópico é definido como a tarefa de associar novas histórias recebidas com tópicos previamente conhecidos pelo sistema, isto é, encontrar mais histórias sobre um determinado tópico (LDC, 1997).

O objetivo de um sistema de rastreamento de tópicos é "rastrear" eventos que o usuário designa como sendo de interesse em histórias futuras. O usuário indica as histórias encontradas sobre um evento que ele gostaria de acompanhar, e o sistema usa

essas histórias para "aprender" e alertar o usuário quando uma nova história é publicada sobre este evento (NIST, 2008).

O rastreamento é uma tarefa supervisionada onde normalmente são indicados de 1 a 4 documentos (sementes) em seu treinamento, e em sua tomada de decisão as histórias podem ser atribuídas a mais de um tópico, ou mesmo nenhum (FRANZ *et al.*, 2001).

YANG *et al.* (2000) adaptaram métodos de classificação supervisionada de texto, sendo o algoritmo *k-Nearest Neighbor* (kNN) e uma abordagem *Rocchio*, para rastrear tópicos. Todos estes métodos mostraram melhorias significativas, de até 71% de redução nas taxas de erro, sobre a performance do algoritmo original do kNN. A combinação desses métodos mostrou bons resultados nos testes realizados em diferentes corpora, sendo sugerido como uma robusta solução para otimização de parâmetros em sistema de rastreamento de tópicos.

2.4.4 – Detecção de Tópico

A detecção de tópico é definida como a tarefa de identificar todos os eventos em um corpus de histórias. Os eventos são definidos por sua associação com histórias, e, por isso, a tarefa é agrupar as histórias, onde cada grupo representa um tópico e todas as histórias neste grupo discutem sobre esse tópico (ALLAN *et al.*, 1998).

Assim, a tarefa de detecção de tópicos deverá construir grupos (*clusters*) de histórias que discutam o mesmo tópico. Essa tarefa é realizada de forma não supervisionada e sem o conhecimento do número de grupos que podem ser gerados (FREY *et al.*, 2001). A grande diferença entre as tarefas de rastreamento e detecção de tópico é que no rastreamento, algumas notícias são conhecidas e a maioria pode ser descartada como totalmente irrelevante, a detecção não tem qualquer informação prévia ou descrição sobre o tópico, e cada história deve ser agrupada em um tópico (FRANZ *et al.*, 2001). Conforme ilustrado na figura 6.

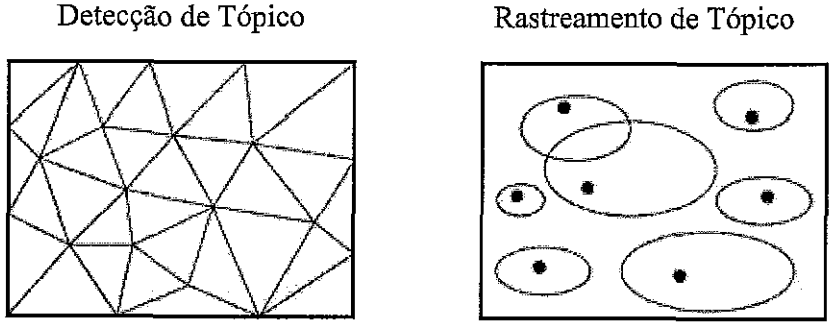

Figura 6 - Diferença entre as tarefas de detecção e rastreamento de tópico (FRANZ *et al*, 2001)

Quando aparecem novos tópicos no fluxo de notícias, o sistema deve automaticamente decidir criar um novo grupo (detecção da primeira história). Embora as tarefas de detecção de tópico e detecção da primeira história estejam relacionadas, as técnicas de detecção de tópico podem não ser suficientes para resolver a detecção da primeira história de forma eficiente. Dessa forma as soluções para uma tarefa podem ajudar a outra, mas não necessariamente resolver o problema (ALLAN, 2002b). A diferença entre as duas tarefas é o retorno, na detecção de tópicos serão apresentados grupos, e na detecção da primeira história irá retornar SIM ou NÃO para se a história discute um novo tópico. Apenas uma parte de um sistema de detecção de tópicos é a de decidir se cada história discute ou não de um novo tópico.

A Dragon Systems (LOWE, 1999) descreve seu sistema de detecção em termos de um modelo de mistura beta-binomial. Esta é uma aproximação de modelagem de língua onde é calculada a distribuição da probabilidade de uma palavra ocorrer um determinado número de vezes em um documento de um tamanho particular. A medida que os documentos são processados, os parâmetros para a distribuição de cada palavra são re-estimados. Foi utilizado um léxico relativamente pequeno, entre 20 e 60 mil palavras. O modelo binomial foi comparado com um modelo multinomial mais custoso, que resultou em pequenas melhorias para um custo médio do TDT2.

2.4.5 – Detecção de Links

A tarefa de detecção de links irá identificar quando duas histórias discutem ou não sobre o mesmo evento. Ela deve emitir como resposta apenas SIM, se as duas histórias discutem o mesmo evento, ou NÃO. Essa tarefa pode servir de base para as

tarefas de detecção da primeira história, rastreamento e detecção de tópico, pois restringe o universo de comparação (FISCUS e DODDINGTON, 2002).

Essa tarefa não foi amplamente adotada pela comunidade de pesquisa, pois sua aplicabilidade não é obvia e não ficou claro como ela deveria ser usada. (ALLAN, 2002a).

Em YANG *et al.* (2002) é utilizada medida de similaridade do cosseno em um modelo TF*IDF de pesos para detectar se duas histórias discutem sobre o mesmo tópico. Sua similaridade é comparada a um marco predefinido para que a decisão seja efetuada de forma binária. Para a avaliação, os valores TF*IDF são inicializados através de uma coleção completa de histórias em inglês, em um corpus de treinamento, onde são incrementalmente atualizados, de forma a adaptar as mudanças no padrão ao longo do tempo.

CAPÍTULO 3

AGRUPAMENTO DE DOCUMENTOS

Agrupamento (*clustering*) de documentos é a atividade de agregar documentos similares, de maneira a melhor discriminar documentos pertencentes a grupos diferentes. É realizado sobre um conjunto de documentos através da divisão desse conjunto de documentos em subconjuntos ou grupos (*clusters*), de forma que documentos dentro de um mesmo grupo são similares, e diferentes dos documentos de outros grupos. Esses métodos buscam ajudar o usuário a entender a estrutura natural em um conjunto de dados.

O agrupamento é a classificação não supervisionada de padrões em grupos (JAIN *et al.*, 1999). É importante entender a diferença entre a classificação não supervisionada e a classificação supervisionada. Na classificação supervisionada, é fornecida uma coleção pré-classificada, onde o problema é rotular um novo documento ainda não classificado. Tipicamente, os padrões rotulados são utilizados para aprender a descrição da classe (treinamento) e esta informação aprendida, por sua vez, é usado para rotular um novo padrão. No caso da classificação não supervisionada, o problema é colocar em grupos um conjunto de padrões não rotulados de forma que os grupos tenham um significado relevante.

Geralmente a tarefa de agrupamento se baseia em escolher uma partição que maximize um critério, ou seja, é um problema de otimização. Enumerar todos os possíveis agrupamentos e escolher aquele com melhor valor da função objetivo é um problema intratável (NP completo). É necessária uma heurística para tornar esta busca eficiente.

Não existe uma técnica de agrupamento universal, capaz de revelar toda a variedade de estruturas que podem estar presentes em conjuntos de dados multidimensionais. Uma solução mais geral consiste em definir medidas de similaridade entre dois clusters assim como um critério global.

Segundo ZAIANE (2002), a tarefa de agrupamento exige métodos que apresentem as seguintes características:

- Ser capaz de lidar com dados com alta dimensionalidade;
- Ser "escalável" com o número de dimensões e com a quantidade de elementos a ser agrupados;
- Habilidade para lidar com diferentes tipos de dados;
- Capacidade de definir agrupamentos de diferentes tamanhos e formas;
- Exigir o mínimo de conhecimento para determinação dos parâmetros de entrada;
- Ser robusto à presença de ruído;
- Apresentar resultado consistente independente da ordem em que os dados são apresentados;

Em geral, algoritmo algum atende a todos esses requisitos e, por isso, é importante entender as características de cada algoritmo para a escolha de um método adequado a cada tipo de dado ou problema (HALKIDI *et al.*, 2001).

A maioria dos algoritmos de agrupamento, são de alguma forma, dependentes de uma distância a qual é utilizada para definir que documentos são similares e quais não são. Existem muitas medidas de distância disponíveis na literatura. Através de um processo iterativo estes algoritmos calculam parâmetros de modo a diminuir o valor da função objetivo iteração a iteração até um estado de convergência ser atingido.

Geralmente, a atividade de agrupamento envolve as seguintes etapas (JAIN e DUBES, 1988):

1. Uma representação padrão (opcionalmente incluindo a extração de recursos e/ou seleção);
2. Definição de um padrão medida de similaridade adequada para o domínio de dados;
3. Agrupamento;
4. Abstração de dados (se necessário);
5. Avaliação do resultado (se necessário).

Agrupamento tem sido um problema amplamente estudado em uma variedade de domínios de aplicação, incluindo reconhecimento de padrões, análise de dados, processamento de imagens e pesquisa de mercado (JAIN *et al.*, 1999).

Os algoritmos de agrupamento geralmente são classificados como de particionamento ou hierárquico que serão abordadas em mais detalhes a seguir. Além das técnicas hierárquicas e não-hierárquicas, outras técnicas como algoritmos evolutivos, agrupamentos *fuzzy*, redes neurais (mapas de *Kohonen*), entre outras, podem ser empregadas para formação de agrupamentos, mas não serão abordadas nesse trabalho.

3.1 – Algoritmos de particionamento

Um algoritmo de agrupamento de particionamento é aquele que particiona todos os dados em grupos, sem organizar estes grupos em uma hierarquia, como é feito no hierárquico.

Para realizar agrupamento com particionamento, vários algoritmos já foram propostos e aplicados em documentos, tais como os algoritmos baseados em grafos, que usa uma matriz de proximidade dos pontos e conceitos de grafos e hiper-grafos para agregar ou dividir pontos (GUHA *et al.*, 1999), algoritmos baseados na teoria da informação (SLONIM *et al.*, 2002) ou algoritmos que usam o conceito de centróide, como o K-means (KANUNGO *et al.*, 2002).

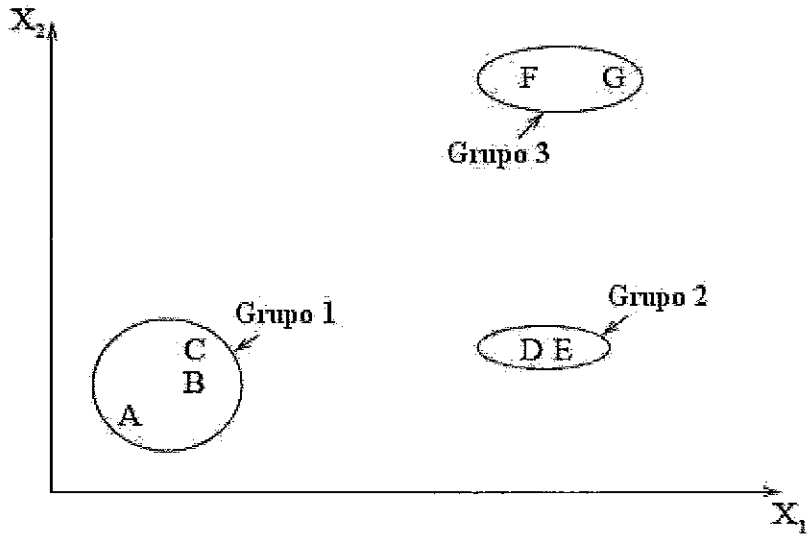

Figura 7 - Exemplo de agrupamento por particionamento

3.2 – Algoritmos hierárquicos

O agrupamento hierárquico associa os dados a uma hierarquia de grupos. Para agrupamento de documentos, a solução hierárquica possui mais vantagens em relação à abordagem de particionamento, por proporcionar uma melhor visão de quais tipos de questões podem ser respondidas pela coleção de documentos. Ela é capaz de dividir a coleção de documentos em diferentes níveis de granularidade e especificidade, expandindo as opções do usuário, ajudando-o a decidir quais grupos lhe são interessantes na coleção de documentos (SAHOO *et al.*, 2006; ZHAO e KARYPIS, 2002).

Algoritmos de agrupamento hierárquico podem ser aglomerativos ou divisivos. Aglomerativo é uma abordagem ascendente (*bottom-up*) e começa afetando cada documento a um grupo distinto e prossegue combinando os documentos e grupos mais similares, até que todos os documentos sejam alocados a um único grupo, ou outro critério de parada seja alcançado. Divisivo é uma abordagem descendente (*top-down*) e começa considerando todos os documentos em um único grupo, escolhendo um grupo, particionando-o em outros grupos e prossegue escolhendo e dividindo até que cada grupo terminal da árvore possua somente um documento, ou outro critério de parada seja alcançado.

A maioria dos algoritmos de agrupamento hierárquico são variantes do *single-link, complete-link*, e *minimum-variance*. Destes, os algoritmos *single-link* e *complete-link* são mais populares (JAIN *et al.*, 1999).

Os grupos gerados nos métodos hierárquicos, são geralmente representados por um diagrama bi-dimensional chamado de dendograma ou diagrama de árvore. Neste diagrama, cada ramo representa um elemento, enquanto a raiz representa o agrupamento de todos os elementos. A figura a seguir traz um exemplo de dendograma.

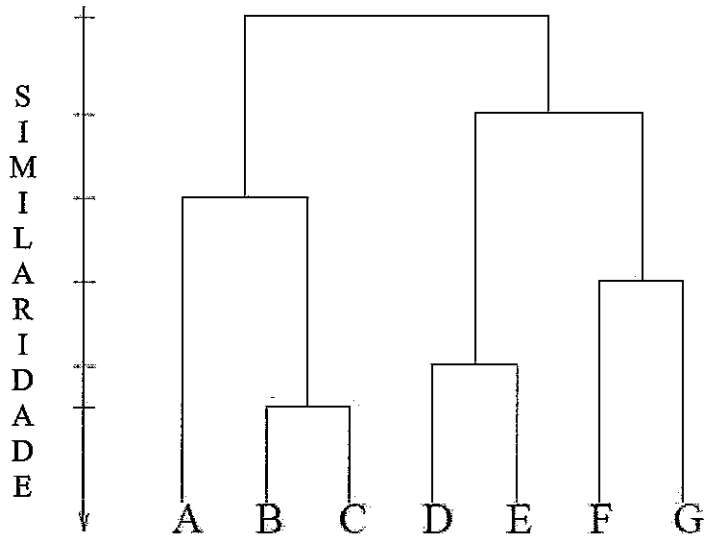

Figura 8 - Exemplo de dendograma
(JAIN *et al.*, 1999)

Através do dendograma e do conhecimento prévio sobre a estrutura dos dados, deve-se determinar uma distância de corte para definir quais serão os grupos formados. Essa decisão é subjetiva, e deve ser feita de acordo o objetivo da análise e o número de grupos desejados.

CAPÍTULO 4
IMPLEMENTAÇÃO

Este capítulo dá uma visão geral das implementações realizadas nesta dissertação. Começamos discutindo a criação do corpus, identificação das classes, sua representação e cálculo de similaridade, e abordamos o processamento para identificação das linhas de pesquisa e o agrupamento dos artigos nas mesmas, finalizando com a etiquetagem dos grupos.

4.1 – Introdução

O trabalho está desenvolvido com base na técnica apresentada por MAKKONEN *et al.*, (2002), onde são propostas quatro classes, que representam lugares, tempo, nomes próprios e o por último, as demais palavras do documento. Assim, são propostas seis classes para representar o artigo científico, que são: Abstract, Autores, Referências, Citação, Keywords e Texto. Cada uma dessas classes é explicada separadamente no item 4.3.

Com base nos resultados apresentados em CSELLE *et al.* (2007), a implementação da solução está baseada em algoritmos de agrupamento hierárquico, que foi adaptado para o processamento usando as classes propostas e uso de heurísticas para a realizar a etiquetagem dos grupos.

Além do desenvolvimento da ferramenta para identificar linhas de pesquisa em artigos científico, faz parte deste trabalho construir um corpus de artigos para realização dos testes.

4.2 – Corpus

Uma das etapas deste trabalho foi construir um corpus de artigos científicos para ser usado para a realização dos testes.

4.2.1 – Seleção dos textos

Como o processamento do TDT é efetuado em linha de tempo, os textos a serem selecionados para o corpus devem possuir uma informação temporal de sua publicação para possibilitar a simulação do desenvolvimento dos assuntos ao decorrer do tempo. De forma a possibilitar um amplo estudo desta linha de tempo, foi determinado que o corpus tivesse artigos compreendendo um período de 10 anos, no presente estudo, de 1999 a 2008.

Foram utilizados alguns critérios na escolha dos congressos. Estes precisavam ter publicações no período determinado, disponíveis em formato eletrônico e que fossem de assuntos ligados à área de pesquisa Banco de Dados, na qual esse trabalho está inserido, de forma a facilitar a análise dos resultados.

Foram escolhidos três congressos:

- *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*;
- *Proceedings of the ACM SIGMOD International Conference on Management of Data*;
- *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Todos os artigos apresentam o texto em inglês.

Os dados foram extraídos da *Association of Computing Machinery's Digital Library of Scientific Literature* (ACM Portal)[1].

4.2.2 – Organização dos dados

Os dados coletados dos artigos foram armazenados em um banco de dados MySQL e as informações disponíveis são:

Sobre os eventos: nome completo, sigla, local onde o congresso foi realizado e a data de quando ocorreu;

Sobre os artigos: texto completo, número de páginas e a sessão do congresso.

[1] http://portal.acm.org/

4.2.3 – Números

Ao total, o corpus possui 3.195 artigos.

Tabela 1 - Total de artigos do corpus por ano e congresso

Ano	SIGIR	SIGMOD	SIGKDD	Total de Artigos
1999	74	80	62	216
2000	70	79	67	216
2001	86	72	71	229
2002	105	75	88	268
2003	103	81	92	276
2004	129	114	110	353
2005	134	116	101	351
2006	148	98	126	372
2007	212	135	110	457
2008	204	127	126	457
Total	1.265	977	953	3.195

Esse total não representa o total de artigos publicados nos anos selecionados em cada congresso, mas é apenas a quantidade de artigos que foi possível coletar.

4.3 – Classes

Outra etapa a ser realizada é a identificação das classes que irão representar o documento.

Foram identificadas 5 classes, além da classe composta por todas as palavras do artigo, que são: abstract, autores, keywords, referências e citação. Segue abaixo uma explicação sobre cada classe e como é calculada a medida de similaridade.

Abstract

A classe Abstract (AB) é composta pelas palavras do resumo do artigo. Sua similaridade é calculada baseada no espaço vetorial, pelo cálculo do cosseno do ângulo formado entre dois vetores.

A sessão *abstract* do artigo representa uma sumarização feita pelo autor para seu artigo. Em um teste no corpus de artigos construído, foi observado que, em média, 85% das palavras do resumo estavam contidas no restante do texto.

Autores

A classe Autores (AT) é composta pelo nome dos autores do artigo. A sua similaridade é medida comparando os autores de um artigo, com os autores dos demais artigos do corpus, buscando fortalecer a ligação entre artigos escritos pelos mesmos autores.

Referências

A classe Referências (RF) representa a ligação entre as referências bibliográficas de dois artigos. Para encontrar essa ligação, foram extraídos do artigo todos os títulos dos trabalhos citados e comparados com os títulos dos trabalhos citados pelos demais artigos do corpus. Dessa forma, vamos aumentar a ligação dos artigos que citam os mesmos trabalhos.

Citação

A classe Citação (CT) é composta pela indicação se artigo é referenciado por outro. Para encontrar essa relação, foi extraído do artigo o título e comparado com as referências bibliográficas dos demais.

Keywords

A classe Keywords (KW) é composta pelas palavras-chave do artigo. Essas palavras-chave são a representação de uma categorização feita pelo autor para seu artigo. Dessa forma, deseja-se fortalecer a ligação entre artigos que possuem as mesmas palavras-chave e representam as mesmas categorias.

Texto

A classe Texto (TX) é composta por todas as palavras do artigo. Sua similaridade é calculada baseada no espaço vetorial, pelo cálculo do cosseno do ângulo formado entre dois vetores.

4.4 – Agrupamento

Esta seção explica o algoritmo de agrupamento para identificar e agrupar os artigos em linhas de pesquisa. O processo é bastante semelhante aos métodos utilizados para o agrupamento de notícias abordado no estudo TDT. Mas o artigo científico é

muito mais rico no conteúdo da informação que o texto de notícia, onde podemos usar essa informação para fortalecer as ligações entre os artigos

O algoritmo usado como base é o algoritmo de agrupamento hierárquico *complete-link clustering*, com medida de similaridade de texto o cosseno do ângulo entre dois artigos e acrescentado outros atributos textuais e não textuais.

Nossa primeira opção de algoritmo foi o algoritmo de agrupamento hierárquico *single-link clustering*, pela experiência relatada em CSELLE *et al.* (2007), mas a comparação entre os dois algoritmos feita em LINGPIPE (2009), mostra que o algoritmo *complete-link clustering* tende a ser mais útil por criar grupos mais bem centrado.

A saída do processamento é uma árvore ou dendrograma anotado com as palavras que caracterizam o grupo. Dessa forma, o usuário pode navegar entre grupos mais ou menos especializados, de acordo com sua necessidade.

Foi por essa característica que foi escolhido o agrupamento hierárquico, permitindo diversas visualizações dos grupos, e também por não existir uma regra ou fórmula que indique a quantidade de grupos que uma coleção irá gerar e que fosse atualizado sempre que um novo documento fosse incluído ao corpus.

O processamento para identificar as linhas de pesquisas e agrupar os artigos se deu em cinco etapas, conforme mostrado na imagem abaixo e detalhado a seguir:

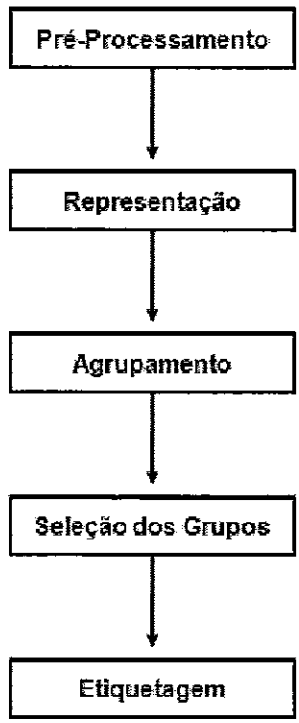

Figura 9 - Arquitetura básica para o agrupamento de documentos

4.4.1 – Pré-processamento

Em geral, para os artigos disponibilizados em formato eletrônico, adota-se o formato PDF (*Portable Document Format*). Todos os artigos coletados que formam o corpus de teste estão nesse formato. Então, o primeiro passo do pré-processamento é a extração da informação textual dos artigos.

Em um segundo passo, é necessária a extração dos dados para o processamento das classes. E como último passo, a limpeza dos textos das classes Abstract e Texto, para a extração de *stopwords* e *stemming*.

Stopwords são termos que usualmente ocorrem em grande parte dos documentos e são geralmente constituídos por artigos, preposições e conjunções. A eliminação de *stopwords* é necessária, pois estes termos não servem para discriminar documentos.

O processo chamado de *stemming* é o de extração do radical das variantes de uma mesma palavra. Assim as variantes passam a ser representadas por um mesmo termo. O algoritmo mais utilizado para realização de *stemming* é o de Porter (PORTER, 1997), que usa uma lista de sufixos e aplica uma série de regras para extrair os sufixos das palavras.

4.4.2 – Representação

Os artigos são representados pelo Modelo de Espaço Vetorial, onde um documento pode ser descrito por um conjunto de palavras-chave, os chamados termos de indexação, que são todo o vocabulário presente na coleção de textos (SALTON *et al.*, 1975).

As classes Abstract e Texto serão representadas por um vetor de atributos (termos ou palavras), e cada posição do vetor possui um valor relacionado ao atributo, ou seja, um peso. Nesse trabalho, foi usado o valor da medida *term frequency-inverse document frequency* (TF-IDF) para avaliar o quanto o termo é importante para o artigo em relação ao corpus (BAEZA-YATES e RIBEIRO-NETO, 1999).

Term Frequency (TF) – Trata-se da freqüência do termo no documento, ou seja, quanto maior, mais relevante é o termo para descrever o documento.

Inverse Document Frequency (IDF) – Inverso da freqüência do termo entre todos os documentos da coleção. Um termo que aparece em muitos documentos não pode ser usado como um bom critério de distinção entre eles.

Assim temos:

$$(TF\_IDF)_{i,j} = tf_{i,j} * idf_{i,j}$$

Onde,

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

Sendo $n_{i,j}$ o numero de ocorrências do termo i no documento j e o denominador corresponde ao número de ocorrências de todos os termos em j.

E,

$$idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

Sendo:

|D| o total de documentos no corpus

$|\{d_j : t_i \in d_j\}|$ o número de documentos onde o termo t_i aparece.

As classes Autores, Referências e Keywords são representadas apenas por suas palavras, onde temos, respectivamente, o nome dos autores, os títulos dos trabalhados citados e as palavras-chave de cada artigo.

A classe Citação é apenas uma indicação da existência ou não de citação entre dois artigos.

4.4.3 – Agrupamento

Para o agrupamento é usado o algoritmo *complete-link clustering* disponível na *suite* de bibliotecas Java para a análise linguística da linguagem humana chamada LingPipe distribuída na internet (LINGPIPE, 2009).

O código que calcula a similaridade entre dois documentos foi alterado de forma a permitir o uso das classes identificadas, onde teremos:

Sim = (α * AB) + (β * AT) + (γ * RF) + (δ * CT) + (φ * KW) + (ω * TX)

Embora tenha sido previsto um peso para cada classe, neste trabalho foi adotado peso 1 para todas as medidas.

Para calcular a similaridade das classes Autores, Referências e Keywords, foi usado o coeficiente de *Jaccard* (MANNING *et al.*, 2008), conforme a fórmula a seguir:

Sejam,

A = conjunto de palavras da classe extraídas do artigo A

B = conjunto de palavras da classe extraídas do artigo B

$$Sim(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

O valor da similaridade de cada classe fica no intervalo [0,1], quanto mais próximo de 1 se encontra o valor, mais similares as classes dos dois artigos.

As classes Abstract e Texto, que são representadas por vetores de termos, têm sua similaridade calculada pelo cosseno do ângulo formado entre dois vetores (BAEZA-YATES e RIBEIRO-NETO, 1999).

$$sim(\vec{d_i}, \vec{d_j}) = \cos\left(\widehat{\vec{d_i}\,\vec{d_j}}\right) = \frac{\vec{d_i} \cdot \vec{d_j}}{|\vec{d_i}| \cdot |\vec{d_j}|} = \frac{\sum_{k=1}^{n}(w_{k,i} \cdot w_{k,j})}{\sqrt{\sum_{k=1}^{n}(w_{k,i})^2} \cdot \sqrt{\sum_{k=1}^{n}(w_{k,j})^2}}$$

Sendo $w_{i,j}$ o peso do temo k no documento d.

O valor da similaridade baseado no cosseno também fica no intervalo [0,1] e quanto mais próximo de 1, mais similares são as classes.

A classe Citação terá sua similaridade calculada conforme abaixo:

Onde,

A = título do artigo A

B = todos os títulos das referências bibliográficas do artigo B

$$Sim(CT) = \begin{cases} 1, se\ A \subset B \\ 0, se\ A \not\subset B \end{cases}$$

Ao final do processamento, será gerado um dendrograma com a hierarquia dos grupos.

4.4.4 – Seleção dos grupos

Quando o conjunto de dados é pequeno, uma simples inspeção visual dos resultados é suficiente para analisar os grupos. No entanto, à medida que o conjunto de dados aumenta, fica inviável realizar este processo de forma manual, sendo necessário o uso de técnicas que automatizam a seleção dos grupos.

Em um dendrograma, alguns pontos podem ser facilmente descartados, já que não representam uma informação ou grupo específico. Por exemplo, a raiz, que é composta por todos os documentos da coleção, representa um grupo muito genérico, não somando informação ao usuário. Da mesma forma temos as folhas, onde representam cada documento.

É conhecido que, para agrupamento de documentos, uma boa solução é aquela que organiza os documentos em uma hierarquia de grupos, para que o usuário possa

escolher quais grupos são relevantes e em qual nível da hierarquia. Esta organização tenta passar a noção de que documentos em nós-pais tratam de um assunto mais geral e os nós-filhos de casos específicos do mesmo assunto. Isto é desejado, mas não uma realidade, pois a variedade de grupos que podem ser gerados por conta da característica aleatória do algoritmo de agrupamento, o que pode ser uma vantagem para navegação entre documentos (TREERATPITUK e CALLAN, 2006), nem sempre reflete uma relação pai-filho entre grupos.

Neste trabalho, a seleção dos grupos é realizada através do cálculo das medidas intra-grupo e inter-grupo. Segundo FUNG *et al.* (2006), um bom grupo deve ter alta similaridade intra-grupo e baixa similaridade inter-grupo, ou seja, os documentos dentro do mesmo grupo devem ser semelhantes e com pouco similaridade com os documentos de outros grupos.

Considerando que um grupo de documentos também possui representação no espaço vetorial. Seja um grupo G, com um total de k documentos d, o centróide é um vetor representante do grupo G e é definido como:

$$centróide(G) = \frac{\sum_{i=1}^{k} d_i}{k}$$

Para obter a medida intra-grupo em um determinado grupo G, com k documentos d, neste trabalho, calcula-se a variância entre as similaridades cosseno de cada documento com o centróide de G, conforme a seguinte expressão:

$$intragrupo(G) = 1 - var\big(\cos\big(d_1, centróido(G)\big), \dots, \cos\big(d_k, centróide(G)\big)\big)$$

Considera-se um grupo significante quando a medida intra-grupo se aproxima de 1, pois indica que os documentos de um mesmo grupo são similares, com baixa variabilidade. Ao aproximar-se de 0, o grupo é considerado não significante.

A medida inter-grupo de um grupo G, é definida, neste trabalho, como o valor da similaridade cosseno entre o centróide de G e o centróide do pai de G na hierarquia, de acordo com a expressão:

$$intergrupo(G) = \cos\Big(centróide(G), centróide\big(pai(G)\big)\Big)$$

Neste caso, um valor próximo de 1 para inter-grupo indica alta similaridade entre o grupo pai e filho e, portanto, continuam tratando sobre informações similares.

Baseado nesses valores, são descartadas todas as ligações acima do ponto onde a medida intra-grupo ou inter-grupo seja inferior a 0,5. Dessa forma, ao final da seleção

dos grupos, teremos vários dendrogramas menores, correspondendo aos grupos baseados em linhas de pesquisa.

4.4.5 – Etiquetagem

Uma boa etiqueta deve ser capaz de descrever um grupo e permitir que através da visualização de etiquetas de grupos diferentes, um usuário possa conhecer os diversos contextos presentes nos documentos pertencentes ao grupo.

A tarefa consiste em atribuir um bom descritor para cada grupo. Os descritores de cluster mais comuns são listas de termos ou frases. A lista de termos é freqüentemente mais utilizada que as frases, porque exige que o usuário infira o conceito implícito pelos termos.

A etiquetagem dos grupos foi baseada na heurística usada por CSELLE *et al.* (2007) para anotar os grupos de e-mails.

Cada grupo foi etiquetado separadamente. Primeiro buscou-se encontrar palavras em comum entre as palavras da classe Keywords dos documentos do grupo, e assim todas as palavras que ocorrem em mais da metade do grupo são definidas como etiqueta do grupo. Depois são adotadas como etiqueta do grupo palavras com maior TF-IDF até que se obtenham cinco etiquetas para cada grupo. Para isso, calcula-se, novamente, o valor de TF-IDF das palavras considerando apenas os documentos do grupo. Isso é útil se muitos documentos dentro do grupo contêm a mesma palavra, mas a palavra nunca ocorra fora do grupo. Esta palavra é muito descritiva para o grupo, embora tivesse um baixo valor de TF-IDF para o corpus.

Nessa etapa também são agregadas outras informações aos grupos, como:

- A distribuição em anos dos artigos que compõem o grupo;
- A lista de autores que mais contribuíram com o grupo;
- A lista de publicações mais citadas no grupo.

4.5 – Protótipo

De forma a ilustrar a utilização da ferramenta, é proposta uma interface web, conforme a imagem a seguir:

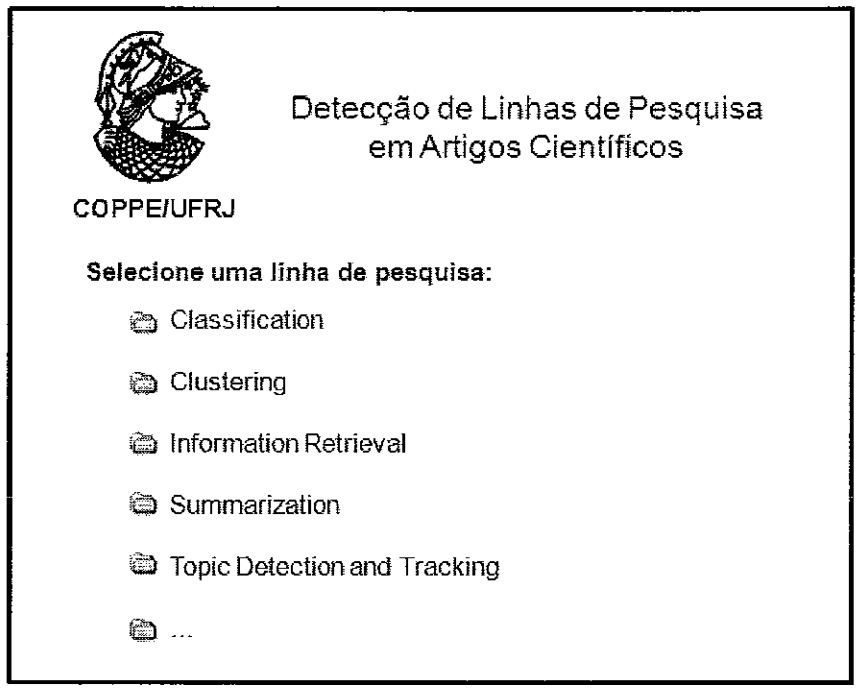

Figura 10 - Tela inicial para consulta das linhas de pesquisa

O usuário, ao selecionar uma linha de pesquisa, nova tela será mostrada contendo o detalhamento da linha de pesquisa.

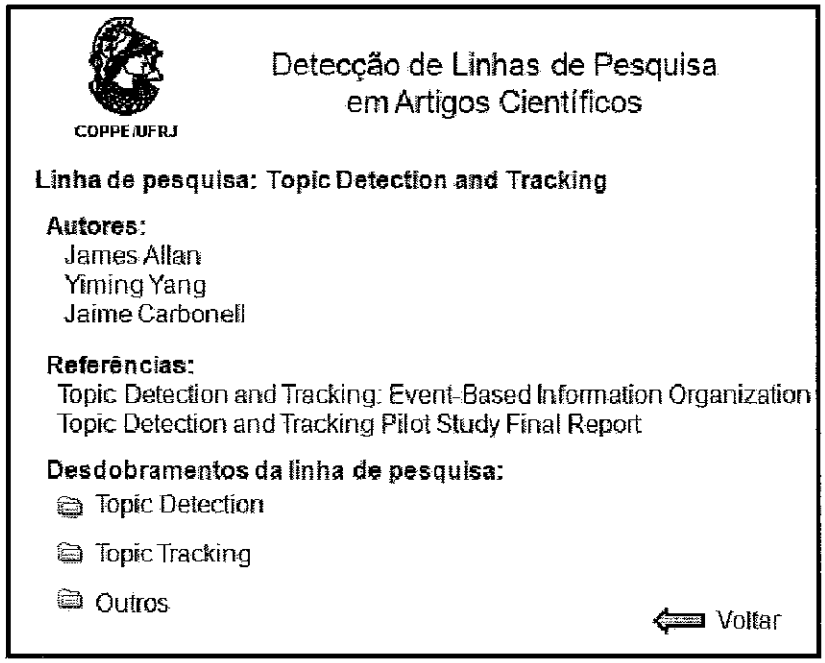

Figura 11 - Tela de detalhamento da linha de pesquisa

A linha de pesquisa TDT usada nas imagens foi gerada manualmente apenas para compor o exemplo.

CAPÍTULO 5

TESTES

Neste capítulo serão descritos os dados utilizados, os testes realizados e como serão analisados os resultados obtidos, baseando-se na metodologia de avaliação proposta, já que o corpus de artigos em uso não foi anotado por especialistas.

5.1 - Dados

Os dados foram armazenados em tabelas de um banco de dados MySQL, conforme o diagrama a seguir:

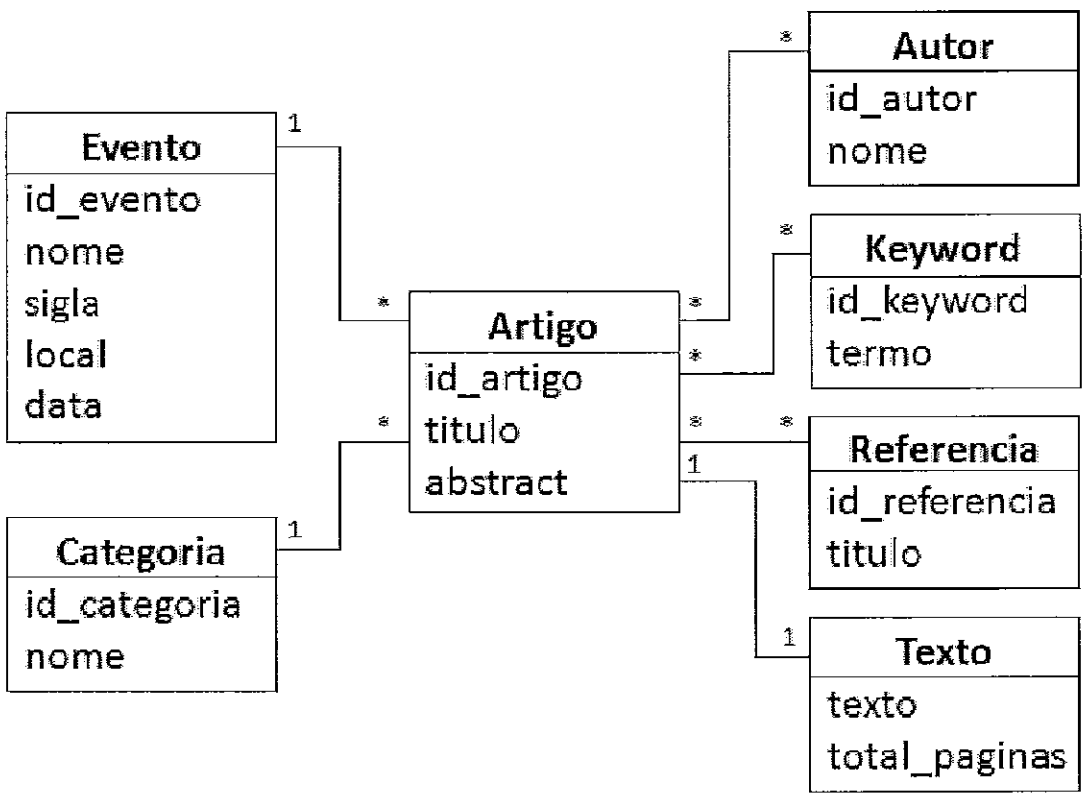

Figura 12 - Diagrama de entidades e relacionamentos do banco de dados usado para armazenar o corpus

Inicialmente, foram preenchidas todas as informações relativas aos artigos científicos que compõem as tabelas Artigo, Evento e Categoria. Em seguida através da extração textual das informações contidas nos arquivos PDF dos artigos foi preenchida a tabela Texto.

A partir da tabela Texto, através da aplicação de uma série de expressões regulares, foram identificadas as principais seções do documento. Após, o texto foi dividido em partes de forma a permitir o processamento das classes conforme explicado no capítulo anterior. Estas partes foram salvas nas tabelas Referencia, Keyword e Autor.

Após o fim da extração dos dados, os mesmos passaram novamente por uma série de expressões regulares e validação humana de forma a filtrar e identificar artigos que não se encaixam em todos os critérios necessários a sua utilização neste trabalho.

Os artigos que apresentaram problema ou não possuíam todas as sessões que são necessárias para a composição das classes, sendo estas: título, autor(es), *keyword(s)*, *abstract*, texto e referências bibliográficas, foram descartados.

Os problemas encontrados e a quantidade de artigos descartados seguem abaixo:

503 artigos não possuíam informações textuais dentro do PDF. Ao tentar extrair os dados, o mesmo só apresentava símbolos.

229 artigos que só possuíam o resumo, geralmente apresentados na sessão de pôsteres ou demonstração de ferramentas nos congressos.

165 artigos não possuíam a sessão *abstract*.

440 artigos não possuíam *keywords*.

Ao final, apenas 58% do corpus pode ser aproveitado, restando 1.858 artigos completos para serem utilizados nos testes.

5.1.1 – Estatística do corpus

Algumas informações podem ser extraídas de forma a se conhecer melhor o corpus, como: a quantidade de artigos por ano, a quantidade de artigos por total de páginas e a quantidade autores, *keywords* e citações únicas.

Tabela 2 - Total de artigos por ano

Ano	Total de Artigos
1999	50
2000	63
2001	71
2002	114
2003	157
2004	214
2005	214
2006	241
2007	356
2008	378
Total	**1.858**

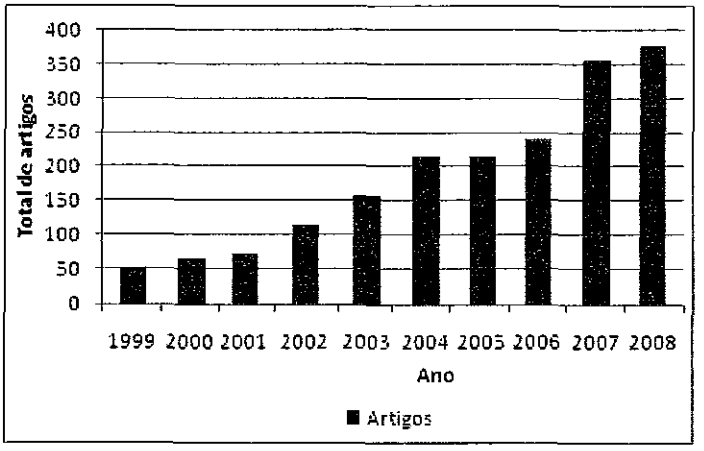

Figura 13 - Gráfico: Ano x Total de artigos

Assim como as notícias utilizadas no estudo de TDT, os artigos científicos possuem tamanho variado.

Tabela 3 - Total de artigos por quantidade de páginas

Qtd. Páginas	Total de Artigos
1	51
2	389
3	44
4	23
5	19
6	212
7	77
8	447
9	142
10	270
11	13
12	131
13	17
14	23
Total	**1.858**

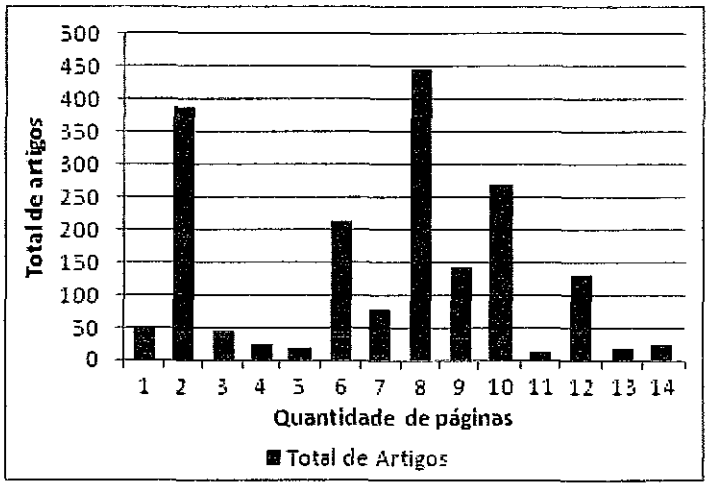

Figura 14 - Gráfico: Quantidade de páginas x Total de artigos

Temos no corpus 3.582 autores diferentes e cada artigo tem em média 3 autores. São 3.371 *keywords* únicos, onde em média cada artigo possui aproximadamente 3 *keywords*. Além disso, são identificadas 29.431 referências, sendo em média 18 referências por artigo.

Tabela 4 - Quantidade de dados únicos e média do corpus

	Quantidade única	Média por artigo
Autor	3.582	3
Keyword	3.371	3
Referência	29.431	18

5.2 – Metodologia de avaliação

Segundo ANDREWS e FOX (2002) algoritmos de agrupamento têm sido avaliados de muitas maneiras, mas há pouco consenso sobre qual é a melhor forma de avaliação. A escolha dos métodos de avaliação freqüentemente depende do domínio em que a investigação está sendo conduzida.

No presente trabalho é adotada a avaliação proposta em NIST (2008) para a tarefa de detecção e rastreamento de tópicos do TDT.

Seu desempenho é avaliado pela medição do quão bem os artigos pertencentes a cada um dos tópicos previamente anotado casam com os artigos que o sistema marcou como sendo correspondente a estes tópicos. A melhor relação será a que possuir um menor custo de detecção, onde o custo de detecção é definido por:

$$C_{Det} = C_{Miss} * P_{Miss} * P_{Topic} + C_{FalseAlarm} * P_{FalseAlarm} * (1 - P_{Topic})$$

Onde,

C_{Miss} é o custo de um erro, ou seja, artigos que pertencem ao tópico, mas não foram marcados. Nos testes consideramos igual a 1.

P_{Miss} é a probabilidade da ocorrência de um erro. É calculada pelo número de artigos que pertencem ao tópico, mas não foram marcados, dividido pelo número total de artigos que pertencem ao tópico.

P_{Topic} é probabilidade a priori de um artigo estar em um dado tópico. Foi adotado $P_{Topic} = 0.02$, conforme indicado nos testes de TDT (NIST, 2008).

$C_{FalseAlarm}$ é o custo de um alarme falso. Nos testes consideramos igual a 1.

$P_{FalseAlarme}$ é a probabilidade da ocorrência de um alarme falso. É calculado pelo número de artigos que não pertencem ao tópico, mas foram marcados como pertencentes, dividido pelo número artigos que não pertencem ao tópico.

Devido à falta de anotação no corpus com relação às linhas de pesquisa ou tópicos, os *keywords* presentes em cada artigo são utilizados como anotação. É sabido que embora os *keywords* representem uma classificação para o artigo, não existe uma

padronização para sua criação, podendo existir vários *keywords* com o mesmo significado, podendo afetar diretamente os resultados.

Todos os *keywords* comuns a mais de 2 artigos foram considerados como um grupo, gerando ao todo 281 grupos.

5.3 – Análise dos resultados

Utilizando um processamento hierárquico para o agrupamento dos artigos, foram obtidas as seguintes relações de quantidade de grupos por similaridade:

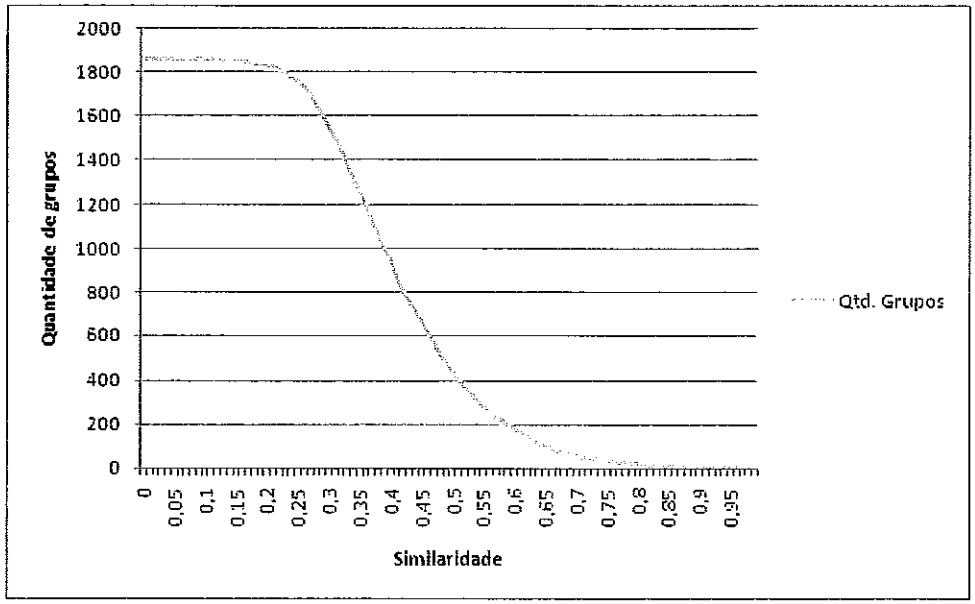

Figura 15 - Gráfico: similaridade x quantidade de grupos

A figura 15 demonstra a quantidade de grupos de artigos presentes em cortes horizontais no dendograma obtido.

De forma a comparar o resultado, foram feitos diversos processamentos e calculados seus custos, conforme apresentado nos gráficos a seguir.

Devido à inexistência de um relacionamento entre os grupos criados a partir da anotação e os grupos criados pelos processamentos, consideramos esta relação por sendo a que detiver o menor custo de detecção C_{Det} entre um grupo e outro.

Os gráficos a seguir foram construídos a partir dos pontos do custo da detecção, custo do erro e custo do alarme falso, onde são plotados como linhas conectadas.

Primeiro foram efetuados processamentos separados para cada classe, Abstract, Referências, Keywords e Texto e calculados os custos, conforme mostrado no gráfico a

seguir. Não foram utilizados os resultados das classes Autores e Citação, por não apresentarem grupos condizentes com as informações anotadas.

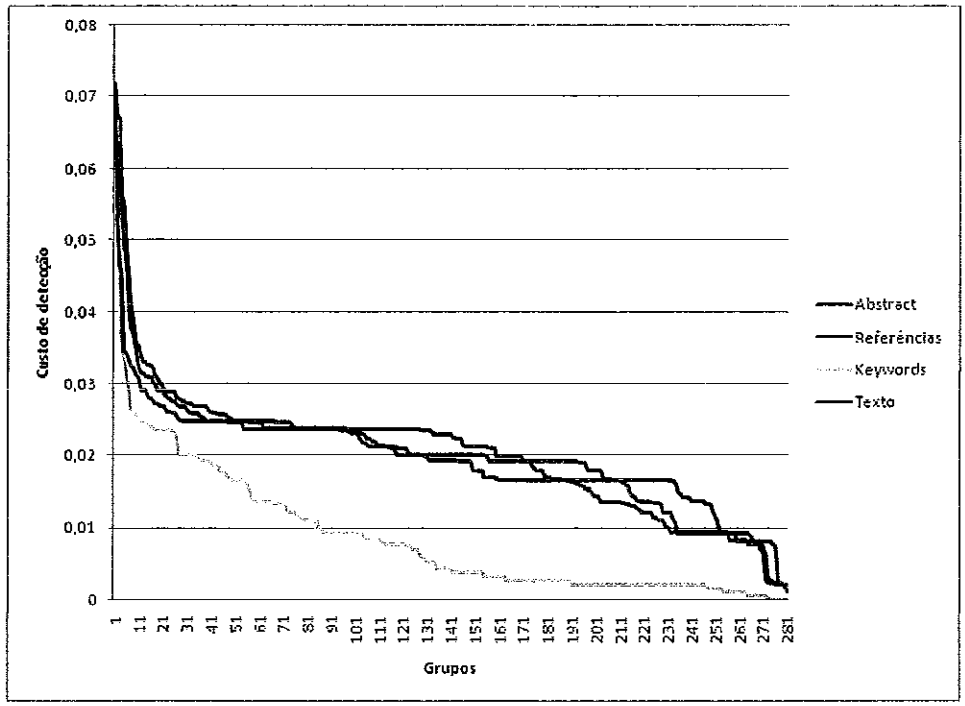

Figura 16 - Gráfico: Custo de detecção (C_{Det}) por classes

As classes Abstract, Referências e Keywords apresentaram resultado bastante semelhante. A classe Keywords apresentou um custo de detecção bem menor que as demais classes.

De forma a avaliar separadamente os custos do erro e dos alarmes falsos, o resultado demonstrado na figura 16 foi desmembrado, criando dois novos gráficos, onde na figura 17 observamos o custo do erro para os grupos anotados, e na figura 18 observamos o custo dos alarmes falsos para os grupos anotados.

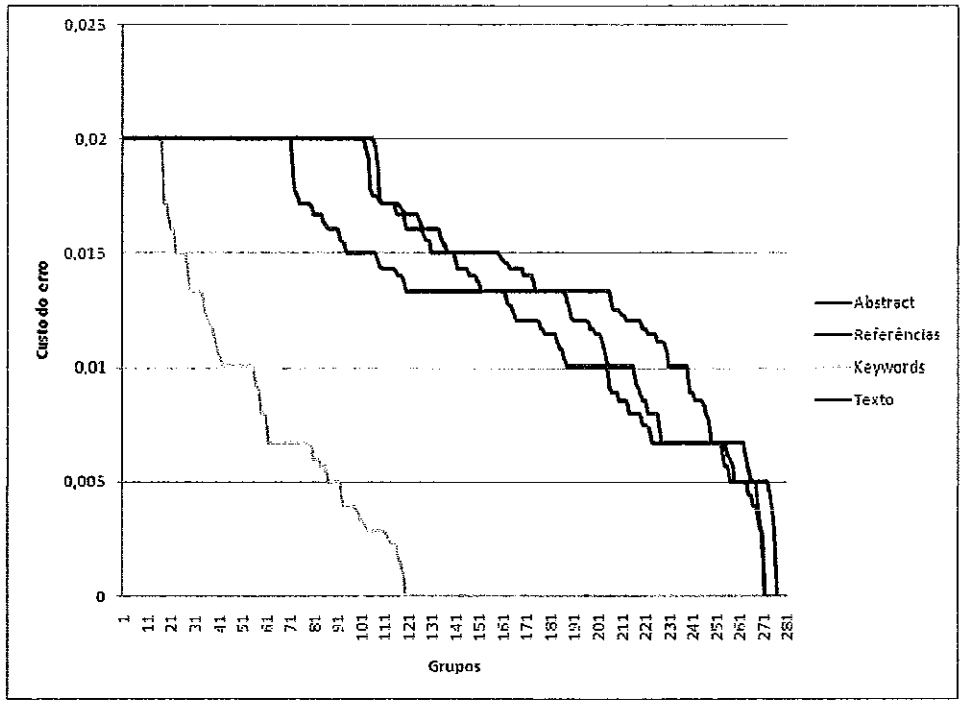

Figura 17 - Gráfico: Custo do erro por classe

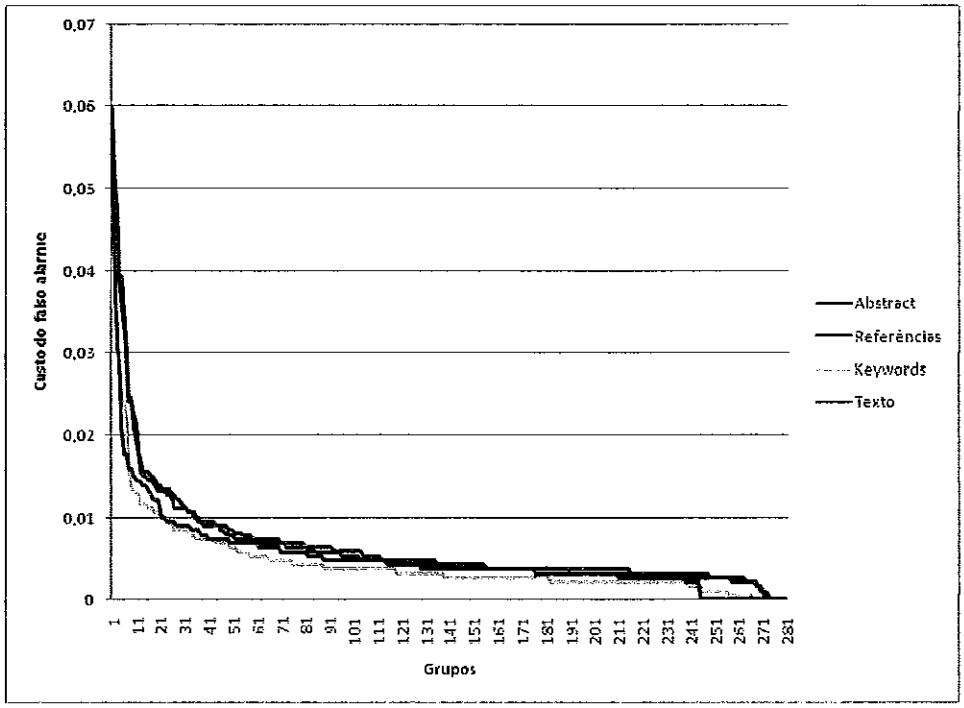

Figura 18 - Gráfico: Custo do falso alarme por classe

Como podemos concluir, a classe Keywords obteve um menor custo de detecção em função da sua baixa taxa de erros, visto que todas as classes apresentaram um custo semelhante de alarmes falsos.

Foram efetuados processamentos usando todas as classes. A figura 19 demonstra a comparação dos resultados dos processamentos entre a classe texto, todas as classes e todas as classes com exceção da classe Keywords.

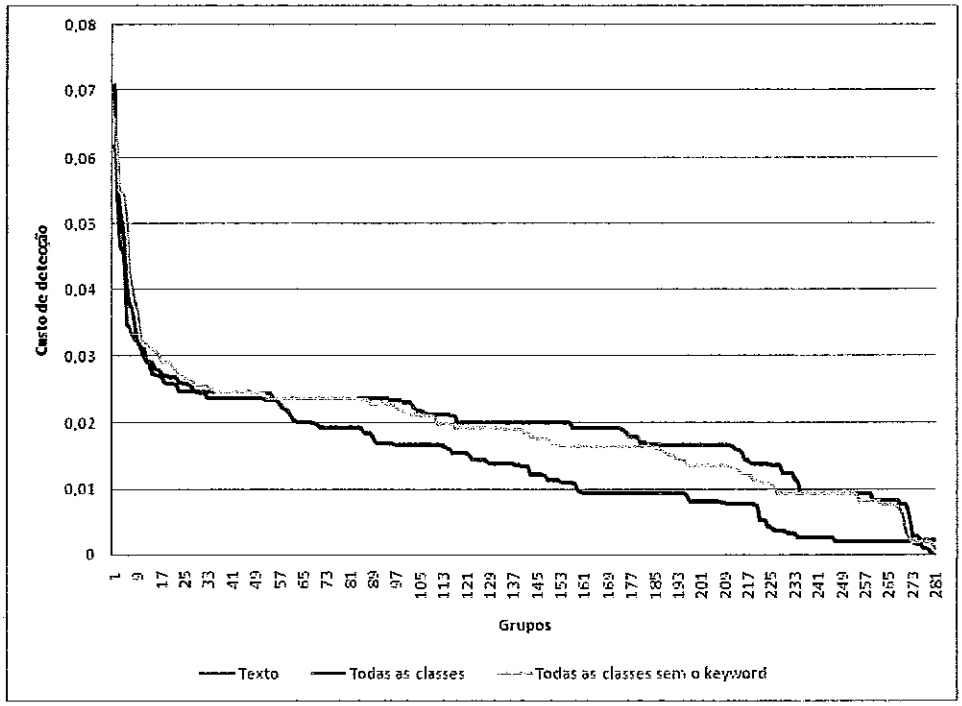

Figura 19 - Gráfico: Custo de detecção (C_{Det})

Pelo gráfico, notamos que ocorre um ganho do processamento realizado com o uso de todas as classes. Mesmo o processamento usando todas as classes sem o *keyword* apresentou um resultado melhor que o processamento usando apenas o texto dos artigos. Como podemos ver nos próximos gráficos, esse ganho ocorreu devido a um melhor desempenho em identificar os artigos do tópico, marcando poucos falsos positivos, conforme a figura 20.

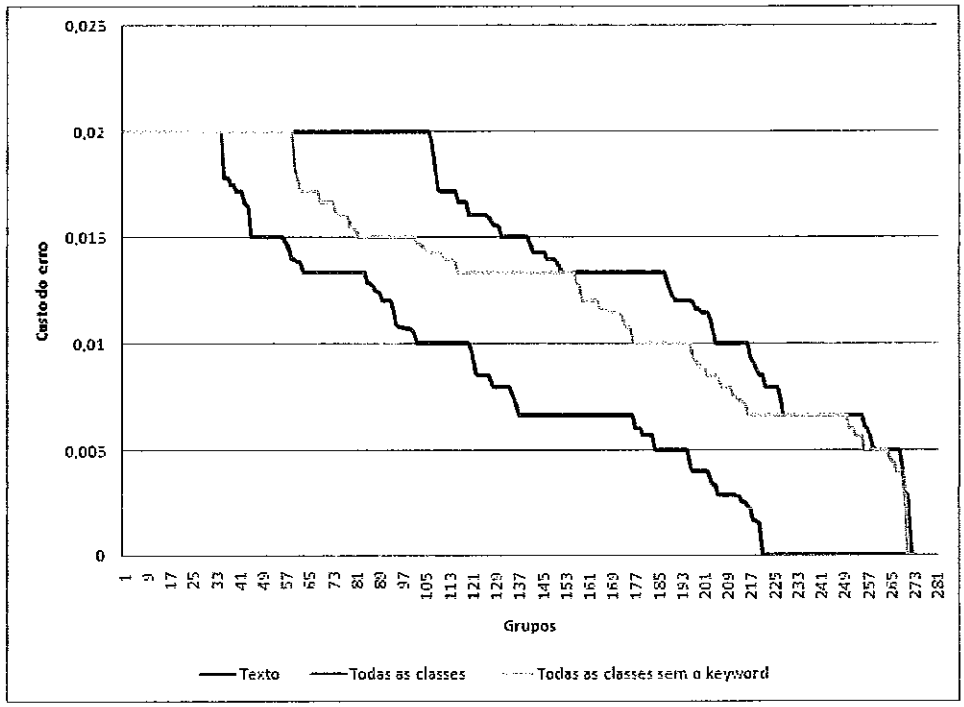

Figura 20 - Gráfico: Custo de erro

O processamento com o uso de todas as classes apresentou o custo de erro bem menor que os demais processamentos.

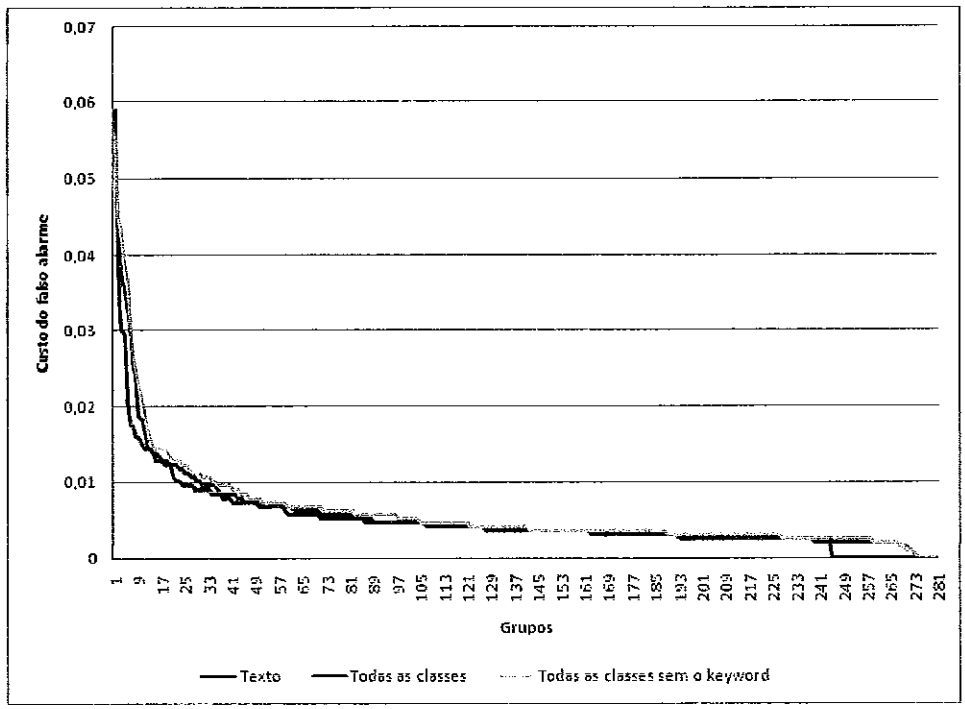

Figura 21 - Gráfico: Custo do falso alarme

O custo do falso alarme foi bem semelhante em todos os processamentos.

Segue na tabela abaixo o valor do maior custo, menor custo e custo médio de detecção, erro e alarme falso para os processamentos usando apenas o texto, todas as classes e todas as classes sem *keyword*.

Tabela 5 - Custos do processamento

	Maior Custo			Menor Custo			Média do Custo		
	Detecção	**Erro**	**Alarme Falso**	**Detecção**	**Erro**	**Alarme Falso**	**Detecção**	**Erro**	**Alarme Falso**
Texto	0,071	0,02	0,059	0,0021	0	0	0,019	0,014	0,005
Todas Classes	0,061	0,02	0,053	0,00052	0	0	0,014	0,008	0,005
Todas Classes sem keyword	0,065	0,02	0,055	0,00105	0	0	0,018	0,012	0,006

Em FENG e ALLAN (2005) são avaliados alguns sistemas de TDT para tarefa de detecção de tópico. O sistema que apresentou melhor resultado, criado pela *Netherlands Organization for Applied Scientific Research* (TNO) teve custo de erro de 0,0196 e custo do alarme falso de 0,0042. Mas no artigo não foi especificado se esse foi o maior, menor ou custo médio do processamento. Mesmo assim obtivemos resultados tão competitivos quanto o apresentado no artigo de TDT. Considerando o processamento com todas as classes, 87,5% dos testes dos 281 grupos anotados tiveram custo de erro menor que o apresentado pelo sistema TNO, e 62,7% do custo de alarme falso.

Analisando os alarmes falsos gerados nos grupos pelo processamento com todas as classes, vemos que são artigos que apresentam *keywords* com assuntos semelhantes, por exemplo, para o grupo anotado com o *keyword* "*Text classification*", nos alarmes falsos encontramos "*Automatic classification*", "*Classification*", "*Classifying with domain knowledge*", "*Collective classification*", "*Hierarchical classification*" e "*Incremental classification mining*".

Ainda usando como exemplo o grupo com o *keyword* "*Text classification*", o grupo gerado usando todas as classes no processamento, apresenta as características a seguir:

1. As etiquetas geradas:

Categories	Classification	Information
Results	Text	

2. Autores mais citados:

Bing Liu	Shantanu Godbole
Dell Zhang	Shourya Roy
Gui-Rong Xue	Wee Sun Lee
Qiang Yang	Yong Yu

3. Referências mais usadas:

A comparative study on feature selection in text categorization

A re-examination of text categorization methods

Inductive learning algorithms and representations for text categorization

RCV1: A new benchmark collection for text categorization research

Text categorization with support vector machines: Learning with many relevant features

4. Quantidade de artigos por ano:

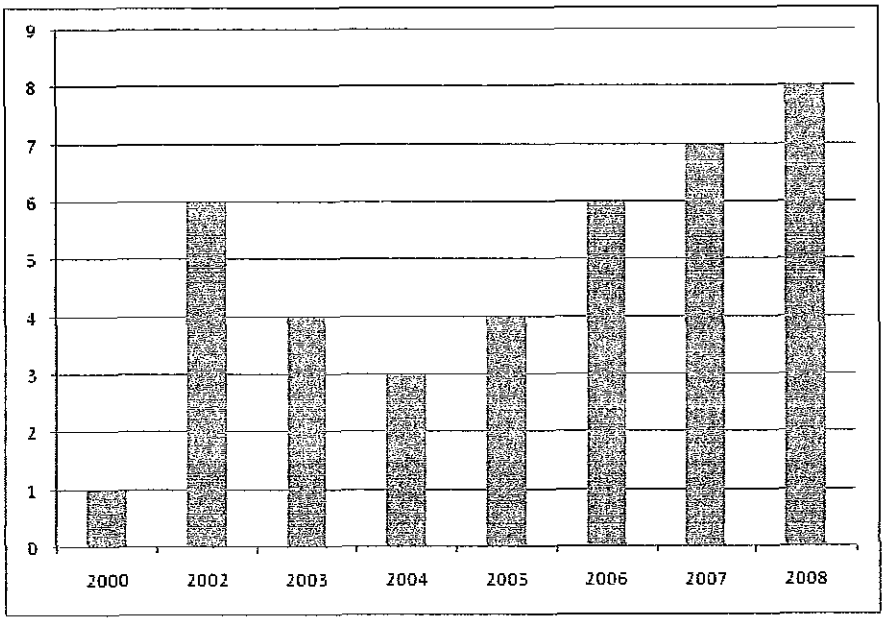

Figura 22 - Gráfico: Evolução do grupo por ano

Os títulos e autores pertencentes aos grupos usados como exemplo estão descritos no Anexo 1, organizados por congresso.

As etiquetas propostas para os grupos gerados a partir do processamento de todas as classes estão organizadas no Anexo 2.

CAPÍTULO 6
CONCLUSÃO

Este trabalho discutiu a identificação de linhas de pesquisa usando publicações científicas, de forma a apresentar um agrupamento dessas publicações de acordo com as linhas de pesquisa identificadas.

Com base em técnicas de TDT, inicialmente elaboradas para trabalhar com notícias, foi proposta uma adaptação para essas técnicas em artigos científicos, usando algoritmos para agrupamento hierárquico de documentos.

Desta forma, foram apresentados todos os passos para implementação da ferramenta proposta e para criação do corpus que foi utilizado nos testes.

Foram apresentados também os resultados encontrados e, de acordo com a avaliação proposta, pode-se confirmar a viabilidade da ferramenta.

Devido a grandes volumes de dados não estruturados gerados diariamente, a importância de ferramentas que organizem documentos e ajudem os usuários a encontrar informação desejada de forma mais fácil e rápida continuará a crescer.

Com o desenvolvimento deste trabalho podemos, ressaltar as seguintes contribuições:

- Adaptação da técnica escolhida de TDT para realização de detecção de linhas de pesquisa em artigos científicos.

- Identificação de classes que caracterizam um artigo científico.

- Criação de um corpus de artigos científicos da área de banco de dados. Os arquivos PDF foram convertidos em texto plano, suas sessões foram separadas e armazenados em um banco de dados, facilitando sua utilização.

6.1 – Trabalhos futuros

Uma possível continuidade do trabalho realizado, seria a implementação da tarefa de TDT para detectar a primeira história, de forma a notificar o surgimento de

uma nova linha de pesquisa ou discussão surgir, facilitando professores e pesquisadores a se manterem atualizados.

A implementação de uma *interface web* para facilitar o uso dos resultados.

A melhoria dos resultados do agrupamento, com base na ampla gama de trabalhos existentes de TDT para notícias. Além da exploração de outras técnicas de agrupamento de documentos e o teste com outros pesos no processamento das classes de forma a melhorar o resultado.

O teste de outras medidas de similaridade, seleção de características e técnicas de etiquetagem de grupo.

A inclusão de uma etapa no processamento para a realização de sumarização dos textos do grupo, de forma a apresentar ao usuário um pequeno texto que informe do que trata o grupo. Algumas vezes, poucas palavras (etiquetas) não indicam de forma precisa o tema do grupo.

O desenvolvimento de uma ferramenta que permita a anotação do corpus de artigos científicos por especialistas, facilitando a execução da avaliação dos resultados.

REFERÊNCIAS BIBLIOGRÁFICAS

ACM: Press Release, 2007, *ACM Digital Library now exceeds one million entries.* Disponível em: <http://campus.acm.org/public/pressroom/press_releases/2_2007/milliondoc.cfm>. Acesso em: 05/08/2008.

ALLAN, J., CARBONELL, J., DODDINGTON, G., *et al.*, 1998, "Topic Detection and Tracking Pilot Study: Final Report". In: *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pp. 194-218, San Francisco, CA, Morgan Kaufmann Publishers.

ALLAN, J., 2002a, "Introduction to Topic Detection and Tracking". In. James Allan, *Topic Detection and Tracking: Event-Based Information Organization*, capítulo 1, Massachusetts, USA, Kluwer Academic Publishers.

ALLAN, J., 2002b, "Detection as Multi-Topic Tracking". In: *Information Retrieval*, v.5, pp. 139-157, Massachusetts, USA, Kluwer Academic Publishers.

ANDREWS, N. O., FOX, E. A., 2007, *Recent Developments in Document Clustering.* In: Technical Report TR-07-35, Computer Science, Virginia Tech.

BAEZA-YATES, R., RIBEIRO-NETO, B., 1999, *Modern Information Retrieval.* New York, Addison Wesley.

BRANTS, T., CHEN, F., FARAHAT, A., 2003, "A System for New Event Detection". In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 330-337, Toronto, Canada.

CHAN, S., PON, R. K., CARDENAS, A. F., 2006, "Visualization and Clustering of Author Social Networks", *Distributed Multimedia Systems Conference*, pp.174-180, Grand Canyon, Arizona.

CIERI, C., STRASSEL, S., GRAFF, D., *et al.*, 2002, "Corpora for Topic Detection and Tracking", In. James Allan, *Topic Detection and Tracking: Event-Based Information Organization*, capítulo 3, Massachusetts, USA, Kluwer Academic Publishers.

CNPQ, 2009, *Censos - Perguntas freqüentes*. Disponível em: <http://dgp.cnpq.br /censos/perguntas/perguntas.htm>. Acesso em: 20/06/2009.

CSELLE, G., ALBRECHT, K., WATTENHOFER, R., 2007, "BuzzTrack: Topic Detection and Tracking in Email". In: *Proceedings of the 12th International Conference on Intelligent User Interfaces*, pp. 190-197, Honolulu, Hawaii, USA.

DARPA Information Processing Technology Office, 2008, *Translingual Information Detection, Extraction and Summarization (TIDES)*. Disponível em: <http://www.darpa.mil/ipto/programs/tides/ >. Acesso em: 27/07/2008.

FENG, A., ALLAN, J., 2005, "Hierarchical Topic Detection in TDT-2004". *CIIR Technical Report*, n. 389, University of Massachusetts.

FENG, A., ALLAN, J., 2007, "Finding and Linking Incidents in News". In: *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pp. 821-830, Lisbon, Portugal.

FISCUS, J. G., DODDINGTON, G. R., 2002, "Topic Detection and Tracking Evaluation Overview", In. James Allan, *Topic Detection and Tracking: Event-Based Information Organization*, capítulo 2, Massachusetts, USA, Kluwer Academic Publishers.

FRANZ, M., WARD, T., MCCARLEY, J. S., *et al.*, 2001, "Unsupervised and Supervised Clustering for Topic Tracking". In: *Proceedings of the 24th Annual*

International ACM SIGIR Conference on Research and development in Information Retrieval, pp.310-317, New Orleans, Louisiana, United States.

FRANZ, M., XU, J. M., 2007, "Story Segmentation of Broadcast News in Arabic, Chinese and English Using Multi-Window Features". In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 703-704, Amsterdam.

FREY, D., GUPTA, R., KHANDELWAL, V., *et al.*, 2001, "Monitoring the News: a TDT demonstration system". In: *Proceedings of the first international conference on Human language technology research*, pp. 1-5, San Diego.

FUKUMOTO, F., SUZUKI, Y., 2007, "Topic Tracking Based on Bilingual Comparable Corpora and Semisupervised Clustering", *ACM Transactions on Asian Language Information Processing*, v. 6, n. 3 (Nov), artigo 11.

FUNG, B. C. M., WANG, K., ESTER, M., 2006, "Hierarchical Document Clustering". In: John Wang, *Encyclopedia of Data Warehousing and Mining*, Idea Group.

GUHA, S., RASTOGI, R., SHIM, K., 1999, "Rock: a Robust Clustering Algorithm for Categorical Attributes". In: *Proceedings of the 15th International Conference on Data Engineering*, pp. 512-521, Washington, USA.

HALKIDI, M., BATISTAKIS, Y., VAZIRGIANNIS, M., 2001, "On Clustering Validation Techniques". *Journal of Intelligent Information Systems*, v. 17, n. 2-3 (Dez), p. 107-145.

HIEW, L., 2006, Assisted Detection of Duplicate Bug Reports, B.Sc. thesis, The University of British Columbia.

JAIN, A. K., DUBES, R. C., 1988. "Algorithms for Clustering Data". *Prentice-Hall Advanced Reference Series*. Prentice-Hall, Inc., Upper Saddle River, NJ.

JAIN, A. K., MURTY, M. N., FLYNN, P. J., 1999, "Data Clustering: A Review", *ACM Computing Surveys (CSUR)*, v. 31, n. 3 (Set), pp. 264-323.

JIN, Y., MYAENG, S. H., JUNG, Y., 2007, "Use of Place Information for Improved Event Tracking", *Information Processing and Management: an International Journal*, v. 43, n. 2 (Mar), pp. 365-378.

KANUNGO, T., MOUNT, D. M., NETANYAHU, N S., *et al.*, 2002, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24 , n. 7 (Jul), pp. 881-892.

KUMARAN, G., ALLAN, J., MCCALLUM, A., 2004. , "Classification Models for New Event Detection", CIIR Technical Report. Disponível em: <http http://ciir-publications.cs.umass.edu/pub/web/getpdf.php?id=511>. Acesso em: 26/08/2008.

LDC - Linguistic Data Consortium, 1997. "The Topic Detection and Tracking (TDT) Pilot Study Evaluation Plan", versão 2.8. pp 1-8. Disponível em: <http://projects.ldc.upenn.edu/TDT-Pilot/TDT.Eval_Plan.97.v2.8.ps>.

LDC - Linguistic Data Consortium, 2008, *Topic Detection and Tracking*. Disponível em: <http://projects.ldc.upenn.edu/TDT/>. Acesso em: 02/08/2008.

LEYDESDORFF, L., RAFOLS, I., 2009, "A Global Map of Science Based on the ISI Subject Categories", *Journal of the American Society for Information Science and Technology*, v. 60, n. 2, pp. 348-362.

LIN, F., LIANG, C., 2008, "Storyline-Based Summarization for News Topic Retrospection", *Decision Support Systems*, v. 45, n. 3 (Jun), pp. 473-490.

LINGPIPE, 2009, *Topic LingPipe: Clustering Tutorial*. Disponível em: <http://alias-i.com/lingpipe/demos/tutorial/cluster/read-me.html>. Acesso em: 06/06/2009.

LOWE, S. A., 1999. "The Beta-Binomial Mixture Model and Its Application to TDT Tracking and Detection". In: *Proceedings of the DARPA Broadcast News Workshop*.

MANNING, C.D.; RAGHAVAN, P.; SCHÜTZE, H., 2008, *An Introduction to Information Retrieval*. Cambridge, Cambridge University Press.

MAKKONEN, J., AHONEN-MYKA, H., SALMENKIVI, M., 2002, "Applying Semantic Classes in Event Detection and Tracking". In: *Proceedings International Conference on Natural Language Processing*, pp. 175-183, Mumbai, India.

NALLAPATI, R., 2003, "Semantic Language Models for Topic Detection and Tracking". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pp.1-6, Canadá.

NEWMAN, M. E. J., 2006, "Finding Community Structure in Networks Using the Eigenvectors of Matrices", *Journal of Physical Review E*, 74(3), 36104.

NIST Speech Group Website, 2008, *Topic Detection and Tracking Evaluation.* Disponível em: <http://www.nist.gov/speech/tests/tdt/>. Acesso em: 02/08/2008.

O'LEARY, D. E., 1997, "The Internet, Intranets, and the AI Renaissance". In: *Computer*, v. 30, IEEE Computer Society Press, pp. 71-78, Los Alamitos, CA, USA.

PORTER, M. F., 1997, "An Algorithm for Suffix Stripping", In: *Readings in Information Retrieval*, Morgan Kaufmann Publishers Inc., pp. 313-316.

SAHOO, N., CALLAN, J., KRISHNAN, R., *et al.*, 2006, "Incremental Hierarchical Clustering of Ttext Documents". In: *Proceedings of the 15th ACM international conference on Information and knowledge management*', pp. 357-366, New York, USA.

SALTON, G., WONG, A., YANG, C. S., 1975, "A vector space model for automatic indexing", *Communications of the ACM*, v. 18, n. 11, pp. 613-620.

SBC, 2006, "Grandes Desafios da Pesquisa em Computação no Brasil – 2006 - 2016". Disponível em: < http://www.sbc.org.br/index.php?language=1&content=downloads&id=272>. Acesso em: 28/08/2008.

SLONIM, N., FRIEDMAN, N., TISHBY, N., 2002, "Unsupervised Document Classification Using Sequential Information Maximization". In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 129-136, New York, USA.

STOLCKE, A., SHRIBERG, E., HAKKANI-TUR, D., *et al.*, 1999, "Combining Words and Speech Prosody for Automatic Topic Segmentation". In: *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pp. 61-64.

TREERATPITUK, P., CALLAN, J., 2006, "Automatically Labeling Hierarchical Clusters". In: *Proceedings of the 2006 international conference on Digital government research*, pp. 167-176, New York, USA,.

THO, Q. T., HUI, S. C., FONG, A. C. M., 2003, "A Web Mining Approach for Finding Expertise in Research Areas". In: *Proceedings of the 2003 International Conference on Cyberworlds*, pp. 310, Washington, USA.

WAYNE, C. L., 2000, "Multilingual Topic Detection and Tracking: Successful Research Enabled by Corpora and Evaluation", Disponível em: <http://www.nist.gov/speech/tests/tdt/Wayne-LREC2000.ps>.

YANG, Y., AULT, T., PIERCE, T., *et al.*, 2000, "Improving Text Categorization Methods for Event Tracking". In: *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 65-72, Atenas, Grécia.

YANG, Y., CARBONELL, J., BROWN, R., *et al.*, 2002, "Multi-Strategy Learning for Topic Detection and Tracking: a Joint Report of CMU Approaches to Multilingual TDT", In. James Allan, *Topic Detection and Tracking: Event-Based Information Organization*, capítulo 5, Massachusetts, USA, Kluwer Academic Publishers.

YANG, Y., PIERCE, T., CARBONELL, J., 1998, "A study of retrospective and on-line event detection". In: *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 28-36, Melbourne, Austrália.

ZAIANE, O. R., Foss, A., Lee, C., *et al.*, 2002, "On Data Clustering Analysis: Scalability, Constraints and Validation". In: *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp. 28-39, London, UK.

ZHAO, Y., KARYPIS, G., 2002, "Evaluation of Hierarchical Clustering Algorithms for Document Datasets. In: *Proceedings of the eleventh international conference on Information and knowledge management*', pp. 515-524, New York, USA.

ANEXO 1

GRUPO DE EXEMPLO

Todos os artigos do grupo usado como exemplo gerado com o uso das classes no processamento, a partir do *keyword* "Text classification".

SIGIR 2000

- Hierarchical classification of Web content
  Susan Dumais, Hao Chen

SIGIR 2002

- Automatic classification in product catalogs
  Ben Wolin
- Bayesian online classifiers for text classification and filtering
  Kian Ming Adam Chai, Hai Leong Chieu, Hwee Tou Ng
- Text genre classification with genre-revealing and subject-revealing features
  Yong-Bae Lee, Sung Hyon Myaeng

SIGIR 2003

- Document-self expansion for text categorization
  Yuen-Hsien Tseng, Da-Wei Juang
- Question classification using support vector machines
  Dell Zhang, Wee Sun Lee
- Robustness of regularized linear classification methods in text categorization
  Jian Zhang, Yiming Yang
- Rule-based word clustering for text classification
  Hui Han, Eren Manavoglu, C. Lee Giles, Hongyuan Zha

SIGIR 2004

- Effectiveness of web page classification on finding list answers
  Hui Yang, Tat-Seng Chua
- Text classification and named entities for new event detection
  Giridhar Kumaran, James Allan

- Web-page classification through summarization
  Dou Shen, Zheng Chen, Qiang Yang, Hua-Jun Zeng, Benyu Zhang, Yuchang Lu, Wei-Ying Ma

SIGIR 2005

- On the collective classification of email "speech acts"
  Vitor R. Carvalho, William W. Cohen
- Text classification with kernels on the multinomial manifold
  Dell Zhang, Xi Chen, Wee Sun Lee
- Using dragpushing to refine centroid text classifiers
  Songbo Tan, Xueqi Cheng, Bin Wang, Hongbo Xu, Moustafa M. Ghanem, Yike Guo

SIGIR 2006

- Constructing informative prior distributions from domain knowledge in text classification
  Aynur Dayanik, David D. Lewis, David Madigan, Vladimir Menkov, Alexander Genkin
- Graph-based text classification: learn from your neighbors
  Ralitsa Angelova, Gerhard Weikum
- The effect of OCR errors on stylistic text classification
  Sterling Stuart Stein, Shlomo Argamon, Ophir Frieder

SIGIR 2007

- Improving text classification for oral history archives with temporal domain knowledge
  J. Scott Olsson, Douglas W. Oard
- Semantic text classification of disease reporting
  Yi Zhang, Bing Liu
- Text categorization for streams
  D. L. Thomas, W. J. Teahan
- Using clustering to enhance text classification
  Antonia Kyriakopoulou, Theodore Kalamboukis

- What emotions do news articles trigger in their readers?
  Kevin Hsin-Yih Lin, Changhua Yang, Hsin-Hsi Chen

SIGIR 2008

- Classifiers without borders: incorporating fielded text from neighboring web pages
  Xiaoguang Qi, Brian D. Davison
- Deep classification in large-scale text hierarchies
  Gui-Rong Xue, Dikan Xing, Qiang Yang, Yong Yu
- Improving text classification accuracy using topic modeling over an additional corpus
  Somnath Banerjee
- Topic-bridged PLSA for cross-domain text classification
  Gui-Rong Xue, Wenyuan Dai, Qiang Yang, Yong Yu

SIGKDD 2002

- A parallel learning algorithm for text classification
  Canasai Kruengkrai, Chuleerat Jaruskulchai
- A refinement approach to handling model misfit in text categorization
  Haoran Wu, Tong Heng Phang, Bing Liu, Xiaoli Li
- Incremental context mining for adaptive document classification
  Rey-Long Liu, Yun-Ling Lu

SIGKDD 2005

- On the use of linear programming for unsupervised text classification
  Mark Sandler

SIGKDD 2006

- Acclimatizing taxonomic semantics for hierarchical content classification from semantics to data-driven taxonomy
  Lei Tang, Jianping Zhang, Huan Liu
- Extracting key-substring-group features for text classification
  Dell Zhang, Wee Sun Lee

- Reducing the human overhead in text categorization
  Arnd Christian König, Eric Brill

SIGKDD 2007

- Feature selection methods for text classification
  Anirban Dasgupta, Petros Drineas, Boulos Harb, Vanja Josifovski, Michael W. Mahoney
- Semi-supervised classification with hybrid generative/discriminative methods
  Gregory Druck, Chris Pal, Andrew McCallum, Xiaojin Zhu

SIGKDD 2008

- An integrated system for automatic customer satisfaction analysis in the services industry
  Shantanu Godbole, Shourya Roy
- Building semantic kernels for text classification using Wikipedia
  Pu Wang, Carlotta Domeniconi
- Scaling up text classification for large file systems
  George Forman, Shyamsundar Rajaram
- Text classification, business intelligence, and interactivity: automating C-Sat analysis for services industry
  Shantanu Godbole, Shourya Roy

ANEXO 2

ETIQUETAS DOS GRUPOS

Segue abaixo a lista das etiquetas propostas para os grupos gerados pelo processamento de todas as classes.

Keyword	Etiquetas
active learning	active, algorithms, learning, methods, performance.
aggregates	aggregates, applications, demonstrate, experiments, framework.
algorithms	algorithm, descriptors, experimental, performance, traditional.
anomaly detection	algorithm, approach, categories, detection, information.
applications	application, components, descriptors, programming, systems
arabic	algorithm, arabic, character, compared, recognition.
association rule	algorithm, association, efficient, large, rule.
association rule mining	algorithm, association, mining, rule, traditional.
authority	authority, information, pages, ranking, web.
automatic image annotation	automatic, analysis, image, annotation, visual.
average precision	average, collection, precision, retrieval, trec.
bagging	advantage, bagging, large, method, noisy.
bayesian network	algorithm, bayesian, categories, efficient, mining.
benchmark	benchmark, consists, evaluation, performance, xml.
bioinformatics	bioinformatics, effective, knowledge, method, mining.
boosting	boosting, classification, information, learning, result.
bregman divergence	algorithm, bregman, clustering, divergence, framework.

categorization	categorization, method, compression, examples, experiments.
clarity	clarity, average, measure, performance, precision.
classification	classification, performance, algorithm, accuracy, problem.
classification rule	rule,classification, class, support, traditional.
classifier ensemble	ensemble, classification, datasets, large, learning, time.
clickthrough data	clickthrough, search, engine, improved, relevance.
cluster-based language models	language, retrieval, models, modeling, framework.
clustering	clustering, algorithm, problem, method, experimental.
co-clustering	clustering, algorithms, framework, method, text.
collaborative filtering	filtering, collaborative, recommendations, model, ratings.
collective classification	collective, inference, accuracy, classifiers, conditions.
compression	large, faster, algorithms, indexes, known.
concept drift	concept, important, model, changes, concepts.
conditional random field	random, features, information, models, markov.
consecutive ones property	discuss, finding, problem, efficient, experiments.
constraints	constraints, problem, provide, algorithms, satisfy.
context	context, information, document, model, experiments.
convex optimization	convex, efficiently, experimental, extended, formulated.
correlation	correlation, number, experimental, algorithm, acoefficient.
cost-sensitive learning	cost-sensitive, algorithm, class, learning, method.
cross-language information retrieval	cross-language, retrieval, information, corpus, english.
data cleaning	cleaning, applications, quality, mining, records.
data fusion	fusion, improve, collection, different, techniques.

data integration	integration, information, problem, multiple, web.
data mining	mining, algorithms, efficient, large, datasets.
data publishing	publishing, anonymity, issue, privacy, sensitive.
data stream	stream, streams, real, problem, applications.
data translation	translation, cross, language, retrieval, information, problem.
database	database, query, relational, tables, applications.
decision tree	tree, decision, algorithm, mining, accuracy.
diffusion of innovations	diffusion, communities, individuals, influence, network.
digital library	library, digital, documents, algorithms, framework.
dimensionality reduction	dimensionality, reduction, analysis, space, classification.
disambiguation	disambiguation, query, retrieval, documents, effectiveness.
disguised missing data	effective, empirical, heuristic, quality, specific.
distributed hash table	large, table, management, number, problem.
distributed information retrieval	distributed, information, retrieval, query, search.
document categorization	categorization, experiments, model, algorithms, classification.
document classification	classification, document, text, empirical, methods.
dynamic programming	programming, dynamic, access, search, architecture.
efficiency	efficiency, automatically, collection, document, explore.
element retrieval	element, xml, retrieval, inex, analysis.
em algorithm	algorithm, large, high, novel, problem.
email	email, messages, text, baseline, including.
ensemble learning	ensemble, classification, large, datasets, learning.
enterprise search	enterprise, documents, finding, trec, collection.
entity resolution	problem, resolution, retrieval, entity, accuracy.
entropy	entropy, experiments, features, information, measure.

evaluation	evaluation, performance, relevance, document, precision.
evaluation metrics	metrics, evaluation, precision, retrieval, compare.
event detection	event, detection, specific, algorithm, corpus.
expectation maximization	algorithm, expectation, maximization, learning, method.
experimentation	experimentation, retrieval, techniques, model, patterns.
expert finding	expert, finding, evaluation, trec, enterprise.
extensibility	adding, database, query, applications, language.
feature extraction	analysis, feature, performance, algorithms, features.
feature selection	selection, feature, information, accuracy, algorithm
federated search	federated, search, documents, different, effectiveness.
filtering	filtering, systems, dataset, different, empirical.
formal models	retrieval, formal, information, documents, performance.
frequent itemset	frequent, itemset, algorithm, database, efficient.
fusion	improve, queries, techniques, fusion, combination.
gene expression data	expression, gene, performance, algorithms, bioinformatics.
genetic programming	programming, framework, method, baseline, combination.
genomic information retrieval	retrieval, improve, trec, information, automatic.
graded relevance	relevance, retrieval, test, different, effectiveness.
graph	graph, methods, real, domains, experimental.
graph mining	graph, mining, datasets, demonstrate, frequent.
graph partitioning	partitioning, graph, experiments, method, applications.
graphical model	graphical, models, probabilistic, time, experiments.
hidden markov model	hidden, markov, models, experimental, framework.
hierarchical clustering	clustering, hierarchical, identify, quality, agglomerative.

high-dimensional data	high-dimensional, real, classification, datasets, method.
image database	image, annotation, large, model, performance.
image retrieval	image, visual, query, retrieval, indexing.
implicit feedback	search, implicit, feedback, engine, relevance.
incomplete judgments	evaluation, incomplete, relevance, complete, retrieval.
index compression	query, indexes, large, time, collection.
indexing	indexing, retrieval, space, document, latent.
information extraction	extraction, features, extracting, retrieval, analysis.
information filtering	filtering, information, users, performance, different.
information integration	integration, web, information, enterprise, experiments.
information need	information, retrieval, design, various, web.
information retrieval	retrieval, information, performance, effectiveness, collection.
information theory	theory, information, ir, retrieval, measure.
integration	integration, web, enterprise, experiments, information.
interactive information retrieval	interactive, information, query, performance, study.
interactive retrieval	interactive, effectiveness, information, performance, query.
internet	internet, document, information, experimental, image.
inverse document frequency	document, frequency, inverse, model, retrieval.
inverted index	index, inverted, query, evaluate, search.
k-means	clustering, means, algorithm, experimental, learning.
kernel method	kernel, algorithms, method, efficiently, experimental.
keyword search	keyword, search, existing, processing, ranking.
knowledge discovery	discovery, knowledge, mining, models, consists.
language model	language, retrieval, information, trec, model.

large datasets	datasets, large, number, applications, mining.
latent semantic analysis	latent, semantic, indexing, standard, collaborative.
latent semantic indexing	latent, semantic, indexing, low, retrieval.
learning	learning, performance, labeled, demonstrate, shown.
learning to rank	rank, experimental, method, learning, training.
link analysis	link, analysis, content, web, experimental.
local search	local, search, information, query, algorithm.
logistic regression	logistic, regression, compared, documents, filtering.
machine leaning	machine, learning, vector, support, text.
markov random field	markov, random, models, experimental, features.
matrix approximation	analysis, applications, decomposition, experimental, previously.
matrix factorization	factorization, method, matrix, clustering, experimental, algorithm, negative
maximum entropy	entropy, maximum, algorithm, features, performance.
maximum likelihood	maximum, likelihood, usually, algorithm, distribution.
metadata	metadata, information, automatically, retrieval, search.
metrics	metrics, evaluation, recall, analysis, level.
missing data	missing, effective, empirical, heuristic, quality.
mixture model	mixture, model, algorithm, number, text.
model selection	accuracy, algorithm, efficient, traditional, selection.
multi-class classification	algorithm, method, classification, multi-class, performance.
music information retrieval	music, queries, signals, vector, browsing.
mutual information	mutual, algorithm, method, important, quality.
naive bayes	bayes, naive, classifier, experiments, application.
named entity	named, entity, information, detection, document.
ndcg	ndcg, cumulative, precision, discounted, trec.
nearest neighbor	nearest, neighbor, experimental, search, techniques.

neural networks	networks, algorithms, neural, analysis, biological.
olap	olap, experiments, efficient, finding, framework.
online learning	learning, online, methods, applications, batch.
ontology	ontology, work, information, clustering, current.
opinion mining	opinion, documents, experimental, performance, relevant.
optimization	optimization, performance, experimental, complex, execution.
outliers	outliers, applications, fast, real, outlier, detecting, discovery
pagerank	pagerank, web, ranking, pages, links.
parameter estimation	estimation, information, parameter, empirical, underlying.
passage retrieval	performance, documents, query, systems, answer.
pattern summarization	patterns, model, summarization, datasets, generated.
peer-to-peer	peer, network, distributed, query, efficient.
performance	performance, retrieval, information, evaluation, experimental.
performance evaluation	evaluation, performance, retrieval, relevant, test.
personalization	personalization, search, query, web, specific.
phrase	phrase, experimental, improves, information, large.
plsa	plsa, analysis, latent, semantic, probabilistic.
pooling	pooling, pool, retrieval, test, judged.
precision	precision, retrieval, average, measures, accurately.
prediction	prediction, art, state, evaluation, method.
predictive modeling	models, predictive, accuracy, demonstrate, build.
principal component analysis	component, principal, analysis, dimensionality, reduction.
privacy	privacy, preserving, mining, individual, experimental.
privacy-preserving data mining	privacy-preserving, mining, distributed, efficient, algorithm.

probabilistic data	probabilistic, efficient, model, natural, algorithms.
probabilistic model	probabilistic, model, effectiveness, learning, patterns.
quantification	quantification, measure, accurately, categories, classifications.
query classification	classification, different, query, documents, according.
query drift	problem, drift, concept, different, feedback.
query expansion	query, expansion, relevance, web, model.
query length	length, retrieval, query, effectiveness, experimental.
query log	query, search, effectiveness, demonstrate, expansion.
query log analysis	query, analysis, search, log, engine.
query optimization	query, optimization, experimental, information, techniques.
query processing	query, processing, database, performance, techniques.
query reformulation	query, search, reformulation, web, terms.
query suggestion	query, search, suggestion, engine, log.
query translation	retrieval, translation, query, dictionary, english.
question answering	answering, question, models, task, evaluation.
question classification	question, classification, promising, support, answers
random sampling	large, experiments, quality, query, retrieval.
random walk	random, walk, information, analysis, experimental.
randomized algorithm	randomized, high, mining, systems, analysis.
rank aggregation	aggregation, ranking, consistently, effective, individual.
ranking	ranking, retrieval, information, learning, function.
ranking svm	ranking, feature, retrieval, algorithm, experimental.
recommender system	systems, filtering, recommender, information, collaborative.
redundancy	redundancy, efficient, analysis, method, mining.

regression	regression, model, algorithm, linear, performance.
regularization	regularization, method, information, algorithm, experimental.
relational database	relational, experiments, database, algorithms, efficient.
relevance	relevance, document, performance, query, term.
relevance feedback	feedback, relevance, performance, query, improve.
relevance models	relevance, performance, retrieval, model, precision.
retrieval	retrieval, effectiveness, framework, information, models.
retrieval heuristics	heuristics, information, performance, existing, search.
sample selection bias	random, sample, assumption, bias, applications.
sampling	sampling, information, query, technique, accuracy.
scalability	scalability, large, query, cluster, performance.
schema mapping	mapping, relational, model, information, tool.
schema matching	matching, schema, integration, attributes, problem.
search	search, web, engine, relevant, query.
search engine	engine, search, web, query, relevant.
search results	search, query, web, evaluate, relevant.
security	mining, techniques, crucial, discuss, knowledge, larger, research
semantic web	semantic, information, vector, experimental, model.
semi-supervised learning	semi-supervised, learning, labeled, unlabeled, training.
sentiment analysis	analysis, classification, experimental, sentiment, documents.
sentiment classification	sentiment, classification, analysis, applications, documents.
similarity measure	similarity, measure, work, document, model.
similarity search	similarity, query, effectiveness, feature, search.
simulation	simulation, query, range, applications, automatic.

singular value decomposition	decomposition, value, analysis, applications, matrix.
smoothing	information, retrieval, modeling, language, trec.
social network	network, social, large, model, real.
social network analysis	social, network, analysis, including, time.
spam	spam, filtering, method, detection, threat.
spectral clustering	clustering, spectral, methods, algorithm, applications.
stability	algorithm, stability, point, random, stable.
statistical model	statistical, model, documents, problem, cross.
statistics	statistics, accurate, given, real, class.
stemming	stemming, retrieval, word, character, information.
streams	stream, mining, algorithms, efficient, problem.
subspace clustering	subspace, experimental, algorithms, cluster, improve.
summarization	summarization, text, method, different, experimental.
supervised learning	learning, supervised, algorithm, classification, models.
support vector machine	support, vector, machine, svm, learning.
tagging	tagging, large, real, scale, systems.
temporal analysis	temporal, large, mining, research, traditional.
temporal data mining	temporal, mining, research, discover,collected.
term dependency	retrieval, term, information, dependency, trec.
term weighting	weighting, term, retrieval, document, algorithm.
test collection	collection, test, retrieval, document, search.
text	text, algorithms, applications, demonstrates, evaluation.
text categorization	classification, categorization, text, method, corpus.
text classification	classification, categories, information, results text.
text data mining	text, mining, applications, discover, information.
text mining	text, mining, task, analysis, large.

text summarization	summarization, text, methods, evaluation, human.
thesaurus	information, experimental, web, retrieval, automatically.
time series	series, time, algorithm, problem, work.
topic detection and tracking	tracking, topic, tdt, detection, evaluation.
topic model	topic, model, demonstrate, information, problem.
topic modeling	topic, document, modeling, text, address.
topic tracking	topic, tracking, detection, tdt, evaluation.
transliteration	japanese, language, english, information, retrieval.
trec	trec, performance, effectiveness, ir, task.
unsupervised learning	unsupervised, learning, algorithm, framework, supervised.
update	update, query, applications, problem, constraints.
user behavior	search, behavior, web, engine, findings.
user feedback	feedback, performance, relevance, information, evaluation.
user interface	interface, information, retrieval, techniques, interfaces.
user modeling	modeling, search, query, different, feedback.
user profiling	analysis, techniques, web, accuracy, profiling
user study	study, information, search, documents, query.
vector space model	space, vector, model, query, retrieval.
visualization	visualization, visual, dimensional, features, knowledge.
web information retrieval	information, retrieval, performance, context, query.
web mining	mining, pattern, frequent, algorithms, techniques.
web search	search, engine, information, query, effectiveness.
wikipedia	wikipedia, document, information, corpus, method.
word sense disambiguation	sense, word, disambiguation, search, document.
wordnet	wordnet, sense, word, document, extracted.
world wide web	wide, world, systems, applications, case.
xml retrieval	retrieval, information, effectiveness, models, performance.

xpath	xml, algorithms, experimental, query, xpath,.
xquery	xquery, applications, processors, semantics, language.