UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

LOURENÇO SELLE JACOBS

Utilizando buscas online para identificar informações relevantes em Mineração de Textos

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação

Orientador: Prof. Dr. Eliseo Reategui
Co-orientador: Prof. Dr. Leandro Krug Wives

Porto Alegre
2016

RESUMO

Mineração de Textos é uma área importante na literatura de Recuperação de Informação, por encontrar informações em dados não estruturados (textos), e que tem sido muito utilizada para descobrir conteúdo em livros, artigos, trabalhos textuais, entre outros, de forma rápida e eficiente. Também a utilização de web sites, blogs, redes sociais, fóruns, e outros, teve um crescimento maciço nos últimos anos, fazendo a Internet atualmente ser a principal fonte de conteúdo textual. Junto com esse aumento, os algoritmos e aplicações capazes de extrair informações e minerar esses textos também tiveram um significativo crescimento. Tais algoritmos são utilizados para auxiliar os usuários a obterem maiores quantidades de informação em um menor espaço de tempo ou mesmo encontrar informações que passaram despercebidas pelo leitor. Tendo em vista o crescimento contínuo de dados em formato textual, o processo de mineração de textos busca as melhores formas de encontrar informações que representam claramente o conteúdo de um documento. Uma de suas aplicações é extrair termos que possuam relevância ao assunto que o texto aborda, orientando o usuário no seu entendimento da forma mais adequada. Ao considerar a grande quantidade de textos que se tem de ler no dia a dia, ferramentas de mineração de textos são de grande auxílio para sintetizar e organizar as informações desses documentos. Apesar dos avanços sobre a pesquisa na área, muitos resultados obtidos no processo se apresentam pouco relevantes ao usuário. Considerando que a mineração de documentos não possui uma base de contexto da qual se possa também extrair informações acerca do assunto versado, a qualidade do resultado da mineração de texto pode ser prejudicada. Uma forma de solucionar tal problema é pós-processar o conhecimento obtido antes de apresentá-lo ao usuário final, buscando encontrar as informações que são mais relevantes ao texto minerado. Este trabalho visa aprimorar a qualidade dos termos retornados pelo processo de mineração, através do uso de métricas de contexto para avaliá-los. Para isso, foram utilizados o motor de buscas do Google e o minerador de textos Sobek, esse último desenvolvido por um grupo de pesquisa da UFRGS. É inserido um contexto junto a cada termo extraído, e então realiza-se uma série de buscas em meio às páginas indexadas pelo Google, que mostram o quão relevante são os termos quando inseridos dentro de um determinado assunto. Dessa forma, o usuário pode encontrar um maior destaque naqueles relacionados com o assunto do texto minerado, sendo possível reordená-los para que sejam priorizados os mais representativos dentro do contexto. Para os testes, foram escolhidos textos sobre educação e computação, com participantes especialistas na área.

Foi solicitado que esses fizessem o reordenamento dos termos já extraídos, de forma que apontassem qual a ordem considerada mais relevante. Como resultado, pôde-se observar que houve melhora após o reordenamento dos termos pelo algoritmo proposto, pois os termos reordenados pelos participantes ficaram, em média, com a ordenação mais próxima daquela mostrada pelo algoritmo do que aquela dada pelo minerador.

Palavras-chave: Mineração. usuário. textos. contexto.

Using online searches to identify relevant information in Text Mining

ABSTRACT

Text Mining is an important area in the literature of Information Retrieval, because it helps find information in non-structural data (texts), and it has been used mostly to discover content in books, articles, textual works, and others, in a rapid and efficient way. Also the usage of web sites, blogs, social networks, forums, and others has had a massive growth over the last few years, making the Internet the main source of textual content nowadays. With this advancement, the algorithms and applications capable of extracting information and mining these texts also have had a significant increase. Such algorithms are used to support users to obtain greater amounts of knowledge in a minor amount of time, or even discover information which have passed unnoticed by the reader. Knowing the continuous evolution of data in textual formats, the process of text mining look to find better ways of discovering knowledge that represents clearly the content of the document. One application of text mining is to extract terms that have relevance to the subject the text addresses, guiding the user to comprehend it in a more adequate way. When considering the great amount of texts to be read on a daily basis, text mining tools are a great help to synthesize and organize the information found in these documents. Despite the advances in the area researches, many results obtained in the process are not of relevance to the user. Considering that the mining of simple documents does not have a context base in which information related to the subject can also be extracted, the quality of the text mining can be injured. One way to solve this issue is to post-process the information obtained before presenting them to the final user, looking for finding knowledge that is more relevant in the mined text. This work intends to improve the quality of the returned terms in the mining process through the usage of context metrics to evaluate them. For this purpose, it was used the Google search engine and the Sobek text mining tool, this last one developed by a research group of UFRGS. The context is joined with every term extracted and then it makes a series of searches between the pages indexed by Google, which show how relevant are these terms when inserted in some specific theme. This way, the user can find more relevance in terms related with the mined text's subject, being able to reorder them to highlight those who are the most representative in the document context. To accomplish the tests, it was chosen texts about education and computation, with the participants being specialists in the area. They were asked to reorder the terms already extracted and point

out which order was considered more relevant. Finally, the proposed algorithm improved the terms' order, since the new order given by the participants was, in average, closer to the one shown by the algorithm in comparison to the one given by the miner.

Keywords: mining, user, texts, context.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

MT	Mineração de Texto
RI	Recuperação de Informações
KD	Descoberta de Conhecimento (Knowledge Discovery)
PLN	Processamento de Linguagem Natural

SUMÁRIO

1 INTRODUÇÃO

Grande parte da literatura da área de Recuperação de Informações (RI), Descoberta de Conhecimento (KD) e Mineração de Textos (MT) têm o objetivo de ajudar a desenvolver métodos para extrair informações relevantes de um documento textual. A partir de um conjunto de dados não estruturados ou semiestruturados (textos), é possível elaborar a sua organização, bem como processá-los para obter os principais termos (ou os mais relevantes) de um documento ou de um conjunto, e apresentá-los ao usuário. Assim, a partir da computação, revela-se a possibilidade de descrever aplicações capazes de realizar tais tarefas.

O processo de MT é baseado em uma série de passos bem definidos, mas que variam de acordo com o estilo de abordagem do algoritmo (podendo ou não seguir todas as etapas). Contudo, grande parte dos algoritmos propostos se baseiam em um pré-processamento (MORAIS; AMBRÓSIO, 2007) para só depois iniciar a mineração propriamente dita. Esse processo de pré-processamento engloba: abordagem e preparação dos dados, seguido de indexação e normalização (para identificar similaridades nas palavras e remover *stopwords*). Já a mineração é realizada com o cálculo de relevância das palavras e a seleção dos termos, terminando com a análise dos resultados em um pós-processamento, onde mais termos podem ser removidos ou adicionados (FELDMAN; SANGER, 2007).

As áreas em que se aplica MT são bem variadas, mas é importante destacar que a quantidade de documentos simples textuais presentes na Internet teve um aumento muito significativo nos últimos anos (REATEGUI et al., 2016). Pela grande proliferação da rede, tanto geográfica quanto socialmente, o aumento do número de redes sociais, blogs e mídia online contribuiu para que a Internet se tornasse a principal fonte de informação nos dias atuais. Ainda assim, pode-se observar a aplicação de MT em outros campos, como áreas médicas, biomédicas, de marketing, de biologia, de tecnologia, entre outras, inclusive sendo aplicada sobre dados textuais multilíngues, e também tendo muito enfoque na sumarização de textos (AGGARWAL; ZHAI, 2012).

Considerando os âmbitos de aplicação de MT citados, é importante ressaltar também os conceitos de redução dimensional (MORAIS; AMBRÓSIO, 2007) ou enriquecimento das informações (SPERETTA; GAUCH, 2008), onde se busca melhorar a precisão dos resultados obtidos. É justamente nessa etapa do processo em que este trabalho está inserido, buscando uma forma de melhorar e refinar os resultados em meio a um pós-processamento de MT junto a um documento.

Das muitas abordagens onde é utilizada MT, inúmeras tentam identificar os termos considerados mais relevantes em um documento. Tag Clouds, por exemplo, apresentam para o usuário uma representação gráfica do texto, utilizando os termos mais recorrentes como partes de uma nuvem de informações. A partir dos termos mostrados, ficam representados em fontes largas e enfatizados aqueles que são mais usados e relevantes. Por vezes, é possível selecionar termos e visualizar uma coleção de tags associadas com aquela selecionada.

Algoritmos que buscam encontrar os termos mais recorrentes de um texto necessitam de uma base de dados externa (tal como uma ontologia ou *corpus*) para aprimorar o resultado da mineração (HOTHO; STAAB; STUMME, 2003)(KOHAVI et al., 2004). Normalmente, esses algoritmos não conseguem obter informações a partir do próprio documento e do contexto em que esse está inserido. Ou seja, como não conseguem detectar o tema de um texto, não podem utilizar tais informações para auxiliar na extração dos termos. Esse tema no qual o texto está inserido é definido neste trabalho como sendo o contexto do documento.

O contexto geralmente pode ser inferido partindo-se de uma ontologia ou de informações previamente conhecidas pelo usuário, entretanto, essas são raramente extraídas do documento em si. Ainda assim, parte-se do princípio de que uma pessoa desenvolve um texto sobre um assunto específico (MORAIS; AMBRÓSIO, 2007). Assim, é possível reduzir o domínio para trabalhar somente com os principais termos conectados ao tema abordado. Por exemplo, se considerarmos um documento sobre *"Os principais causadores de ataques cardíacos"*, temos que priorizar um termo como "colesterol", ao invés de algum como "hospital", porque o primeiro está mais fortemente relacionado ao tema do documento.

Para aprimorar o processo de MT e a relação de termos encontrados, este trabalho propõe um algoritmo que busca realizar a análise do texto minerado a partir de determinado contexto. Esse contexto é definido utilizando-se os dois termos que aparecem como os mais relevantes do texto, considerando-se determinado algoritmo de mineração, e realizando uma investigação da relação dos demais termos junto ao contexto escolhido. Para isso, é calculada uma relação entre quando o termo está isolado e quando esse está inserido no contexto do documento. O algoritmo utiliza a ferramenta de mineração de textos Sobek para encontrar os termos mais recorrentes e formar o contexto do documento. A partir desses termos, são realizadas buscas online com o aporte do motor de busca Google para identificar a relevância de cada termo extraído em relação ao tema em questão.

Este trabalho está organizado com a seguinte estrutura: no Capítulo 2, há uma visão geral sobre MT, destacando os principais conceitos envolvidos, explicando o que é definido como entrada e saída de um algoritmo de MT e citando suas principais aplicações; no Capítulo 3 são citadas algumas ferramentas de mineração, e também é explicado o funcionamento do minerador Sobek, descrevendo as vantagens e desvantagens da sua utilização no cenário de MT e também no contexto que envolve o algoritmo desenvolvido; o Capítulo 4 descreve a proposta de pesquisa deste trabalho, iniciando com o conceito da importância de utilizar o contexto de um documento simples, prosseguindo com a ideia que engloba o tema proposto e como o utilizamos no pós-processamento de MT; o Capítulo 5 apresenta os experimentos realizados e o processo de validação do algoritmo; por fim, o capítulo 6 apresenta as conclusões, tendo como base os resultados, e os trabalhos futuros dentro desta área de pesquisa.

2 MINERAÇÃO DE TEXTO

MT é considerado um processo de extração de conhecimento e padrões não-triviais de documentos textuais não-estruturados (TAN et al., 1999). Também é descrito como uma variação da mineração de dados (NAVATHE; RAMEZ, 2000), onde se tenta encontrar e extrair de bases de dados maiores informações interessantes ao usuário. Como a mineração de dados presume que esses estão armazenados em formatos estruturados, MT busca realizar uma análise deles em um pré-processamento, identificando se são semiestruturados e não-estruturados através da avaliação de padrões desses dados (FELDMAN; SANGER, 2007). Gupta e Lehal (GUPTA; LEHAL, 2009) também afirmam que ferramentas de mineração de dados têm o propósito de lidar com dados estruturados na base de dados, enquanto MT ocorre sobre coleções de documentos textuais, e-mails, documentos HTML, entre outros, considerados conjuntos de dados semiestruturados e não-estruturados. Feldman e Sanger (FELDMAN; SANGER, 2007) afirmam ainda que, apesar de algumas informações serem implícitas, e de certa forma até escondidas nos documentos, MT pode transformar sua estrutura implícita em uma representação explícita. Por isso, muitos autores afirmam (PRADO, 2007) (BARION; LAGO, 2015) que MT ocorre descobrindo informações implícitas e não-triviais em bancos de dados textuais. Este capítulo apresenta conceitos e aplicações de mineração de texto, trazendo também detalhes sobre a estrutura geral do processo de mineração.

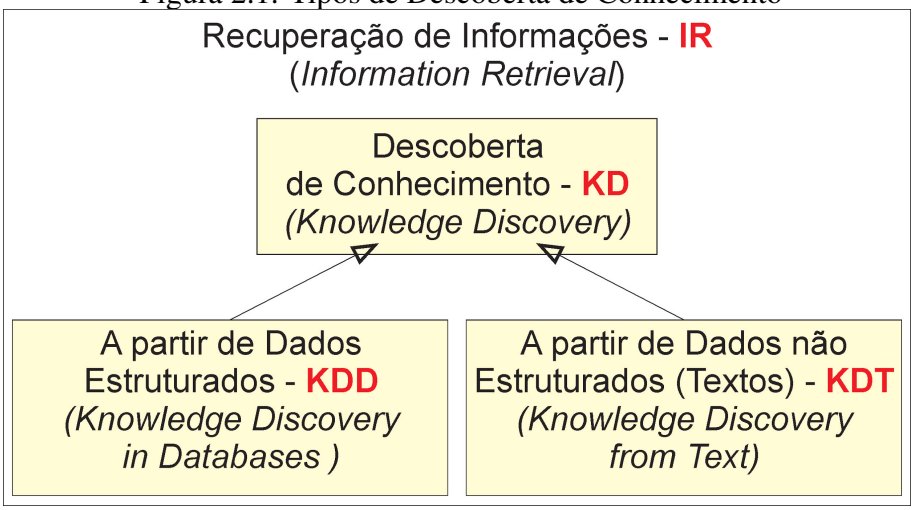
2.1 Conceitos

A Mineração de Textos (MT) pode ser descrita como Processo de Descoberta de Conhecimento (KD), ou Descoberta (ou Recuperação) de Informações (RI) (GUPTA; LEHAL, 2009). KD significa receber informações relevantes e agregá-las ao conhecimento prévio do usuário, mudando seu estado para resolver algum problema atual (WIVES, 2005). Hotho, Nürnberger e Paaß (HOTHO; NÜRNBERGER; PAASS, 2005) descrevem KD em bases de dados como um processo definido por vários passos aplicados iterativamente, onde por vezes precisa-se de um feedback do usuário. É possível observar na Figura 2.1 um diagrama de relacionamento das áreas relacionadas à RI.

Por fim, MT também é considerado um método que realiza o Processamento de Linguagem Natural (PLN), uma área da Ciência da Computação que estuda o desenvolvimento de programas de computador que analisam, reconhecem e/ou geram textos em

linguagens humanas, ou linguagens naturais (PERNA; DELGADO; FINATTO, 2010). Mais especificamente, Gonzalez e Lima (GONZALEZ; LIMA, 2003) definem que PLN envolve o uso de um computador para se tratar diversos aspectos da comunicação de seres humanos (sintaxe, gramática, sons, palavras, discursos, etc.).

Figura 2.1: Tipos de Descoberta de Conhecimento



Fonte: (MORAIS; AMBRÓSIO, 2007), p 2.

As características mais comuns utilizadas para demarcar textos são (FELDMAN; SANGER, 2007):

- Caracteres: normalmente são letras, podendo ser números ou símbolos, e que formam os demais marcadores do texto. Estes são muito usados para identificar o idioma do texto, mas não apresentam informações relevantes para compreender os dados;
- Palavras: cria-se, para cada palavra diferente dentro do texto, um "token" de dado. Assim, gera-se uma base de dados com milhares desses, extremamente densa e inviável de ser analisada. Normalmente, utiliza-se alguma heurística em pós ou pré-processamento para impedir que todas as palavras sejam listadas como resultado;
- Termos: podem ser constituídos de palavras ou de um conjunto de palavras (expressões, palavras separadas por hífen, frases, etc.) dentro do texto. Geralmente, os termos extraídos possuem também uma relevância semântica, diferente da utilização de somente palavras. E também, com frequência, se utilizam heurísticas para um pós-processamento, onde se reduz ainda mais a quantidade de termos que possam ajudar na compreensão do texto;
- Conceitos: estes elementos são a única forma resultante da análise em que se pode utilizar fontes externas ao texto para encontrá-los, como base de dados, ontologias e

outros textos. Dentro de MT, a relação de cada informação com o texto e as demais informações, por vezes, é obtida considerando o escopo do documento e sua relação com outros documentos.

Dentro de MT, são comuns abordagens com heurísticas e algoritmos voltados à análise da distribuição e frequência dos marcadores do documento, assim como várias associações de conceitos entre documentos, para poder extrair informações mais relevantes e descobrir relacionamentos de uma coleção como um todo (FELDMAN; SANGER, 2007). Abordagens que consideram a relação com outros documentos permitem encontrar informações de forma implícita, indo além da busca de termos ou conceitos mais relevantes, e permitindo criar uma representação mais rica e abrangente do conteúdo. Por exemplo, em um conjunto de documentos textuais, há um que cita um determinado conjunto de termos relacionados a uma ontologia específica que também estão presentes em diferentes documentos, permitindo estabelecer um relacionamento entre os diferentes textos. Ou seja, existe um relacionamento entre esses documentos, podendo ser tanto pelo significado do texto quanto pelo problema que eles apresentam. Assim, como citado antes por Feldman e Sanger (FELDMAN; SANGER, 2007), pode-se estabelecer uma análise mais completa de cada texto ao considerá-los como um conjunto de textos.

Este processo pode encontrar também um relacionamento entre os termos, permitindo uma representação ainda mais completa das informações extraídas em um documento simples (REATEGUI et al., 2016). A forma como os relacionamentos entre termos são descobertos varia de acordo com o algoritmo do minerador utilizado, podendo representar diferentes informações. Muitas vezes, um algoritmo pode expressar ligações entre alguns termos, enquanto outro apresenta ligações entre termos diferentes no mesmo documento. A partir dessa perspectiva, pode-se afirmar que nenhum dos relacionamentos está incorreto pois o significado do relacionamento não é único.

Gupta e Lehal (GUPTA; LEHAL, 2009) citam a utilização do relacionamento de termos para ordenar documentos, contribuindo na sua classificação e identificação daqueles mais relacionados a um tópico específico. Junto a esse princípio, citam também a ligação entre conceitos que conectam documentos relacionados, ajudando o usuário a identificar informações que talvez não pudessem ser encontradas com métodos tradicionais. Por exemplo, uma aplicação de MT pode encontrar facilmente uma relação entre os tópicos X e Y, e também entre Y e Z, mas além disso, pode também encontrar uma relação entre X e Z, algo que seria difícil para a pessoa, tendo em vista o grande número de documentos que precisariam ser ordenados e organizados.

2.2 Aplicações

MT vem sendo amplamente utilizado para extrair informações de grandes volumes de documentos textuais, principalmente devido a cerca de 80% do conteúdo presente na Internet estar na forma de texto (CHEN, 2001). Por isso, é necessário que os algoritmos sejam capazes de extrair rapidamente informações ao mesmo tempo em que analisam o seu conteúdo.

Aggarwal e Zhai (AGGARWAL; ZHAI, 2012) definem algumas classes de problemas encontrados no contexto de MT:

- Sumarização de texto: resumir ou sumarizar um conjunto grande de documentos textuais por tópicos. Essa função é dividida em sumarização abstrata (onde se sintetiza as informações que não necessariamente aparecem no texto) e sumarização de extração (onde se monta um resumo com as informações extraídas do texto);
- Extração de informação de dados textuais (*Information Extraction*): extração de entidades e suas relações no texto, que revela informações com maior valor semântico do que apenas uma representação de várias palavras juntas;
- Redução dimensional para MT: um método para representar dados subjacentes em formatos comprimidos para recuperação ou indexação (JOLLIFFE, 2002);
- Mineração em dados textuais multilíngues: com a proliferação de ferramentas que recuperam informações da Internet para outras aplicações, tornou-se útil aplicar mineração em diferentes idiomas, ou mesmo utilizar o conhecimento de um idioma para outro. Exemplos dessas aplicações podem ser encontrados ao se pesquisar um grupo de documentos de diferentes idiomas que versem sobre o mesmo tópico;
- MT em Redes Sociais: uma das fontes mais comuns de texto na *Web* são as redes sociais, que permitem que pessoas se expressem de forma rápida e livre, em um contexto de um largo alcance (AGGARWAL; WANG, 2011). Redes sociais são hoje muito exploradas por sites comerciais pela influência em usuários e pelo marketing direcionado. O processo de mineração em redes sociais requer uma habilidade de minerar dados muito dinâmicos que contém informações pobres e vocabulários sem padrões;
- Mineração de opinião dos dados textuais (*Opinion Mining*): uma quantidade considerável de textos em *Web sites* ocorre no contexto de avaliações de produtos ou opiniões de diferentes usuários. Minerar esses textos para revelar e sumarizar opi-

niões sobre um tópico tem muitas aplicações como, por exemplo, inteligência de negócio (*Business Intelligence*) ou auxiliar um consumidor a otimizar decisões (separando opiniões relevantes de "spam", que só acrescentam ruídos ao processo de mineração);

- MT em dados médicos: técnicas de MT têm um papel importante para auxiliar pesquisas médicas a acessarem, de forma eficiente, o conhecimento por baixo de grandes quantidades de literatura. Sequências de genes e estruturas proteicas são exemplos de estudos que podem ser suplantados com mineração em dados em biomedicina.

Além das áreas citadas por Aggarwal e Zhai (AGGARWAL; ZHAI, 2012), na área empresarial, MT é utilizado para analisar dados de recursos humanos (resultados, opiniões e satisfação dos empregados, currículos, etc.) e também classificar informações dos contatos dos clientes. Na área comercial, há grande utilização para descoberta e análise de mercado, para fazer avaliações de investimento e ações. No âmbito educacional, podemos utilizá-la para processar e analisar uma quantidade grande de trabalhos e publicações - o que é cada vez mais necessário com o crescimento do ensino a distância (EAD) (AZEVEDO; BEHAR; REATEGUI, 2011). Ainda no contexto educacional, há o desenvolvimento de aplicações que fornecem informações para criar estratégias pedagógicas específicas para cada grupo de alunos. Por exemplo, em Klemann, Reategui e Lorenzatti (KLEMANN; REATEGUI; LORENZATTI, 2009) é citada a ferramenta de MT Sobek (a qual será aprofundada no capítulo seguinte) para auxiliar alunos no processo de produção textual.

Alguns outros exemplos do uso de MT são: mídia, telecomunicações, tecnologia da informação (TI), mercado financeiro, educação e cursos à distância, entre muitos outros (BOLASCO et al., 2005). Também é importante ressaltar a utilização de MT na recuperação de informações da Internet (citado anteriormente) - que é um campo muito utilizado atualmente. Assim, podemos fazer a extração de palavras-chave (keywords) em um site, para que o usuário possa fazer uma escolha mais direcionada e personalizada do conteúdo, que lhe seja mais interessante (GUPTA; LEHAL, 2009).

Outra área de pesquisa relacionada à MT é como apresentar ao usuário os resultados extraídos. É necessário fornecer apenas informações e padrões que possam interessar a eles, pois os padrões extraídos pelos algoritmos podem ser muito complexos e não lhes fazerem sentido (REZENDE et al., 2003). Das formas de visualizar o documento, a literatura da área apresenta alguns modelos como árvores sintáticas, modelo de vetor espacial,

árvores de decisão, tabelas, etc., mas a que se mostra de mais fácil compreensão é a representação por grafos, pois possibilita uma fácil e rápida visualização dos termos extraídos e das ligações entre eles, indicando suas relações dentro do texto (CHEIN; MUGNIER, 2008). Assim, o usuário pode observar o significado de cada dado extraído de uma forma mais simples e de fácil compreensão.

A forma de visualização em grafos é muito utilizada para ajudar na compreensão de uma série de problemas comuns, tais como alocação de recursos, comunicação e navegação *Web*, compressão de dados, ordenamento, entre outros. Ela permite organizar informações e estabelecer relação entre elas (CHEIN; MUGNIER, 2008). Normalmente, o mapeamento dos dados ocorre atribuindo um valor ou "peso" à uma relação entre termos, que representa se a relação entre eles é forte ou fraca, e também atribuindo um termo do documento a um vértice do grafo. Tal estrutura de representação dos dados permite levar em consideração a parte semântica do texto (IGLESIAS, 2011).

2.3 O Processo

Para definir o processo de MT, Morais e Ambrósio (MORAIS; AMBRÓSIO, 2007) citam uma série de etapas. Essas se iniciam com a seleção dos documentos textuais até a realização da verificação final dos resultados, passando pela análise (semântica ou estatística) do texto, a preparação dos dados, a indexação, o cálculo de relevância de cada termo e a sua seleção.

2.3.1 Abordagem de Dados

Na análise semântica, se utilizam técnicas de PLN e Processamento Linguístico, que Rosa (ROSA, 1997) define como o estudo dos significados sentenciais das palavras, dependendo do contexto em que se encontram. Mas, seguindo Morais e Ambrósio (MORAIS; AMBRÓSIO, 2007), para realizar PLN é preciso, pelo menos, conhecimento morfológico (da estrutura das palavras), sintático (de listas de palavras a serem combinadas para gerar frases), semântico (do significado das palavras), pragmático (conhecimento do contexto da língua), do discurso (frases precedentes que afetam a interpretação da próxima) e do mundo (conhecimento geral do domínio). Essas técnicas ajudam a encontrar a importância de cada palavra dentro de suas orações (CORDEIRO et al., 2005). Ou-

tras publicações também citam esses conhecimentos como níveis de PLN (GONZALEZ; LIMA, 2003) (BULEGON; MORO, 2010), inclusive Gonzalez e Lima (GONZALEZ; LIMA, 2003) consideram um conhecimento fonético ou fonológico (do relacionamento das palavras com os sons que produzem) antes do morfológico.

Na análise estatística, conta-se quantas vezes o termo está presente no texto. Inicia-se essa parte do processo com uma codificação inicial dos dados, tendo uma indicação de especialistas ou com critérios definidos sobre os dados a serem selecionados. Em uma segunda etapa dessa análise, faz-se a estimativa dos dados, onde se procura um modelo adequado através de um método de estimativa ou um algoritmo de aprendizado. E no final dessa análise, tem-se os modelos de representação de documentos, sendo uma abordagem que ignora pontuação, ordem ou estrutura da palavra, contando apenas o número de vezes que ela aparece.

2.3.2 Preparação de Dados

Após abordar os dados, inicia-se o processo de KD, no qual se tenta fazer a redução dimensional, selecionando apenas as informações interessantes e que melhor expressem o conteúdo dos documentos. Para isso, alguns autores dizem que o primeiro passo para preparar os dados é a RI (EBECKEN; LOPES; COSTA, 2003), utilizando modelos para a representação de grandes coleções de textos. Os documentos relevantes são selecionados de acordo com a quantidade de palavras semelhantes retornadas da consulta. Assim, utiliza-se uma Função de Similaridade, capaz de identificar uma relação entre os termos (WIVES, 2002).

Dentre os modelos utilizados nos métodos de recuperação de informação, temos alguns destacados por Wives (WIVES, 2002) e Ebecken, Lopes e Costa (EBECKEN; LOPES; COSTA, 2003):

1. Modelo Booleano: manipula um conjunto de documentos com “and”, “or” e “not (expressões booleanas). Por exemplo, o usuário indica previamente um conjunto de termos que o documento precisa conter (faz-se uma intersecção) para que esse seja apresentado ao usuário;
2. Modelo espaço-vetorial: cada documento possui um vetor de duplas (palavra, peso), e cada vetor possui todos os termos da coleção, que podem ou não estar no documento. A partir disso, atribui-se o peso (um grau de importância) que pode ser

calculado de diferentes formas (normalmente baseia-se no número de ocorrências);

3. Modelo probabilístico: também conhecido como Modelo Bayesiano, pois usa uma base matemática estatística do Teorema de Bayes, onde se calcula a probabilidade de ocorrer A, sendo que a única evidência que se tem é B;
4. Modelo difuso (*Fuzzy*): Neste modelo, os documentos também são representados por vetores de palavras, onde cada uma possui um grau de relevância. Assim, é criado um universo onde todos os elementos estão presentes em todos os conjuntos. Cada elemento tem uma relevância em cada um dos conjuntos, podendo ser zero - sem relevância alguma - ou um valor de relevância alto, sendo este valor um indicativo de quão importante para o documento é cada palavra;
5. Modelo busca direta: usa-se uma busca de *strings* em um documento ou em um conjunto de documentos, e como resultado são mostrados ao usuário apenas os documentos que contém as *strings* buscadas;
6. Modelo *Clusters*: consiste em encontrar os documentos que tratem de conteúdos similares, contando a quantidade de palavras semelhantes nos documentos e indexando por tópico (ou *cluster*);
7. Modelo lógico: torna-se necessário modelar os documentos com lógica predicativa, para que a aplicação possa decidir melhor a relevância de cada documento, incorporando semântica ao processo de recuperação. É pouco utilizado fora do ambiente acadêmico devido ao grande esforço no trabalho de criá-lo;
8. Modelo contextual ou conceitual: este modelo considera que cada documento possui um contexto, assumindo que o texto, assim como a consulta do usuário, foi desenvolvido sobre um assunto específico. Uma vez identificado o contexto, pode-se recuperar informações relevantes em nível do tema abordado no texto, e não somente de palavras isoladas, como é feito nos demais modelos citados.

2.3.3 Indexação e Normalização

Nesta parte do processo, é gerado um índice: uma estrutura que contém as características de cada documento abordado. Ebecken, Lopes e Costa (EBECKEN; LOPES; COSTA, 2003) dizem que esta etapa facilita a identificação de similaridade de significado entre as palavras. Em MT, esse processo é automático e tem início com a identificação dos termos, podendo ser simples (em que se utiliza um dicionário de sinônimos ou um

analisador léxico para identificar as palavras e eliminar símbolos de caracteres de controle de arquivo ou de formatação) ou ser compostos (WIVES, 2002).

Essa fase também envolve (quando necessário) a remoção de *stopwords*, que são palavras que não agregam informação ao usuário e não devem ser recuperadas pelo processo. *Stopwords* geralmente são pronomes, artigos, preposições, advérbios, etc. Essas são inseridas em uma "stoplist" para não serem utilizadas em uma consulta. Por fim, é realizada uma normalização por *stemming* (que significa reduzir a palavra a sua raiz), onde se tenta eliminar prefixos, sufixos (de verbos ou de advérbios), gênero, número, grau, aumentativo ou diminutivo, plural, etc. Ou seja, eliminar variações morfológicas das palavras (WIVES, 2002).

2.3.4 Cálculo de Relevância

Uma vez que todas as palavras não têm mesma importância no texto, o cálculo de relevância é realizado para destacar as que tenham mais significado. Alguns substantivos e complementos podem ser considerados mais relevantes que os demais termos (WIVES, 2002). É comum esse cálculo basear-se no número de vezes que uma palavra aparece no texto, alguns exemplos desse cálculo de frequência são: frequência absoluta (que simplesmente representa a quantidade de vezes que o termo aparece, como já mencionado anteriormente), frequência relativa (que é calculada dividindo a frequência absoluta pelo número total de palavras do documento) ou frequência inversa (que considera a divisão do número de vezes que o termo aparece no texto pelo número de documentos - em um conjunto - em que ele está contido, para ajudar a discriminar os que têm baixa frequência mas podem ser bastante representativos).

2.3.5 Seleção de Termos

Após o pré-processamento e o cálculo de relevância, considerando os termos que foram selecionados nessas etapas anteriores, são utilizadas técnicas baseadas no peso do termo, de forma a ressaltar seu destaque dentro do tema do documento, para então escolher ou filtrar somente os que possam interessar ao usuário. A seguir, algumas das técnicas utilizadas para selecionar os termos:

- filtragem baseada no peso do termo: se estabelece um limiar - *threshold* - e se corta

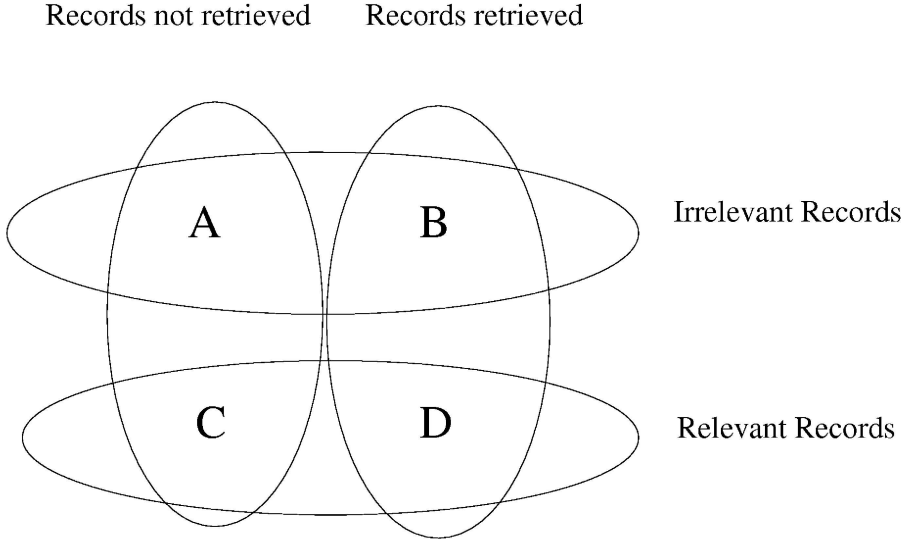
os termos que tenham peso abaixo - definido pelo cálculo de relevância na etapa anterior;

- seleção baseada no peso do termo: também conhecida como truncagem (WIVES, 2002) - é uma técnica que seleciona N termos, que contenham um número de características encontradas no documento - ordenadas por sua relevância;
- seleção por análise de coocorrência: leva em consideração os termos que aparecem em vários documentos ao mesmo tempo;
- seleção por análise de linguagem natural: utiliza análises sintática e semântica para identificar os termos mais importantes.

2.3.6 Análise de Resultados

Essa etapa do processo envolve métricas que avaliam a capacidade do processo de MT em recuperar o máximo de itens relevantes, e também a capacidade de filtrar itens irrelevantes. Para o âmbito em questão, tais itens são considerados como termos ao invés de documentos. Para exemplificar, pode-se observar o diagrama de Venn da Figura 2.2 que ilustra bem essa questão, junto com as métricas citadas posteriormente, onde (MALIK, 2006): A - número de termos irrelevantes não retornados; B - número de termos irrelevantes retornados; C - número de termos relevantes não retornados; e D - número de termos relevantes retornados.

Figura 2.2: Conjuntos do Universo de Documentos



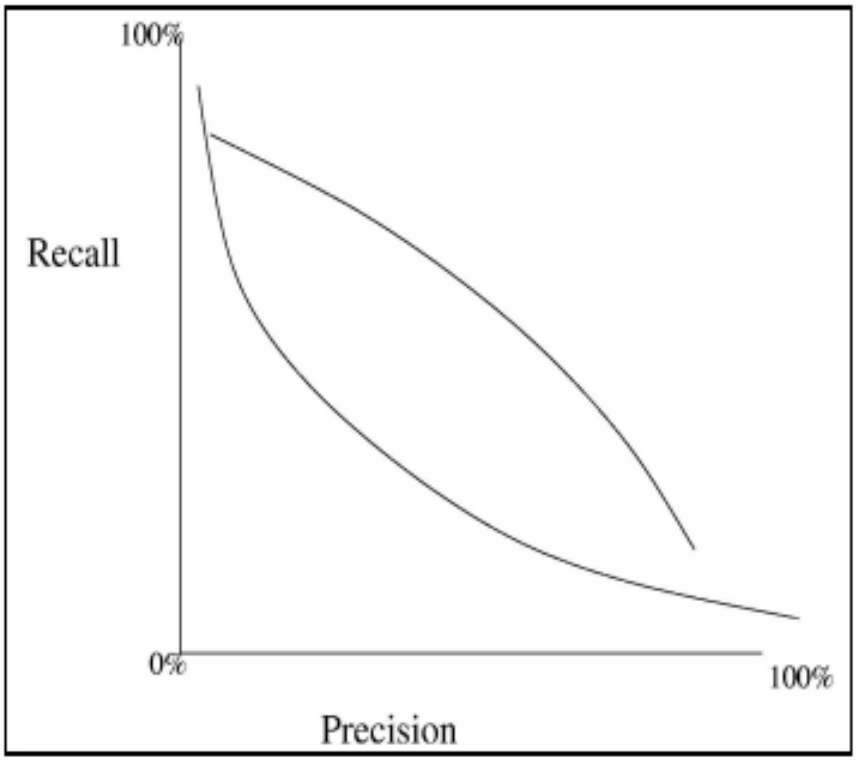
Fonte: (MALIK, 2006), p 82.

Algumas dessas métricas são:

- *Recall*: utiliza uma fórmula onde se divide o número de termos relevantes recuperados por uma estimativa de possíveis relevantes do texto. Visa medir a habilidade do sistema em recuperar itens relevantes e é, normalmente, expressado como um valor percentual (usando a Figura 2.2, a equação seria $Recall = \frac{D}{D+C}$);
- *Precision*: também utiliza uma fórmula e é expressado em valor percentual, dividindo o número de termos relevantes encontrados no texto pelo número total desses que foram retornados, tanto relevantes quanto irrelevantes (usando a Figura 2.2, a equação seria $Precision = \frac{D}{D+B}$);
- *Fall-out*: semelhante ao *Recall*, calcula a quantidade de termos irrelevantes (usando a Figura 2.2, a equação seria $Fall\text{-}out = \frac{B}{A+B}$);
- *Effort*: uma métrica usada para medir o esforço do usuário até obter um bom grau de precisão, através do número de termos relevantes na base de dados que ainda não foram recuperados.

Precision e *Recall* são métricas bastante utilizadas para medir sistemas de RI, as quais são inversamente proporcionais, significando que: à medida que a precisão do sistema aumenta, diminui sua capacidade de retornar documentos relevantes (MALIK, 2006), como se pode ver pelo gráfico na imagem 2.3.

Figura 2.3: Gráfico da Proporção entre Recall e Precision



Fonte: (MALIK, 2006), p 83.

Algumas aplicações foram desenvolvidas baseadas nas métricas citadas acima como, por exemplo, *Precision@K* e *Recall@K*, onde K indica a quantidade de itens para qual a aplicação está sendo avaliada. Assim como *AvgP@K*, que mostra um valor médio calculado entre várias precisões medidas.

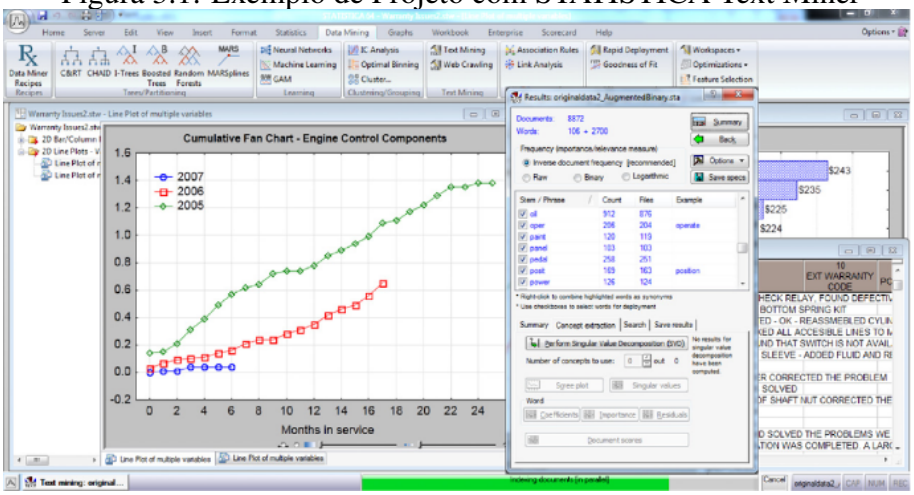
3 FERRAMENTAS DE MINERAÇÃO DE TEXTO

Atualmente, existem muitas ferramentas que realizam o processo de MT, assim como também muitas que integram MT indiretamente em alguma atividade específica como, por exemplo, melhorar uma atividade de negócios que uma empresa propõe. Portanto, aqui são citadas algumas dessas ferramentas, explicando suas principais características e seu processo de funcionamento.

3.1 *STATISTICA Text Miner*

É uma extensão do *STATISTICA Data Miner* e inicia seu processo com o usuário apontando a aplicação para uma base de dados (textos), como um diretório ou caminho, e essa pode percorrer estruturas hierárquicas de arquivos ou *Web sites* para retornar documentos de um tipo específico. O programa então indexa o *corpus* do documento, levando em conta sinônimos e *stopwords*, para então apresentar ao usuário algumas opções de combinação de palavras ou termos, e aplicar técnicas de extração de conceitos.

Figura 3.1: Exemplo de Projeto com STATISTICA Text Miner



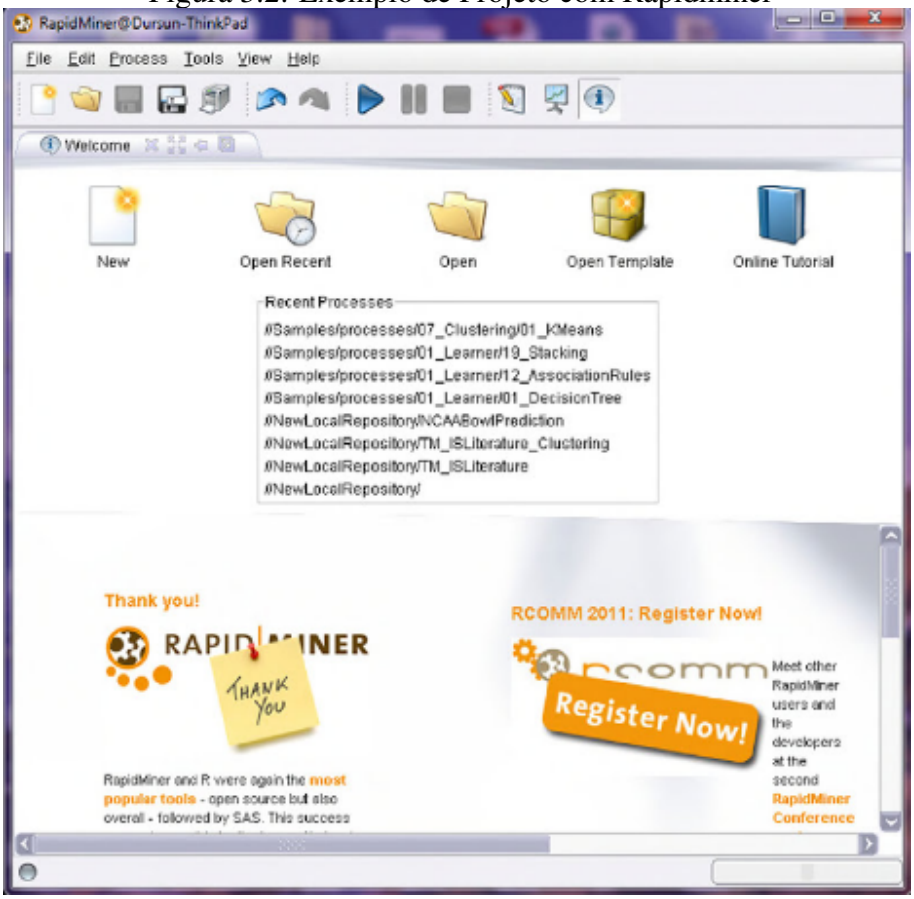
Fonte: (MINER et al., 2012), p 115.

3.2 *RapidMiner*

Uma ferramenta open-source que permite ao usuário escolher uma coleção de arquivos textos para importar, ou somente um arquivo Excel. A partir desses documentos, faz-se a transformação de todos os caracteres para diminutivo (para identificar as pala-

vras iguais), para então realizar uma *tokenização*, que divide todo o texto em *tokens*, que podem ser palavras/termos únicos. Após, é aplicada a remoção de *stopwords* para retirar palavras/termos que não acrescentam informação nova para, por fim, criar uma matriz onde o relacionamento entre o documento e o termo/palavra é representado por um índice numérico de TD-IDF (ou frequência inversa, como visto na seção 2.3.4). Após selecionar os termos, é definida a clusterização com um algoritmo *k-means clustering* (que usa uma técnica para agrupar elementos naturais usando uma medida de distância simples multidimensional).

Figura 3.2: Exemplo de Projeto com Rapidminer



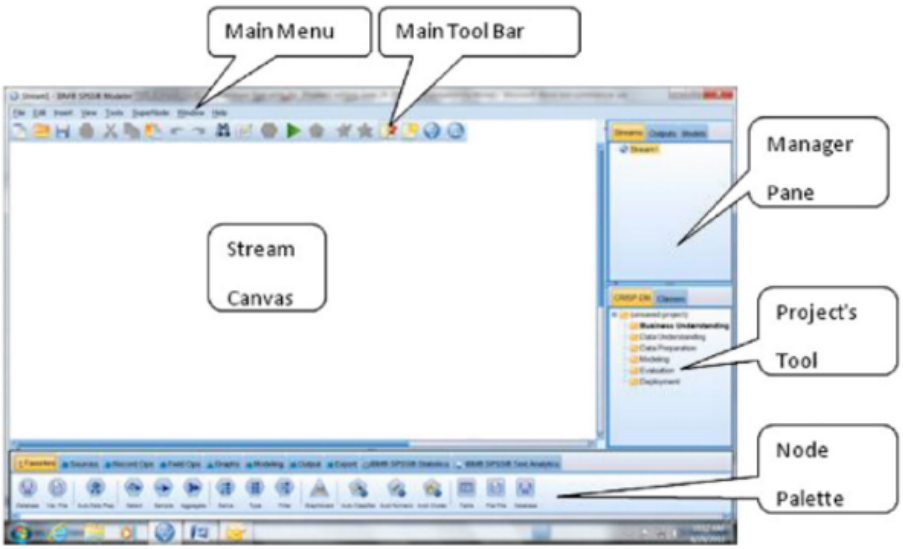
Fonte: (MINER et al., 2012), p 378.

3.3 *IBM SPSS Modeler Premium*

Desenvolvido para ajudar a descobrir e analisar dados não estruturados e criar modelos de categorização relevantes, combinando análise textual às visões de clientes. Por exemplo, muitos clientes deram *reviews* reclamando de um ruim atendimento de recepção, então se utiliza uma análise dos *feedbacks* dos clientes como um modelo para

realizar um balanço de satisfação com mais precisão. O processo se inicia com a coleta e análise de dados textuais, passando por um pré-processamento (para limpar, transcrever, validar, transformar os dados, etc.), selecionar as sugestões, processá-las para apresentar uma matriz (onde afirmações são linhas e sugestões são colunas), desenvolver modelos de classificação, escolhendo e aplicando uma variedade de algoritmos para comparar e avaliar resultados com amostras.

Figura 3.3: Exemplo de Projeto com IBM SPSS Modeler

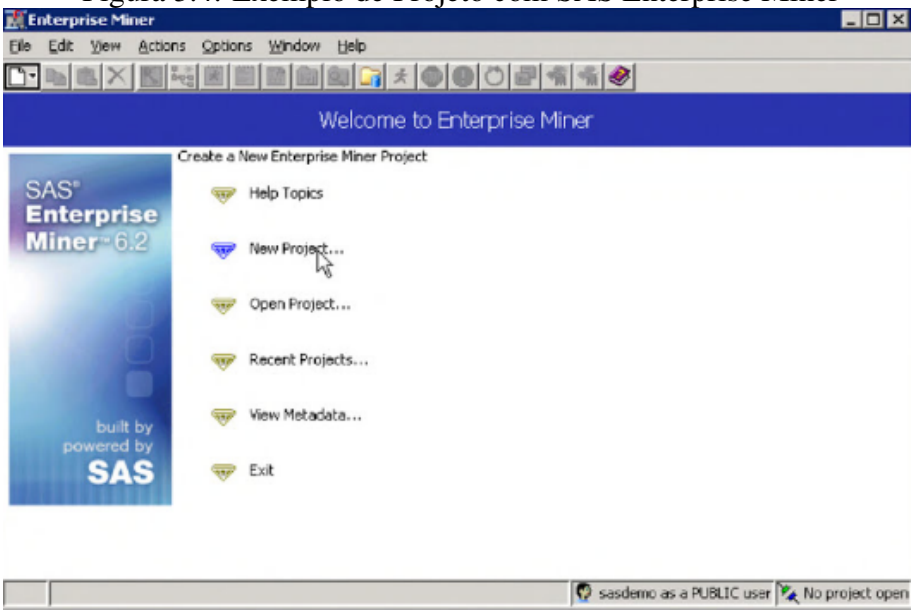


Fonte: (MINER et al., 2012), p 93.

3.4 *SAS Enterprise Miner*

Este é um software que permite descobrir conceitos em grandes coleções de documentos, agrupando-os em *clusters*, classificando-os em categorias pré-definidas e integrando dados textuais a dados estruturados para enriquecer as tentativas de predição de modelos. Ele suporta várias fontes de dados textuais, como arquivos de texto locais, bases de dados externas ou arquivos da *Web*. Esse minerador tem uma utilização mais complexa do que o normal, trabalhando com diagramas de fluxo onde cada nodo tem uma função diferente para lidar com o documento e a base de dados. Então, o usuário escolhe entre: nodo de filtragem de texto (analisa os conceitos que são interligados, como sinônimos, descartando alguns e mantendo os que possam incrementar o modelo), nodo de tópico de texto (permite a divisão dos documentos em grupos dependendo do significado), nodo *parser* de texto (permite analisar o documento para detalhar informações de termo, frases e outras entidades na coleção) entre outros.

Figura 3.4: Exemplo de Projeto com SAS Enterprise Miner



Fonte: (MINER et al., 2012), p 102.

3.5 Outras Ferramentas de Mineração

Apesar das ferramentas citadas acima, podemos apontar mais algumas open-source, como:

- *Carrot*[2]: tem como característica a *clusterização* de documentos buscados em categorias apropriadas, como os resultados de buscas ou documentos abstratos;
- *OpenNLP*: baseada em PLN em textos, utiliza tarefas como *tokenização*, segmentação e fragmentação de frases, análise de sinônimos, etc. Normalmente, essas são tarefas utilizadas em um primeiro momento, para então passar para serviços de processamento de texto mais avançados;
- *General Architecture for Text Engineering (GATE)*: utiliza um sistema de extração de informação chamado *ANNIE (A Nearly-New Information Extraction System)*, que utiliza módulos com *tokenização*, marcadores para frases e correferências, divisor de frases, etc.

3.6 A Ferramenta Sobek

Dentre as ferramentas e métodos de MT, para desenvolver o processo proposto por este trabalho, foi utilizada a ferramenta Sobek, desenvolvida pelo grupo Gtech de pesquisa da UFRGS. Por isso, ela será mais detalhada dentro desta seção.

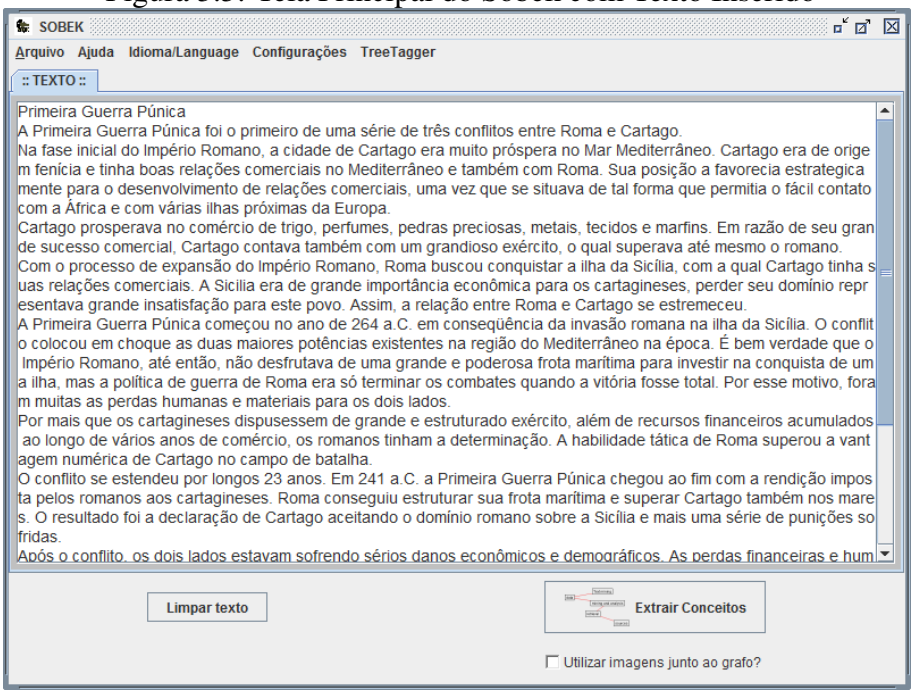
Inicialmente, percebeu-se a necessidade de auxiliar o professor no acompanhamento e gerenciamento de grandes volumes de produções textuais à distância (MACEDO et al., 2009). Assim, a ferramenta Sobek foi desenvolvida com o objetivo de fornecer aos professores instrumentos para auxiliá-los na análise de produções textuais dos estudantes (REATEGUI et al., 2016).

A ferramenta Sobek utiliza um algoritmo definido por Schenker (SCHENKER, 2003), que se baseia em análises estatísticas dos textos, e representa as informações extraídas em um modelo de grafo de distância n-simples, baseado na ideia de que cada palavra relevante no texto deve estar conectada a N palavras relevantes subsequentes (KLEMANN; REATEGUI; LORENZATTI, 2009). Assim, o usuário pode utilizar uma interface gráfica simples e amigável (como visto na Figura 3.5), onde ele pode escrever ou colar um texto que será minerado e terá os principais termos extraídos. A seguir, após apertar o botão de extração (“Extrair Conceitos”), ocorre a representação gráfica desses termos e suas relações, sendo os termos os vértices (os maiores são os de maior frequência) e as relações as arestas (como visto na Figura 3.6), juntamente com a demonstração, ao clicar em cima do termo no grafo, de suas ocorrências no texto.

É importante destacar que a apresentação dos resultados para o usuário é um dos diferenciais da ferramenta Sobek, quando comparada com outros softwares mineradores de texto (como, por exemplo, *TextAlyser*, *WordCounter* e *TagCrowd*), justamente por apresentar os termos em um grafo com o destaque dos relacionamentos entre esses (REATEGUI et al., 2011). Outro ponto que se destaca é o da sua interface ser simples e de fácil manuseio, diferente do *RapidMiner* e *IBM ManyEyes*, por exemplo, que podem exigir dos usuários maior esforço ou treinamento para sua operação (REATEGUI et al., 2016).

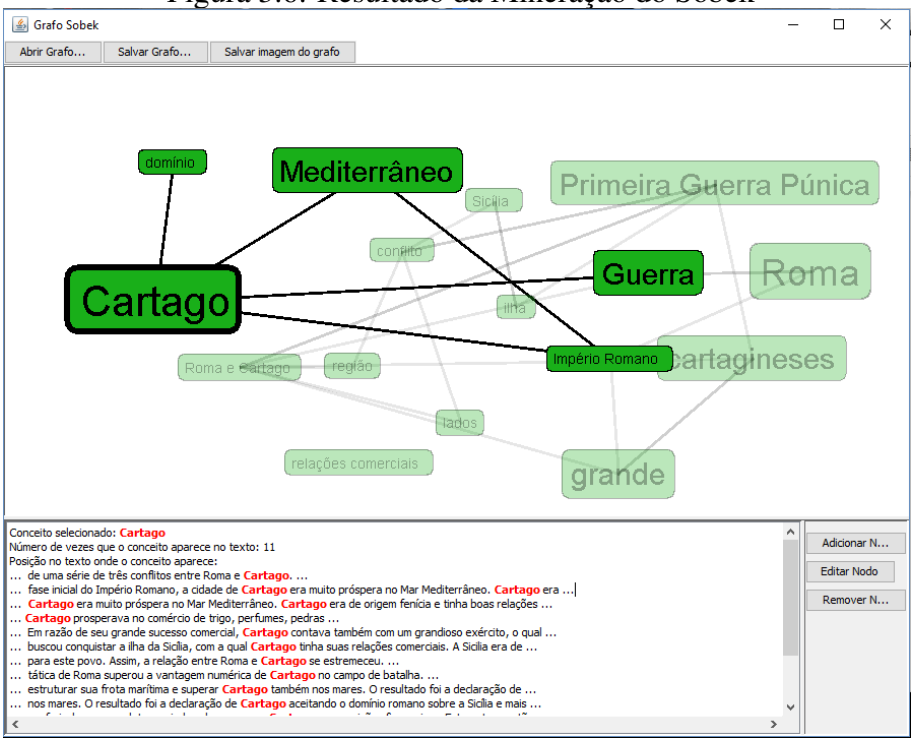
As Figuras 3.5 e 3.6 mostram um exemplo da utilização do Sobek. Inicialmente, é mostrada a imagem da tela inicial do minerador, com um texto inserido a ser minerado (Figura 3.5). Após ser pressionado o botão "Extrair Conceitos", é exibido um grafo onde são destacados os termos mais recorrentes do documento (nos nodos), com as ligações entre eles (Figura 3.6).

Figura 3.5: Tela Principal do Sobek com Texto Inserido



Fonte: Autor

Figura 3.6: Resultado da Mineração do Sobek



Fonte: Autor

A partir da representação por grafo, é possível ter-se uma ideia de como a ferramenta pode auxiliar o usuário na compreensão do texto. A sequência de termos apresentada possibilita a identificação de relações entre estes, levando o usuário a refletir sobre

o texto por meio da interpretação do grafo (KLEMANN; REATEGUI; LORENZATTI, 2009). Pela Figura 3.6, por exemplo, ao perceber o destaque do conceito “Cartago”, pode-se chegar em “Guerra” e então em “Roma”, o que demonstra a ideia que o texto traz sobre o “conflito” (também um conceito extraído) entre as duas cidades. Fica fácil de perceber, pelo conceito “Primeira Guerra Púnica” (casualmente o título do texto), que é o principal tema abordado, conflito que ocorreu na região do “Mediterrâneo”.

Outro ponto interessante a ser ressaltado é o fato de se poder escolher o idioma do texto que será inserido (português ou inglês) e algumas opções que tornam a mineração mais personalizada ao usuário como, por exemplo, poder escolher entre:

- Número médio de conceitos: define um valor médio dos termos que serão exibidos como resultado da mineração;
- *Stopwords*: permite personalizar mais a lista de *stopwords*, adicionando ou removendo termos que o usuário gostaria ou não de ver no resultado da mineração;
- Frequência mínima: pode-se escolher o número de vezes que um termo deve aparecer no texto, para que esse deva ser relacionado em meio ao resultado da extração.

Apesar da versão do programa Sobek desktop operar dessa forma, a versão utilizada neste trabalho é um *webservice* em que nem todas as opções citadas acima estão ativadas. Ou seja, não utiliza uma representação por grafo nem a possibilidade de escolher as configurações. O resultado na resposta é apenas uma mensagem que contém todos os termos e o número de ocorrências desses no documento.

3.6.1 Algoritmo do Sobek

No primeiro passo do algoritmo, para identificar os principais termos, um texto T é dividido em um conjunto P de palavras, utilizando espaços e pontuações como divisores. O conjunto P então é mapeado em termos que podem consistir de uma palavra (chamados de "termos simples") ou de várias palavras (chamados de "termos compostos"). O mapeamento é um processo estatístico que considera a frequência com que cada palavra é encontrada no texto. Quando um subconjunto de palavras é repetido com certa frequência, um termo composto é criado, e assim, essas palavras são removidas do conjunto P (apesar de ainda poderem figurar entre os termos simples, contanto que o Sobek as identifique).

Como exemplo, considera-se a sequência de palavras "AA BB CC". Se for esco-

lhido um cenário onde os termos podem ser compostos de até três palavras, então se teria o seguinte conjunto de *strings*: *S ='AA'; 'AA BB'; 'AA BB CC'; 'BB'; 'BB CC'; 'CC'* (termos com mais de três palavras não são considerados, pois não são muito frequentes e sua computação não justificaria os benefícios).

Uma vez que a identificação dos termos é completa, os elementos de S com frequência maior que a mínima considerada são selecionados para uma avaliação futura. A frequência é determinada como um limiar (*threshold*), onde considera-se que o conjunto de termos retornados tem um tamanho mínimo (o valor mínimo da frequência é 2; caso contrário, todas as palavras do texto estariam no grafo resultante).

Durante o processo de identificação dos termos, três funções são usadas para remover termos e palavras que não acrescentam informações ao grafo:

- Remoção de *stopwords*: palavras que são, em sua maioria, artigos e preposições, que não possuem significado relevante;
- *Stemming*: utilizado para reduzir redundâncias e remover termos com o mesmo significado ou grafias semelhantes (como sufixos, por exemplo);
- *Thesaurus*: identificação de sinônimos, que permite ao Sobek eliminar, do grafo resultante, termos que têm significados semelhantes.

Apesar de o número de termos poder ser definido pelo usuário, Novak e Cañas (NOVAK; CAÑAS, 2006) afirmam que entre 15 e 25 termos são suficientes para identificar a ideia central do texto. Dessa forma, o Sobek tem uma configuração inicial de um conjunto de 20 termos.

Em um segundo momento, é realizada a identificação dos relacionamentos entre os termos. Cada termo selecionado pertence ao conjunto C de todos os termos. Um relacionamento entre c_i e c_j implica em uma conexão entre eles (e que eles estão em posições próximas em T). Isso pode representar diferentes tipos de relacionamentos. Uma análise do texto T relaciona c_i e c_j quando esses não estão a mais de z palavras de distância e não há paradas entre eles no texto.

Dependendo do tamanho de C, um termo c_i pode ser relacionado com muitos outros termos, o que produziria um grafo em que as conexões seriam sem significado. Para reduzir o número de conexões, um valor máximo de *links* é permitido para cada termo. Entretanto, termos com alta frequência não possuem o mesmo número de conexões que os de baixa frequência, ou seja, o valor de conexões é proporcional à frequência do termo no texto. Apenas aquele com a frequência máxima têm o número de conexões

máximo. Os demais termos têm suas conexões definidas pela sua frequência, multiplicada pelo número de *links* do mesmo, tudo isso dividido pela frequência do termo que mais aparece no texto.

Não existe um limite mínimo do número de relações entre termos para que uma seja considerada um link no grafo. Através de *reviews* de usuários do Sobek, chegou-se em parâmetros padrões de 7 *links* e 5 palavras de distância para haver um relacionamento entre dois termos. Valores maiores que esses trariam ligações entre termos pouco relacionados e produziria muitas conexões, o que seria difícil de interpretar.

3.6.2 Aplicações e Utilizações

Como descrito no início da seção anterior, a ferramenta Sobek foi criada com o intuito de ajudar educadores a classificarem e analisarem textos de forma mais rápida e automática, no âmbito da educação à distância. Além desta, outras aplicações na área da Educação foram desenvolvidas, como em Acosta, Behar e Reategui (ACOSTA; BEHAR; REATEGUI, 2014), que propõem um sistema de recomendação que se utiliza do minerador Sobek para identificar palavras-chave no texto do aluno e recomendar-lhe conteúdos relacionados encontrados na *Web*. Já Klemann, Reategui e Rapkiewicz (KLEMANN; REATEGUI; RAPKIEWICZ, 2011) mostraram como utilizar o Sobek no apoio à produção textual, em que os alunos tinham de ler um documento, visualizar o grafo deste resultante da mineração pelo Sobek, e a partir de então deviam iniciar o desenvolvimento de um novo texto. Ainda no âmbito educacional, Azevedo, Reategui e Behar (AZEVEDO; REATEGUI; BEHAR, 2014) utilizaram essa ferramenta para integrá-la a um minerador de fóruns de discussão, chamado de *ForumMiner*. Este tinha o objetivo de analisar posts de alunos em fóruns de discussões para identificar o índice de relevância das contribuições de cada um.

Outros trabalhos fazem referência ao emprego da ferramenta Sobek na área educacional, tais como: sumarização de texto (REATEGUI; EPSTEIN, 2015), letramento (ROCKENBACH et al., 2014) (LANGA et al., 2012), apoio à aprendizagem em língua estrangeira (COSTA; REATEGUI, 2012) (PINHO et al., 2013), entre outros. Além destes, existem outras aplicações do minerador Sobek, como por exemplo no apoio à busca e classificação de consultas anteriores num ambiente de telemedicina, mais especificamente em um sistema de telessaúde implantado no sul do país (DAMASCENO et al., 2014).

4 O ESTUDO DESENVOLVIDO

Neste capítulo é descrito o algoritmo que propomos para tratar os resultados de MT, de forma a encontrar uma nova ordem dos termos resultantes no processo.

4.1 Conceitos

Primeiramente, são detalhados os conceitos que foram abordados para desenvolver um novo algoritmo de reordenamento de termos, para então explicar o funcionamento do processo criado.

4.1.1 Conceito de Contexto do Documento

Dos conceitos de MT apresentados, segue-se a ideia da utilização do contexto em que o documento está inserido, para que se possa definir com mais precisão os termos mais relevantes entre aqueles extraídos. A ideia trata de considerar que cada texto foi construído em torno de um tema, de um contexto.

Partindo desse princípio, é preciso inicialmente definir essa ramificação que engloba o assunto do texto, para então identificar uma relação deste com cada um dos termos extraídos. Ou seja, para qualquer conceito identificado dentro do documento textual, é preciso obter um indicativo da sua relevância para o tema em questão.

Para exemplificar, considera-se um documento textual que trate do assunto "medicina". Quando realizado o processo de mineração textual no documento, são extraídos os termos mais frequentes: "coração", "leito", "hospital", "médico", "doença", "maca", "injeção", "bancos" e "soro", definidos sem nenhum tipo de relevância ou importância junto ao texto, ou seja, todos os termos têm pesos iguais após o processo de extração. Contudo, o conceito "coração" está mais proximamente relacionado ao contexto da "medicina" do que o conceito "bancos". A possibilidade de identificar este índice de relevância de cada termo extraído de um texto com base em sua proximidade com determinado contexto é o principal propósito deste trabalho.

Uma das questões envolvidas neste processo é a definição do contexto de maneira manual ou automática. É importante compreender que a escolha do contexto é um ponto fundamental para a extração dos conceitos mais relevantes, pois uma definição inadequada

pode trazer uma interpretação equivocada do tema em questão. Por exemplo, em um processo de mineração de um texto intitulado "*A importância dos primeiros anos escolares*", se os termos definidos como contexto forem "escola" e "professor", os demais termos extraídos terão um indicador de relevância maior ou menor em função desta escolha.

Neste trabalho, optou-se por utilizar o processo de identificação do contexto de maneira automática, a partir do índice de relevância estabelecido pelo sistema de MT. Os dois termos com o maior índice de relevância são tomados como o contexto do documento minerado. Essa escolha teve por objetivo possibilitar o desenvolvimento da técnica sem exigir a intervenção do usuário no processo de mineração, como seria o caso se este tivesse que ler todo o texto, entendê-lo, identificar o tema, para então escolher o contexto do processo. Assim, como veremos na próxima seção, a escolha do contexto também é definida no processo de mineração, depois da extração dos termos mais frequentes/relevantes do texto.

4.1.2 Conceitos do Processo

Para melhor trabalhar com a ideia de MT e KD em um documento, optou-se por agregar um novo processo na etapa de pós-processamento. Ou seja, este processo atua sobre uma lista de termos já extraídos previamente. Como forma de gerar a lista de termos a partir de um documento textual, foi decidido utilizar o minerador Sobek para receber o texto do usuário, realizar a mineração e extrair os principais termos. Portanto, a lista de termos extraídos será fornecida pelo Sobek, a fim de tornar o processo mais automatizado e rápido, além de permitir eliminar a necessidade de intervenção do usuário no processo de KD.

A partir dos termos, obtemos o contexto do documento utilizando os principais termos extraídos pelo Sobek. Esse contexto é utilizado para reordenar os demais termos extraídos, buscando encontrar aqueles mais relevantes para o texto. Como resultado, a lista de termos considerados relevantes é ordenada com base na informação obtida pelo termo quando associado ao contexto destacado. Assim, aqueles que apresentam maior relação são priorizados quando apresentados ao usuário. Na subsecão 4.1.3, é apresentado o algoritmo em pseudocódigo do processo proposto.

4.1.3 Algoritmo em Pseudocódigo

```
1  Algoritmo de Reordenamento de Termos()
2
3  /*
4  Entrada:
5  listaTermos: lista de termos extraidos do texto
6
7  Saida:
8  listaReordenada: lista de termos reordenada
9
10 Listas:
11 LC: lista de resultados com contexto
12 L: lista de resultados sem contexto
13 LC_i.nr: num. resultados da busca do termo i no Google, com contexto
14 L_i.nr: num. resultados da busca do termo i no Google, sem contexto
15 */
16 Inicio:
17 {
18    contexto := listaTermos_0 + "+" + listaTermos_1
19    listaTermos.remove(listaTermos_0, listaTermos_1)
20    Para cada i E listaTermos faca {
21        L_i <- searchGoogle(listaTermos_i) // busca resultados sem contexto
22        LC_i <- searchGoogle(contexto + "+" + listaTermos_i) // busca
              resultados com contexto
23        listaReordenada <- LC_i.nr / L_i.nr // insere em ordem
24    Retorna listaReordenada
25 }
26 Fim
```

4.2 Descrição do Algoritmo

Previamente ao processo de reordenação dos termos, é realizada a mineração textual de um documento a ser definido pelo usuário, através da comunicação com um *web-service* desenvolvido para este fim. Esse documento é minerado pelo Sobek, que extraí os termos mais recorrentes do texto.

Uma vez que o Sobek tenha extraído todos os termos relevantes, uma *string* é retornada, contendo os termos ordenados pelo número de ocorrências destes no texto (*string:*

"termo:número de ocorrências"). Considerando que os dois primeiros termos extraídos do texto são muito representativos no tema do documento, esses dois são utilizados para definir o contexto do documento.

Nessa parte do processo, é importante ressaltar uma diferença entre termos simples e compostos, pois não podemos considerar termos compostos como sendo a união de termos simples. Por exemplo, a análise do termo composto "aquecimento global" é diferente da análise dos termos "aquecimento" e "global". Tendo isso em vista, é necessário garantir que termos compostos sejam identificados como uma sequência de caracteres e não como palavras. Uma vez que se insira uma palavra no campo de buscas do Google entre aspas, é garantido que o número de resultados apresente apenas páginas que tenham aquela palavra exatamente como foi digitada, impedindo que o Google tente corrigi-la, ou faça a busca com as palavras separadas. Se apenas uma das palavras ou trecho digitado está entre aspas, os resultados irão garantir que esses estejam presentes nas páginas apresentadas como resposta. Caso haja mais palavras no campo, que não estão entre aspas, essas podem ou não aparecer nos resultados.

Tendo a lista de termos, é realizada uma busca, para cada um desses individualmente, onde são obtidos os números de resultados sem contexto de páginas encontradas pelo Google. Essa informação é mantida na estrutura de cada termo. Dessa maneira, a quantidade de páginas resultantes está mostrando um universo de todos os documentos possíveis na Internet (das páginas indexadas pelo Google) para determinado termo sozinho.

Depois de obter esse número de resultados para cada elemento da lista, são realizadas novas buscas no Google utilizando o contexto encontrado anteriormente. Isso resulta na busca de strings no formato: "contexto+termo". Por exemplo, considerando que o contexto de um documento é "aquecimento global" e "poluição", as novas buscas serão realizadas utilizando a *string*: ["Aquecimento global"+poluição+termo].

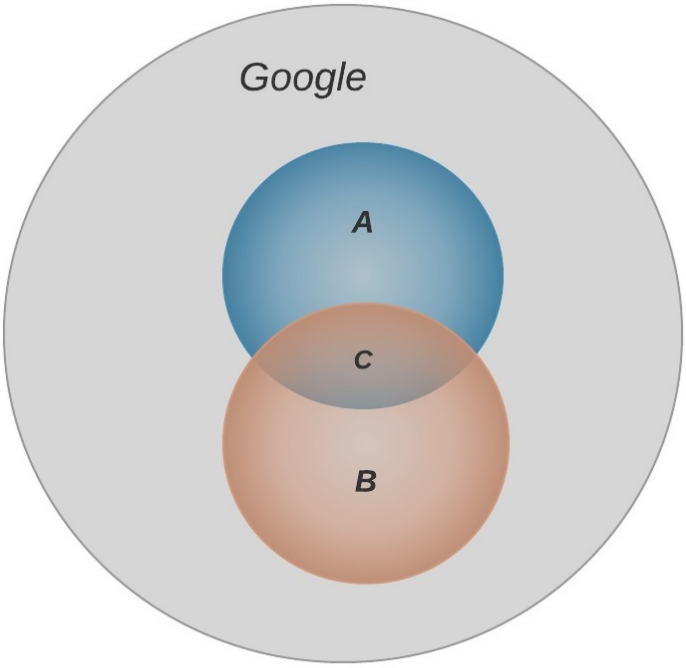
Com base nos resultados obtidos durante a operação de busca pelas ocorrências de páginas onde estão presentes o termo isolado e o termo junto ao contexto, foi desenvolvida uma equação que demonstra a relação do termo com o texto em questão. O valor resultante dessa equação indicará a relevância do termo para o tema do documento. Ou seja, obtêm-se um percentual de afinidade entre o termo extraído e o assunto abordado no texto, conforme a Equação 4.1.

$$Relação = \frac{N^o\ Resultados\ COM\ Contexto}{N^o\ Resultados\ SEM\ Contexto} \tag{4.1}$$

Essa representação nos permite abordar os termos como sendo mais relacionados (maior percentual) ou menos relacionados (menor percentual) ao cenário do texto.

Considerando A o conjunto de todas as páginas indexadas pelo Google que contém o termo “x” (termo isolado) e considerando B o conjunto de todas as páginas indexadas pelo Google que contém o termo “y” (contexto). É possível inferir que o conjunto C, que contém todas as páginas indexadas pelo Google, que possuem o termo “x” e “y”, é um subconjunto de A e B. Sendo assim, podemos afirmar que o tamanho do conjunto C é menor ou igual ao de A e B, como pode ser visto no exemplo da figura 4.1. Isso resulta no valor máximo possível para o relacionamento do termo (valor 1). Quanto mais próximo de 1 é esse valor, mais próximo o termo está do contexto, o que significa esse ser mais relevante para o assunto abordado no documento.

Figura 4.1: Teoria dos Conjuntos dos Termos e Contexto



Fonte: Autor

4.3 Reordenamento dos Termos

A proposta final desse método consiste em reordenar os termos extraídos pelo minerador Sobek, de forma que os mais relevantes (os que têm maior valor na Equação 4.1) para o contexto do documento recebam mais destaque quando forem apresentados ao usuário. Na Tabela 4.1 é mostrado o caso de um texto sobre a Segunda Guerra Mun-

dial, onde há todos os termos extraídos do Sobek na coluna à esquerda. Os termos estão definidos pela ordem do nº de aparições (frequência) no documento, que fica na coluna à direita. Consideramos também que o contexto extraído (os dois termos mais frequentes retornados pelo Sobek) é formado pelos termos “guerra” e “Alemanha”, como pode-se ver na Tabela 4.1.

Tabela 4.1: Termos em ordem retornada pelo Sobek

Termo	**Frequência**
guerra	28
alemanha	10
hitler	10
aliados	9
nazistas	9
fascistas	8
eua	7
dia	5
país	5
conflito	4
grande	4
judeus	4
mortos	4
polônia	4
"potências do eixo"	4
urss	4

Fonte: Autor

A partir dos termos retornados pelo Sobek, o contexto é definido e são então ordenados os demais termos conforme a relação da Equação 4.1. Com esse valor, ocorre o reordenamento dos termos (exceto os dois primeiros que formam o contexto e permanecem como os principais termos extraídos), destacando aqueles que obtiveram um valor maior de relacionamento com o tema do documento.

Na Tabela 4.2, temos a mesma Tabela 4.1 após aplicação do processo de reordenamento. Nas tabelas daqui pra frente, setas azuis indicam termos que subiram de posição, vermelhas as que caíram e com traço as que não mudaram.

Como se pode perceber, com o contexto definido, a relação do termo com o assunto do documento prioriza uma ordem diferente daquela que o minerador considera. Em alguns casos, a relação é forte e mais próxima do valor 1 (100%), enquanto outros não têm uma relação tão forte e o valor se aproxima de zero. Assim, esse processo considera que termos como “potências do eixo”, “nazistas” e “judeus” são mais relevantes em relação ao tema “Segunda Guerra Mundial” (a partir do contexto “guerra” e ”Alemanha”) do que termos como “país”, “dia” e “grande”.

Tabela 4.2: Termos reordenados com buscas no Google com Contexto

Termo	**Relação**
guerra	-
alemanha	-
"potências do eixo"↑	0,5188679245
nazistas ↑	0,2653333333
judeus ↑	0,1115044248
fascistas –	0,0633858268
polônia ↑	0,0511351351
urss ↑	0,0284375000
conflito ↑	0,0283443709
aliados ↓	0,0199206349
mortos ↑	0,0163837638
eua ↓	0,0048387097
hitler ↓	0,0045454545
país ↓	0,0007587169
dia ↓	0,0004415094
grande ↓	0,0003266667

Fonte: Autor

Para explicar o resultado obtido, é possível observar o resultado para dois termos realocados após o processo de identificação de contexto, os termos “nazistas” e “dia”. O primeiro termo a ser analisado (“nazistas”) foi considerado mais relevante após o processo de reordenação. Grande parte dos documentos que contêm o termo “nazistas” também contêm os termos “guerra” e “Alemanha". Ou seja, o termo “nazistas” está bastante relacionado ao contexto encontrado. Por outro lado, não ocorre o mesmo com o termo “dia”, pois o universo de documentos que contém este termo sozinho é muito maior do que aquele que contém o termo associado ao contexto (“guerra” e ”Alemanha”). Ou seja, isso indica que a relação entre o termo e o contexto é baixa e este termo é realocado para posições inferiores na lista.

É importante ressaltar aqui que as buscas ocorrem de forma a garantir que os termos inseridos no campo de busca estejam presentes em todos os documentos resultantes. Para isso, é utilizada a opção “ao pé da letra” (*verbatim*, em inglês) na URL de busca do Google, de forma a garantir que a palavra inserida não sofrerá alterações durante a pesquisa. Ela é utilizada também para garantir que todas as palavras inseridas estejam presentes nos resultados da busca.

A busca de termos no Google ocorre com a string URL definida abaixo:

String url = "https://www.google.com.br/search?hl=pt-br&tbs=li:1&q=";

Onde o trecho *“https://www.google.com.br/search?”* indica a URL da busca no Google; *“hl=pt-br”* indica que serão pesquisadas apenas as páginas em português do

Brasil; *"tbs=li:1"* indica que a busca será feita "ao pé da letra", como explicado anteriormente; e a última parte, *"q="*, é onde vão estar concatenados os termos que queremos buscar, ficando *"q=termo"* (quando for buscado isoladamente) ou *"q=contexto1+contexto2+termo"* (quando for buscado junto ao contexto).

4.4 Trabalhos Relacionados

Com a ideia de basear-se em fontes externas para validar os resultados de MT - como o Google e a Internet, ontologias, um *corpus* com um conjunto de documentos específico - é possível encontrar alguns trabalhos relacionados.

O artigo de Chuang e Chien (CHUANG; CHIEN, 2004) traz similaridades com a proposta aqui apresentada, no qual é mostrada uma forma de utilizar buscas na *Web*, através de mecanismos de busca conhecidos, para ampliar e enriquecer o domínio de alguns trechos de textos menores, ou que tipicamente não contêm informações suficientes para que se possa extrair dados adequados e confiáveis. Assim, a ideia base é explorar a Web para adequar contextos de segmentos textuais menores do texto e parear tais frases àquelas extraídas de grandes quantidades de páginas indexadas.

No trabalho de Hearst e Pedersen (HEARST; PEDERSEN, 1996), sobre reexaminar a hipótese de clusterização dos resultados da RI, é apresentada uma proposta similar. A partir dos termos identificados em um documento, é feita uma busca em um *corpus* que retorna uma quantidade de documentos similares, ordenados por um *rank*, que podem também ser divididos em *clusters*, se solicitado pelo usuário. Um exemplo prático é aquele no qual é apresentado o documento "*Is the automobile industry making an honest effort to develop and produce an electric-powered automobile?*", do qual são extraídos os termos "auto", "car", "vehicle" e "electric", para com esses realizar uma busca em um corpus grande (>3GB), composto de revistas, jornais online e documentos governamentais, resultando em um conjunto dos melhores 100, 250, 500 ou 1000 resultados, ordenados por relevância, e que podem ser classificados em 5 *clusters* diferentes (podendo ser menos, mas sempre levando em conta o desejo do usuário).

Outro trabalho que mostra *clusterização* de documentos para facilitar a procura em mecanismos de buscas da Internet é o de Zamir e Etzioni (ZAMIR; ETZIONI, 1998). Nesse artigo, é introduzida a técnica *Suffix Tree Clustering (STC)*, a qual utiliza um motor de buscas online para prover resultados sobre os quais será aplicado o algoritmo STC e, com isso, *clusterizar* esses de forma relevante e rápida.

Já em Beeferman e Berger (BEEFERMAN; BERGER, 2000) é mostrado também um algoritmo para *clusterizar* documentos buscados na Internet, baseado na técnica de *clusterização hierárquica de aglomeração (HAC)*, a qual, simplesmente, encontra os dois documentos mais semelhantes e os agrupa. A diferença proposta é não se basear no conteúdo desses documentos ao procurá-los - tendo em vista que já existiam dois bilhões de páginas acessíveis por motores de busca na Internet na data em que o artigo foi publicado, em 2000 - porque demoraria muito tempo ao considerar uma quantidade grande de páginas. Assim, ao invés do conteúdo, se utilizam informações de coocorrência em várias transações para orientar seus agrupamentos. Por exemplo, dois usuários distintos podem realizar buscas como “chitas” e “gatos selvagens”, cada um, e chegarem na mesma página dentre as oferecidas pelo buscador, o que sugere que as buscas são semelhantes. Da mesma forma que se pode deduzir que, quando dois usuários distintos fazem a mesma busca e chegam em páginas diferentes, essas páginas estão fortemente relacionadas (ex: www.fundz.com e www.mutualfundsite.com).

Também é visto em Wen, Nie e Zhang (WEN; NIE; ZHANG, 2002) uma abordagem de agrupamento das buscas realizadas online, de forma a manter os registros dessas para cada usuário. Assim, pode-se cruzar as buscas feitas e os documentos escolhidos para leitura e, como no trabalho anterior, é possível relacionar buscas diferentes que cheguem em um mesmo documento, assim como, relacionar um conjunto de páginas que são frequentemente buscadas com termos distintos.

Diferente desses trabalhos citados, a metodologia aqui proposta explora os resultados das pesquisas online para identificar uma relação entre os termos previamente extraídos por um minerador com o contexto do documento. Assim, é possível identificar a ordem dos principais termos dentro do tema abordado pelo texto - separando aqueles que são mais fortemente ligados daqueles que são mais genéricos ou não tão relacionados com o âmbito em questão.

5 EXPERIMENTOS E RESULTADOS

Como forma de validação do processo apresentado, 16 especialistas na área de informática na educação participaram de um estudo para ordenação de termos identificados como importantes em textos. Todos os especialistas ou estavam cursando o doutorado em informática na educação, ou já tinham concluído seus cursos de doutoramento. Foram selecionados 3 textos na área de especialização dos participantes, textos tratando de temas relativos à utilização da informática e novas tecnologias na educação. A partir desses, os testes ocorreram da seguinte forma:

- Cada participante recebeu 3 formulários do Google, cada um contendo um dos 3 textos selecionados;
- Cada formulário também possuía uma imagem com os termos extraídos do texto em questão pelo Sobek. A imagem mostrava de forma embaralhada os termos extraídos, todos com mesmo tamanho para que não houvesse nenhuma influência da imagem sobre percepção de importância do termo no texto;
- Cada formulário continha tantos campos em branco quanto o número de termos mostrados na imagem;
- Para cada formulário, o participante devia ler o texto e ordenar os termos da imagem de acordo com a sua percepção de relevância no texto.

Ao final da atividade, cada participante tinha ordenado os termos de acordo com sua relevância para cada um dos três textos. Essa ordenação serviu para identificar se o processo aqui proposto de fato é capaz de produzir melhor ordenamento de termos que o processo de mineração de textos da ferramenta Sobek, que utiliza apenas dados estatísticos relacionados à frequência dos termos no texto.

5.1 Métricas de Validação

Para avaliar os dados obtidos dos formulários citados, foram utilizados dois tipos de métricas. A primeira é um dos métodos citados na seção de avaliação de resultados do capítulo 2, *Precision@K*, que indica uma precisão do algoritmo em detectar os termos mais relevantes. A segunda é o método *Spearman's Rank-Order Correlation*, utilizado para avaliação com um coeficiente de correlação entre dois rankings.

Para ambas as métricas, utilizou-se as ordens dos termos fornecidas nas respos-

tas dos testes, juntamente com as ordens dadas pelo algoritmo de reordenamento e pelo minerador Sobek. Assim, para a resposta de cada participante, foi feita a comparação com a lista ordenada pelo algoritmo, onde foi medida a precisão com *Precision@K* e a correlação com *Spearman's Rank-Order Correlation*. A partir dos valores obtidos, são calculados a média e o desvio padrão entre todos os valores retornados com a precisão, e depois com a correlação.

5.1.1 Precision@K

Esta é uma métrica que computa um percentual dos melhores K documentos, dentro daqueles retornados de uma coleção por um método de RI, ignorando aqueles em posição abaixo de K. Para utilizá-la com o reordenamento dos termos, pode-se considerar a precisão ao comparar duas listas em ordem diferentes, calculando quando os termos das duas listas estão presentes nas posições de 1 à K. Por exemplo:

A= pedra, areia, cascalho, madeira, ferro, arame
B= cascalho, madeira, pedra, areia, ferro, arame

Se considerarmos $K = 3$, das três primeiras posições, e que 2 palavras estão entre essas posições em ambas as listas ("pedra"e "cascalho"), não importando suas posições exatas, a precisão p então é definida como $p = 2$.

Para os testes de validação do algoritmo, foi utilizado o valor de K como sendo a metade do tamanho da lista *(n/2)*, tendo em vista a necessidade de se criar um limiar (*threshold*) em que possam ser desconsiderados os termos mais genéricos, que não agregam tanta representatividade no texto. No caso de o número de termos ser ímpar, ocorre o arredondamento para cima (por exemplo, caso hajam 15 termos, o valor de K é 8).

5.1.2 Spearman's Rank-Order Correlation

A ideia da correlação de Spearman é calcular a "distância" entre termos de duas listas com os mesmos elementos, através de um coeficiente ρ, que pode variar no intervalo:

$$-1 \leq \rho \leq 1$$

Assim, pode-se definir intervalos onde se mede a chamada “força de correlação”:

- 0.00 - 0.19 $\rightarrow$ "muito fraco"
- 0.20 - 0.39 $\rightarrow$ ”fraca”
- 0.40 - 0.59 $\rightarrow$ ”média”
- 0.60 - 0.79 $\rightarrow$ ”forte”
- 0.80 - 1.00 $\rightarrow$ ”muito forte”

A seguir são apresentadas duas listas de termos para exemplo:

A = árvore, papel, madeira, folha, sol, chuva
B = papel, árvore, folha, sol, madeira, chuva

Com isso, calcula-se a distância *D* de A para B, onde *D* é definido em valores absolutos, em que soma-se 1 a cada posição distante de um termo de A para o mesmo em B:

D(árvore) = 1
D(papel) = 1
D(madeira) = 2
D(folha) = 1
D(sol) = 1
D(chuva) = 0

Após, é utilizada a Equação 5.1 para calcular o coeficiente ρ de correlação:

$$\rho = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{n(n^2-1)} \tag{5.1}$$

Considerando o exemplo anterior, é preciso obter o quadrado do somatório das distâncias:

$$\sum_{i=1}^{N} d_i^2 = 1+1+4+1+1+0 = 8$$

Substituindo isso na Equação 5.1, e considerando que n=6, do nº de elementos em cada lista:

$$\rho = 1 - \frac{6x8}{6x(6^2-1)} = 1 - \frac{48}{210} \cong 0,77$$

Para um valor de, aproximadamente, 0.77, a correlação indicada é alta, ou seja, as listas têm elementos colocados em posições aproximadas.

5.2 Resultados

A seguir, são mostradas as Tabelas 5.1, 5.2 e 5.3. Em cada uma, é feita a comparação da ordem retornada pelo minerador Sobek (com base na frequência de aparições do termo) com a da ordem dada pelo algoritmo (com base no relacionamento dos termos com o contexto) indicando quais termos foram reposicionados. Em seguida, para ambos os métodos de validação citados e para cada texto, é apresentada uma tabela, primeiro com a precisão K=n/2 (Tabelas 5.4, 5.5 e 5.6) e depois com o método de Spearman (Tabelas 5.7, 5.8 e 5.9). Em ambos os casos, é indicado o participante com a precisão ou correlação da sua lista quando comparada com a lista dada pelo algoritmo, e quando comparada com a dada pelo Sobek. Por fim, é mostrada a média e o desvio padrão entre todos os valores.

Tabela 5.1: Ordem dos Termos no Texto 1

Ordem do Sobek	**Frequência**	**Ordem do Algoritmo**	**Relação**
alunos	**5**	**alunos**	*Contexto*
digital	**5**	**digital**	*Contexto*
educação	5	“novas tecnologias” ↑	0.236410
"novas tecnologias"	5	tecnologias ↑	0.012460
parte	5	professores ↑	0.009476
aula	4	educação ↓	0.002766
escola	4	aula ↓	0.002686
faz	4	escola ↓	0.002412
professores	4	faz ↓	0.001852
tecnologias	4	uso ↑	0.000477
uso	4	parte ↓	0.000325

Fonte: Autor

Tabela 5.2: Ordem dos Termos no Texto 2

Ordem do Sobek	**Frequência**	**Ordem do Algoritmo**	**Relação**
família	**5**	**família**	*Contexto*
videogames	**5**	**videogames**	*Contexto*
crianças	4	“sintomas de vício” ↑	0.114124
jogos	4	viciados ↑	0.015644
alguns	3	efeitos ↑	0.007508
azar	3	alguns ↓	0.004864
efeitos	3	negativos ↑	0.004018
“Estados Unidos”	3	pesquisa ↑	0.003173
games	3	crianças ↓	0.002729
negativos	3	"Estados Unidos"↓	0.001879
pesquisa	3	jogos ↓	0.001792
“sintomas de vício”	3	azar ↓	0.001255
viciados	3	games ↓	0.000141

Fonte: Autor

Tabela 5.3: Ordem dos Termos no Texto 3

Ordem do Sobek	**Frequência**	**Ordem do Algoritmo**	**Relação**
“informática na educação”	**12**	**“informática na educação”**	*Contexto*
computação	**9**	**computação**	*Contexto*
conhecimento	5	conhecimento –	0.000806
educação	5	educação –	0.000478
outros	5	vezes ↑	0.000288
áreas	5	áreas –	0.000268
cada	4	distintas ↑	0.000220
deste	4	deste –	0.000204
diferentes	4	diferentes –	0.000109
distintas	4	outros ↓	0.000104
mundo	4	cada ↓	0.000054
vezes	4	mundo ↓	0.000039

Fonte: Autor

5.2.1 Resultados do Precision@K

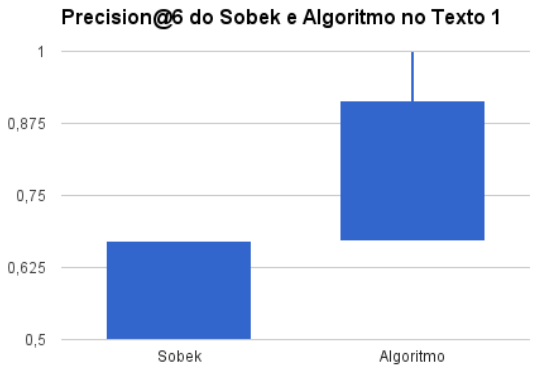
Em todas as tabelas daqui pra frente, foram destacados (com as células pintadas em laranja) apenas os resultados em que a ordenação piorou em comparação com a do minerador Sobek, o que significa que, após a reordenação do algoritmo, a ordem está mais distante daquela apontada pelo participante da pesquisa. No primeiro texto, foram obtidas, com *Precision@6*, as precisões mostradas na Tabela 5.4; no segundo texto, com Precision@7, foram obtidas as precisões mostradas na Tabela 5.5; e no terceiro texto, novamente com Precision@6, foram obtidas as precisões mostradas na Tabela 5.6.

Tabela 5.4: Precisões com Precision@6 dos Testes no Texto 1

Participante	Precisão do Algoritmo	Precisão do Sobek
#1	1.00	0.67
#2	0.67	0.67
#3	0.67	0.50
#4	0.83	0.50
#5	1.00	0.67
#6	0.83	0.50
#7	0.67	0.50
#8	0.67	0.50
#9	1.00	0.67
#10	0.67	0.67
#11	1.00	0.67
#12	0.83	0.50
#13	0.67	0.50
#14	0.83	0.50
#15	0.83	0.50
#16	0.67	0.67
–	**Média: 0.802083**	**Média: 0.572916**
–	**Desvio Padrão: 0.134612**	**Desvio Padrão: 0.082679**

Fonte: Autor

Figura 5.1: Box Plot da Tabela 5.4



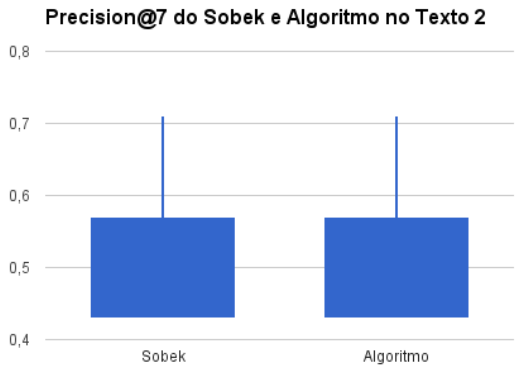
Fonte: Autor

Tabela 5.5: Precisões com Precision@7 dos Testes no Texto 2

Participante	Precisão do Algoritmo	Precisão do Sobek
#1	0.57	0.57
#2	0.43	0.57
#3	0.57	0.57
#4	0.71	0.71
#5	0.43	0.43
#6	0.57	0.43
#7	0.43	0.57
#8	0.43	0.57
#9	0.57	0.57
#10	0.71	0.43
#11	0.57	0.57
#12	0.43	0.43
#13	0.57	0.43
#14	0.57	0.57
#15	0.57	0.43
#16	0.57	0.71
–	**Média: 0.5446428**	**Média: 0.5357142**
–	**Desvio Padrão: 0.090615**	**Desvio Padrão: 0.09449**

Fonte: Autor

Figura 5.2: Box Plot da Tabela 5.5



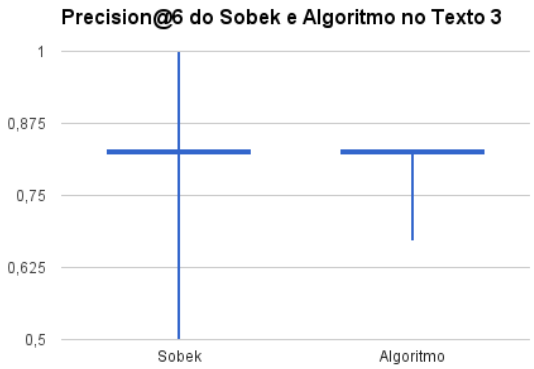
Fonte: Autor

Tabela 5.6: Precisões com Precision@6 dos Testes no Texto 3

Participante	Precisão do Algoritmo	Precisão do Sobek
#1	0.83	0.83
#2	0.83	0.83
#3	0.83	0.83
#4	0.83	0.83
#5	0.83	0.83
#6	0.83	0.83
#7	0.83	0.83
#8	0.83	0.83
#9	0.83	0.83
#10	0.67	0.50
#11	0.67	0.67
#12	0.83	0.83
#13	0.83	0.83
#14	0.83	0.83
#15	0.83	0.83
#16	0.83	1.00
–	**Média: 0.812500**	**Média: 0.812500**
–	**Desvio Padrão: 0.055119**	**Desvio Padrão: 0.099913**

Fonte: Autor

Figura 5.3: Box Plot da Tabela 5.6



Fonte: Autor

Utilizando K=n/2, observa-se a melhora mais significativa no primeiro texto, no qual a média é consideravelmente maior do que a apresentada pelo Sobek. Também é

possível observar que todas as precisões da lista desenvolvida pelo método deste trabalho foram maiores ou iguais às do minerador Sobek, quando comparadas de forma individual. Nesse caso, se pode afirmar que a *Precision@6* consegue apontar os 6 principais termos extraídos com mais de 80% de confiança, sendo esses 6 termos muito próximos aos considerados principais pelos participantes da pesquisa. Aqui, o algoritmo funciona bem pois tem sucesso desconsiderando termos como "faz", "uso" e "parte", os quais também são colocados nas últimas posições pela grande maioria dos participantes na pesquisa.

Já no segundo texto, com *Precision@7*, os resultados foram muito próximos aos do Sobek, com uma leve melhora. Para quatro listas, os valores que o minerador obteve apresentam uma precisão maior do que o algoritmo (assinalados na Tabela 5.5). Na maioria das respostas dos participantes, pode-se notar que termos como "alguns" ou "Estados Unidos" aparecem nas últimas posições de relevância, enquanto o algoritmo coloca esses termos nas posições 6 e 10 da lista (posições mais intermediárias), respectivamente.

Por último, no terceiro texto, as médias são rigorosamente iguais e os valores de cada resposta são também praticamente iguais, sendo diferentes em apenas dois casos. Com isso, pode-se deduzir que as listas do algoritmo e do Sobek são muito semelhantes, ou seja, o minerador já tem, com *Precision@6*, um resultado muito semelhante àqueles encontrados nas respostas da pesquisa. Aqui é importante ressaltar que o algoritmo não piora os resultados ao aplicar o reordenamento de termos utilizando as buscas com contexto. Os resultados com a ordem apresentada pelo minerador já podem ser considerados muito bons (média > 80%), e a utilização do algoritmo aqui continua a garantir que a ordem é relevante, sem grandes alterações nas posições dos termos.

5.2.2 Resultados de Spearman's Rank-Order Correlation

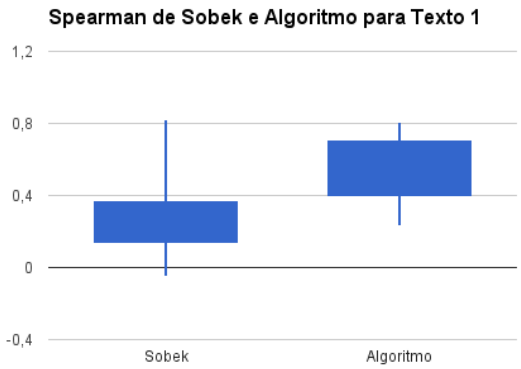
A seguir, são apresentadas as tabelas contendo os resultados considerando a correlação de Spearman. Após, cada correlação é analisada em relação à lista apresentada pelo Sobek e a lista reordenada pelo algoritmo. A Tabela 5.7 apresenta a correlação de Spearman medida com cada resposta no primeiro texto; a Tabela 5.8 para o segundo texto; e a Tabela 5.9 para o terceiro.

Tabela 5.7: Coeficientes de Correlação para o Texto 1

Participante	Coef. de Correlação do Algoritmo	Coef. de Correlação do Sobek
#1	0.70	0.17
#2	0.37	0.34
#3	0.32	-0.05
#4	0.71	0.16
#5	0.71	0.39
#6	0.41	0.02
#7	0.23	-0.03
#8	0.31	0.05
#9	0.74	0.46
#10	0.47	0.35
#11	0.81	0.59
#12	0.69	0.37
#13	0.54	0.82
#14	0.71	0.23
#15	0.54	0.16
#16	0.35	0.22
–	**Média: 0.538068**	**Média: 0.221022**
–	**Desvio Padrão: 0.181909**	**Desvio Padrão: 0.179148**

Fonte: Autor

Figura 5.4: Box Plot da Tabela 5.7



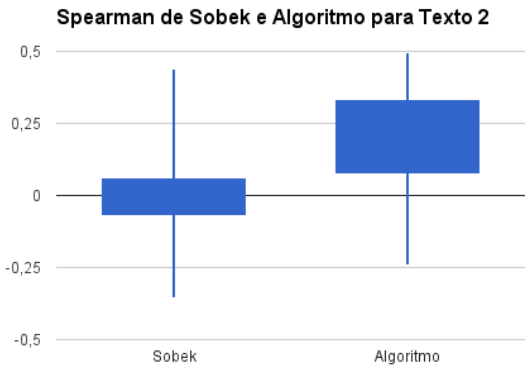
Fonte: Autor

Tabela 5.8: Coeficientes de Correlação para o Texto 2

Participante	Coef. de Correlação do Algoritmo	Coef. de Correlação do Sobek
#1	-0.033	-0.159
#2	0.330	0.330
#3	0.093	-0.005
#4	0.401	0.082
#5	-0.242	0.066
#6	0.472	-0.038
#7	0.252	0.033
#8	0.104	0.038
#9	0.494	0.439
#10	-0.055	-0.231
#11	0.286	0.038
#12	0.335	0.043
#13	0.317	-0.357
#14	0.027	0.060
#15	0.121	-0.291
#16	0.225	0.022
–	**Média: 0.195741**	**Média: 0.004464**
–	**Desvio Padrão: 0.198752**	**Desvio Padrão: 0.196108**

Fonte: Autor

Figura 5.5: Box Plot da Tabela 5.8



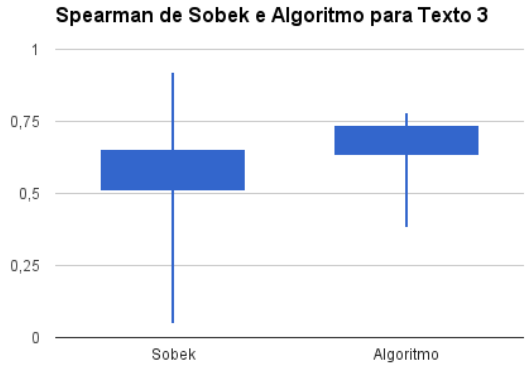
Fonte: Autor

Tabela 5.9: Coeficientes de Correlação para o Texto 3

Participante	Coef. de Correlação do Algoritmo	Coef. de Correlação do Sobek
#1	0.78	0.65
#2	0.68	0.52
#3	0.73	0.65
#4	0.61	0.58
#5	0.61	0.60
#6	0.66	0.51
#7	0.70	0.64
#8	0.73	0.60
#9	0.76	0.45
#10	0.47	0.05
#11	0.38	0.14
#12	0.69	0.67
#13	0.70	0.66
#14	0.64	0.51
#15	0.75	0.70
#16	0.75	0.92
–	**Média: 0.665209**	**Média: 0.555506**
–	**Desvio Padrão: 0.103954**	**Desvio Padrão: 0.200416**

Fonte: Autor

Figura 5.6: Box Plot da Tabela 5.9



Fonte: Autor

Os resultados dos testes com correlação de Spearman também indicam que a reordenação dos termos pelo algoritmo apresenta resultados melhores que a ordenação apre-

sentada pelo Sobek. Isso significa que a aplicação do algoritmo sobre a lista de termos coloca esses em uma ordem mais próxima àquelas dadas pelos participantes da pesquisa.

No primeiro texto pode-se perceber esse aprimoramento na ordenação dos termos de forma acentuada, onde tanto a média de correlacionamento quanto os coeficientes individualmente calculados são maiores quase que na totalidade. Em apenas uma lista da pesquisa se obteve um coeficiente de relação menor do que a apontada pelo Sobek.

Já no segundo texto, a correlação média fica um pouco baixa ($< 20\%$), ainda assim sendo melhor que aquela indicada pelo Sobek ($< 1\%$), onde apenas duas listas indicam uma correlação mais forte com a ordenação do minerador. Aqui, pode-se concluir que a ordem dada pelo minerador é bastante divergente daquelas indicadas pelos participantes, e que o algoritmo consegue colocar os termos em uma ordem que está mais de acordo com a mostrada pelos testes.

No terceiro texto, a média é um pouco maior também para a ordenação do algoritmo, com apenas uma lista tendo correlação maior com a ordem do minerador, o que é importante e confirma a melhora que ocorre. Pode-se afirmar no final o que já podia ser notado com os resultados com *Precision@K*:

- No primeiro texto é onde ocorre uma melhora mais significativa entre os três. Nele, observa-se uma aumento, com *Precision@6*, de mais de 20%, enquanto que a correlação de Spearman mostra uma melhora na média do coeficiente de mais de 30%;
- No segundo texto, para *Precision@7*, o resultado é mediano tanto para a ordenação de termos do Sobek quanto para a do algoritmo, entretanto o coeficiente de correlação de Spearman é baixo para as duas, apesar de ser maior com o algoritmo. Esses resultados apontam que os primeiros sete termos das duas listas são semelhantes (o que é mostrado pelo *Precision@K*), independente da ordem. Já em Spearman, como as posições são comparadas e as distâncias entre os termos são calculadas, pode-se deduzir que essas sete palavras na primeira metade da lista estão em posições diferentes entre as apontadas pelo minerador e pelo algoritmo. Como a média do coeficiente de Spearman no algoritmo é maior que a do Sobek, também pode-se afirmar que a ordem entre os sete primeiros é mais próxima daquelas mostradas nas pesquisas após utilizar o algoritmo do que a ordem do minerador;
- No terceiro e último texto, os resultados são iguais e altos com *Precision@6* ($> 80\%$), o que indica que os seis primeiros termos são muito semelhantes entre a lista do algoritmo, a lista do Sobek e as listas dos participantes. Semelhante ao caso do segundo texto, o que muda mais entre as listas é a ordem dessas seis palavras

iniciais, pois a média do coeficiente de correlação é diferente: para o Sobek a média é de 55%, enquanto para o algoritmo é de 66%. Ou seja, o algoritmo consegue encontrar uma ordenação mais de acordo com a escolhida pelos participantes do que o minerador.

6 CONCLUSÃO E TRABALHOS FUTUROS

Após avaliar os resultados do algoritmo proposto, é possível perceber que a utilização do mecanismo de busca do Google para identificar termos melhor inseridos em determinado contexto é promissora. A ideia de criar um processo rápido, que possa utilizar buscas online para complementar as informações do texto mostra-se relativamente simples. Com ele, se utilizam os próprios termos obtidos do minerador para definir o contexto, e a partir disso encontra-se um valor de relacionamento para os demais termos, sem que nenhuma informação seja perdida ou o processo se torne mais lento. Na prática, os resultados dos testes mostram efetivamente que existe uma melhora significativa, em ambas as métricas utilizadas.

Ainda assim, é possível planejar algumas melhorias neste processo. O minerador Sobek poderia dispor, por exemplo, de uma opção selecionável como "Utilizar buscas online e reordenar os termos extraídos". O usuário poderia determinar se desejaria ou não realizar a mineração com as buscas online, tendo-se em vista que, apesar de ser um processo ágil comparado a outros que fazem MT, fazer buscas deste tipo com e sem contexto para cada termo, pode levar em média 01 min 10 seg a mais, o que poderia onerar um pouco o processo de mineração.

No futuro, planeja-se ainda obter e testar a utilização de repositórios específicos, nos quais se possa fazer buscas com o contexto e determinar a relação dos termos em *corpus* ou ontologias especializadas no assunto abordado. Nestas situações o contexto seria determinado pelos principais termos dados pelo minerador, mas o repositório seria mais específico para o tema determinado. Ainda assim, seria interessante testar a utilização de mais termos para formar o contexto, ao invés de apenas dois termos - como no estudo aqui apresentado.

Repositórios:

ContextMining (JAVA):

`https://github.com/lsjacobs/ContextMining`

Spearman's Rank Order Correlation (JAVA):

`https://github.com/lsjacobs/Spearman-Rank-Order-Correlation`

Precision at K (Python):

`https://github.com/lsjacobs/Precision_at_K`

Textos e Respostas dos Testes:

`https://drive.google.com/open?id=0B6DKNvfpiXWfajdJTV93UFQ5eDQ`

REFERÊNCIAS

ACOSTA, O.; BEHAR, P.; REATEGUI, E. B. Content recommendation in an inquiry-based learning environment. In: **Frontiers in Education Conference (FIE), 2014 IEEE**. [S.l.: s.n.], 2014. p. 1–6.

AGGARWAL, C.; ZHAI, C. **Mining Text Data**. Springer New York, 2012. ISBN 9781461432234. Available from Internet: <https://books.google.com.br/books?id=vFHOx8wfSU0C>.

AGGARWAL, C. C.; WANG, H. Text mining in social networks. In: **Social Network Data Analytics**. [S.l.]: Springer, 2011. p. 353–378.

AZEVEDO, B. F.; REATEGUI, E.; BEHAR, P. A. Analysis of the relevance of posts in asynchronous discussions. **Interdisciplinary Journal of Knowledge and Learning Objects**, v. 10, p. 107–121, 2014. ISSN 0306-4573.

AZEVEDO, B. F. T.; BEHAR, P. A.; REATEGUI, E. B. Análise das mensagens de fóruns de discussão através de um software para mineração de textos. In: **Anais do Simpósio Brasileiro de Informática na Educação**. [S.l.: s.n.], 2011. v. 1, n. 1.

BARION, E. C. N.; LAGO, D. Mineração de textos. **Revista de Ciências Exatas e Tecnologia**, v. 3, n. 3, p. 123–140, 2015.

BEEFERMAN, D.; BERGER, A. Agglomerative clustering of a search engine query log. In: ACM. **Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.], 2000. p. 407–416.

BOLASCO, S. et al. Understanding text mining: A pragmatic approach. In: **Knowledge mining**. [S.l.]: Springer, 2005. p. 31–50.

BULEGON, H.; MORO, C. M. C. Mineração de texto e o processamento de linguagem natural em sumários de alta hospitalar. **Journal of Health Informatics**, v. 2, n. 2, 2010.

CHEIN, M.; MUGNIER, M.-L. **Graph-based knowledge representation: computational foundations of conceptual graphs**. [S.l.]: Springer Science & Business Media, 2008.

CHEN, H. Knowledge management systems: a text mining perspective. Knowledge Computing Corporation, 2001.

CHUANG, S.-L.; CHIEN, L.-F. A practical web-based approach to generating topic hierarchy for text segments. In: ACM. **Proceedings of the thirteenth ACM international conference on Information and knowledge management**. [S.l.], 2004. p. 127–136.

CORDEIRO, A. D. et al. Gerador inteligente de sistemas com auto-aprendizagem para gestão de informações e conhecimento. Florianópolis, SC, 2005.

COSTA, P. d. S. C.; REATEGUI, E. B. Oportunidades de letramento através de mineração textual e produção de fanfictions. **Revista Brasileira de Linguística Aplicada**, SciELO Brasil, v. 12, n. 4, p. 835–859, 2012.

DAMASCENO, F. et al. Supporting teleconsulting with text mining: Continuing professional development in the telehealthrs project. In: **Collaboration and Technology**. [S.l.]: Springer International Publishing, 2014, (Lecture Notes in Computer Science, v. 8658). p. 97–104.

EBECKEN, N. F.; LOPES, M. C. S.; COSTA, M. C. Mineração de textos. **Sistemas inteligentes: fundamentos e aplicações. São Carlos: Manole**, p. 337–370, 2003.

FELDMAN, R.; SANGER, J. **The text mining handbook: advanced approaches in analyzing unstructured data**. [S.l.]: Cambridge University Press, 2007.

GONZALEZ, M.; LIMA, V. L. S. Recuperação de informação e processamento da linguagem natural. In: **XXIII Congresso da Sociedade Brasileira de Computaçao**. [S.l.: s.n.], 2003. v. 3, p. 347–395.

GUPTA, V.; LEHAL, G. S. A survey of text mining techniques and applications. **Journal of emerging technologies in web intelligence**, v. 1, n. 1, p. 60–76, 2009.

HEARST, M. A.; PEDERSEN, J. O. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In: ACM. **Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.], 1996. p. 76–84.

HOTHO, A.; NÜRNBERGER, A.; PAASS, G. A brief survey of text mining. In: **Ldv Forum**. [S.l.: s.n.], 2005. v. 20, n. 1, p. 19–62.

HOTHO, A.; STAAB, S.; STUMME, G. Ontologies improve text document clustering. In: IEEE. **Data Mining, 2003. ICDM 2003. Third IEEE International Conference on**. [S.l.], 2003. p. 541–544.

IGLESIAS, A. V. Graph representation of documents content and its suitability for text mining tasks. Institutt for datateknikk og informasjonsvitenskap, 2011.

JOLLIFFE, I. **Principal component analysis**. [S.l.]: Wiley Online Library, 2002.

KLEMANN, M.; REATEGUI, E.; LORENZATTI, A. O emprego da ferramenta de mineração de textos sobek como apoio à produção textual. In: **Anais do Simpósio Brasileiro de Informática na Educação**. [S.l.: s.n.], 2009. v. 1, n. 1.

KLEMANN, M.; REATEGUI, E.; RAPKIEWICZ, C. Análise de ferramentas de mineração de textos para apoio a produção textual. In: **Anais do Simpósio Brasileiro de Informática na Educação**. [S.l.: s.n.], 2011. v. 1, n. 1.

KOHAVI, R. et al. Lessons and challenges from mining retail e-commerce data. **Machine Learning**, Springer, v. 57, n. 1-2, p. 83–113, 2004.

LANGA, N. R. et al. Apoio ao letramento infantil por meio de construção de narrativas empregando uma ferramenta de mineração textual. **RENOTE**, v. 11, n. 3, 2012.

MACEDO, A. L. et al. Using text-mining to support the evaluation of texts produced collaboratively. In: **WCCE**. [S.l.]: Springer, 2009. (IFIP Advances in Information and Communication Technology, v. 302), p. 368–377.

MALIK, R. **CONAN: Text Mining in the Biomedical Domain**. [S.l.]: Utrecht University, 2006.

MINER, G. et al. **Practical text mining and statistical analysis for non-structured text data applications. 2012**. [S.l.]: Elsevier Academic Press, 2012.

MORAIS, E. A. M.; AMBRÓSIO, A. P. L. Mineraçao de textos. **Relatório Técnico–Instituto de Informática (UFG)**, 2007.

NAVATHE, S. B.; RAMEZ, E. Data warehousing and data mining. **Fundamentals of Database Systems**, p. 841–872, 2000.

NOVAK, J. D.; CAÑAS, A. J. The theory underlying concept maps and how to construct them. **Florida Institute for Human and Machine Cognition**, v. 1, p. 2006–2001, 2006.

PERNA, C. L.; DELGADO, H. K.; FINATTO, M. J. **Linguagens Especializadas em Corpora: modos de dizer e interfaces de pesquisa**. [S.l.]: EDIPUCRS, 2010.

PINHO, I. da C. et al. The use of text mining to build a pedagogical agent capable of mediating synchronous online discussions in the context of foreign language learning. In: IEEE. **2013 IEEE Frontiers in Education Conference (FIE)**. [S.l.], 2013. p. 393–399.

PRADO, H. A. D. **Emerging Technologies of Text Mining: Techniques and Applications: Techniques and Applications**. [S.l.]: IGI Global, 2007.

REATEGUI, E.; EPSTEIN, D. Using text mining to support text summarization. In: **Anais do Simpósio Brasileiro de Informática na Educação**. [S.l.: s.n.], 2015. v. 26, n. 1, p. 1217.

REATEGUI, E. et al. Sobek: a text mining tool for educational applications. International Conference on Data Mining, 2011.

REATEGUI, E. et al. Mineração textual e letramento: aplicações iniciais da ferramenta sobek com alunos do ensino fundamental. Congresso Brasileiro de Informática da Educação, 2016.

REZENDE, S. O. et al. Mineração de dados. **Sistemas inteligentes: fundamentos e aplicações**, v. 1, p. 307–335, 2003.

ROCKENBACH, D. et al. Story maker. In: SPRINGER. **International Conference on Serious Games**. [S.l.], 2014. p. 86–91.

ROSA, J. L. G. O significado da palavra para o processamento de linguagem natural. **Trabalho apresentado no ZLV Seminário do GEL, Unicamp, Campinas**, 1997.

SCHENKER, A. Graph-theoretic techniques for web content mining. 2003.

SPERETTA, M.; GAUCH, S. Using text mining to enrich the vocabulary of domain ontologies. In: IEEE. **Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on**. [S.l.], 2008. v. 1, p. 549–552.

TAN, A.-H. et al. Text mining: The state of the art and the challenges. In: **Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases**. [S.l.: s.n.], 1999. v. 8, p. 65–70.

WEN, J.-R.; NIE, J.-Y.; ZHANG, H.-J. Query clustering using user logs. **ACM Transactions on Information Systems**, v. 20, n. 1, p. 59–81, 2002.

WIVES, L. Recursos de text mining. **Disponível por www em http://www. inf. ufrgs. br/˜ wives/portugues/textmining. html (15 de maio de 2001)**, 2005.

WIVES, L. K. **Tecnologias de Descoberta de Conhecimento em Textos Aplicadas à Inteligência Competitiva. 2002. 116 f**. Thesis (PhD) — Dissertação (Mestrado em Ciência da Computação)–Instituto de Informática, UFRGS, Porto Alegre, 2002.

ZAMIR, O.; ETZIONI, O. Web document clustering: A feasibility demonstration. In: ACM. **Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval**. [S.l.], 1998. p. 46–54.