Rodrigo Ramos Nogueira

# NEWSMINER: UM SISTEMA DE DATA WAREHOUSE BASEADO EM TEXTOS DE NOTÍCIAS

**Sorocaba, SP**

**12 de Maio de 2017**

Rodrigo Ramos Nogueira

# NEWSMINER: UM SISTEMA DE DATA WAREHOUSE BASEADO EM TEXTOS DE NOTÍCIAS

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação (PPGCC-So) da Universidade Federal de São Carlos como parte dos requisitos exigidos para a obtenção do título de Mestre em Ciência da Computação. Linha de pesquisa: Inteligência Artificial e Banco de Dados.

Universidade Federal de São Carlos – UFSCar

Centro de Ciências em Gestão e Tecnologia – CCGT

Programa de Pós-Graduação em Ciência da Computação – PPGCC-So

Orientadora: Profa. Dra. Sahudy Montenegro González

Coorientadora: Profa. Dra. Tiemi Christine Sakata

Sorocaba, SP

12 de Maio de 2017

Nogueira, Rodrigo Ramos

NEWSMINER: UM SISTEMA DE DATA WAREHOUSE BASEADO EM TEXTOS DE NOTÍCIAS / Rodrigo Ramos Nogueira. -- 2017.
97 f. : 30 cm.

Dissertação (mestrado)-Universidade Federal de São Carlos, campus Sorocaba, Sorocaba
Orientador: Profa. Dra. Sahudy Montenegro González
Banca examinadora: Prof. Dr. Marcio Katsumi Oikawa, Prof. Dr. Tiago Agostinho de Almeida
Bibliografia

1. OLAP. 2. Categorização de Notícias. 3. Fontes de dados da Web. I. Orientador. II. Universidade Federal de São Carlos. III. Título.

Ficha catalográfica elaborada pelo Programa de Geração Automática da Secretaria Geral de Informática (SIn).
DADOS FORNECIDOS PELO(A) AUTOR(A)

**UNIVERSIDADE FEDERAL DE SÃO CARLOS**

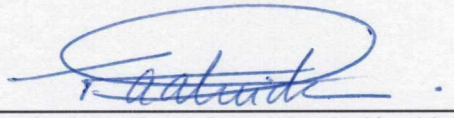
Centro de Ciências em Gestão e Tecnologia
Programa de Pós-Graduação em Ciência da Computação

## Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Rodrigo Ramos Nogueira, realizada em 12/05/2017:

Profa. Dra. Sahudy Montenegro González
UFSCar

Prof. Dr. Marcio Katsumi Oikawa
UFABC



Prof. Dr. Tiago Agostinho de Almeida
UFSCar

*Dedico este trabalho aos meus pais Ari e Alberci pelo incentivo incansável, pelo amor, pela dedicação, e, principalmente, por acreditarem em mim. Ao meu irmão Ariovaldo(in memorian) que mesmo em nossos momentos de dificuldades, sempre me incentivou, apoiou e me deixou como lição a importância de nunca parar de estudar.*

# Agradecimentos

Agradeço,
a Deus por tudo em minha vida e por ser meu guia nos dias difíceis;
a CAPES pelo apoio financeiro durante o mestrado;
aos meus pais Ari e Alberci, por me darem apoio e amor durante esta etapa;
a minha avó Maria das Dores, por mesmo sem compreender os motivos de minha ausência nos últimos dois anos, sempre dedicou-me um amor incondicional;
aos meus amigos pelo apoio emocional e incentivo de sempre e em especial nestes últimos 2 anos, pelas risadas e principalmente por sempre estarem ao meu lado dando o incentivo necessário para que eu chegasse ao fim desta jornada;
aos meus colegas Johannes e Charles, por todas as vezes que me auxiliaram no desenvolvimento desta projeto, pela amizade que construímos e por todas as risadas que pudemos dar ao decorrer do mestrado;
aos meus colegas de laboratório, com os quais tive a oportunidade de compartilhar a vida acadêmica, mas também tivemos ótimos momentos de convivência e descontração;
as minhas orientadoras Profa. Sahudy e Profa. Tiemi, pessoas que tenho grande respeito e admiração, primeiramente, lhes agradeço por me escolherem e confiarem em mim para desenvolver este trabalho. Também lhes agradeço imensamente pela orientação, dedicação, paciência, por sempre me tratarem com horizontalidade e disponibilidade durante toda a minha estadia como acadêmico desta Universidade, agradeço por tudo e sempre lembrarei de vocês com carinho e admiração.

.

*"Demore o tempo que for para decidir o que você quer da vida, mas depois que decidir não recue ante nenhum pretexto"(Friedrich Nietzsche)*

# Resumo

As aplicações de mineração de dados e textos oriundos da Internet têm sido alvo de recentes pesquisas. E, em todos os casos, as tarefas de mineração de dados necessitam trabalhar sobre dados limpos, consistentes e integrados para obter os melhores resultados. Sendo assim, ambientes de *Data Warehouse* são uma valiosa fonte de dados limpos e integrados para as aplicações de mineração. A tecnologia de *Data Warehouse* tem evoluído no sentido de recuperar e tratar dados provenientes da Web. Em particular, os *sites* de notícias são fontes ricas em textos, que podem compor um *corpus* linguístico. Inserindo o *corpus* em um ambiente de *Data Warehouse*, as aplicações poderão tirar proveito da flexibilidade que um modelo multidimensional e as operações OLAP fornecem. Dentre as vantagens estão a navegação pelos dados, a seleção da parte dos dados considerados relevantes, a análise dos dados em diferentes níveis de abstração, e a agregação, desagregação, rotação e filtragem sobre qualquer conjunto de dados. Este trabalho apresenta o ambiente de *Data Warehouse Newsminer*, que fornece um conjunto de textos consistente e limpo, na forma de um *corpus* multidimensional para consumo por aplicações externas e usuários. A proposta inclui uma arquitetura que integra a coleta textos de notícias em tempo próximo do tempo real, um módulo de enriquecimento semântico como parte da etapa de *ETL*, que acrescenta propriedades semânticas aos dados coletados tais como a categoria da notícia e a anotação *POS-tagging*, e a disponibilização de cubos de dados para consumo por aplicações e usuários. Foram executados dois experimentos. O primeiro experimento é relacionado à escolha do melhor classificador de categorias das notícias do módulo de enriquecimento semântico. A análise estatística dos resultados indicou que o classificador *Perceptron* atingiu os melhores resultados de F-medida, com resultado bom de tempo de processamento. O segundo experimento coletou dados para avaliar o pré-processamento de notícias em tempo real. Para o conjunto de dados coletados, os resultados indicaram que é possível atingir tempo de processamento *online*.

**Palavras-chaves**: Aplicações de Mineração de Textos, Dados da Web, *sites* de notícias, Corpora multidimensional, Enriquecimento semântico, Categorização de notícias, *Data Warehouse*, OLAP.

# Abstract

Data and text mining applications managing Web data have been the subject of recent research. In every case, data mining tasks need to work on clean, consistent, and integrated data for obtaining the best results. Thus, Data Warehouse environments are a valuable source of clean, integrated data for data mining applications. Data Warehouse technology has evolved to retrieve and process data from the Web. In particular, news websites are rich sources that can compose a linguistic corpus. By inserting corpus into a Data Warehousing environment, applications can take advantage of the flexibility that a multidimensional model and OLAP operations provide. Among the benefits are the navigation through the data, the selection of the part of the data considered relevant, data analysis at different levels of abstraction, and aggregation, disaggregation, rotation and filtering over any set of data. This paper presents *Newsminer*, a data warehouse environment, which provides a consistent and clean set of texts in the form of a multidimensional corpus for consumption by external applications and users. The proposal includes an architecture that integrates the gathering of news in real time, a semantic enrichment module as part of the ETL stage, which adds semantic properties to the data such as news category and POS-tagging annotation and the access to data cubes for consumption by applications and users. Two experiments were performed. The first experiment selects the best news classifier for the semantic enrichment module. The statistical analysis of the results indicated that the *Perceptron* classifier achieved the best results of F-measure, with a good result of computational time. The second experiment collected data to evaluate real-time news preprocessing. For the data set collected, the results indicated that it is possible to achieve online processing time.

**Key-words**: Text Mining applications, Web data, News websites, Multidimensional corpora, Semantic enrichment, News categorization, Data Warehouse, OLAP.

# Lista de ilustrações

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| API | *Application Programming Interface* |
| DW | *Data Warehouse* |
| ETL | *Extract, Transform and Load* |
| FT | *Full Text* |
| HTML | *Hypertext Markup Language* |
| KNN | *K Nearest Neighbors* |
| KP | *Keyphrases* |
| NBB | *Naïve Bayes Bernoulli* |
| NBM | *Naïve Bayes Multinomial* |
| OLAP | *On-line Analytical Processing* |
| OLTP | *On-line Transaction Processing* |
| PA | *Passive Agressive* |
| PER | *Perceptron* |
| RL | Regressão Logística |
| ROC | *Rocchio* |
| SGBD | Sistema Gerenciador de Banco de Dados |
| SGD | *Stochastic Gradient Descent - SGD* |
| SVM | *Support Vector Machine* |
| SW | *Stopwords* |

# Sumário

# 1 INTRODUÇÃO

A Internet é a grande responsável pela explosão de informação. Até o mês de abril de 2017, os *Web sites* atingiram o número aproximado de 1 bilhão e 117 milhões[1], dos quais 12.500 são *sites* de notícias. Apenas o jornal americano *The Washington Post*, considerado um dos maiores jornais do mundo, publica, em média, diariamente, 500 textos entre artigos e notícias, acessados por milhões de pessoas em todo o mundo[2]. Neste cenário de grande volume de informação, a mineração de dados oriundos da Internet tem sido alvo de recentes pesquisas, pois reúne em seu ambiente, quase a totalidade dos tipos de estruturas, simples ou complexas, que existem, atendendo a diversas necessidades e possuindo diversos conteúdos e formatos.

As tarefas de Mineração de Dados necessitam trabalhar sobre dados limpos, consistentes e integrados para obter os melhores resultados. Para isto, os dados das fontes, geralmente, precisam passar por uma etapa de pré-processamento (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Sendo assim, ambientes de *Data Warehouse* (DW) são uma valiosa fonte de dados limpos e integrados para o processamento analítico *online* (OLAP) e as aplicações de Mineração de Dados. Nesse caso, a tarefa de mineração é dedicada apenas a sua funcionalidade em si, não necessitando realizar pré-processamento (MANSMANN et al., 2014).

No contexto de dados oriundos da Internet, para poder prover um conjunto consistente de dados para as aplicações de Mineração de Dados e Textos, os ambientes de *Data Warehouse* têm como desafio trabalhar com dados não estruturados, escritos em linguagem natural. A tecnologia de *Data Warehouse* tem evoluído no sentido de recuperar e tratar tais dados, transformando-os e armazenando-os em sua estrutura multidimensional para que possam ser analisados (VICTOR; REX, 2016; MANSMANN et al., 2014).

Em particular, os *sites* de notícias são fontes ricas em textos, que podem compor um *corpus* linguístico. Um *corpus* de texto é um conjunto grande e estruturado de textos, que pode servir como base para aplicações de Mineração de Dados e Textos. Caso esse *corpus* seja armazenado em um ambiente de *Data Warehouse*, as aplicações poderão tirar proveito da flexibilidade que um modelo multidimensional e as operações OLAP fornecem. Essas vantagens são a navegação pelos dados, permitindo a seleção da parte dos dados considerados relevantes; e a análise dos dados em diferentes níveis de abstração. Os operadores OLAP possibilitam a agregação, desagregação, rotação e filtragem sobre

[1] *Internet live stats*. Disponível em <http://www.internetlivestats.com>, acessado em 10/04/2017.

[2] *"How Many Stories Do Newspapers Publish Per Day?"*. Disponível em <https://www.theatlantic.com/technology/archive/2016/05/how-many-stories-do-newspapers-publish-per-day/483845/>, acessado em 15/06/2016.

qualquer conjunto de dados.

A exploração de textos de notícias permitiu a Atkinson et al. (2008) gerar alertas sobre eventos sísmicos georreferenciados, em tempo real. Nesse sistema, na medida que novas notícias sobre furacões, tornados e outros eventos são extraídas, os alertas são gerados simultaneamente. O aspecto temporal em textos de notícias serve para se obter e analisar tendências e variações. Por exemplo, Moraes et al. (2016) estuda as ocorrências de palavras do ano de 1851 até 2015, e, estabelece uma correlação entre publicações relativas a eleições e pena de morte, para trazer a reflexão de que o tema de pena de morte é muito debatido nos períodos eleitorais.

Porém, existem desafios relacionados à coleta de textos da *Web*. Um deles é a complexidade de manipulação, pois os textos são armazenados na fonte como hipertexto em linguagem *HTML*. Neste formato, torna-se difícil a compreensão e interpretação, tanto por humanos, como computacionalmente. Um segundo desafio é a exploração das fontes de notícias em tempo real ou próximo do tempo real (*near real-time*), pois trabalhar sobre um *corpus* atualizado pode ser importante para algumas aplicações, como aplicações de *e-business* e corretagem de ações. Outro desafio é a falta de completude dos dados coletados. Por exemplo, até dezembro de 2016, de cerca de 200.000 notícias coletadas, 27.5% não tem categoria definida pela fonte. Um último desafio é a criação de anotações para o conjunto de textos. As anotações tornam o *corpus* útil para fazer pesquisas linguísticas. Elas acrescentam semântica ao *corpus*. Um exemplo de anotação é etiquetar a parte do discurso de cada palavra, ou *POS-tagging* (substantivo, adjetivo, verbo, etc.). Outro exemplo é armazenar a categoria de cada texto do conjunto.

## 1.1 Objetivos

O objetivo deste trabalho tem como base a seguinte questão: *é possível fazer uma integração entre as fontes de dados Web e as aplicações de Mineração de Dados e Textos por meio de um ambiente de Data Warehousing?*

Para responder essa questão, este trabalho apresenta o ambiente *Newsminer*, que fornece um conjunto de dados consistentes e limpos, na forma de um *corpus* multidimensional para consumo por aplicações externas e usuários. O *corpus multidimensional* é um conjunto de textos armazenado de acordo com um modelo multidimensional, que permite explorar a multidimensionalidade em diferentes níveis de abstração.

Os objetivos específicos são:

- propor um modelo multidimensional que consiga armazenar o conjunto de textos e a característica temporal das notícias;
- coletar textos de notícias em tempo próximo do tempo real;

- acrescentar anotações semânticas, linguísticas ou não, dinamicamente no ambiente;
- disponibilizar os dados para consultas multidimensionais de aplicações e usuários;
- por último, propor uma arquitetura que consiga a integração dos objetivos anteriores no ambiente *Newsminer*.

## 1.2 Contribuições

As contribuições desta dissertação são:

- A proposta de um modelo multidimensional de textos de notícias, que permite a exploração de notícias por tempo, categorias e palavras/termos dos textos.
- A inclusão de um *Web Crawler* como parte da etapa de *ETL* da arquitetura do *Newsminer*, permitindo a obtenção de textos de notícias em tempo próximo do tempo real.
- A proposta de um módulo de enriquecimento semântico como parte da etapa de *ETL* da arquitetura do *Newsminer*, que acrescenta propriedades semânticas aos dados coletados, tais como a categoria da notícia e a anotação *POS-tagging*.
- A proposta de um classificador de categorias de notícias, baseado em 17 categorias de notícias da ontologia *IPTC - International Press Telecommunications Council* (SMOOT, 2003), como parte do módulo de enriquecimento semântico para a adição dinâmica da categoria das notícias coletadas. Para descobrir a categoria das notícias, foram avaliados nove métodos de classificação, geralmente usados na literatura, e escolhido aquele que representa a melhor generalização de textos de notícias.
- A análise de diferentes métodos de pré-processamento: texto completo, com remoção de *stopwords*, extração de substantivos e extração de termos compostos (*keyphrases*). O propósito dessa análise é determinar a melhor configuração que permita obter os melhores resultados na categorização de notícias. O método de pré-processamento selecionado é incluído como parte do processo de *ETL* do *Newsminer* e seus resultados armazenados no DW.
- Um novo conjunto de dados (*dataset*) de textos de notícias em inglês, chamado de *Newsminer Collection*, categorizadas para futuras pesquisas. Este conjunto de dados foi utilizado nos experimentos do Capítulo 5.
- A disponibilização de consultas multidimensionais em uma interface *Web* e via API para consumo pelas aplicações e usuários. O propósito é a exploração e navegação pelos dados contidos no *Data Warehouse*, permitindo a seleção flexível de parte ou

de todos os dados e a análise dos dados em diferentes níveis de abstração. Algumas consultas multidimensionais foram pré-definidas e podem ser parametrizadas com filtros de categoria e períodos de tempo. Elas são:

- *Ranking* de termos: retorna os termos que mais ocorrem em todo o banco ou em uma categoria em específico. Por exemplo, o termo que mais ocorreu em todo o banco de dados foi `year`, na categoria de esportes foi `game`, enquanto na categoria de política os dois termos de maior ocorrência foram `president` e `trump`[3].
- Associação entre termos: dado um termo, retorna os termos que mais ocorreram em todo o banco de dados ou em uma categoria em específico. Por exemplo, os cinco termos de maior ocorrência, na categoria de economia, de janeiro de 2010 até dezembro de 2015, associadas ao termo `dollar` são `investor`, `economy`, `market`, `bank`, `related`, e `markets`. Já, na categoria de política, no mesmo período, os termos mais associados à `dollar` são `statement`, `government`, `buy`, `administration` e `america`.
- Categorização de textos: dado um texto de entrada, esta consulta retorna a qual categoria este texto pertence. Por exemplo, dado o texto de entrada *Soccer players on wrong side of anti-doping officials*, o *Newsminer* retorna que este texto pertence à categoria de esporte.
- Busca textual: esta consulta tem como entrada um conjunto de termos e retorna quais os *Top K* textos que esses termos aparecem em conjunto. Por exemplo, tendo como entrada *Wolverine old man logan movie*, esta consulta retorna as principais notícias em que estes termos aparecem em conjunto.
- Cubo de dados: pensando em flexibilizar as consultas de usuários e aplicações, o *Newsminer* oferece uma consulta exploratória, que se baseia em todos os campos contemplados no modelo multidimensional. Com esta consulta, o usuário poderá escolher quais dados deseja recuperar, explorando a multidimensionalidade do modelo, aplicando filtros e gerando diversos cubos de dados, a partir dos dados armazenados na base do *Newsminer*.

## 1.3 Organização do trabalho

Para facilitar a leitura e compreensão desta dissertação, seus capítulos estão organizados da seguinte maneira.

[3] Para todos os exemplos, as informações foram extraídas do ambiente em mar/2017.

- O Capítulo 2 apresenta a fundamentação teórica do trabalho, que inclui os conceitos básicos utilizados ao longo desta pesquisa sobre *Data Warehouse*, representação e pré-processamento de textos e os métodos de classificação de notícias.
- O Capítulo 3 faz e a revisão da literatura recente, dividida em quatro tópicos relacionados.
- O Capítulo 4 mostra em detalhes o ambiente *Newsminer*, explicando sua arquitetura e características.
- O Capítulo 5 descreve os dois experimentos realizados. Para cada um, descrevem-se as configurações adotadas e discutem-se os resultados. O primeiro experimento é relacionado à escolha do melhor classificador de categorias das notícias. O segundo experimento coleta dados para avaliar o pré-processamento (*ETL*) de notícias em tempo real. Por último, este capítulo apresenta cenários de uso do ambiente.
- Finalmente, a Conclusão expõe as considerações finais sobre o trabalho desenvolvido, assim como, os trabalhos futuros.

# 2 FUNDAMENTAÇÃO TEÓRICA

O principal objetivo desta dissertação é apresentar um ambiente que integra fonte de dados de notícias, para permitir a realização de consultas multidimensionais e fornecer dados consistentes e limpos para as aplicações externas consumirem. Para isto, a arquitetura do ambiente *Newsminer* baseia-se em *Data Warehouse* e inclui categorização de textos para acrescentar semântica. Por conseguinte, este capítulo expõe os conceitos básicos relacionados a esses tópicos, além das técnicas de processamento de textos utilizadas no pré-processamento das notícias. Esses conceitos são essenciais para a compreensão desta dissertação.

## 2.1 *Data Warehouse* e *OLAP*

Com o crescente aumento do volume das informações organizacionais armazenadas nos sistemas gerenciadores de banco de dados, os sistemas tradicionais *OLTP - Online Transaction Processing* (Processamento de Transações em Tempo Real) já não dispõem de suporte para retornar as consultas em um tempo hábil para a tomada de decisão. Tendo como finalidade a realização de operações analíticas, os sistemas *OLAP - Online Analytical Processing* (Processamento Analítico em Tempo Real) permitem a exploração de dados armazenados em ambientes de *Data Warehouse*. *Data Warehouses*, explicado de uma maneira simples, são bancos de dados analíticos, projetados para armazenar os dados de fontes diversas, já transformados e preparados para a realização de consultas através de operações *OLAP* (HAN; KAMBER; PEI, 2011; KIMBALL; ROSS, 2011; INMON, 2005).

*Data Warehouse* e *OLAP* estão diretamente ligados, principalmente por terem suas arquiteturas propostas em conjunto. Há uma sinergia entre estas duas tecnologias, uma vez que *Data Warehouse* é utilizado para armazenar dados em uma estrutura voltada à análise e *OLAP* é utilizado para consultar dados de maneira analítica.

A arquitetura tradicional utilizada em ambientes de *Data Warehouse* foi proposta por Kimball e Ross (2011) e é composta por quatro camadas principais. As quatro camadas que compõem esta arquitetura são ilustradas na Figura 1 e estão descritas na sequência:

- Fontes Provedoras: em um ambiente de *Data Warehouse*, os dados são oriundos de diversas fontes. Esta camada contém todos os dados possíveis de serem armazenados no *Data Warehouse* (relacional, orientado a objetos, não estruturados, textual, Web, etc.) que possam ajudar a cumprir as tarefas de análise.

- Área de Trabalho: nesta camada são realizados os processos de integração das fontes

de dados, bem como as transformações necessárias para armazenar os dados de acordo com o modelo definido para *Data Warehouse*.

- Área de Apresentação de Dados: esta camada trata do *Data Warehouse* em si, onde os dados já coletados e transformados serão armazenados em um banco de dados multidimensional.

- Ferramentas de Acesso aos Dados: a quarta camada é onde as ferramentas de visualização fazendo uso do servidor *OLAP* submetem requisições de acesso aos dados armazenados.

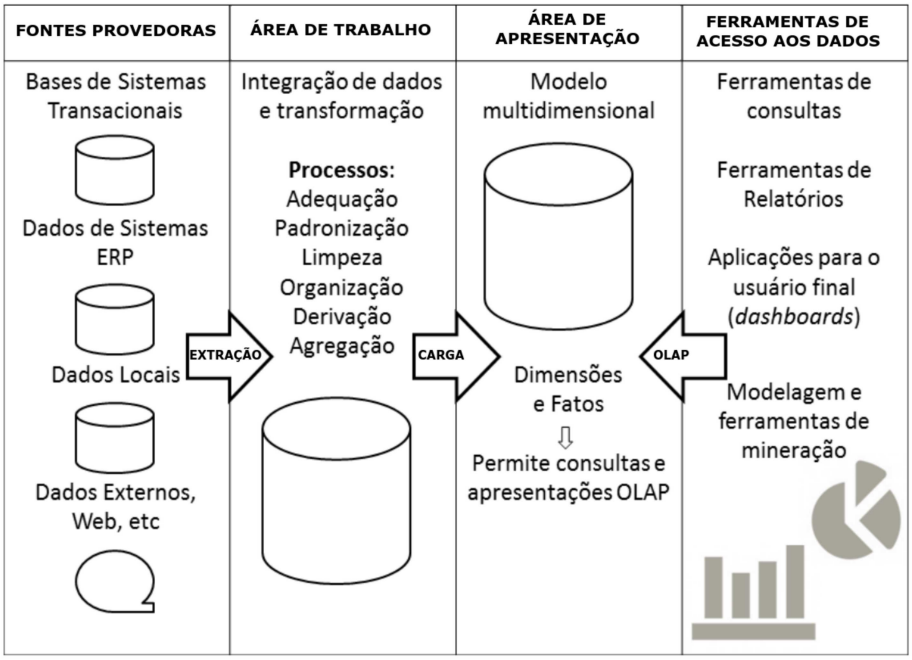



Figura 1: Arquitetura de um ambiente *Data Warehouse*. Adaptado de Kimball e Ross (2011)

As próximas seções descrevem cada uma das camadas da arquitetura de *Data Warehouse* em um maior nível de detalhes, bem como os demais elementos que as compõem, a iniciar pelo processo de *ETL*.

### 2.1.1 Extração, transformação e carga

A *ETL - Extract Transform and Load* (Extração, Transformação e Carga) é responsável por fazer a integração entre as camadas de um *Data Warehouse* a partir das fontes de dados, além de tratar e armazenar estes dados em um banco de dados multidimensional no *Data Warehouse*. É comum que cerca de 60% a 80% de um projeto de implementação de um *Data Warehouse* seja dedicado a etapa de *ETL* (NAGABHUSHANA, 2006). A

etapa de *ETL* é responsável por extrair, limpar, transformar e carregar os dados das fontes no *Data Warehouse*. Cada etapa da *ETL* tem as seguintes funções:

- Extração: é responsável por extrair os dados das fontes, ou seja é processo de recuperação dos dados necessários das fontes de origem, que podem ser as tabelas reais ou simplesmente cópias que foram carregadas no *Data Warehouse*. A etapa de extração deve ser capaz de ler e compreender os dados da fonte e copiar apenas os dados necessários.

- Transformação: esta etapa, literalmente, transforma os dados coletados das fontes de acordo com os definidos no modelo do *Data Warehouse*. Nesta etapa, são realizados pré-processamentos, nos quais são identificados os dados duplicados, integração entre os dados, substituição de valores, limpeza de campos e toda a transformação necessária para adequar as fontes de dados. Um exemplo de uma transformação comum realizada por processos *ETL* é relacionado com campos de sexo, em alguns sistemas são "M" para Masculino e "F" para Feminino, porém em outros está guardado como "H" para Masculino e "M" para Feminino, em outro ainda, podemos encontrar "1" para Masculino e "2" para Feminino, cabendo transformá-los para um único formato. A etapa de transformação também é responsável por resolver desafiadores problemas oriundos das fontes de dados, como ausência de informação, valores inválidos, ausência de integridade referencial, violação de regras de negócios, cálculos inválidos, duplicação de informação, inconsistência de dados e falhas na modelagem das fontes de dados (GOLDSCHMIDT; PASSOS, 2005).

- Carga: uma vez que os dados foram coletados e transformados, a etapa de carga é responsável por armazenar os dados no *Data Warehouse*. Na etapa carga, o *Data Warehouse* é alimentado com novos dados, de forma que as estruturas de dados (por exemplo, tabelas no caso de bancos relacionais) sejam atualizadas para conter os novos dados. Normalmente, o *Data Warehouse* é colocado *off-line* durante a carga de forma que nenhum usuário possa consultar o *Data Warehouse* simultaneamente (JENSEN; PEDERSEN; THOMSEN, 2010). Como o armazenamento de dados em ambientes de *Data Warehouse*, normalmente, envolve grandes quantidades de dados, a etapa de carga sempre ocorre em um período regular, por exemplo, diariamente.

Na definição de Kimball e Ross (2011), o sistema de *ETL* é análogo ao da cozinha de um restaurante, onde os chefes pegam matérias-primas e as transformam em deliciosas refeições para os clientes. Deste modo, a etapa de *ETL* tem grande importância em ambientes de *Data Warehouse*, afinal é responsável por servir o ambiente de *Data Warehouse* com dados consistentes e limpos. O conceito de *Data Warehouse* é explicado na próxima seção.

### 2.1.2 Data Warehouse

*Data Warehouse* tem como tradução "armazém de dados". É uma estrutura de banco de dados analítico, permitindo através da sua estrutura de armazenamento de dados, a realização de consultas. Um *Data Warehouse* contém as mesmas informações contidas nas fontes de dados, no entanto, já tratadas e preparadas para a realização de consultas.

Na definição de Inmon (2005), um *Data Warehouse* é uma coleção de dados, orientado a um assunto, integrado, com tempo variável e não volátil, para suporte ao gerenciamento dos processos de tomada de decisão. Cada uma destas características são descritas brevemente a seguir.

- Orientado a assunto: diz-se que um *Data Warehouse* é orientado a assunto pelo fato de que este sempre estará relacionado a um tema sobre qual consultas são realizadas. Isto significa que ele sempre será direcionado a um tema, seja uma loja, um *e-commerce*, locadora, ou como neste projeto, textos de *sites* de notícias.
- Integrado: dada a pluralidade das fontes de dados, um *Data Warehouse* tem o dever de ter estes mesmos dados integrados e consolidados.
- Variável em relação ao tempo: o fator temporal pode ser determinante na análise nos dados armazenados em um *Data Warehouse*. Por exemplo, em um *Data Warehouse* de vendas se obtém os meses nos quais há uma maior queda nas vendas e os meses em que há um número maior de vendas. Sendo assim, faz-se necessário que os dados de *Data Warehouse* sejam armazenados em relação ao tempo.
- Não volátil: em um *Data Warehouse* há uma periodicidade na carga dos dados e estes dados não são excluídos e nem alterados, apenas carregados.

Em resumo, pela definição de Kimball e Ross (2011), um *Data Warehouse* é uma cópia de dados de transação (dados armazenados em sistemas transacionais de origem), especificamente estruturado para consulta e análise. Para fazer possível o armazenamento de dados, o esquema de dados baseia-se no modelo multidimensional, abordado na próxima seção.

### 2.1.3 Modelagem multidimensional

A modelagem multidimensional é uma técnica de modelagem de bancos de dados que se destina apoiar as consultas realizadas pelo usuário final em um *Data Warehouse* (KIMBALL; ROSS, 2011). A técnica de modelagem multidimensional trata da elaboração de um projeto lógico de um banco de dados, que tem sua aplicação destinada à análise de dados. Utilizando a modelagem multidimensional, se estabelece a estrutura de dados sob qual o cubo de dados será analisado.

Por meio da modelagem multidimensional, obtém-se o modelo multidimensional. O modelo multidimensional é estruturado de uma maneira que permite que os dados armazenados possam ser recuperados de maneira rápida e eficaz pelo usuário.

O modelo multidimensional é composto por três componentes principais: as tabelas de fato, as tabelas com dimensões e as métricas. As métricas são valores, normalmente, aditivos, armazenados na tabela fato. A tabela fato é a tabela principal em um modelo dimensional onde as métricas são armazenadas. As tabelas dimensões são companheiras integrais da tabela fato, contendo em seus atributos as descrições do negócio e uma chave primária para ser identificada pela tabela fato.

Os atributos que compõem uma dimensão podem ser apenas descritivos ou formar uma hierarquia. As hierarquias são representadas pela composição de vários atributos em uma dimensão, onde cada atributo representa um nível em uma hierarquia. Por exemplo, em uma dimensão que tenha os atributos `dia`, `mês` e `ano`, estes atributos compõem uma hierarquia de tempo que permite navegar pelos níveis `data -> mês -> ano`.

Um dos modelos que representa a modelagem multidimensional de dados é o modelo estrela. O modelo estrela tem esse nome por conta de sua representação, no qual a disposição das tabelas traz similaridade a uma estrela. Ao redor da tabela de fato estão as dimensões, que descrevem as perspectivas de análise sobre as métricas. A Figura 2 ilustra um exemplo do modelo em estrela com dados deste projeto, onde a tabela fato `FATOOCORRÊNCIA` está disposta ao centro e contém chave estrangeira para todas as tabelas dimensões, permitindo com que a métrica `NUMEROOCORRENCIAS` possa ser analisada de todas as dimensões.

Para a manipulação dos dados, os usuários visualizam as informações explorando o cubo de dados, descrito na próxima seção.

### 2.1.4 Cubo de dados

Os dados em um *Data Warehouse* podem ser explorados por diversas perspectivas. O cubo de dados é uma abstração da representação analítica dos dados armazenados multidimensionalmente. É uma metáfora para como os dados são vistos de acordo com as dimensões. Em um *Data Warehouse*, o cubo é utilizado para representar uma consulta multidimensional.

A Figura 3 relaciona a modelagem multidimensional com sua representação analítica na forma de um cubo de dados. Cada tabela dimensão do modelo multidimensional é um eixo do cubo, no qual os valores contidos nas células representam as métricas contidas na tabela fato do modelo. Especificamente, a consulta que está sendo visualizada no cubo é o número de ocorrências (métrica) por palavras, ao longo das categorias e em relação ao tempo (bimestres).

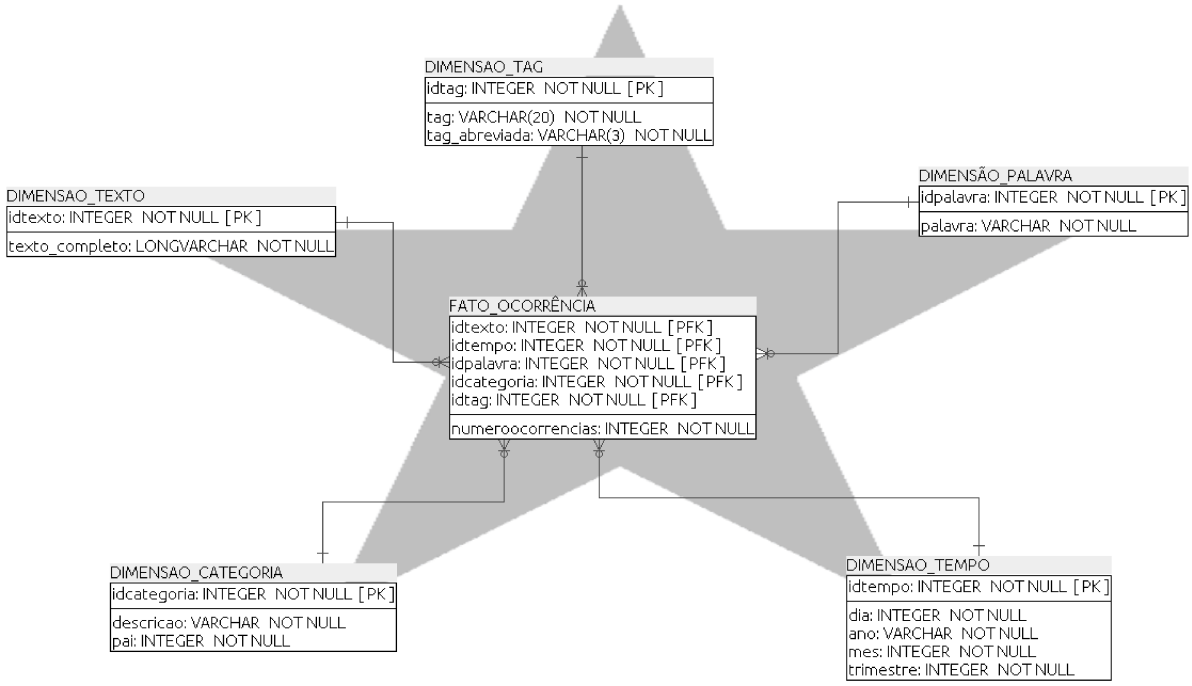



Figura 2: Exemplo de modelo multidimensional em estrela

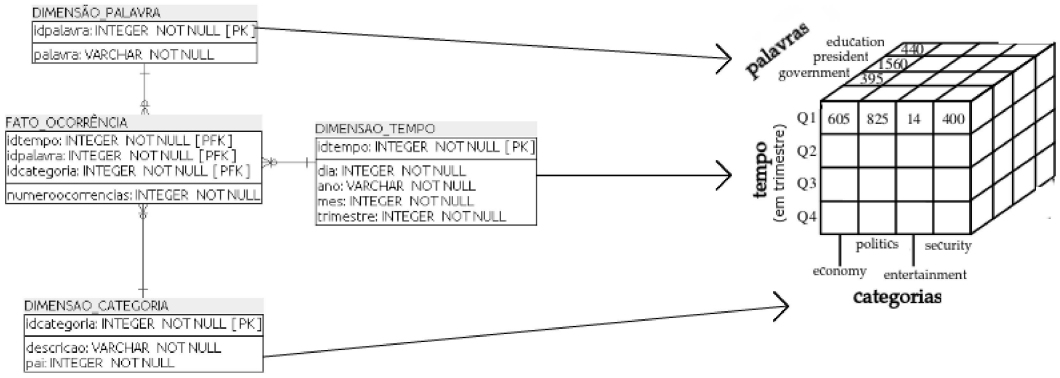



Figura 3: Relação entre cubo de dados e o modelo multidimensional

Após os dados serem armazenados, os cubos de dados são explorados por intermédio das operações *OLAP*, explicadas na próxima seção.

### 2.1.5 OLAP e suas operações

*OLAP* é o processamento *online* de dados com foco em análise para tarefas de tomada de decisão. As análises ocorrem em tempo real, ou seja, são executadas de maneira eficiente, quando comparadas com as mesmas consultas executadas nos sistemas transacionais. Servidores *OLAP* sempre empregam uma visão multidimensional dos dados, fazendo com que haja grande aplicabilidade quando integrados a bancos de dados multidimensionais.

As ferramentas *OLAP* são utilizadas para a análise interativa de dados multidimensionais de granularidades variadas, o que facilita a generalização de dados eficaz e mineração de dados. Muitas outras funções de mineração de dados, tais como associação, classificação, previsão e agrupamento, podem ser integradas com as operações de *OLAP* para aumentar a mineração interativa e a obtenção do conhecimento em vários níveis de abstração (HAN; KAMBER; PEI, 2011).

Servidores *OLAP* permitem obter uma visão analítica das informações mediante o acesso rápido, consistente e interativo à informação. Utilizando as operações *OLAP*, é possível navegar pelas hierarquia dos dados, explorando o cubo de dados e a multidimensionalidade. As operações *OLAP* são descritas a seguir:

- *Slice*: retorna valores específicos de uma dimensão do cubo, pode-se dizer que fatia uma parte do cubo a ser visualizado. No exemplo da Figura 4, uma operação *slice* no cubo de dados, relacionado ao modelo da Figura 2, é a seleção do número de ocorrências por palavras e categorias apenas do bimestre `Q1`.

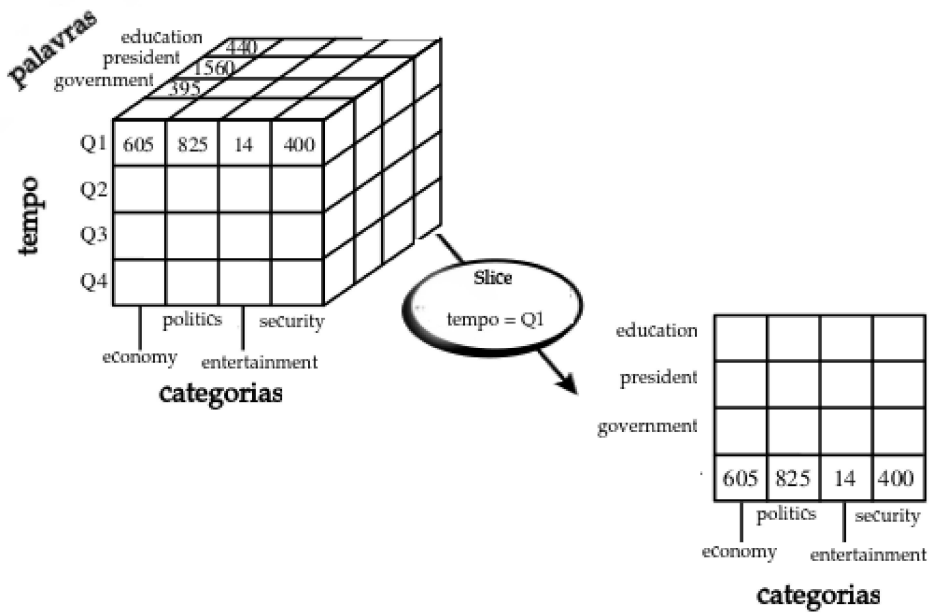



Figura 4: Operação de *Slice*

- *Dice*: são *slices* consecutivos, permitindo cortar o cubo, gerando um subcubo. No exemplo da Figura 5, é exemplificado um *dice*, no qual são selecionados os tempos `Q1` e `Q2`, bem como as categorias `politics` e `economy`.

- *Drill-down*: este operador navega a partir de dados, obtendo um maior nível de detalhamento. Na Figura 6, o exemplo no cubo mostra as ocorrências de palavras organizadas em quatro bimestres, no qual é realizado um *drill-down* que expande os dados do cubo, trazendo as mesmas informações organizadas por meses.

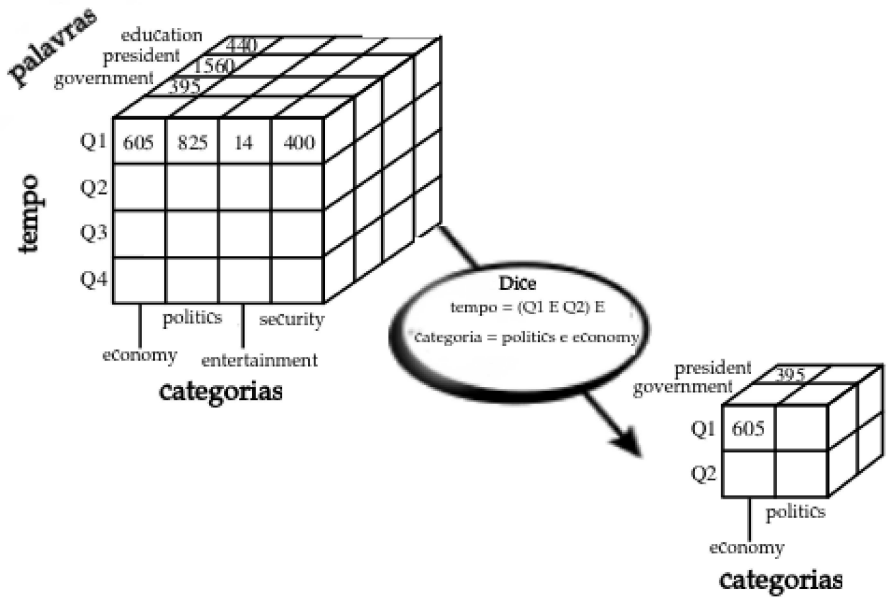



Figura 5: Operação de *Dice*

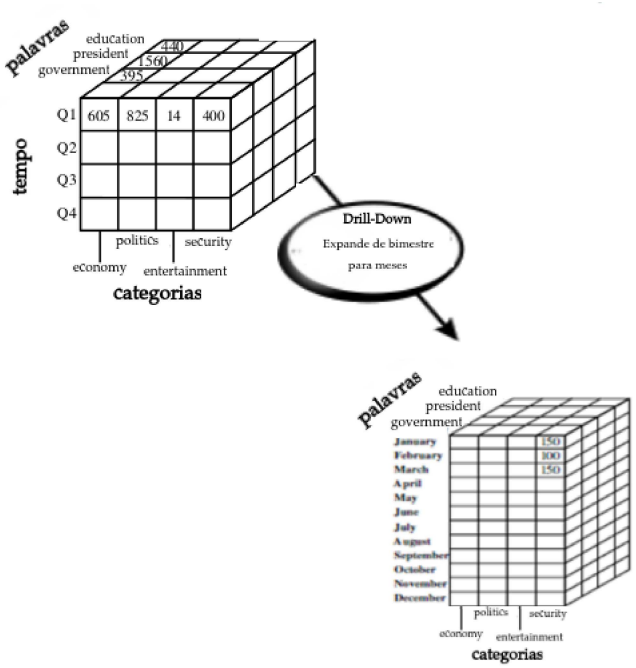



Figura 6: Operação de *Drill-Down*

- *Roll-Up*: oposto do operador *drill-down*, sumariza as informações, diminuindo o nível de detalhes. A operação de *roll-up* executa a agregação em um cubo de dados. A Figura 7 mostra que a hierarquia está definida em palavras e textos. A operação de *roll-up*, no exemplo, agrega os dados de palavras para textos, gerando um cubo resultante com o total de ocorrências por textos, ao invés do cubo inicial que agrupava por palavra.

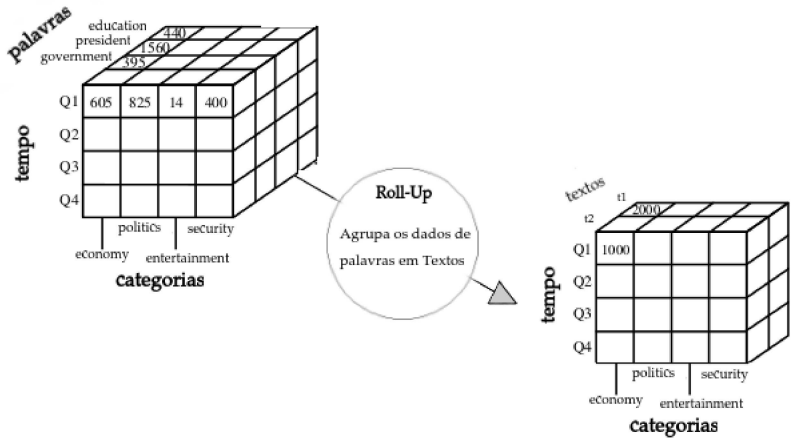



Figura 7: Operação de *Roll-up*

## 2.2 Representação e Pré-Processamento de Texto

As fontes de dados podem apresentar diferentes características, dimensões ou formatos. Podem conter tanto dados limpos, como ruídos e imperfeições, com valores incorretos, inconsistentes, incompletos ou ausentes (FACELI et al., 2011). As técnicas de pré-processamento de texto são constantemente utilizadas em cenários de *Data Warehouse* e Mineração de Texto com a finalidade melhorar a qualidade dos dados através da remoção ou minimização dos problemas citados.

Uma vez obtendo-se uma melhor qualidade dos dados, se amplifica a possibilidade da eficácia na tomada de decisões, melhor desempenho das consultas e dos métodos de categorização que utilizam estes dados. Sendo assim, esta seção aborda os métodos de pré-processamento e representação de textos.

### 2.2.1 Representação computacional de textos

O texto é uma fonte de dados rica em conteúdo e informação, que pode ser explorado a partir de suas diversas características semânticas. Todavia, para que seja compreendido pelos métodos computacionais, um texto deve estar representado da maneira apropriada.

#### 2.2.1.1 *Bag of words*

Quando se trata da representação computacional de dados textuais, a técnica denominada *Bag of Words* (Bolsa de Palavras) é utilizada por diversos trabalhos da literatura, inclusive na representação de textos extraídos de *sites* de notícias (LI et al., 2011a; LIN et al., 2014; JONNALAGEDDA et al., 2016).

Na representação *Bag of Words*, os documentos são simplesmente tratados como coleções de palavras desordenadas. Os valores que são atribuídos a cada palavra, geralmente,

são a ocorrência da palavra em um documento ou a frequência com que esta palavra aparece em um texto (LI et al., 2011b). A ocorrência é binária, ou seja, para cada palavra, se descreve se ela ocorreu no texto ou não. Quando se trabalha com frequência é considerado o número de vezes que cada palavra aparece em um documento. Para melhor entendimento, considere `D1` e `D2`:

- D1: *the player do not will play another game, because is the final game*
- D2: *is the final game play*

Se for utilizado *Bag of Words* por ocorrência, os textos `D1` e `D2` são representados pela matriz da Tabela 1.

Tabela 1: *Bag of Words* por ocorrência

| - | the | player | do | not | will | play | another | game | because | is | final |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

O mesmo exemplo, utilizando textos `D1` e `D2` em *Bag of Words* por frequência dos termos, é apresentado pela Tabela 2.

Tabela 2: *Bag of Words* por frequência

| - | the | player | do | not | will | play | another | game | because | is | final |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 |
| D2 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

#### 2.2.1.2 *TF-IDF*

A combinação de `tf` e `idf` define a importância (peso) de um termo para um documento dentro do corpus (GOKER; DAVIES, 2009). A frequência (*Term of Frequency - TF*) e a frequência invertida (*Inverse Document Frequency - IDF*) são baseadas na representação vetorial de *Bag of Words*. Esta técnica é amplamente utilizada para representação de textos em projetos de processamento de linguagem natural e mineração de textos. Mais informações podem ser encontradas em (LESKOVEC; RAJARAMAN; ULLMAN, 2014).

#### 2.2.1.3 *N-grams*

*n-gramas* é uma sequência contígua de $n$ itens de uma determinada sequência de texto. Por exemplo, na frase "*for something completely different*", um unigrama pode ser "something", um bigrama "something completely", e um trigrama "something completely different" (XUE; SUN, 2004).

### 2.2.2 Pré-processamento de textos

Quando se trabalha com textos em ambientes de *Data Warehouse*, a etapa de pré-processamento se torna essencial. Nela, os textos são transformados da sua estrutura original, para um formato que podem ser processados de maneira mais simples e eficiente, de acordo com a sua devida finalidade. O pré-processamento de textos é amplamente utilizado, também, antes da aplicação de técnicas mineração de dados.

A fase de pré-processamento pode ser longa, demandando tempo e recursos computacionais. De acordo com a literatura Losarwar e Joshi (2012), mais de 80% do tempo necessário para realizar qualquer projeto de mineração de dados do mundo real normalmente é gasto em pré-processamento de dados.

As etapas de pré-processamento têm como finalidade remover ambiguidades do texto, bem como diminuir a alta dimensionalidade dos atributos de textos. A seguir, descrevem-se os métodos de pré-processamento aplicados nesta pesquisa.

#### Remoção de *stopwords*

As *stopwords* são palavras removidas dos textos, geralmente no início do processo. Elas não acrescentam semântica no texto. Alguns exemplos são preposições, artigos, conjunções e outros. Caso as *stopwords* sejam removidas, o texto não perderá seu contexto.

Posterior à remoção de *stopwords*, a dimensionalidade de atributos de um texto será diminuída, e, normalmente, esta etapa otimiza o desempenho dos métodos computacionais que utilizam estes dados.

A próxima seção descreve o método de *POS Tagging*. Este método pode ser combinado com a remoção de *stopwords* a fim de se obter melhores resultados em cenários de processamento de linguagem natural e mineração de textos.

#### Classificação morfológica

*POS Tagging*, do inglês *Part Of Speech Tagging*, pode ser traduzido como "Etiquetador morfológico". É uma técnica de rotulamento de palavras, na qual um etiquetador atribui uma categoria morfo-sintática a cada palavra. A tarefa de *POS Tagging* é geralmente designada para fazer a classificação gramatical de uma palavra, ou seja, dado um texto é possível atribuir uma classificação para cada palavra (verbo, substantivo, adjetivo, etc.). Esta rotulação pode ser feita manualmente ou utilizando alguma técnica que a faça automaticamente.

Para a realização de tarefas de *POS Tagging*, o padrão universal, proposto por Petrov, Das e McDonald (2011), define a divisão das classes contidas em um texto em 12 componentes. Este padrão universal é utilizado por diversos algoritmos e bibliotecas, e é ilustrado pela Tabela 3.

Tabela 3: Padrão Universal de *POS Tagging*

| Rótulo | Significado |
|---|---|
| *VERB* | Verbos |
| *NOUN* | Substantivos |
| *PRON* | Pronomes |
| *ADJ* | Adjetivos |
| *ADV* | Advérbios |
| *ADP* | Preposições e Aposições |
| *CONJ* | Conjunções |
| *DET* | Determinantes |
| *NUM* | Números cardinais |
| *PRT* | Partículas ou outras palavras funcionais |
| *X* | Outros: palavras estrangeiras, erros de digitação, abreviaturas |
| . | Pontuação |

Dado o texto de exemplo "*And now for something completely different*" e aplicando a este uma técnica de *Pos Tagging*, a rotulação será como ilustra a Tabela 4.

Tabela 4: Exemplo de rótulo *Pos Tagging*

| Ordem | Palavra | Rótulo |
|---|---|---|
| 1 | *And* | *CONJ* |
| 2 | *Now* | *ADV* |
| 3 | *for* | *ADP* |
| 4 | *something* | *NOUN* |
| 5 | *completely* | *ADV* |
| 6 | *different* | *ADJ* |

O processo de *POS Tagging* pode ser utilizado isoladamente, ou aplicado em conjunto com outros métodos. Dentre eles, o processo de extração de *keyphrases*, descrito na próxima subseção.

### Extração de termos compostos

As técnicas de extração de termos compostos (*keyphrases*) possibilitam a extração das principais frases ou palavras de um texto. Esta técnica permite sintetizar um texto, facilitando não somente trabalho humano, mas também a execução de processos computacionais em cima destes dados. Os termos *key segments, key terms* e *keywords* também são utilizados na literatura para descrever a tarefa de extração de *keyphrase* (BELIGA, 2014). Em cenários de textos de notícias, esses métodos inerentemente escalam grandes coleções e podem ser aplicados em muitos contextos para enriquecer sistemas de recuperação de informação e ferramentas de análise (ROSE et al., 2010).

A aplicação da técnica de extração de *keyphrases* permite o reconhecimento de palavras compostas, ou seja, que um conjunto de palavras sejam reconhecidas como um

único termo. São exemplos de palavras compostas reconhecidas como um único termo: *Hillary Clinton*, *fast-food*, *heart attack*, *movie stars*, *address book*.

Dentre os algoritmos que podem ser utilizados na tarefa extração de *keyphrases*, se destaca o *Rapid Automatic Keyword Extraction (RAKE)* (ROSE et al., 2010).

Os métodos de pré-processamento até aqui vistos têm aplicações em diversos cenários de processamento de textos. Dentre eles, a categorização de textos, explanada na próxima seção.

## 2.3 Métodos de categorização de textos

A categorização de textos, também conhecida como classificação de textos, é a atividade de rotular textos com suas respectivas categorias temáticas a partir de um conjunto de dados pré-definidos. Tendo como finalidade validar a melhor alternativa para realizar a categorização de textos de notícias, foram estudados os principais métodos contidos na literatura, descritos com mais detalhes nas próximas seções.

Os métodos de classificação de textos podem ser de aprendizado *online* ou *offline*, de acordo com a capacidade de construir e atualizar o classificador. Os métodos de aprendizado *online* podem atualizar o modelo de predição (ou classificador) a cada novo documento de maneira incremental, sem necessidade de refazer o treinamento com todos os documentos. Já, os métodos de aprendizado *offline*, a cada nova amostra, o modelo precisa ser recalculado com todos os documentos do treinamento, incluindo a nova amostra.

Cada método de classificação de texto descrito nesta seção inclui a maneira em que atualiza o modelo de predição. Esta propriedade é importante para o ambiente do *Newsminer*, dado que em um cenário de *Data Warehouse*, onde o volume de dados processado é grande, é interessante que o classificador escolhido seja do tipo *online*.

### 2.3.1 *KNN*

O *KNN (K-Nearest Neighbors)*, método do vizinho mais próximo, é um método que trabalha de acordo com a proximidade, ou seja, rotula um determinado objeto de acordo com os objetos mais próximos. O número de objetos a serem agrupados são os `K` vizinhos mais próximos.

A ideia básica do *KNN* é a de determinar a categoria de um determinado objeto baseado em similaridades entre os documentos no espaço. Para calcular similaridade, usa-se uma métrica de distância. O método *KNN* percorre todo conjunto de dados, computando a distância de cada elemento em relação ao documento que esta sendo consultado. Uma vez calculadas as distâncias, a categoria será determinada pelos `K` documentos mais próximos ao documento consultado. O KNN em relação a outros métodos apresenta uma deficiência

no que se refere ao aprendizado *online* ou incremental, pois a cada nova amostra, todas as distâncias precisam ser recalculadas. Portanto, ele é um método de aprendizado *offline*.

Para maiores informações sobre o método *KNN*, consulte Bijalwan et al. (2014).

### 2.3.2 *Naïve Bayes*

*Naïve Bayes* é um método probabilístico, que na categorização de texto, calcula a probabilidade de um documento pertencer a uma categoria, de acordo com a ocorrência das palavras deste documento estarem nas categorias existentes.

Estudos comparativos entre algoritmos de classificação consideram que o classificador *Naïve Bayes* é um classificador simples e com um bom desempenho, quando comparado com a árvore de decisão, redes neurais, entre outros (MCCALLUM; NIGAM et al., 1998). O classificador *Naïve Bayes* também exibe alta precisão e velocidade quando aplicada a grandes bases de dados.

O método *Naïve Bayes* tem diversas variações, das quais o *Naïve Bayes Multinomial* e o *Naïve Bayes Bernoulli* se destacam na realização de tarefas de categorização de textos (DUDA; HART, 1973). O método *Naïve Bayes* é considerado um método de aprendizado *online*, uma vez que suas probabilidades podem ser atualizadas a cada novo documento de maneira incremental, sem necessidade de refazer o treinamento com todos os documentos.

Para maiores informações sobre o método *Naïve Bayes*, consulte Duda e Hart (1973).

### 2.3.3 Regressão logística

O método de regressão logística é uma adaptação da tradicional regressão linear (WALKER; DUNCAN, 1967) com saída binária. A regressão logística é capaz de gerar hipóteses com saídas discretas, e mapeia a relação entre uma variável dependente categórica com uma ou mais variáveis independentes.

A regressão logística pode ser utilizada para modelar uma relação não-linear entre a variável resposta e as variáveis explicativas. Entretanto, esta relação não-linear tem que ser explicitada pelo desenvolvedor do modelo. O método de regressão logística utiliza a probabilidade de algum evento que ocorre como uma função de regressão linear, para descobrir a categoria de uma determinada variável. Este método é considerado um método de aprendizado *offline*.

Para maiores informações sobre o método de regressão logística, consulte Walker e Duncan (1967).

### 2.3.4 Perceptron

A versão inicial do algoritmo do *Perceptron* foi proposto na década de 1950 porRosenblatt (1958). Este algoritmo, em sua versão inicial teve como objetivo replicar o funcionamento da mente humana. A implementação inicial é uma implementação para a aprendizagem de redes neurais simples, de uma camada.

O método perceptron tem diversas variações, desde o *Voted-Perceptron* proposto por Freund, Schapire e Abe (1999) até versões mais complexas que dão origem às redes neurais. Como o erro do *Perceptron* pode ser calculado a cada iteração, pode-se dizer que este método tem a capacidade de aprender de maneira incremental.

Para maiores informações sobre o método de *Perceptron*, consulte Freund, Schapire e Abe (1999).

### 2.3.5 Rocchio

O classificador *Rocchio* divide o espaço vetorial em distintas regiões ao redor de centroides. Este método é baseado no *feedback* de relevância de *Rocchio* para recuperação de documento, no qual pode retroalimentar o processo com informações positivas ou negativas (ROCCHIO, 1971).

No caso do classificador de *Rocchio*, cada categoria apresenta seu próprio centroide ou protótipo. O centroide da categoria é definido positivamente pela média dos documentos pertencentes à categoria e, negativamente pela média dos demais documentos. Este método é considerado de aprendizado *offline*.

Para maiores informações sobre o classificador *Rocchio*, consulte Rocchio (1971).

### 2.3.6 Máquina de vetor de suporte

As máquinas de vetor de suporte *(SVM - Support Vector Machines)* (VAPNIK, 1995) compõem uma técnica de aprendizado de máquina, amplamente utilizada e com excelentes resultados na classificação de textos.

O *SVM* é um novo método para a classificação de dados lineares e não lineares. Para o cálculo do *SVM*, são considerados o *bias*, o tamanho do vetor, o vetor de teste e o vetor de suporte.

As máquinas de vetor de suporte são embasadas pela teoria de aprendizado estatístico. Essa teoria estabelece uma série de princípios que devem ser seguidos na obtenção de classificadores com boa generalização, definida como a sua capacidade de prever corretamente a classe de novos dados do mesmo domínio em que o aprendizado ocorreu. Este método é considerado de aprendizado *offline*.

Para maiores informações sobre o método *SVM*, consulte Vapnik (1995).

### 2.3.7 Passive agressive

O classificador *Passive Agressive* é oriundo de uma família de classificadores baseados no método *SVM*. Este método tem como principal objetivo a mesma robustez do método *SVM* em uma implementação que possa aprender de maneira incremental.

A principal implementação do método *Passive Agressive* foi no proposta por Crammer et al. (2006). Neste trabalho, o núcleo do algoritmo *Passive Agressive* é composto por uma máquina de vetor de suporte com uma implementação próxima dos tradicionais métodos *SVM*. A solução do *Passive Agressive* não é apenas uma, mas sim uma família de algoritmos que realizam o aprendizado *online*. Estes algoritmos têm a capacidade de classificar tanto classes binárias quanto em exemplos de multi-classes.

Para maiores informações sobre o método *Passive Agressive*, consulte Crammer et al. (2006).

### 2.3.8 Gradiente descendente estocástico

O Gradiente Descendente Estocástico (*Stochastic Gradient Descent* - SGD) é uma variação do tradicional método Gradiente Descendente (*Gradient Descent*) (ZINKEVICH, 2003). Estes métodos são muito utilizados em problemas de classificação, permitindo ajustar os parâmetros de uma função a fim de minimizar a diferença entre uma hipótese e um conjunto de treinamento.

Em ambos os gradientes, o objetivo é o mesmo: atualizar um conjunto de parâmetros de forma iterativa para minimizar uma função de erro. Todavia, a principal diferença entre estes é que *Gradient Descent* percorre todas as amostras de um conjunto de treinamento para fazer o ajuste de parâmetro. Por sua vez, o *Stochastic Gradient Descent* utiliza somente uma amostra de treinamento para fazer o ajuste de um parâmetro em uma iteração específica (ZINKEVICH, 2003).

Para maiores informações sobre o método *SGD*, consulte Zinkevich (2003).

## 2.4 Sumário

Neste capítulo, foram apresentados os conceitos relacionados a ambientes de *Data Warehouses*, que serão utilizados na proposta da arquitetura do *Newsminer* no Capítulo 4. A representação e o pré-processamento de textos são utilizados na etapa de ETL dessa arquitetura. Também, foram brevemente descritos os métodos de classificação que serão utilizados no Experimento 1, exposto no Capítulo 5. Cada método foi rotulado como de aprendizado *online* ou *offline*, dado que a propriedade de aprendizado *online* é importante

na escolha do método que irá fazer parte do ambiente do *Newsminer*. Em um cenário de *Data Warehouse*, onde (1) existe um grande volume de dados a ser processado, tanto na etapa de ETL quanto para o consumo das aplicações e usuários, e (2) visa-se o consumo próximo do tempo real a partir das fontes de dados (*sites* de notícias), é desejável que o modelo de predição do classificador possa ser atualizado de maneira incremental. Os métodos *offline* não foram descartados, pois outros parâmetros serão considerados na escolha. Eles são: o desempenho quanto à qualidade da classificação e quanto ao tempo de processamento.

# 3 ESTADO DA ARTE

Este capítulo apresenta os trabalhos da literatura científica atual que estão relacionados com o objetivo desta dissertação. A revisão da literatura foi dividida em quatro tópicos: (i) integração entre *Data Warehousing* e Mineração de Dados, (ii) *OLAP* em tempo real, (iii) coleta de dados de notícias em tempo real, e (iv) categorização de textos de notícias. Para cada seção, os trabalhos foram descritos seguindo a ordem cronológica.

## 3.1 Integração entre *Data Warehousing* e Mineração de dados

Os processos de mineração de dados têm a necessidade de obtenção de dados de qualidade, quando isto não acontece é gasto grande parte do tempo realizando o pré-processamento destes dados. Este tempo pode ser economizado, caso a fonte de dados seja proveniente de um ambiente de *Data Warehouse*.

Em contrapartida, os algoritmos de mineração de dados podem ser aplicados na etapa de *ETL* com a finalidade de agregar conhecimento e enriquecer os dados armazenados em um ambiente de *Data Warehouse*. As técnicas de descoberta de conhecimento têm sido empregadas para enriquecer semanticamente os dados e metadados em ambientes de Data Warehouse durante o processo *ETL*. Neste trabalho, utiliza-se o termo *enriquecimento semântico* cujo embasamento provém da abordagem descrita em (MANSMANN et al., 2014). Os autores introduzem de uma camada de enriquecimento de dados responsável por detectar novos elementos estruturais nos dados usando a mineração de dados e outras técnicas. No caso, os elementos descobertos podem ser métricas, dimensão ou nível hierárquico e podem representar propriedades estáticas ou dinâmicas dos dados.

Visando complementar a arquitetura de um *Data Warehouse* para textos, Prasad e Ramakrishna (2010) utiliza algoritmos de categorização de textos integrados com a etapa de *ETL*. Tal integração permite que o enriquecimento semântico ocorra, implicando que dados não estruturados sejam analisados e, posteriormente armazenados em um *Data Warehouse* com os devidos campos reconhecidos automaticamente.

As propostas descritas em (REHMAN et al., 2012) e (MANSMANN et al., 2014) utilizam uma arquitetura de *Data Warehouse* em cinco camadas. A primeira camada é a fonte de dados. Nesse caso, é um modelo de banco de dados relacional para o *Twitter*. Na segunda camada, o processo de *ETL* foi integrado com técnicas de mineração de dados, com o propósito de permitir o enriquecimento semântico por meio da descoberta de novos elementos textuais, o que gerou novas dimensões de análise. O *Data Warehouse* é responsável por armazenar multidimensionalmente dados oriundos do *Twitter*. A camada

de ferramentas de acesso a dados, por sua vez, integra as ferramentas de análise com métodos de mineração de dados que realizam a classificação de mensagens utilizando análise de sentimento.

Dentro do contexto do *Twitter* utilizado por Mansmann et al. (2014), a aplicação de técnicas de Mineração de Texto obteve o enriquecimento semântico dos dados armazenados. No conjunto de dados original, apenas dois campos são armazenados: descrição de usuário e *Tweet*, ambos dados de texto. O campo *Tweet* pode incluir nomes de usuários, *URLs* de *sites* externos, fotos e vídeos. Estes dois campos são fundamentais para a análise semântica, fornecendo informações valiosas sobre os usuários e suas opiniões.

No projeto desenvolvido por Tao et al. (2013), os dados de agências de notícias foram recebidos e disponibilizados em um sistema *Web* denominado *EventCube*. Neste sistema, o *OLAP* é integrado com métodos de mineração de dados, obtendo enriquecimento semântico com a descoberta de novos segmentos, permitindo construir termos, encontrar hierarquias, construir índices e calcular métricas. Após uma etapa de pré-processamento, o processo de análise é passado ao usuário, para que o mesmo possa conduzir os componentes analíticos.

Renso, Roncato e Trasarti (2014) desenvolveram um *Trajectory Data Warehouse*, que armazena dados de trajetória de pessoas, permitindo uma análise futura. As análises deste projeto têm como objetivo responder perguntas como *qual é a distância média percorrida pelas pessoas que usam o transporte público para visitar pelo menos uma atração cultural?*. Tendo como finalidade otimizar as consultas para responder análise, o processo *ETL* inclui uma etapa de enriquecimento semântico de dados de mobilidade, cujo objetivo é associar informação semântica do domínio de aplicação com os dados da trajetória.

Em seu trabalho, Victor e Rex (2016) utiliza *sites* de universidades como fonte de dados para alimentar um *Data Warehouse*. Neste cenário, os algoritmos de mineração de dados são integrados as consultas *OLAP* que, ao final, realizam *ranking* de *sites*.

Conforme visto nesta seção, a integração de *Data Warehouse* e *OLAP* com Mineração de Dados pode trazer benefícios para essas áreas de pesquisa. No entanto, o maior desafio dos dias atuais é lidar com a velocidade de atualização de informações. Visando solucionar esta problemática, existem as aplicações *Data Warehouse* em tempo real, abordadas na próxima seção.

## 3.2 OLAP em tempo real

Na maioria dos sistemas comerciais, as ferramentas de ETL extraem os dados dos sistemas de origem periodicamente (por exemplo: diariamente ou semanalmente). Nos dias de hoje, os usuários do DW querem as informações de BI atuais e atualizadas. Em alguns

segmentos de negócio, como *e-business*, corretagem de ações, telecomunicações *online* e sistemas de saúde, a informação relevante deve ser entregue o mais rápido possível para os gestores do conhecimento ou sistemas de decisão que contam com ele para reagir próximo do tempo real, de acordo com os novos e mais recentes dados capturados pelo sistema de informação de uma organização (SANTOS; BERNARDINO, 2008).

*OLAP* em tempo real e próximo do tempo real é definido em (GARCIA; TANAKA; BAIÃO, 2015) como:

> Tempo real é uma definição que contempla *OLAP* no acesso aos dados históricos, assim como dados operacionais recentemente atualizados, quase em linha de tempo com as transações. O objetivo é alcançar a menor latência possível, ou seja, a menor diferença entre o tempo em que o dado chegou à base de dados operacional e está disponível para análise no *Data Warehouse*.

Para atingir tempo real em um ambiente de DW, o tempo decorrido entre o evento e sua consequente ação (chamado de *latência de dados*) precisa ser minimizado. Na maior parte dos casos, o processo de aquisição de dados introduz a maior latência de dados (VAISMAN; ZIMÁNYI, 2012). Os sistemas *OLTP*, bem como as fontes de dados, já operam em tempo real. No entanto, os dados mudam a todo momento, não permitindo uma análise concisa, fazendo com que o desafio de um ambiente de *OLAP* em tempo real seja obter dados em tempo real e que esses dados sejam consistentes.

Abello et al. (2015) propõem uma solução para cenários de análise em tempo real, envolvendo dados da *Web*.Esta proposta consiste em uma arquitetura para *Data Warehouse* que integra fontes de dados estruturados e não estruturados. Ela serviu de embasamento para o desenvolvimento da arquitetura do *Newsminer* e é ilustrada na Figura 8. As três camadas definidas são:

- *Fontes de Dados*: camada que consiste em todos os dados possíveis de qualquer natureza (relacional, orientada a objetos, semi-estruturados, não estruturados , textual, etc.) que podem ser integrados, visando atingir as tarefas analíticas.
- *Integração dos Dados*: camada responsável pela realização da transformação e limpeza dos dados coletados, bem como seu armazenamento em um formato adequado ao banco de dados analítico que foi projetado.
- *Análise*: camada que contém uma série de ferramentas para extrair informação e conhecimento a partir dos dados integrados e apresentá-los na forma de relatórios ou outros tipos de visualização.

Em seu trabalho, Dehne et al. (2015) apresentam o *CR-OLAP*, um sistema baseado na computação em nuvem e que realiza *OLAP* em tempo real. Para este fim, é utilizado

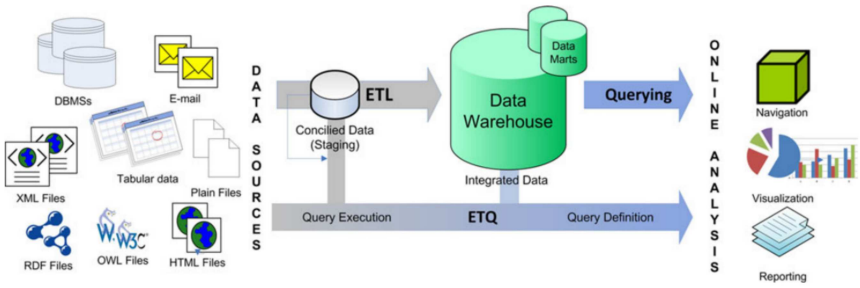



Figura 8: Arquitetura Data Warehouse. Fonte: (ABELLO et al., 2015)

uma infraestrutura de nuvem escalável, composta por vários servidores de *commodities*. Sendo assim, conforme os dados aumentam, o número de servidores também aumenta dinamicamente para manter o processamento. Os testes mostram que o *CR-OLAP* se equilibra bem com o número crescente de processadores, mesmo para consultas complexas, com resultados que podem ser considerados como resposta em tempo real.

Com o objetivo de atingir a execução de *OLAP* em tempo real, ou próximo do tempo real, a partir das fontes de dados de *sites* de notícias, a próxima seção aborda uma técnica para coleta de dados *online*, o *Web Crawler*.

## 3.3 Coleta de notícias em tempo real

*Web Crawler*, também denominado rastreador *Web*, *Web Spiders* ou *Web robots*, é um programa ou *script* de computador que percorre páginas da Internet rastreando e armazenando determinado conteúdo. Suas aplicações podem variar desde projetos acadêmicos até grandes buscadores como o *Google*, que utilizam esses *robots* para varrer toda a Internet. *Web Crawlers* são os componentes principais dos motores de busca, a partir de sua utilização, é possível reunir facilmente informações na Internet (BAHRAMI; SINGHAL; ZHUANG, 2015).

(SILVA et al., 2011) desenvolveu um sistema de coleta de notícias dos *sites* do governo brasileiro. Este projeto fornece subsídios para o governo brasileiro avaliar suas ações de comunicação. Para isto, utilizou *J2EE (Java 2 Enterprise Edition)* fazendo uso do padrão *MDA (Model Driven Architecture)*.

Para realizar a coleta de notícias, Zhiqiang, Wei e Guixian (2015) desenvolveram um *Web Crawler* com linguagem de programação *Python* em conjunto com a biblioteca *Scrapy*, para coletar notícias de *sites* tibetanos. A finalidade foi o desenvolvimento de um motor de busca integrado com o *SOLR*, um projeto de código aberto para buscas e indexação de alto desempenho.

Tendo como finalidade a construção de um sistema de leitura de notícias rápido

e automático[1], Bai et al. (2015) desenvolveu um *Web Crawler*. Para este fim, utilizou a linguagem *Java* para coletar notícias de 120 *sites* chineses, divididos em 19 categorias.

Há casos em que não se faz necessário o desenvolvimento de um *crawler* para a obtenção de dados de um *site*. No caso de Iglesias et al. (2015), os autores utilizaram a *Article Search API v2* [2], que pertence ao jornal americano *The New York Times*, que permite, através de parâmetros, coletar notícias armazenadas desde o ano de 1850. No desenvolvimento de um sistema integrador de notícias, que realiza a recomendação de textos, Jonnalagedda et al. (2016) utilizou a *API* de dados abertos do *Twitter* para coletar opinião dos usuários para realizar a tarefa de recomendação de textos de notícias.

Em um projeto que realiza o *ranking* de *sites* da *Web* utilizando um ambiente de *Data Warehouse*, Victor e Rex (2016) desenvolveu um *Web Crawler* para coletar dados de diversos *sites* de universidades. Neste projeto o *Web Crawler* tem um papel fundamental, pois além de coletar os dados de um *site* em específico, coleta dados de outros *sites* contidos nele, *URLS*, por exemplo, que são importantes para a construção do *ranking*.

Após a coleta de dados via *Web Crawler*, estes dados podem ser utilizado para diversas tarefas computacionais, dentre estas, a categorização de textos, que é o caso deste trabalho de pequisa. A revisão da literatura sobre categorização de textos de notícias é explorada na próxima seção.

## 3.4 Categorização de textos de notícias

Os *sites* de notícias são o terceiro maior veículo de informação mais acessado da Internet, perdendo apenas para aplicativos de mensagens e redes sociais[3]. Esta informação reflete a importância do uso de *sites* de notícias e seu impacto no cotidiano das pessoas.

A categorização de textos de notícias é a tarefa de rotular uma determinada notícia, baseando-se em um conjunto de textos previamente rotulado. Esta seção tem como finalidade apresentar trabalhos que realizam a categorização de textos de notícias. Os trabalhos escolhidos para serem descritos nesta seção utilizam a ontologia *IPTC*, pois esta será a ontologia abordada no *Newsminer*.

No desenvolvimento de um *SNRS - Social Network Based on Recomendation Systems* (Sistema de Recomendação Baseado em Redes Sociais), Agarwal e Singhal (2013) identificam as entidades nomeadas nos itens de notícias usando a ferramenta *NER - Named Entity Recognizer* (Reconhecimento de Entidade Nomeada) e, em seguida, identifica o contexto correto da entidade usando a categoria de notícias com base na classificação

[1] *Kuaiwenbao System*. Disponível em <http://www.kuaiwenbao.com>, acessado em 15/01/2016.
[2] *Article Search API v2*. Disponível em <https://developer.nytimes.com>, acessado em 15/03/2016.
[3] Estudo sobre uso de tecnologias. Disponível em <http://www.brasil.gov.br/governo/2014/12/cerca-de-48-dos-brasileiros-usam-internet-regularmente>, acessado em 15/12/2015

semântica de notícias. Para este fim, a ontologia *IPTC* foi utilizada para categorizar 50 notícias coletadas via *RSS*, obtendo uma F-medida de 86%.

Para a realização de categorização de notícias, no trabalho desenvolvido por Liparas et al. (2014), foram consideradas quatro categorias da ontologia *IPTC*: *Business, finance; Lifestyle and leisure; Science and technology* e *Sports*. Para compor o conjunto de dados, foram coletados 1043 textos, utilizando notícias dos *sites* da *BBC*, *The Guardian* e *Reuter*. Na etapa de pré-processamento foram removidas as *stopwords*. O método de categorização utilizado foi o *Random Forest*, que obteve 83.2% de F-medida e 84.4% de acurácia.

Mangal e Goyal (2014) utiliza o método *Naïve Bayes* para rotular 141 textos de notícias divididos em 4 categorias: *Terror Attack News, Murder Related News, Accidental News* e *Suicide News*. A F-medida do método nesse cenário foi de 74% e acurácia de 72%.

Mohiuddin, Ahmed e Ismail (2015) propõem um sistema para classificação em tempo real. O pré-processamento é feito em 5 etapas: análise léxica, *steamming*, remoção de *stopwords*, seleção de termos indexados e *thesauri*. O conjunto de dados é composto por notícias de 12 *sites*, que foram coletadas fazendo uso da *API* do *Facebook*, dispostas nas categorias *Crime, International, National, Sports, Entertainment, Politics* e *General*. Para realizar a categorização, foram utilizados os métodos *NaïveBayes* e *J48*, obtendo acurácia de 61.28% e 52.14%, respectivamente.

Dehankar, Wagh e Chatur (2015) realiza a categorização de páginas *Web* utilizando 9 categorias de *sites* de notícias baseado na ontologia *IPTC*: *Hotel, Hospital, Computer, Sports, Academic Institutions, Bank, Tours and Travel* e *Domestic Applications*. O método utilizado foi o *Naïve Bayes Apriori*, que utiliza regra de suporte e confiança para eliminar termos. Nesse trabalho a acurácia foi de 61.98% e a F-medida de 75.91%.

No trabalho desenvolvido por Gong e Song (2016), é realizado uma avaliação da aplicação de *thesaurus* e seu impacto na classificação de textos de notícias. Para este fim, foram utilizadas 1500 mensagens, distribuídas em 24 categorias, para teste e outras 100 para treino. Os resultados de acurácia e F-medida foram 76.1% e 76.4%, respectivamente, sem a aplicação do *thesaurus*, e 77.8% e 78%, respectivamente, após a aplicação do mesmo.

## 3.5 Sumário

Tendo como finalidade sumarizar os principais trabalhos relacionados, vistos neste capítulo, e sua associação com o *Newsminer*, a Tabela 5 destaca os recursos empregados por esses trabalhos. As linhas representam os trabalhos relacionados e as colunas a característica de interesse: obtenção de dados em tempo real mediante o uso de *Web Crawler (WC)*, *sites* de notícias (SN) como fonte de dados, adição de semântica por meio de ontologia (IPTC), desenvolvimento em ambiente de *Data Warehouse/OLAP*(DW), categorização de

textos (CT), e enriquecimento semântico (ES) na etapa de ETL.

Tabela 5: Comparação dos trabalhos da literatura com o *Newsminer*

| Trabalho | WC | SN | IPTC | DW | CT | ES |
|---|---|---|---|---|---|---|
| Prasad e Ramakrishna (2010) | | | | ■ | ■ | ■ |
| Silva et al. (2011) | ■ | ■ | | | | |
| Rehman et al. (2012), Mansmann et al. (2014) | | | | ■ | ■ | ■ |
| Agarwal e Singhal (2013) | | ■ | ■ | | ■ | |
| Tao et al. (2013) | | ■ | | ■ | | ■ |
| Renso, Roncato e Trasarti (2014) | | | | ■ | ■ | ■ |
| Liparas et al. (2014) | | ■ | ■ | | ■ | |
| Mangal e Goyal (2014) | | ■ | | | ■ | |
| Iglesias et al. (2015) | | ■ | ■ | | | |
| Mohiuddin, Ahmed e Ismail (2015) | | ■ | | | ■ | |
| Bai et al. (2015) | ■ | ■ | | | | |
| Dehankar, Wagh e Chatur (2015) | | ■ | ■ | | ■ | |
| Zhiqiang, Wei e Guixian (2015) | ■ | ■ | | | | |
| Victor e Rex (2016) | ■ | | | ■ | | ■ |
| Jonnalagedda et al. (2016) | | ■ | | | | |
| *Newsminer* | ■ | ■ | ■ | ■ | ■ | ■ |

Na Tabela 5, pode se observar a relação entre os trabalhos da literatura entre si, bem como em relação ao *Newsminer*. Os trabalhos da literatura integram, no máximo, três dos seis recursos definidos, sendo que o *Newsminer* se destaca por trazer uma proposta que consegue integrar todos eles. Os trabalhos mais próximos do *Newsminer* são os desenvolvidos em (REHMAN et al., 2012), (MANSMANN et al., 2014), que propõem uma integração entre OLAP e Mineração de Dados a partir de dados da Web, e o descrito em (VICTOR; REX, 2016), que utiliza um *Web Crawler* para coletar textos de notícias, armazenando-os em um *Data Warehouse*. No entanto, todos os trabalhos se tornam importantes, pois foram utilizados como embasamento teórico e científico para o desenvolvimento desta pesquisa.

O próximo capítulo descreve o *Newsminer*, um ambiente de *Data Warehouse*, que armazena um corpus baseado em textos de notícias, coletados da *Web* via *crawler*. O *Newsminer* visa a integração com aplicações de descoberta de conhecimento, como as de mineração de dados, com o propósito de oferecer um conjunto de dados consistente e limpo em um esquema de dados multidimensional.

# 4 NEWSMINER

Como descrito no Capítulo 1, o intuito desta dissertação de mestrado é responder a questão de pesquisa: *é possível fazer uma integração entre as fontes de dados Web e as aplicações de Mineração de Dados e Textos por meio de um ambiente de Data Warehousing?* Este capítulo descreve o ambiente *Newsminer*, que visa essa integração por meio da proposta de uma arquitetura de *Data Warehouse*.

## 4.1 Objetivos do ambiente

O objetivo do ambiente *Newsminer* é fornecer um conjunto de dados consistentes e limpos, na forma de um *corpus* multidimensional para consumo por aplicações externas e usuários. Para isto, o ambiente é sustentado por uma arquitetura que visa integrar:

- um modelo multidimensional que armazene o conjunto de textos e a característica temporal das notícias;
- a coleta de textos de notícias em tempo próximo do tempo real;
- anotações semânticas de maneira dinâmica no ambiente;
- qualquer cubo de dados de consultas multidimensionais requisitadas por aplicações e usuários.

Com o *corpus* sendo armazenado seguindo um modelo multidimensional, definiu que fosse chamado de *corpus multidimensional*, pois as aplicações/usuários poderão explorar a multidimensionalidade e os diferentes níveis de abstração associados ao conjunto de textos.

## 4.2 Arquitetura

A arquitetura do *Newsminer* é baseada na arquitetura de *Data Warehouse* proposta por Abello et al. (2015), ilustrada na Figura 8 do Capítulo 2. Abello et al. (2015) propõem um processo denominado *ETQ (Extract, Transform, Query)*, que realiza consultas dinâmicas em variadas fontes de dados (sistemas OLTP, Web, etc.), visando alimentar a demanda *online* dos usuários. Assim, os resultados do processamento OLAP podem integrar dados de todas as fontes relevantes às consultas realizadas. No caso do *Newsminer*, todas as fontes de dados são dinâmicas, pois são notícias coletadas a partir de *websites*. A arquitetura do *Newsminer* é ilustrada na Figura 9.

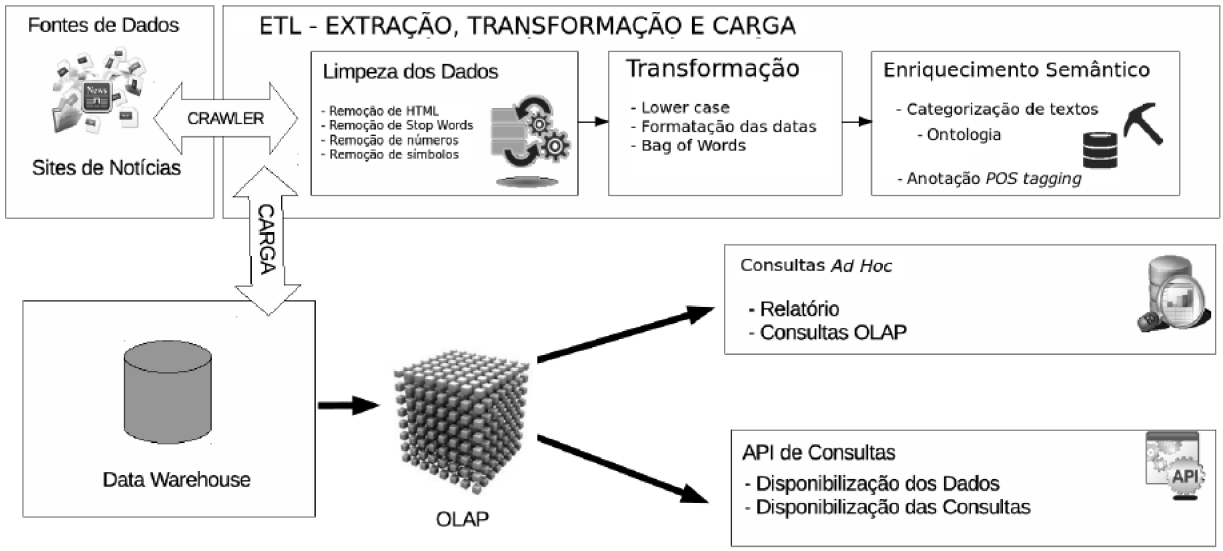



Figura 9: Arquitetura do *Newsminer*

De maneira geral, para que as notícias sejam armazenadas de acordo com o modelo multidimensional e os dados disponibilizados para as aplicações, a coleta de notícias é realizada previamente, bem como o seu pré-processamento, compondo a etapa de *ETL*.

O processo de *ETL*, juntamente com os demais elementos que compõem Cada elemento é descrito com maior nível de detalhamento na sequência.

- Fonte de dados: as fontes de dados são *sites* de notícias em inglês. Para fundamentar a escolha dos *sites* utilizados, foi utilizada como referência os dados do *site* Alexa[1] que atua ordenando os *sites* da Internet, de acordo com o número de acessos.

- ETL: esta etapa é responsável por coletar os dados das fontes de origem, prepará-los e realizar o armazenamento no banco de dados multidimensional. A etapa de *ETL* do *Newsminer* é formada pelos seguintes itens:

  1. *Web crawler*: é responsável por percorrer os *sites* e coletar os textos notícias.
  2. Limpeza de dados: nesta etapa é realizada a remoção de caracteres não alfabéticos, que podem impedir a consistência e a qualidade dos dados armazenados.
  3. Transformação: os dados de textos de notícias são em sua essência não estruturados. Esta etapa realiza o tratamento específico para que os textos possam ser armazenados de acordo com o modelo multidimensional e na forma do *corpus* linguístico.
  4. Enriquecimento Semântico: com o intuito de acrescentar anotações semânticas ao corpus, foram definidos, inicialmente, dois tipos: (1) anotação *POS Tagging*

[1] *Alexa: Keyword Research,Analysis and Ranking.* Disponível em <http://www.alexa.com>, acessado em 05/01/2017

e (2) a descoberta da categoria para aquelas notícias que não a possuem quando trazidas da fonte.

5. Carga: o processo de carga tem a finalidade de armazenar os dados já limpos e transformados no banco de dados multidimensional.

- Banco de dados multidimensional (*Data Warehouse*): é responsável por armazenar os dados segundo o modelo multidimensional proposto.
- Servidor *OLAP*: permite a exploração do modelo a fim de fornecer consultas multidimensionais dispondo dos dados armazenados.
- Consultas *Ad Hoc*: permite visualizar e explorar os resultados das consultas pré-definidas, permitindo o uso de parâmetros ajustados pelo usuário.
- API de consulta: esta etapa é responsável por fornecer dados para aplicações externas.

Para maior compreensão da arquitetura e funcionamento do *Newsminer*, as próximas seções descrevem, com maior nível de detalhamento, cada um de seus elementos.

### 4.2.1 ETL: Web Crawler

Com o propósito de conseguir dados em tempo real, foi desenvolvido um *Web Crawler*. Ele é responsável por percorrer os *sites* de notícias, coletando os textos, juntamente com as categorias e data de publicação. Visando garantir melhor a qualidade dos dados, o *crawler* utiliza filtros nos metadados contidos no *site* (imagens, vídeos, *tags HTML*, etc.). Estes filtros garantem que sejam coletados apenas os textos de notícias.

Para o desenvolvimento do *Web Crawler* foi utilizada a linguagem de programação *Python*[2] em conjunto com a biblioteca *Scrapy*[3]. O funcionamento do *Scrapy* é dividido em duas etapas principais, o *crawling* que é responsável por seguir *URLs* de uma página a outra e o *scraping* que extrai conteúdo das páginas. Atualmente, os *sites* de notícias para os quais o *Web Crawler* foi configurado são sete: The Guardian[4], CNN[5], BBC[6], Fox News[7], NyPost[8], China Daily [9] e NBC[10].

O funcionamento do *Web Crawler* do *Newsminer* é ilustrado pela Figura 10. A tabela de `URLs` armazena uma lista de *sites*: o campo `VISITADO` indica se os dados de um

[2] *Python home page*. Disponível em <https://www.python.org/>, acessado em 21/04/2015
[3] *Scrapy home page*. Disponível em <https://scrapy.org>, acessado em 15/03/2016
[4] www.theguardian.com
[5] www.cnn.com
[6] www.bbc.com
[7] www.foxnews.com
[8] nypost.com
[9] www.chinadaily.com.cn
[10] www.nbcnews.com

determinado *site* já foram coletados, o campo `CATEGORIA` armazena o código da categoria da notícia, a `DATA` armazena a data de publicação da notícia e o campo `ESTILO` contém o estilo *CSS* que permitirá a análise dos metadados e coletar apenas o texto da notícia do *site* visitado. A etapa de *crawling* verifica se o *site* não foi visitado e percorre a *Web* obtendo páginas. Por sua vez, a etapa *scraping* é responsável por analisar os metadados contidos no *site*, e, baseado no estilo contido na tabela de *URLs*, coletar apenas o texto e a categoria da notícia. Também, armazena todas as *URLs* que direcionam para outros *sites* e a data destes. Posterior à coleta, os dados são pré-processamento e armazenados no banco de dados multidimensional.

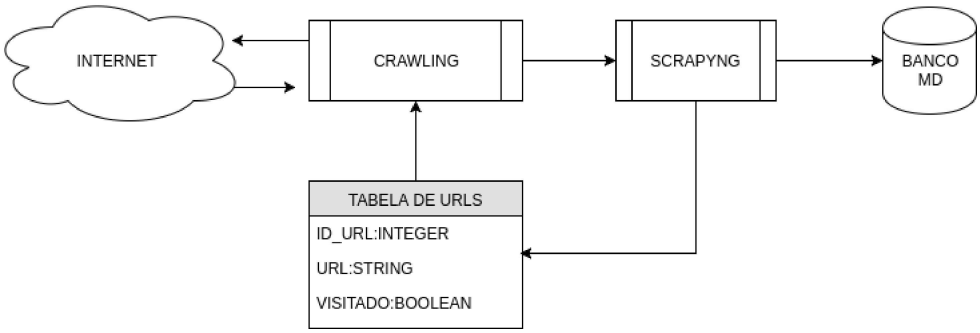



Figura 10: Funcionamento do *Web Crawler* do *Newsminer*

Os atributos da tabela de `URLs` podem ser utilizados para coletar notícias específicas com a aplicação de filtros. Com isso, por exemplo, se houver poucas notícias armazenadas da categoria de esportes, o *Web Crawler* pode ser parametrizado para coletar mais notícias dessa categoria, deixando a base de notícias balanceada com relação às categorias.

Dentro do ambiente, está previsto a ativação automática do *Web Crawler* quando não houver dados armazenados pertinentes a uma consulta submetida. Para isto, o *Web Crawler* será parametrizado para coletar dados relacionados na *Web*. Uma vez que os dados de notícias são coletados dos *sites*, trazem consigo as características dos textos *HTML*. Dessa maneira, é necessário aplicar técnicas de limpeza de dados. A próxima seção descreve os métodos de limpeza aplicados na etapa e *ETL* do *Newsminer*.

### 4.2.2 ETL: Limpeza dos dados

A etapa da limpeza de dados é importante, pois é nela que são removidos os dados desnecessários, que além de ocupar espaço em disco, podem atrapalhar o desempenho dos métodos computacionais que utilizam os dados armazenados.

A limpeza dos dados passa por três fases: a remoção de *tags HTML*, remoção de caracteres não alfabéticos e a remoção de *stopwords*. Como as fontes de dados utilizadas para alimentar o *Newsminer* são *sites*, os dados não são estruturados e contêm *tags HTML*.

Desse modo, faz-se necessária a remoção dessas *tags*, tornando o texto mais limpo. A Figura 11 mostra um texto antes e depois da remoção de *tags*.

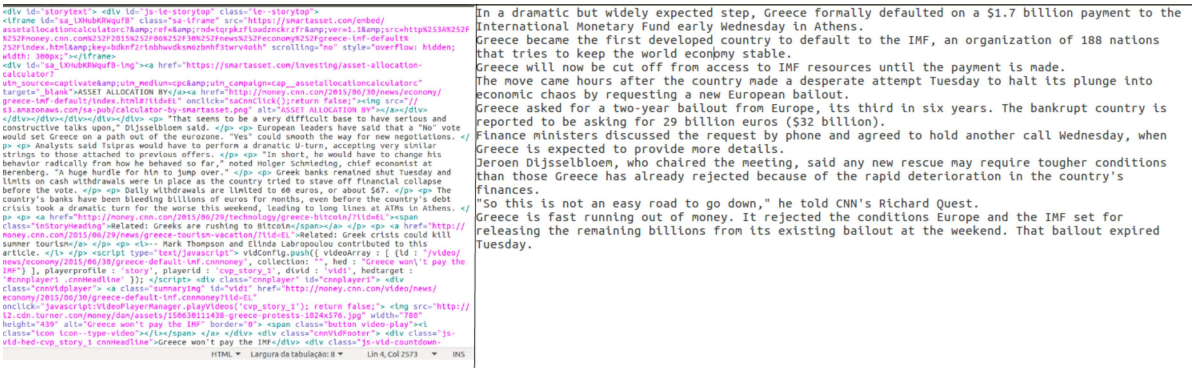



Figura 11: Exemplo de um texto antes e depois da remoção de *tags HTML*

Com o texto sem *tags HTML*, é realizada a remoção de todos os caracteres não alfabéticos, ou seja, números e símbolos. A última etapa de limpeza é a remoção das *stopwords* baseada na lista fornecida pela biblioteca *NLTK*[11].

Após a realização da limpeza dos dados, é executada a transformação dos dados. Esta etapa é descrita na próxima seção.

### 4.2.3 ETL: Transformação

Na fase de transformação, os dados coletados são preparados para serem adequados ao formato do banco de dados do modelo multidimensional. Nesta etapa, também, podem ser aplicados métodos de pré-processamento que tenham como finalidade a otimização do desempenho computacional das tarefas que venham a ser executadas utilizando estes dados.

As *URLs* das notícias coletadas pelo *Web Crawler* permitem em sua estrutura[12] reconhecer a data de publicação. Porém, esta data precisa ser transformada para ser armazenada no modelo multidimensional do *Newsminer*. Na etapa de transformação da *ETL* é realizado o reconhecimento das datas, padronização e armazenamento de acordo com o modelo.

A estratégia de padronização utiliza uma técnica simples: a transformação dos textos para caracteres minúsculos (*lower case*). Em seguida, é contabilizado o número de ocorrências dos termos que aparecem no texto. Consequentemente, cada texto será transformado na representação *Bag of Words*, sendo compreendido como um documento

[11] *NLTK home page*. Disponível em <http://www.nltk.org>, acessado em 01/04/2017.

[12] Exemplo de *URL*. Disponível em <http://edition.cnn.com/2017/04/16/europe/turkey-referendum-results-erdogan/index.html>, acessado em 16/04/2017

com frequências de termos. Se o termo existir no banco de dados, o processo de carga atualiza sua frequência.

A próxima transformação está relacionada com escolher o melhor método de pré-processamento que oferece a melhor categorização de notícias, na próxima etapa de enriquecimento semântico.

### 4.2.4 ETL: Métodos de pré-processamento de texto

A aplicação do pré-processamento de dados na etapa de *ETL* tem como objetivo auxiliar o método de categorização de textos a obter melhores resultados no enriquecimento semântico e na realização de consultas multidimensionais. Esses métodos de pré-processamento de textos serão avaliados em conjunto com os métodos de categorização de textos no Capítulo 5. Após a avaliação, apenas a transformação que obteve melhores resultados para a categorização de notícias será executada dentro do ambiente.

O pré-processamento de dados garante a redução da dimensionalidade dos textos coletados, e possibilita que os dados sejam armazenados já pré-processados no banco multidimensional. Os tipos de pré-processamento considerados são:

1. remoção de *stopwords*: armazena o texto completo e todos os termos obtidos após a etapa de limpeza;
2. substantivos: armazena o texto completo e todos os termos do tipo substantivo obtidos após a etapa de limpeza. Para isso, utiliza-se uma biblioteca de *POS - Tagging*, que permite reconhecer a classe gramatical das palavras;
3. termos compostos: armazena o texto completo e todos os termos compostos obtidos após a etapa de limpeza. Para isto, uma biblioteca de extração de *keyphrases* é utilizada para reconhecer os termos compostos.

### 4.2.5 ETL: Enriquecimento semântico

Neste trabalho, o embasamento do termo *enriquecimento semântico* provém da abordagem descrita em (MANSMANN et al., 2014). Os autores introduzem uma camada de enriquecimento de dados responsável por detectar novos elementos estruturais usando mineração de dados e outras técnicas. No caso, os elementos descobertos podem ser métricas, dimensão ou nível hierárquico e podem representar propriedades estáticas ou dinâmicas dos dados.

A proposta de um módulo de enriquecimento semântico acrescenta propriedades semânticas aos dados coletados. Inicialmente, foram definidos dois tipos: a categoria da notícia e a anotação *POS Tagging* para cada termo do texto. Para obter a anotação *POS Tagging*, é aplicado um algoritmo que reconhece a classe morfológica de cada termo.

Até dezembro de 2016, cerca de 200.000 notícias foram coletadas pelo *Newsminer*. Dessas, cerca de 27.5% não tem categoria definida pela fonte. Por isso, a categorização das notícias coletadas tornou-se alvo de enriquecimento semântico. A adição de um classificador de notícias, baseado em 17 categorias de notícias da ontologia *IPTC* (SMOOT, 2003), permite oferecer para as aplicações e usuários conjuntos dos textos categorizados. Para descobrir a categoria das notícias, foram avaliados nove métodos de classificação, geralmente usados na literatura. O método escolhido é aquele que representa a melhor generalização para textos de notícias e foi incorporado à etapa de *ETL* do *Newsminer*. Esses resultados são discutidos no Capítulo 5.

Com as descobertas semânticas, o conjunto de textos, armazenado de acordo com o modelo multidimensional, é enriquecido.

### 4.2.6 Banco de dados multidimensional de textos de notícias

O banco de dados multidimensional do *Newsminer* consolida e armazena os textos de notícias que foram coletados e pré-processados. Sendo assim, fornece os recursos para observar as notícias por diferentes perspectivas e realizar consultas multidimensionais.

O modelo multidimensional, ilustrado na Figura 12, representa o projeto lógico do banco de dados multidimensional do *Newsminer*.

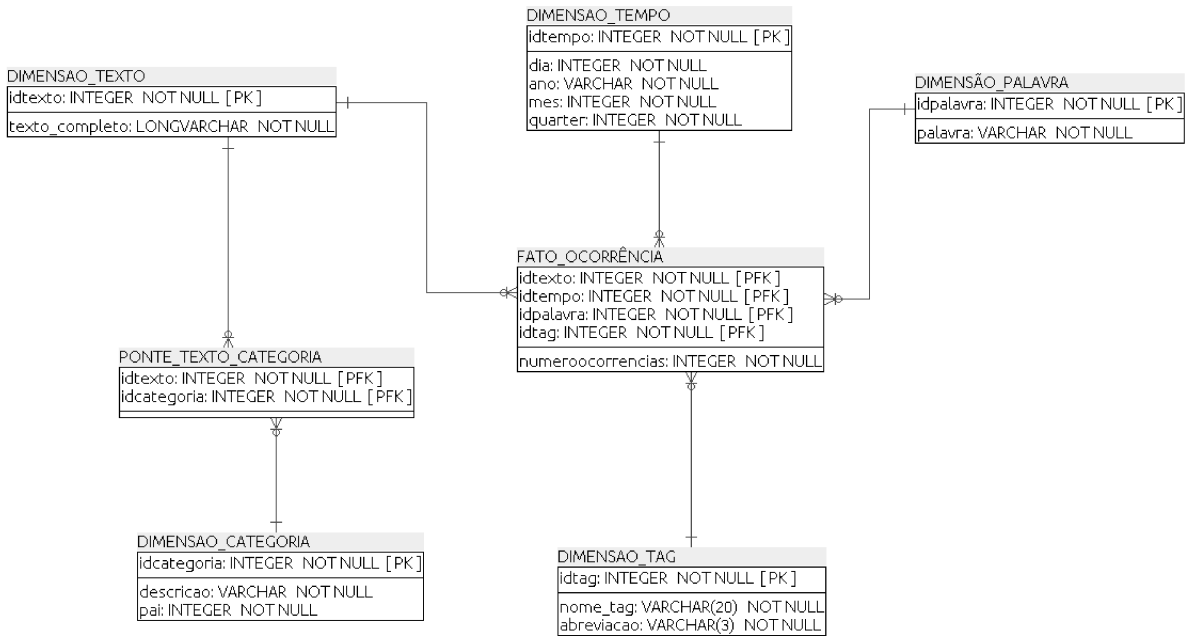



Figura 12: Modelo multidimensional do *Newsminer*

A tabela fato é denominada `FATO_OCORRENCIA` e as demais tabelas são as dimensões. A tabela `DIMENSAO_PALAVRA` é responsável por armazenar todos os termos. A `DIMENSAO_CATEGORIA`, por sua vez, armazena todas as categorias baseando-se na ontologia

*IPTC*. A tabela `PONTETEXTOCATEGORIA` foi acrescentada ao modelo para resolver o relacionamento muitos-para-muitos entre a tabela `DIMENSAO_TEXTO` e a tabela `DIMENSAO_CATEGORIA`. A métrica `numeroocorrencias` contida na tabela `FATO_OCORRENCIA` representa a quantidade de ocorrências de um termo, associado a um texto (que pode estar associado a uma ou mais categorias), em uma determinada data. Por último, para oferecer diferentes níveis de abstração, foram definidas hierarquias para as dimensões `DIMENSAO_TEXTO` ( `texto`< `idcategoria`) e `DIMENSAO_TEMPO` ( `data (dia)`< `mes`< `ano`).

O modelo proposto visa oferecer às aplicações uma nova forma de explorar um conjunto de textos, por isso, chamado de *corpus multidimensional*.

### 4.2.7 OLAP e consultas multidimensionais

O processamento das consultas é executado utilizando um servidor *ROLAP*. *Newsminer* utiliza o sistema gerenciador de banco de dados relacional *PostgreSQL*[13] Este servidor armazena os dados coletados e permite a exploração dos textos de notícias armazenados no *Newsminer*.

As consultas multidimensionais podem ser feitas em uma interface *Web* disponibilizada dentro do ambiente ou via programação, mediante uma API. Ambos os meios são oferecidos para consumo pelas aplicações e usuários.

A Figura 13 ilustra a interface da página inicial do *Newsminer*[14]. A interface visual das consultas do *Newsminer* foi desenvolvida utilizando HTML 5 em conjunto com o *framework Bootsrap*[15], tais tecnologias garantem a responsividade do sistema, ou seja, garante a adaptação da interface aos dispositivos que acessarem o sistema.

A partir da interface, os usuários podem navegar pelos dados, selecionar cubos de dados e analisar os dados em diferentes níveis de abstração. Para a realização das consultas, o ambiente serve-se das notícias armazenadas em seu banco de dados e permite que sejam obtidos dados da *Web* em tempo real via *Web Crawler*. Alguns tipos de consultas multidimensionais foram pré-definidos. As consultas podem ser parametrizadas com filtros de categoria e períodos de tempo. Os tipos de consultas são:

- *Ranking* de termos: retorna os termos que mais ocorrem em todo o banco ou em uma categoria em específico. Por exemplo, na categoria `Politics`, no intervalo de tempo entre janeiro de 2015 a dezembro de 2016, os dois termos de maior ocorrência foram `clinton` e `trump`, os sobrenomes dos candidatos à presidência dos Estados Unidos no período selecionado.

[13] *PostgreSQL home page*. Disponível em <http://www.postgresql.org>, acessado em 15/02/2017.
[14] Disponível em <http://lasid.sor.ufscar.br/newsminer/>, acessado em 12/05/2017.
[15] *Bootstrap home page*. Disponível em <http://www.getbootstrap.com>, acessado em 15/01/2017.


Figura 13: Página inicial do *Newsminer*

- Associação entre termos: dado um termo, retorna os termos que mais ocorreram em todo o banco de dados ou em uma categoria em específico. Por exemplo, na categoria `Economy, Business and Finance`, os três termos que estão mais associados a `dollar` são: `economy`, `market` e `investors`.

- Categorização de textos: dado um texto de entrada, esta consulta retorna a qual categoria este texto pertence. Por exemplo, dado o texto "*Brazilian President Michel Temer is struggling with a loss on the political and in the economy*", a consulta retornará que o texto pertence à categoria `Politics`.

- Busca textual: esta consulta tem como entrada um conjunto de termos e retorna quais os *Top K* textos que esses termos aparecem em conjunto.

- Cubo de dados: pensando em flexibilizar as consultas de usuários e aplicações, o *Newsminer* oferece uma consulta exploratória, que se baseia em todos os campos contemplados no modelo multidimensional. Com esta consulta, o usuário poderá escolher quais dados deseja recuperar, explorando a multidimensionalidade do modelo, aplicando filtros e gerando diversos cubos de dados, a partir dos dados armazenados na base do *Newsminer*.

A *API* permite que aplicações externas consumam os dados armazenados utilizando as mesmas consultas acima.

As próximas seções apresentam cada tipo de consulta com maiores detalhes.

### *Ranking* de termos

A consulta de *ranking* de termos ordena os termos de maior ocorrência, conforme mostra a Figura 14. A consulta pode ser configurada com os seguintes parâmetros:

- Período: define data de início e de fim para a ocorrência dos termos. Caso este filtro não seja selecionado será considerado todo o período armazenado.
- Categoria: ao selecionar este filtro, serão retornados apenas termos que ocorreram em textos da categoria selecionada. Caso este filtro não seja selecionado, serão considerados termos que ocorreram em todas as categorias.
- Número de resultados: determina a quantidade de termos que serão retornados.

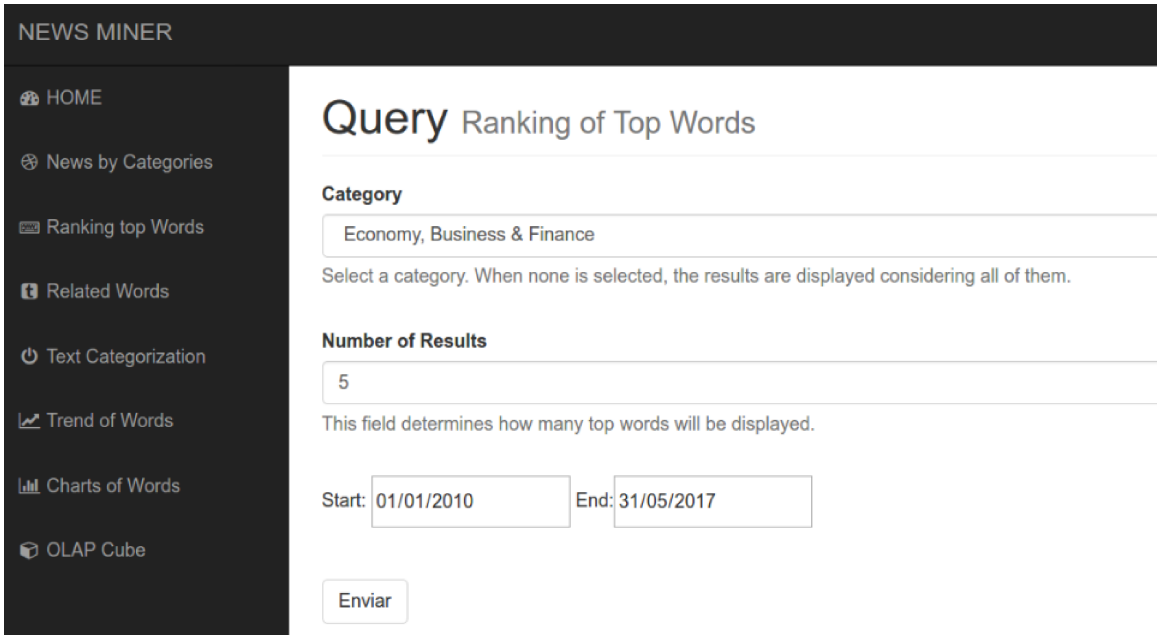



Figura 14: Tela da consulta de *ranking* de termos

A partir dos resultados obtidos pela consulta de *ranking* de termos, é possível se obter um gráfico com a tendência (*trend*) de termos. Este tem como objetivo fornecer uma visualização ao longo do tempo [16].

A Figura 15 ilustra o gráfico de *trend* de termos no *Newsminer*. Neste gráfico, o eixo `x` representa o período de tempo e o eixo `y`, o número de ocorrências de determinado termo.

[16] Projetos similares realizam *trend* de termos e palavras, em cenários de dados diferentes. Por exemplo, o *Google N-Gram*. Disponível em <https://books.google.com/ngrams/>, acessado em 20/12/2016, que utiliza base de dados histórica de livros. O *FiveThirrtEigth*, disponível em <https://projects.fivethirtyeight.com/reddit-ngram/>, acessado em 20/12/2016, que utiliza a base de dados históricos da rede social *Reddit*.

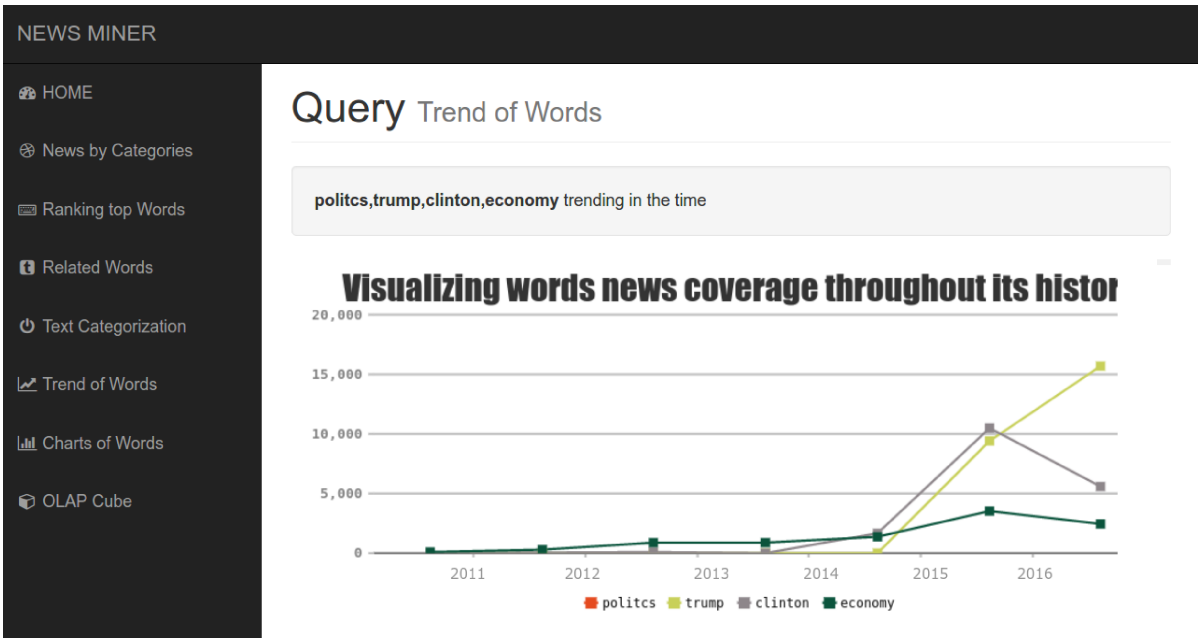



Figura 15: Tela de exibição do gráfico de *trend* de termos

## Associação entre termos

Explorando as tabelas `FATOOCORRENCIAS` e `PALAVRA` é possível determinar os *Top K* termos que ocorrem mais vezes associados a este conjunto de termos. A Figura 16 mostra a interface de realização desta consulta, que pode ser configurada de acordo com os seguintes parâmetros:

- Termo: este é um parâmetro obrigatório, trata-se do termo que será associado aos demais.
- Período: este filtro pode ser configurado com a data de início e de fim. Caso não seja selecionado, será considerado todo o período total.
- Categoria: ao selecionar este filtro, serão retornados apenas termos que ocorreram em textos da categoria selecionada. Caso este filtro não seja selecionado, será realizada a associação com termos de todas as categorias.
- Número de resultados: determina os $K$ termos que tem maior número ocorrência de acordo com todos os parâmetros.

Uma das maneiras de visualizar esta consulta é através de um grafo de associação entre termos [17].

[17] Esta consulta é similar a outros projetos que utilizam bases de dados em inglês para realizar consultas visuais de análise semântica, como o *GraphWords*, disponível em <http://graphwords.com/>, acessado em 15/12/2016, e o *VisuWords*, disponível em <http://visuwords.com/>, acessado em 15/12/2016. Ambos realizam associação de palavras baseados em um dicionário de palavras em inglês denominado *Thesaurus* (Disponível em <http://www.thesaurus.com/>, acessado em 15/12/2016).

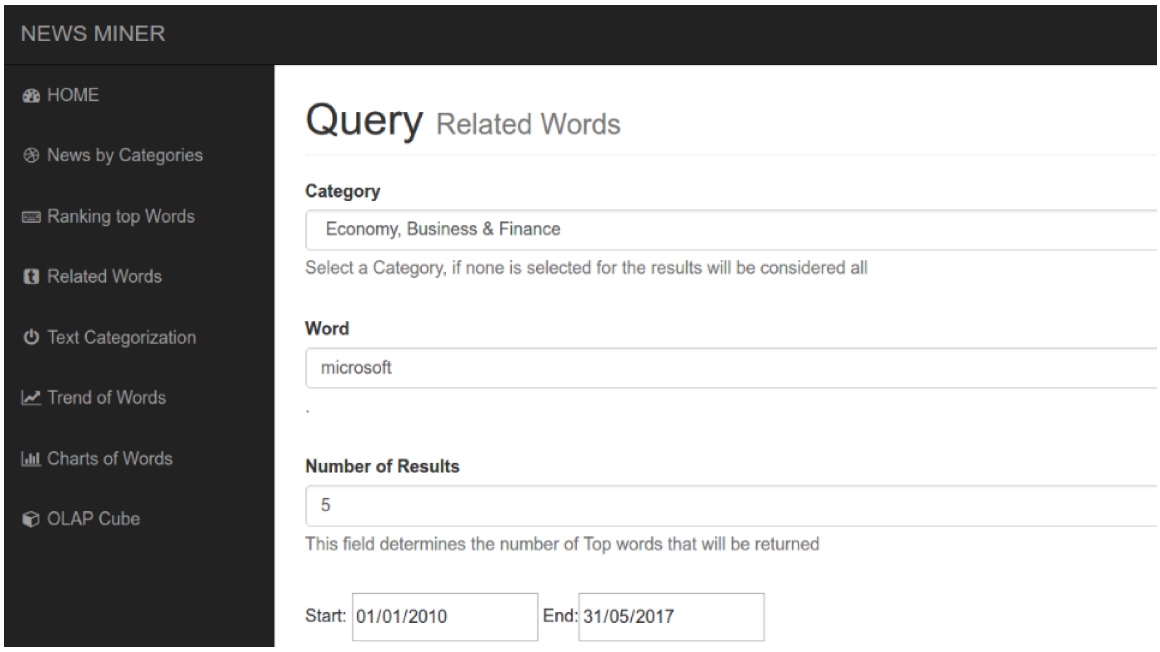



Figura 16: Tela de consulta de associação entre termos

Uma consulta, com os termos `hillary`, `trump`, `ecomony`, `politic` como entrada, retornou o grafo ilustrado pela Figura 17. No grafo de associação entre termos, cada nó representa um termo e o tamanho de um nó é proporcional ao número de ocorrências.Já as arestas representam a associação entre dois termos, e a largura de uma aresta está ligada ao número de documentos em que ocorreram associadas, ou seja, quanto mais espessa é uma aresta, mais vezes os termos ligados por ela ocorreram juntos.

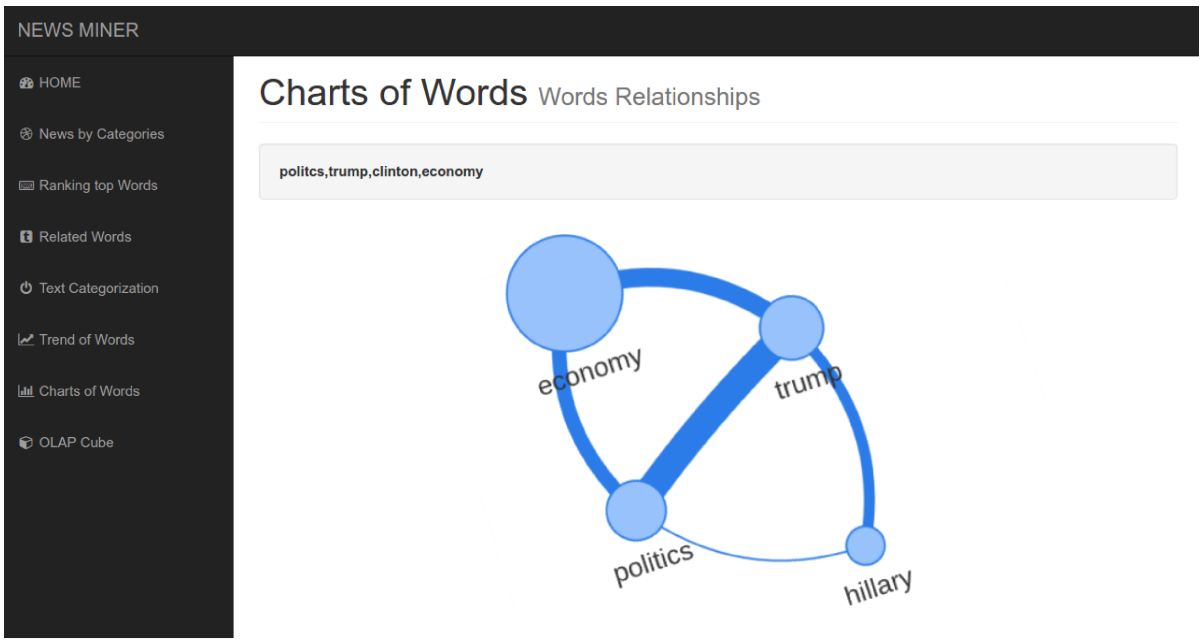



Figura 17: Tela de exibição do grafo de associação entre termos

### Categorização de textos

A categorização de texto do *Newsminer* retorna, dado um texto informado, qual sua categoria. Por exemplo, dado a frase *Neymar: Barcelona's Brazil star creates his dream player soccer*, a consulta retornará que esta frase é da categoria `Sport`. O consulta de categorização do *Newsminer*, baseado no exemplo, é ilustrada pela Figura 18.

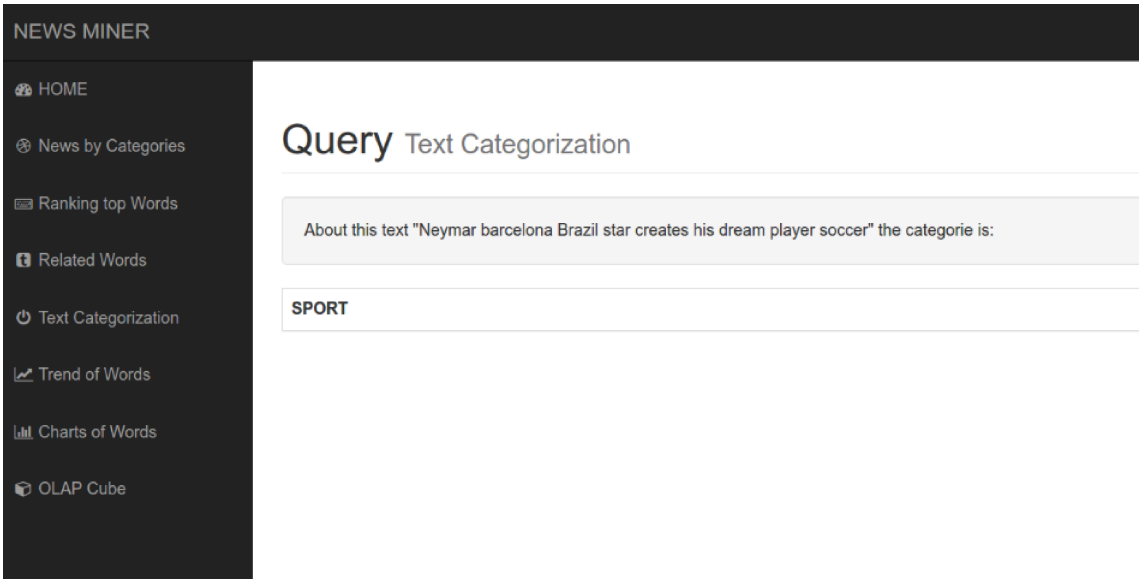



Figura 18: Tela da consulta de categorização de textos

### Exploração do cubo de dados

Dentro do *Newsminer*, o cubo de dados é uma consulta que permite explorar os dados armazenados no *Newsminer*, por todas as perspectivas e através de parâmetros fornecidos pelo usuário. A Figura 19 ilustra a interface dessa consulta.

Esse tipo de consulta visa flexibilizar as consultas. Observa-se que a interface fornece todos os campos contemplados no modelo multidimensional. Com esta consulta, o usuário poderá escolher quais dados deseja recuperar, explorando a multidimensionalidade do modelo, aplicando filtros e gerando diversos cubos de dados.

## 4.2.8 API de consultas e consumo

Tendo como finalidade a exploração dos dados armazenados no *Newsminer* por aplicações externas, foi desenvolvida uma *Application Programming Interface - API* (Interface de Programação de Aplicações). A utilização de *APIs* permite a integração e interoperabilidade entre aplicações com um baixo nível de acoplamento quando comparado ao acesso direto às bases de dados. Somando a estes fatores o fato de que, ao contrário dos bancos de dados que permitem apenas um acesso local, com o uso de *APIs* é possível compartilhar informações em escala global através da *Web*, tornando heterogêneo o consumo

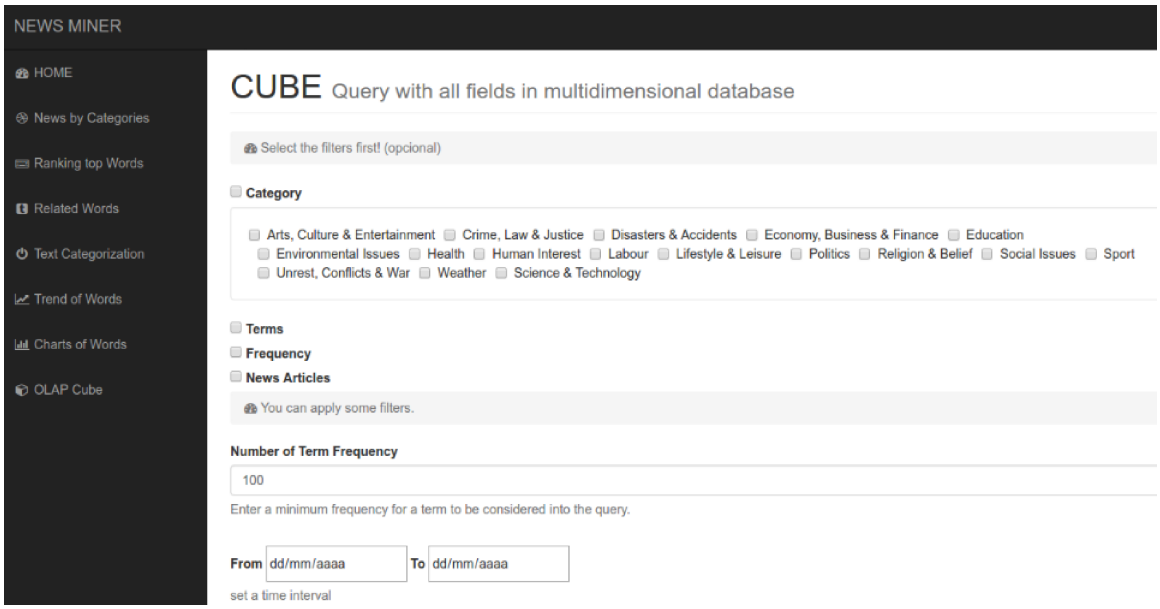



Figura 19: Tela da consulta exploratória do cubo de dados

dos dados e a forma de interação entre cliente e servidor (NGUYEN; NGUYEN; HUYNH, 2016).

A *API* do *Newsminer* está disponível para uso em <http://lasid.sor.ufscar.br/newsminer/api/>. Em específico, a *API* do *Newsminer* permite acesso aos dados armazenados e sua exploração através das consultas multidimensionais. Tal *API* foi desenvolvida utilizando o conceito de *REST - Representational State Transfer* (Transferência de Estado Representacional) (FIELDING, 2000), sendo assim permite com que o acesso aos dados seja realizado via protocolo *HTTP*. Através dos parâmetros fornecidos, esta API irá realizar consultas que retornarão os resultados em um documento *JSON*[18]

Pode-se acessar à *API* do *Newsminer*, por exemplo, utilizando a seguinte *URL* <http://lasid.sor.ufscar.br/newsminer/api/?r=categories>. Tal requisição significa um retorno com todas as categorias e seus respectivos códigos identificadores.

[ "category":"Religion & Belief","id":"12

, "category":"Social Issues","id":"14

, "category":"Sport","id":"15

]

Com a *URL* <http://lasid.sor.ufscar.br/newsminer/api/?r=topwords&c=15&l=3> executa uma consulta de exemplo. Esta consulta retorna as *Top 3* palavras da categoria de código igual a 15, que como visto no exemplo anterior, é *Sport*. A listagem abaixo ilustra o retorno da consulta para esta requisição.

[18] *JSON* - Documentação Oficial. Disponível em <http://www.json.org/json-pt.html>, acessado em Maio de 2017.

["word":"game","count":"2124

, "word":"time","count":"2021

, "word":"team","count":"1848

]

Anteriormente foi descrita a consulta de associação de palavras, tal consulta também pode ser realizada via *API* e explorada através de seus parâmetros. Um exemplo da realização da consulta de associação entre palavras são as 3 palavras mais associadas à palavra *soccer*, na categoria de *Sport*, no período de 05/05/2015 até 06/06/2016. Essa consulta pode ser realizada via *API*, com a seguinte *URL* <http://lasid.sor.ufscar.br/newsminer/api/?r=wordassoc&w=soccer&l=3&c=15&s=2015-05-05&e=2016-06-06>, cuja requisição retornará a seguinte listagem.

["word":"team","count":"17

, "word":"players","count":"16

, "word":"club","count":"14

]

Os cubos de dados podem ser gerados pela *API*. Para isso, devem ser fornecidos as dimensões a serem exploradas e as métricas utilizadas. Por exemplo, a *URL* <http://lasid.sor.ufscar.br/newsminer/api/?r=cube&d=text&d=category&l=3> retorna os textos armazenados e suas respectivas categorias, limitando-se a três resultados, como ilustra a listagem a seguir.

[ news:"enni karlsson and i are standing by...",category:"Sport

, news:"find all over the area. many farms use ...",category:"Politics

, news:"but in fact we get loads of people...",category:"Sport

]

Outra requisição que é oferecida é a ativação do *Web Crawler*. Ela pode ser utilizada caso, ao explorar o cubo de dados, a aplicação não encontre um número suficiente de resultados. Deste modo, a *Web* poderá ser explorada em busca de novos dados, que serão armazenados no *Newsminer*. Um exemplo da consulta anterior com a ativação do *Web Crawler* se dá pela seguinte *URL* <http://lasid.sor.ufscar.br/newsminer/api/r=cube&d=text&d=category&l=3&crawler=1>. Na nova listagem, pode-se notar que os dados de retorno são diferentes, isto porque os dados recém coletados já fazem parte da base de dados do *Newsminer*.

"cube": [ [news:"you could spend a few days here very easily...",category:9], [news:"support an art school, a cinema, a theatre ...",category:12], [news:"ishermen and farmers come and

go to there ...",category:3] ]

Sendo assim, esta *API* poderá beneficiar as aplicações que venham a consumir os dados do *Newsminer*, que estarão limpos e prontos para realização de consultas encima do *corpus* multidimensional. Os parâmetros para a utilização da *API*, conforme visto nos exemplos anteriores, são os seguintes:

- `r`: significa *request* (requisição). Está relacionada à maneira que os dados serão explorados. Os valores possíveis são: (*topwords*, *association*, *categorization*, *datacube*).
- `c`: significa *category* (categoria). Esse parâmetro permite que a aplicação selecione dados de uma ou mais categorias específicas.
- `s,e`: remetem-se à *start* e *end* (datas de início e fim). Deste modo, permitindo com que os resultados obtidos possam ser filtrados por períodos.
- `w`: quando for necessário retornar informações sobre determinado termo ou palavra (*word*), deverá ser fornecido via esse parâmetro.
- `d`: é um parâmetro específico para consultas que realizam o cubo de dados, significa dimensão(*dimension*). Cada vez que for informado, esse parâmetro indicará as dimensões utilizadas na consulta: *text, category,time,word.*
- `l`: significa *limit* (limite). Este parâmetro limita o número de resultados. Limite igual a 100 significa que serão retornados 100 textos, por exemplo.
- `crawler`: este parâmetro pode receber o valor igual a 1, isso indicará que o *Web Crawler* será ativado para responder a requisição.

Futuramente, outros parâmetros poderão ser acrescentados para proporcionar maior flexibilidade para as aplicações via *API*.

## 4.3 Sumário

Este capítulo descreve o ambiente *Newsminer* e sua arquitetura. As fontes de dados escolhidas para a implementação foram *sites* de notícias em inglês. Dentre os módulos da arquitetura se destacam, na *ETL*, o módulo de coleta *online* de notícias por meio de um *Web Crawler*, e o módulo de enriquecimento semântico que detecta novos elementos estruturais nos dados, como a descoberta da categoria das notícias usando técnicas de mineração de dados e a ontologia *IPTC*, e a anotação *POS Tagging* dos termos dos textos. Por último, a arquitetura contempla a disponibilização de consultas multidimensionais em uma interface *Web* e via *API* para consumo pelas aplicações e usuários. Com isto, oferece

a exploração dos dados e da multidimensionalidade do modelo, e a seleção flexível de parte ou de todos os dados em diferentes níveis de abstração.

Atualmente, o ambiente *Newsminer* disponibiliza notícias em 17 categorias, desde o ano de 2012 até o ano de 2017, sendo atualizado em tempo real.

# 5 AVALIAÇÃO EXPERIMENTAL

Este capítulo descreve e analisa os experimentos realizados e, na última seção, cenários de uso do *corpus* multidimensional. O intuito dos experimentos é responder as seguintes questões:

- Experimento 1: na etapa de Enriquecimento Semântico da *ETL*, procura-se achar *qual método de categorização de textos aliado a tipos de pré-processamento que melhor identifica a categoria de uma notícia*
- Experimento 2: na etapa de *ETL*, *é possível executar o processo desde a coleta das notícias até a carga no banco de dados em um tempo considerado próximo do tempo real?*

Para realização de todos os experimentos descritos neste capítulo foi utilizado um Servidor *IBM* Torre X3430 com processador *Intel Xeon X3430* (quatro núcleos - 2,4GHz) e 8GB de memória RAM.

## 5.1 Experimento 1: Avaliação dos métodos de categorização de textos

Este experimento tem como objetivo avaliar os métodos de categorização e de pré-processamento de textos. Os resultados deste experimento irão fundamentar a escolha do método de categorização de texto e de pré-processamento que serão incorporados à *ETL* (particularmente, ao módulo de enriquecimento semântico) da arquitetura do *Newsminer*.

A avaliação dos métodos de categorização de textos tem como finalidade a obtenção de um método com a melhor capacidade de generalização e predição para o conjunto de textos de notícias armazenado no *Newsminer*. Embasado nos resultados obtidos, será selecionado o método de categorização de textos que fará parte da arquitetura do *Newsminer*, tanto para o enriquecimento semântico na etapa de *ETL*, quanto na consulta de categorização de textos.

Os métodos de categorização são avaliados em combinação com os métodos de pré-processamento de textos. Assim, um método de pré-processamento de texto, também, será selecionado e aplicado em todos os textos de notícias antes de serem armazenados no banco de dados multidimensional. A obtenção de bons resultados com o método de pré-processamento implicará em dados de melhor qualidade para realização de consultas

multidimensionais e de melhores resultados para as aplicações que venham a consumir estes dados.

As próximas seções (da Seção 5.1.1 até a Seção 5.1.6) compõem a metodologia para execução deste experimento.

### 5.1.1 *Datasets*

Foram utilizados dois *datasets* durante este experimento: o *Newsminer Collection* (dados coletados pelo próprio *Newsminer*) e a base pública denominada *20Newsgroups*. Estes dois *datasets* são detalhados nas próximas subseções.

#### 5.1.1.1 *Newsminer Collection*

A base de dados do *Newsminer* contém textos de notícias distribuídos pelas 17 categorias definidas da ontologia *IPTC* (SMOOT, 2003). Um subconjunto desses dados foi selecionado para ser utilizado na experimentação e é chamado de *Newsminer Collection*. Os dados foram coletados no Laboratório de Sistemas Inteligentes e Distribuídos (LaSID) da Universidade Federal de São Carlos, campus Sorocaba.

Esta base de dados contém 7000 textos de notícias, do ano de 1990 até 2016, e estão igualmente distribuídos entre as categorias de *Art, Culture and Entertainment, Sport, Lifestyle and Leisure, Politics, Economy, Business and Finance,Environmental Issues* e *Science and Technology*. O número de notícias e termos por categoria é ilustrado na Tabela 6.

Tabela 6: Composição do *Newsminer Collection*

| **Categoria** | **Nº de Notícias** | **Nº de Termos** |
|---|---|---|
| *Art* | 1000 | 143333 |
| *Economy Business and Finance* | 1000 | 140279 |
| *Lifestyle and Leisure* | 1000 | 190026 |
| *Politics* | 1000 | 175966 |
| *Environmental Issues* | 1000 | 193645 |
| *Sport* | 1000 | 153708 |
| *Science and Technology* | 1000 | 142061 |

Os textos de notícias que compõem o *dataset* foram coletados via o *Web Crawler*, conforme descrito no Capítulo 4. O *Newsminer Collection* está publicamente disponível em <http://lasid.sor.ufscar.br/newsminer/datasets>.

#### 5.1.1.2 20Newsgroups

O *dataset* do *20Newsgroups (twenty news groups)*[1] é uma coleção de dados públicos, comumente utilizada em trabalhos científicos da área de classificação e agrupamento de textos. Por estes motivos, o *dataset* do *20NewsGroups* foi utilizado para validar os resultados obtidos pelo *Newsminer*.

O *20Newsgroups* contém, aproximadamente, 20.000 documentos, divididos uniformemente em 20 grupos de notícias, cada um correspondente a uma categoria diferente. Algumas das categorias estão próximas (por exemplo: comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), enquanto que outras são completamente distantes (por exemplo: rec.autos / soc.religion.christian). As categorias do *20NewsGroups* são mostradas na Tabela 7. É possível observar nesta tabela que as categorias que estão na mesma célula representam categorias próximas.

Tabela 7: Categorias do *dataset* do *20NewsGroups*

| | | |
|---|---|---|
| comp.graphics<br>comp.os.ms-windows.misc<br>comp.sys.ibm.pc.hardware<br>comp.sys.mac.hardware<br>comp.windows.x | rec.autos<br>rec.motorcycles<br>rec.sport.baseball<br>rec.sport.hockey | sci.crypt<br>sci.electronics<br>sci.med<br>sci.space |
| sci.cryptsci.electronicssci.medsci.space | talk.politics.misc<br>talk.politics.guns<br>talk.politics.mideast | talk.religion.misc<br>alt.atheism<br>soc.religion.christian |

### 5.1.2 Métodos de categorização de textos

Na arquitetura do *Newsminer*, a categorização de textos ocorre em dois momentos: na etapa de *ETL* e na realização das consultas. Este experimento tem como objetivo avaliar o desempenho dos métodos de classificação de textos nos *datasets* e, baseado neste estudo, fundamentar a utilização do melhor método, o qual será aplicado na arquitetura do *Newsminer*.

Os métodos avaliados são aqueles que segundo a literatura científica obtiveram um bom desempenho na categorização de textos. A Tabela 8 mostra quais foram os métodos utilizados neste experimento e sua respectiva abreviação, que será utilizada posteriormente na descrição dos resultados. A descrição dos métodos pode ser encontrada no Capítulo 2 (Seção 2.3).

[1] *20NewsGroups*. Disponível em <http://qwone.com/~jason/20NewsGroups/>, acessado 15/04/2017

Tabela 8: Métodos de categorização de textos avaliados

| **Método** | **Abreviação** |
|---|---|
| K-Vizinhos próximos | KNN |
| Máquinas de vetores de suporte | SVM |
| Naïve Bayes Bernoulli | NBB |
| Naïve Bayes Multinomial | NBM |
| *Passive Agressive* | PA |
| *Perceptron* | PER |
| Regressão Logística | RL |
| *Rocchio* | ROC |
| *Stochastic Gradient Descent* | SGD |

### 5.1.3 Métodos de pré-processamento de textos

A aplicação de métodos de pré-processamento na arquitetura do *Newsminer* tem como objetivo obter melhor desempenho das consultas multidimensionais e na categorização de textos de notícias. Baseando-se nos trabalhos da literatura científica, os seguintes métodos de pré-processamento de textos foram avaliados neste experimento:

- *Full Text* (FT) : texto completo, ou seja, é o texto de um *site* de notícias após a remoção de caracteres numéricos e das *tags HTML*.

- *Stopwords* (SW): este método de pré-processamento é realizado após o *Full Text* (FT), no qual são removidas as *stopwords*. Para realizar esta etapa, a lista de *stopwords* da biblioteca *NLTK* [2] é utilizada.

- *Noun* (NN): nesta etapa é aplicado o algoritmo de *POS Tagging*, que é responsável por reconhecer as classes morfológicas de cada termo. No procedimento de *POS Tagging* utilizado para esta avaliação, são considerados apenas os termos que são substantivos. Esta etapa, também, é executada com auxílio da biblioteca *NLTK*.

- *Keyphrases* (KP): este método de pré-processamento realiza a extração de *keyphrases*. Consiste em realizar a descoberta de termos compostos através da aplicação do algoritmo *RAKE* [3].

Para a realização dos experimentos foram aplicados os métodos de pré-processamento de texto nos dois *datasets*. Sendo assim, foram geradas quatro (4) bases de dados para cada *dataset*, cada uma com um determinado pré-processamento (*FT*, *SW*, *NN* e *KP*).

[2] *NLTK*. Disponível em <http://www.nltk.org/book/ch02.html>, acessado em 21/03/2016.
[3] *RAKE*. Disponível em <https://pypi.python.org/pypi/python-rake/1.0.5>, acessado em 21/03/2016

### 5.1.4 Ajustes de parâmetros

Esta etapa tem como objetivo obter os melhores parâmetros possíveis para a execução dos métodos de categorização de textos empregados nos experimentos para cada um dos *datasets Newsminer Collection* e *20NewsGroup*. Os valores dos parâmetros de cada método de classificação foram ajustados através da técnica de *grid search*, que testa valores de cada parâmetro dentro de uma faixa específica de busca, dado um intervalo. Para realização dos experimentos de categorização de texto, bem como para os ajustes de parâmetros, foi utilizada a biblioteca *Scikit-learn* (PEDREGOSA et al., 2011).

Os resultados obtidos pela técnica de *grid search* para o ajuste de parâmetros são ilustrados pela Tabela 9. A primeira coluna representa os métodos, a segunda os parâmetros testados, a terceira coluna representa o intervalo de valores utilizados no teste e as duas últimas o melhor parâmetro obtido para o *Newsminer Collection – NewsMC* e o *20NewsGroup – 20NewsG*, respectivamente.

Tabela 9: Resultados dos ajustes de parâmetros

| **Método** | **Parâmetro** | **Valores** | ***NewsMC*** | ***20NewsG*** |
|---|---|---|---|---|
| *SVM* | *C* | $10^7$ até $10^{-7}$, passo 1 na potência | 2.0 | 10.0 |
| *KNN* | *k* | $10^7$ até $10^{-7}$, passo 1 na potência | 10 | 2 |
| *NBM* | *alpha* | $10^7$ até $10^{-7}$, passo 1 na potência | 1 | 0.1 |
| *NBB* | *alpha* | $10^7$ até $10^{-7}$, passo 1 na potência | 1 | 1e-05 |
| *RL* | *penalty* | *L1* e *L2* | L2 | L2 |
| *ROC* | *alpha* | $10^7$ até $10^{-7}$, passo 1 na potência | 0.01 | 0.1 |
| *PA* | *alpha* | $10^7$ até $10^{-7}$, passo 1 na potência | 0.001 | 0.01 |
| *PER* | *alpha* | $10^7$ até $10^{-7}$, passo 1 na potência | 1e-07 | 1e-05 |
| *SGD* | *alpha* | $10^7$ até $10^{-7}$, passo 1 na potência | 0.01 | 0.1 |

### 5.1.5 Métricas de avaliação

As métricas de avaliação têm como finalidade avaliar a capacidade de generalização dos métodos de categorização e pré-processamento de textos. Foi utilizada a tradicional métrica de avaliação em cenários de classificação e categorização de texto: a *F-medida* (também utilizada na literatura como *F-score, F-measure, F1*).

A métrica da F-medida é a média harmônica das medidas *precisão* `P` (também chamado de valor preditivo positivo) e *revocação* `R` (sensibilidade) (XIONG; WU; CHEN, 2009). A fórmula do cálculo da F-medida é expressa por:

$$F = 2\frac{P * R}{P + R} \tag{5.1}$$

Para a avaliação dos métodos foi realizada a validação cruzada (MITTERMAYER; KNOLMAYER, 2006). A técnica de validação cruzada consiste em dividir a base de dados

em fatias, a cada fase de treino um fatia é removida. O treino então é realizado com as demais fatias e o teste realizado com a fatia que foi removida.

O processo de validação cruzada foi realizado em dez (10) execuções para cada classificador e método de pré-processamento. Ao final de cada execução, foi realizado um procedimento para que a ordem dos registros ficasse aleatória para a próxima execução. Após as dez execuções, foi computada a média da F-medida obtida pelas execuções e a mediana do tempo de execução.

### 5.1.6 Análise estatística

Em cenários de categorização de textos, as análises estatísticas são utilizadas para fundamentar as escolhas dos melhores resultados. Estas análises têm como papel importante, principalmente, a análise de dados que obtêm valores muito próximos. Tais validações têm como finalidade fundamentar se há diferença, estatisticamente significante entre determinados valores.

Para fins de validação estatística, os resultados obtidos pelos experimentos foram avaliados com o emprego do *Teste T* (SENN; RICHARDSON, 1994) com uma confiança de 95%. Neste caso, o *Teste T* foi realizado aos pares a fim de verificar qual é o melhor ou se há equivalência estatística entre os pares. A hipótese nula, neste caso, afirma que os métodos possuem desempenhos equivalentes.

### 5.1.7 Resultados e discussão

A Tabela 10 ilustra os resultados obtidos por método de classificação para cada base do *Newsminer Collection* após aplicação do pré-processamento. Cada linha indica o método de pré-processamento (base de dados) utilizado para o treinamento e teste e cada coluna corresponde a um método de classificação. Os valores representam a média da F-medida das execuções.

Tabela 10: Resultados dos classificadores para o *Newsminer Collection*. Os valores da F-medida destacados em negrito correspondem aos melhores resultados após a avaliação estatística.

| | NBB | NBM | SVM | KNN | RL | PA | Rocchio | SGD | Perceptron |
|---|---|---|---|---|---|---|---|---|---|
| FT | 0,89 | 0,89 | **0,92** | 0,85 | 0,88 | 0,82 | 0,85 | 0,85 | **0,91** |
| SW | 0,92 | 0,91 | **0,93** | 0,85 | 0,89 | 0,87 | 0,86 | 0,88 | **0,92** |
| NN | 0,90 | 0,91 | **0,92** | 0,84 | 0,89 | 0,86 | 0,85 | 0,87 | **0,91** |
| KP | 0,88 | 0,90 | **0,92** | 0,86 | 0,88 | 0,85 | 0,87 | 0,86 | **0,91** |

Os resultados obtidos evidenciaram o bom desempenho de alguns dos métodos avaliados. O desvio padrão entre os resultados obtidos usando os diferentes métodos foi baixo ($< 0,02$), para todos os resultados. O método que apresentou melhor resultado foi

o SVM. Testes estatísticos (teste de Friedman (DEMsAR, 2006) com confiança de 95%) mostrou que apenas o *Perceptron* obteve resultados estatisticamente equivalentes ao SVM.

Os resultados dos métodos de classificação para o *20NewsGroup* estão mostrados pela Tabela 11. Neste caso, o método de classificação se mostrou ter o melhor desempenho foi o *Perceptron*. Segundo testes estatísticos, nenhum outro método de classificação teve desempenho equivalente ou superior ao *Perceptron*.

Tabela 11: Resultados dos classificadores no *dataset* do *20NewsGroup*. Os valores da F-medida destacados em negrito correspondem aos melhores resultados após a avaliação estatística.

| | NBB | NBM | SVM | KNN | RL | PA | Rocchio | SGD | Per |
|---|---|---|---|---|---|---|---|---|---|
| FT | 0,768 | 0,731 | 0,789 | 0,635 | 0,767 | 0,712 | 0,750 | 0,774 | **0,829** |
| SW | 0,763 | 0,738 | 0,791 | 0,632 | 0,770 | 0,759 | 0,794 | 0,798 | **0,845** |
| NN | 0,756 | 0,740 | 0,786 | 0,627 | 0,767 | 0,748 | 0,767 | 0,771 | **0,810** |
| KP | 0,765 | 0,728 | 0,796 | 0,654 | 0,770 | 0,742 | 0,767 | 0,768 | **0,837** |

Pela Tabela 10, é possível observar que o pré-processamento SW foi o que obteve melhor resultado médio. Com o intuito de avaliar se SW é o melhor método de pré-processamento que oferece os melhores resultados para os melhores métodos de classificação SVM e Perceptron, a Tabela 12 mostra os resultados obtidos pelas 10 execuções para o *Newsminer Collection*.

Tabela 12: Resultados obtidos das 10 execuções pelos métodos Perceptron e SVM para o *Newsminer Collection*.

| Perceptron | | | | SVM | | | |
|---|---|---|---|---|---|---|---|
| FT | SW | NN | KP | FT | SW | NN | KP |
| 0,92 | 0,92 | 0,91 | 0,91 | 0,92 | 0,93 | 0,92 | 0,92 |
| 0,92 | 0,92 | 0,91 | 0,91 | 0,92 | 0,93 | 0,92 | 0,92 |
| 0,91 | 0,92 | 0,91 | 0,91 | 0,92 | 0,93 | 0,92 | 0,92 |
| 0,91 | 0,93 | 0,91 | 0,91 | 0,92 | 0,93 | 0,92 | 0,92 |
| 0,91 | 0,92 | 0,91 | 0,91 | 0,92 | 0,93 | 0,92 | 0,92 |
| 0,91 | 0,92 | 0,91 | 0,91 | 0,92 | 0,93 | 0,92 | 0,92 |
| 0,91 | 0,92 | 0,91 | 0,91 | 0,92 | 0,93 | 0,92 | 0,92 |
| 0,92 | 0,92 | 0,91 | 0,91 | 0,93 | 0,93 | 0,92 | 0,92 |
| 0,91 | 0,92 | 0,91 | 0,91 | 0,93 | 0,93 | 0,92 | 0,92 |
| 0,91 | 0,92 | 0,91 | 0,91 | 0,92 | 0,93 | 0,92 | 0,92 |

Para concluir sobre a superioridade dos resultados obtidos pelo pré-processamento SW, foi realizado o teste estatístico *Test T* pareado, com 95% de grau de confiança entre o resultado do pré-processamento SW e cada um dos outros resultados do método de classificação. O resultado da análise estatística sugere que o pré-processamento SW é superior que os outros métodos de pré-processamento tanto para o método Perceptron quanto para o método SVM.

### 5.1.8 Configuração final do *Newsminer*

Os resultados do Experimento 1 deram subsídios para a definição do método de classificação e do método de pré-processamento para compor a arquitetura do *Newsminer*.

O método de classificação selecionado é executado como parte do enriquecimento semântico na etapa de *ETL*, bem como para determinar as categorias das notícias nas consultas. Na arquitetura do *Newsminer*, o método de classificação de textos será responsável por categorizar notícias coletadas em tempo real em uma base de dados que, até março de 2017, já armazena mais de 200.000 mil notícias. O método de pré-processamento é aplicado, na *ETL*, aos textos de notícias que provêm das fontes e aos textos inseridos como parte das consultas.

Com relação ao pré-processamento, foi selecionado o método de remoção de *stopwords*. Tal método, além de reduzir a dimensionalidade dos atributos originais em 50%, também foi o método que obteve os melhores resultados neste primeiro experimento.

Quanto à categorização, os métodos *Perceptron* e *SVM* mostraram-se superiores estatisticamente aos outros sete métodos e equivalentes entre si. Sendo assim, para escolher um classificador para a arquitetura do *Newsminer*, foram consideradas outras duas métricas: tempo de execução e capacidade de aprendizado *online*.

A mediana dos tempos de execução para cada método é mostrada na Tabela 13. Os valores do tempo de processamento consistem do tempo de execução da fase de treinamento do classificador somado ao tempo de execução da fase de testes, para o *Newsminer Collection*, considerando o pré-processamento com remoção de *stopwords*.

Tabela 13: Mediana do tempo de execução total dos métodos de categorização de textos

| **Método** | **Tempo (minutos:segundos)** |
|---|---|
| PER-SW | 13:56 |
| SVM-SW | 20:38 |

Portanto, o método escolhido para compor o *Newsminer* foi o *Perceptron*, que obteve um bom tempo de execução. O *Perceptron* tem, também, como característica o aprendizado *online*, atualizando o modelo de predição de maneira incremental. Esta propriedade é de grande importância para categorizar notícias em tempo real. Uma vez que o conjunto todo não vai ser processado a cada execução, é possível afirmar que o tempo da execução incremental vai ser menor que 13:56 minutos.

## 5.2 Experimento 2: Avaliação da *ETL* em tempo real

A definição de Garcia, Tanaka e Baião (2015) de tempo real é a menor latência possível entre a fonte de dados e a apresentação dos dados em um *Data Warehouse*. De

acordo com Vaisman e Zimányi (2012), o processo de aquisição de dados introduz a maior latência. Por isso, este experimento tem como objetivo verificar se é possível executar o processo de *ETL*, pertencente à arquitetura do *Newsminer*, em tempo real (ou próximo do tempo real). O intuito é mostrar que as aplicações conseguem ativar o processo de *ETL* (via *API*) para coletar novas notícias e, assim, obter um conjunto de textos atuais e atualizados. Para este fim, foram realizados testes de coleta de textos dos *sites* de notícias via *Web Crawler*, pré-processamento dos textos coletados e carga no banco de dados multidimensional.

Para este experimento foram selecionadas as 7.000 *URLs* das notícias utilizadas no Experimento 1 (conjunto de textos do *Newsminer Collection*). Inicialmente, foram definidas três métricas: (1) tempo de execução do *Web Crawler*, (2) tempo de pré-processamento das notícias e (3) tempo de contagem de ocorrências e armazenamento/carga de dados no banco de dados.

O tempo de execução do *Web Crawler* corresponde ao tempo de extração. Porém, há diversos fatores que impactam no tempo de extração, como velocidade da banda larga do computador que está coletando, velocidade da banda larga do servidor que hospeda o *site*, sobrecarga no tráfego da rede, dentre outros. Por causa destes fatores, para complementar o tempo de extração, foram utilizadas outras métricas: a métrica de tamanho da página coletada, que dada uma *URL* indica o tamanho do arquivo *HTML* em *bytes*, e a métrica de palavras, que indica a quantidade de palavras de um texto coletado.

Para o tempo de pré-processamento da *ETL* foi escolhido o tempo de limpeza e extração das *stopwords*. Conforme selecionado pelo Experimento 1, o método de pré-processamento a ser aplicado nas notícias armazenadas no *Newsminer* foi a extração de *stopwords*. Para avaliar este método, foi considerado o tempo de execução do momento em que a notícia foi coletada até serem extraídas todas as *stopwords*. Após extrair as *stopwords*, é realizado o procedimento que contabiliza o número de ocorrências (OC) de cada termo por notícia. Assim, é coletado o tempo de contagem de ocorrências. Adicionalmente, a carga/atualização dos dados verifica se o termo já esta armazenado no banco de dados, se não existir, o insere. Posteriormente, a ocorrência deste termo, já contabilizada, é armazenada na tabela `FATO_OCORRENCIA`. O tempo desse último passo é atribuído ao tempo de armazenamento/carga de dados no banco de dados.

Para cada *URL*, o procedimento foi executado dez (10) vezes. Quanto ao tempo de execução, foi considerada a mediana do tempo de todas as execuções. A Tabela 14 ilustra os resultados obtidos durante o processo de *ETL*. A primeira coluna indica o número de notícias coletadas (Nº) e as demais são: tamanho da página (TP), número de palavras (NP), tempo de coleta do *Web Crawler* (WC), tempo de extração de *stopwords* (SW) e carga das ocorrências no banco de dados (OC).

Nota-se que foram coletados os tempos para diferentes conjuntos de notícias, cujo

Tabela 14: Resultados da Extração, Transformação e Carga do *Newsminer*. Unidade de medida formatada em 00:00:0000, representando horas:minutos:milissegundos

| **Nº** | **TP** | **NP** | **WC** | **SW** | **OC** | **Total** |
|---|---|---|---|---|---|---|
| 100 | 418004 | 97 | 00:00:01926 | 00:00:00092 | 00:00:00110 | 00:00:02128 |
| 200 | 423960 | 124 | 00:00:01953 | 00:00:00115 | 00:00:00103 | 00:00:02170 |
| 300 | 426048 | 136 | 00:00:02007 | 00:00:00127 | 00:00:00111 | 00:00:02245 |
| 400 | 425873 | 120 | 00:00:01984 | 00:00:00111 | 00:00:00111 | 00:00:02206 |
| 500 | 425194 | 124 | 00:00:02015 | 00:00:00129 | 00:00:00111 | 00:00:02255 |
| 600 | 414490 | 258 | 00:00:01964 | 00:00:00126 | 00:00:00113 | 00:00:02203 |
| 700 | 413692 | 276 | 00:00:01961 | 00:00:00111 | 00:00:00111 | 00:00:02184 |
| 800 | 415040 | 259 | 00:00:01954 | 00:00:00115 | 00:00:00110 | 00:00:02179 |
| 900 | 415786 | 249 | 00:00:01954 | 00:00:00112 | 00:00:00110 | 00:00:02176 |
| 1000 | 418513 | 181 | 00:00:02006 | 00:00:00113 | 00:00:00113 | 00:00:02231 |
| 2000 | 418234 | 190 | 00:00:01986 | 00:00:00113 | 00:00:00110 | 00:00:02209 |
| 3000 | 417957 | 189 | 00:00:01976 | 00:00:00098 | 00:00:00110 | 00:00:02184 |
| 4000 | 414725 | 185 | 00:00:01965 | 00:00:00118 | 00:00:00116 | 00:00:02199 |
| 5000 | 399373 | 277 | 00:00:01967 | 00:00:00106 | 00:00:00106 | 00:00:02179 |
| 6000 | 378873 | 338 | 00:00:01917 | 00:00:00107 | 00:00:00108 | 00:00:02131 |
| 7000 | 346929 | 371 | 00:00:01810 | 00:00:00111 | 00:00:00109 | 00:00:02030 |

tamanho foi aumentando de maneira gradativa, primeiro em intervalos de 100 e depois em intervalos de 1000 notícias. O tempo médio de coleta por notícia armazenada e pré-processada do *Newsminer*, independente do volume coletado, é de 2 segundos (que pode ser observado na tabela).

O número médio de notícias que um jornal americano publica [4] por dia é 500. Segundo Vaisman e Zimányi (2012), pode-se dizer que a etapa de *ETL* do *Newsminer* está preparada para suportar tempo real para alguns tipos de aplicações[5], considerando o processamento de 500 notícias e o número de fontes que o ambiente trabalha atualmente (sete). O tempo geral é de 117 minutos (3500 notícias em quase 2 horas ou 17 minutos a cada 500 notícias). No entanto, é possível das aplicações coletarem notícias diariamente em tempo próximo do tempo real, e, assim, obterem um conjunto de textos atuais e atualizados. Para melhorar esse tempo, podem-se aplicar técnicas de processamento distribuído de dados ou de paralelismo.

Um fator analisado foi o quanto o aumento gradativo do número de notícias implica no tempo de execução da *ETL*. Para ilustrar a proporcionalidade entre as métricas em decorrência da escala do número de notícias coletadas, o gráfico da Figura 20 ilustra os valores da Tabela 14, após realizada a normalização *min-max* em uma escala de 0 até 1[6].

[4] *"How Many Stories Do Newspapers Publish Per Day?"*. Disponível em <https://www.theatlantic.com/technology/archive/2016/05/how-many-stories-do-newspapers-publish-per-day/483845/>, acessado em 15/06/2016.

[5] Por exemplo, não é o caso para aplicações de detecção de fraudes, pois a precisão deve ser na ordem de minutos.

[6] Foram considerados como valores mínimo e máximo de intervalo para a normalização, os valores

O eixo `x` do gráfico indica o número de notícias coletadas (de 100 até 7000) e o eixo `y` os valores das métricas em relação ao número de notícias coletadas.

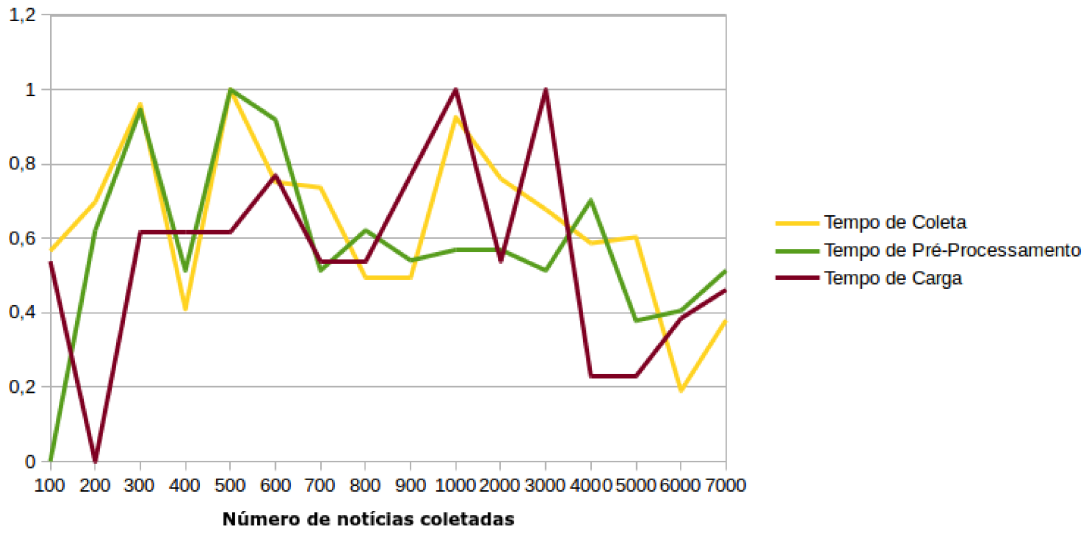



Figura 20: Tempo de execução da *ETL*

O gráfico mostra que, mesmo o número de notícias aumentando gradativamente, não há alteração significativa das métricas conforme o número de notícias coletada. Isto indica que, o aumento do número de notícias não afeta as médias de tempo de execução. Logo, a etapa de *ETL* do *Newsminer* é estável quando se trata de escalabilidade de notícias coletadas. Esta afirmação é válida quando considerado o número máximo de notícias que foram coletadas para o experimento. Deste modo, poderá ser realizada uma carga de até 7000 notícias sem afetar o tempo médio de processamento da *ETL*.

## 5.3 Cenários de uso: consultas multidimensionais sobre o *corpus*

Esta seção tem como objetivo ilustrar alguns tipos de consultas que podem ser explorados a partir do modelo do *Newsminer*. Para a execução das consultas, foi gerada uma base com a versão do *Newsminer* de 10 de fevereiro de 2017 [7]. Esta versão contém 60.863 textos de notícias, separados em todas a 17 categorias da ontologia *IPTC*, em um total de 1.476.956 de termos do ano de 1900 até 2017. Para cada consulta, foram coletados os tempos de dez execuções. A mediana do tempo de execução foi mostrada junto aos resultados das consultas.

mínimo e máximo dos tempos coletados. Por isso, os tempos (de coleta, de pré-processamento e de carga) normalizados para 100 textos é 0, que correspondem aos valores mínimos de cada tempo.

[7] *Newsminer*. Disponível em <http://lasid.sor.ufscar.br/newsminer/>, acessado em 12/05/2017.

### 5.3.1 Consulta de *ranking* de termos

A primeira consulta a ser explorada é a consulta de *ranking* de termos. Para este cenário, serão consideradas todas as categorias e todo o período, ou seja, os termos que mais ocorreram em todas as notícias armazenadas. Neste experimento, foram considerados os *Top 5* termos de maior ocorrência. O resultado desta consulta é mostrado pela Tabela 15.

Tabela 15: Resultados da consulta de *ranking* de termos no *Newsminer*

| Termo | Nº de ocorrências |
|---|---|
| `world` | 4114 |
| `new` | 3692 |
| `just` | 3513 |
| `like` | 3299 |
| `work` | 3200 |
| **Mediana do tempo**: 7.6 segundos | |

### 5.3.2 Consulta de associação de termos

Baseando-se nos resultados dos *Top 5* termos que mais ocorreram, foi realizada a consulta de associação de termos. Novamente, foram desconsiderados os filtros de tempo e de categoria. Logo, para cada termo consultado, serão retornados os *Top 5* termos que mais ocorreram associados à ele, em todo o período e em todas as categorias. A Tabela 16 mostra os resultados obtidos para a consulta de associação de termos. Para cada termo consultado, apresentam-se cinco pares: termo associado, número de ocorrências em conjunto. Os resultados, para cada termo consultado, foram ordenados de maneira decrescente do número de ocorrências.

Tabela 16: Resultados da consulta de associação entre termos do *Newsminer*

| `world` | | `new` | | `just` | | `like` | | `work` | |
|---|---|---|---|---|---|---|---|---|---|
| `new` | 2981 | `just` | 3807 | `new` | 3807 | `just` | 3782 | `new` | 2358 |
| `just` | 2786 | `like` | 3756 | `like` | 3782 | `new` | 3756 | `like` | 2247 |
| `like` | 2781 | `world` | 2981 | `way` | 2827 | `way` | 2880 | `just` | 2218 |
| `way` | 2200 | `way` | 2889 | `world` | 2786 | `world` | 2781 | `world` | 2151 |
| `work` | 2151 | `work` | 2358 | `going` | 2392 | `think` | 2356 | `way` | 1858 |
| **Mediana do tempo**: 7.6 segundos | | | | | | | | | |

Ao realizar a consulta de associação entre termos, observa-se que os *Top 5* termos de maior ocorrência mostrados pela Tabela 15 são frequentemente associados entre si. Por exemplo, os termos `world` e `new` ocorrem em 4114 e 3692 textos, respetivamente. E, ocorrem juntos em 2981 textos de notícias.

Tendo como objetivo explorar as dimensões do modelo multidimensional, as próximas subseções descrevem as consultas com aplicação de filtros de tempo e categoria.

### 5.3.3 Explorando a dimensão Tempo

Em ambientes de *Data Warehouse*, a dimensão temporal exerce um papel fundamental. Além de controlar as cargas de dados por período, de maneira incremental, também é possível explorar o tempo como fator determinante na análise dos dados. Para fins de explorar o aspecto temporal, a consulta de *ranking* de termo foi repetida aplicando o filtro de período. Para cada execução da consulta, foram selecionados os anos de 2010 a 2016.

Os resultados obtidos são ilustrados na Tabela 17. O cabeçalho da tabela é composto pela posição (POS) em que o termo mais ocorre e o ano de referência da ocorrência. As linhas apresentam o *ranking* de termos de cada ano, na ordem decrescente do número de ocorrências.

Tabela 17: Resultados da consulta de *ranking* de termos no *Newsminer* explorando a dimensão `Tempo`

| POS | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 |
|---|---|---|---|---|---|---|---|
| 1º | `apple` | `work` | `market` | `good` | `like` | `world` | `world` |
| 2º | `world` | `world` | `new` | `company` | `new` | `world` | `said` |
| 3º | `ipad` | `new` | `big` | `new` | `company` | `new` | `just` |
| 4º | `iphone` | `company` | `company` | `investors` | `world` | `just` | `word` |
| 5º | `company` | `money` | `investors` | `just` | `just` | `like` | `related` |
| **Mediana do tempo**: 2.8 segundos | | | | | | | |

Os resultados obtidos ao realizar-se a consulta de *ranking* de termos explorando a dimensão de `Tempo` mostram que há variação a cada ano e, também, são diferentes dos resultados obtidos quando realizada a consulta de *ranking* de palavras sem a aplicação de filtros (Tabela 15). O ano de 2010 é o que chama atenção no que se refere aos termos mais recorrentes, pois ao contrário dos demais anos, os termos `apple`, `ipad` e `iphone` aparecem entre os mais recorrentes para este ano. Tais ocorrências refletem o grande número de notícias desse ano relacionadas à *Steve Jobs* e os lançamentos tecnológicos de sua empresa, *Apple* [8].

### 5.3.4 Explorando a dimensão Categoria

Cada texto está associado a uma categoria. A exploração desta dimensão se torna interessante para se obter resultados específicos, onde conteúdos são direcionados a determinadas categorias. Podem-se explorar termos que ocorrem em uma categoria, várias, ou em todas as categorias.

Para avaliar o comportamento do modelo multidimensional na dimensão de categoria, na realização da consulta de *ranking* de termos, foram consideradas as categorias de

[8] *Apple unveils iPhone*. Disponível em <money.cnn.com/2010/06/07/technology/newappleiphone>, acessado em 15/03/2017

`Arts, Culture and Entertainment` e `Sport`. Foram obtidos os *Top 5* termos de maior ocorrência para cada categoria e em conjunto. A Tabela 18 mostra os resultados obtidos por este experimento, onde a coluna *Entertainment* representa os *Top 5* termos de maior ocorrência na categoria *Arts, Culture and Entertainment*, a coluna *Sport* representam os *Top 5* termos na categoria `Sport` e a última coluna, representa os resultados obtidos para as categorias em conjunto.

Tabela 18: Resultados da consulta de *ranking* de termos no *Newsminer* explorando a dimensão `Categoria`

| Pos | `Entertainment` | `Sport` | `Entertainment` e `Sport` |
|---|---|---|---|
| 1 | `art` | `team` | `world` |
| 2 | `just` | `game` | `work` |
| 3 | `life` | `win` | `related` |
| 4 | `london` | `season` | `like` |
| 5 | `artist` | `play` | `just` |
| **Mediana do tempo**: 3.2 segundos | | | |

Os resultados mostram que, para a categoria de `Sport`, tem-se termos relacionados a competições esportivas. Já, para a categoria de `Entertainment`, os termos estão no contexto de arte e entretenimento. Complementarmente, ao realizar a consulta considerando termos que ocorreram nas duas categorias, se nota um resultado similar à consulta mostrada pela Tabela 15, quando não foi aplicado nenhum filtro, pois os termos resultantes são genéricos.

Ainda explorando as categorias, com base nos *Top 5* termos de maior ocorrência para a categoria de `Sport`, foi realizada a consulta de associação entre termos que é mostrada na Tabela 19. Para cada termo consultado, apresentam-se cinco pares: termo associado, número de ocorrências em conjunto. Os resultados, para cada termo consultado, foram ordenados de maneira decrescente do número de ocorrências.

Tabela 19: Resultados da consulta de associação entre termos no *Newsminer* explorando a dimensão `Categoria`

| `team` | | `game` | | `win` | | `season` | | `play` | |
|---|---|---|---|---|---|---|---|---|---|
| `game` | 570 | `team` | 570 | `game` | 558 | `team` | 562 | `game` | 544 |
| `season` | 562 | `season` | 562 | `team` | 534 | `game` | 562 | `team` | 474 |
| `win` | 534 | `win` | 558 | `season` | 517 | `win` | 517 | `season` | 541 |
| `player` | 474 | `play` | 544 | `play` | 461 | `players` | 477 | `win` | 461 |
| `players` | 471 | `players` | 531 | `players` | 432 | `play` | 451 | `players` | 443 |
| **Mediana do tempo**: 2.3 segundos | | | | | | | | | |

Assim como aconteceu com os resultados da Tabela 16, observa-se uma forte correlação entre os termos de maior ocorrência. Isto significa que estes termos, além de aparecerem na maioria dos documentos, também aparecem em conjunto. Um exemplo desta

correlação é vista com os três termos `season`, `game` e `team`, que aparecem em conjunto em 562 textos de notícias.

### 5.3.5 Explorando as dimensões `Tempo` e `Categoria`

A multidimensionalidade é um fator característico e determinante de ambientes de *Data Warehouse*. Através desta característica é possível explorar e analisar um modelo multidimensional por múltiplas perspectivas. Até o momento foram exploradas as dimensões `Tempo` e `Categoria` por separado. Neste experimento, são exploradas estas duas dimensões em conjunto.

Para a realização deste experimento foram considerados todos o textos da categoria de `Politics`. Esta categoria foi selecionada como exemplo dada a sua relevância em debates atuais e pelo fato de ser uma categoria dinâmica no que se refere aos conteúdos relacionados à ela. O termo com maior número de ocorrências (1280) da categoria de `Politics`, dos anos de 2012 até 2016, foi `president`. Utilizando esse termo, a consulta de associação de termos foi executada para explorar o fator temporal. Os resultados obtidos são ilustrados pela Tabela 20.

Tabela 20: Resultados da consulta de associação entre termos para a categoria de `Politics`, explorando as dimensões `Tempo` e `Categoria`

| *Top 5* termos relacionados à `president` | | | | | |
|---|---|---|---|---|---|
| **Pos** | **2012** | **2013** | **2014** | **2015** | **2016** |
| 1º | `obama` | obama | obama | state | `trump` |
| 2º | `romney` | barack | country | campaign | campaign |
| 3º | growing | government | house | republican | republican |
| 4º | iraq | `obama` | republican | `obama` | state |
| 5º | war | congress | state | `clinton` | `clinton` |
| **Mediana do tempo**: 3.6 segundos | | | | | |

Pode-se observar que há uma variação de termos de maior ocorrência relacionados ao termo da pesquisa (`president`). Por exemplo, no ano de 2012, um termo que ocorreu foi `romney`. Ao se analisar as notícias relacionadas a este termo, nota-se que são relacionadas à eleição de 2012, quando empresário Willard Mitt Romney foi o candidato republicano à presidência dos Estados Unidos. Também, nota-se que, de 2013 até 2015, aparecem os termos `obama` e `barack`, sobrenome e nome do presidente em exercício *Barack Obama* nesse período. No ano de 2015, o termo `clinton` começa a aparecer, e, em 2016, o termo que aparece mais associado a `president` é `trump`. Ambos os termos correspondem aos sobrenomes da candidata derrotada nas eleições de 2016, *Hillary Clinton* e do presidente eleito, *Donald Trump*.

## 5.4 Sumário

Este capítulo apresentou duas avaliações experimentais. Para responder as questões formuladas no início do capítulo, tem-se as seguintes considerações.

- Experimento 1: conclui-se que vale a pena utilizar um método de categorização de notícias para enriquecer semanticamente o conjunto de textos, pois o melhor método de categorização de notícias teve resultados superiores a 90%. O método escolhido foi o *Perceptron*, aliado ao método de pré-processamento de remoção de *stopwords*. Portanto, esses métodos foram incorporados à *ETL* do ambiente *Newsminer*.

- Experimento 2: conclui-se que até o número de 7000 notícias, o tempo do processo de *ETL* acontece em quase quatro horas. Observa-se aqui que ao incluir o tempo de processamento da categorização de notícias, o tempo da *ETL* é de 246 minutos (4 horas e 10 minutos). Porém, como o classificador *Perceptron* suporta a aprendizado *online*, os artigos de notícias são incorporados no modelo de previsão de forma incremental e não sendo todo o conjunto processado a cada vez, por isso é possível que o tempo de execução não exceda de 13:56 minutos. Esses resultados precisam ser melhorados, mas são compatíveis com algumas das aplicações apresentadas em (VAISMAN; ZIMÁNYI, 2012).

Também, foram apresentados diferentes cenários de uso, que mostra o potencial de relacionamentos entre os termos do *corpus* e a exploração da multidimensionalidade em diferentes níveis de abstração. Para observar esse potencial, foram realizadas consultas com valores de dimensões específicos. Por exemplo, utilizando a categoria de `Politics`, os resultados já recuperaram informação útil ao seu contexto (a política). Agregando mais dimensões às consultas, obteve-se uma análise por diversas perspectivas. Ao explorar o conjunto de dados armazenado no ambiente *Newsminer*, foi possível obter resultados relevantes ao contexto da categoria selecionada e ao período em que as notícias ocorreram.

# CONCLUSÃO

Cenários compostos por textos de *sites* de notícias em tempo real são um desafio, tanto para o pré-processamento e armazenamento de dados em larga escala, dado ao seu volume e variedade de fontes de dados, tanto para a realização de tarefas de aprendizado de máquina. O objetivo de oferecer um *corpus* multidimensional às aplicações de Mineração de Dados e Textos, a partir de fontes Web, foi atingido. Utilizando o *Web Crawler* obteve-se um processo de coleta de notícias, que, integrado à etapa de *ETL* permite a realização de consultas multidimensionais em tempo real. Realizando a aplicação do método de extração de *stopwords* foi possível reduzir o tamanho de textos em 50%. O método de categorização de textos utilizado, *Perceptron*, tem um papel importante, pois permite que seja realizado o enriquecimento semântico dos dados e que isto ocorra de maneira incremental. E, a combinação de todas as tarefas citadas aqui, permitiu com que fosse concretizado o desenvolvimento do *Newsminer* como um ambiente de *Data Warehouse* exploratório de notícias em tempo próximo do tempo real, também fornecendo um novo *corpus* de textos e termos, hoje, na língua inglesa.

## Contribuições

Por fim, as contribuições desta pequisa são as seguintes:

- o *corpus* multidimensional baseado em notícias de *websites*;
- um novo *dataset* de textos de notícias em inglês, o *Newsminer Collection*, categorizado para futuras pesquisas. Este conjunto de dados foi utilizado no Experimento 1 do Capítulo 5.
- a incorporação de um *Web Crawler* como parte da etapa de *ETL*, permitindo a obtenção de textos em tempo próximo do real;
- o enriquecimento semântico como parte da arquitetura do ambiente, que descobre a classe morfológica dos termos e a categoria de notícias na etapa de *ETL*, auxiliando-se do classificador *Perceptron* aliado ao pré-processamento de textos de remoção de *stopwords* e a ontologia *IPTC*;
- a disponibilização do *corpus*, permitindo a exploração multidimensional de dados via consultas, na interface *Web*, e via *API*;

- via avaliação experimental, determinou-se que o método de classificação de textos, aliado ao método de pré-processamento de textos, que melhor categoriza textos de notícias foi o *Perceptron* para bases cujas *stopwords* sejam removidas.

## Trabalhos Futuros

Os trabalhos futuros que poderão ser desenvolvidos a partir desta pesquisa são listados a seguir:

- **Coletar textos de notícias em outras línguas:** pode ser criado um *corpora* multi-lingual. Para isto, faz-se necessário que estejam disponíveis técnicas de pré-processamento para a língua em si.

- **Implementação de uma consulta visual:** a grande maioria dos sistemas de visualização de dados multidimensionais utiliza a abordagem de consultas *pivô*, que permite arrastar campos e agregações.Para o *Newsminer*, tem-se como sugestão buscar uma versão *open source* da biblioteca *KENDO UI* [9].

- **Importação de bases de dados para dentro do *Newsminer***: ao permitir a importação de outras bases de dados para dentro do *Newsminer* poder-se-á garantir a exploração multidimensional de textos, palavras e categorias em outros cenários como, por exemplo, livros, revistas, artigos científicos, atas de reunião, entre outros.

- **Classificação de sub-classes**: de acordo com a ontologia *IPTC* para cada classe há suas sub-classes, sendo assim, é interessante a investigação de um método que permita realizar a categorização por sub-classes.

- **Paralelizar o *Newsminer***: desde a coleta de notícias até o armazenamento no banco multidimensional, o *Newsminer* tem um curto espaço de tempo. Por isso, a paralelização da etapa de *ETL* se torna interessante, assim, poderá abranger um maior número de *sites* coletados simultaneamente.

- **Enriquecer a *API***: outros parâmetros podem ser acrescentados para proporcionar maior flexibilidade para as aplicações via *API*. Por exemplo, data de início e fim de coleta quando ativado o *Web Crawler*.

- **Avaliação de bancos de dados NoSQL:** visando realizar uma avaliação de desempenho, será interessante avaliar o desempenho atual das consultas e compará-las com tecnologias *NoSQL*.

[9] *KENDO UI*. Disponível em <http://demos.telerik.com/kendo-ui/pivotgrid/index>, acessado em 20/04/2017

- **Implementação de recursos visuais:** tendo como objetivo ampliar o número de consultas visuais, poderão ser desenvolvidos trabalhos na área de visualização, como, por exemplo, nuvem de palavras.

- **Métricas de relevância de termos:** disponibilizar os dados no ambiente *Newsminer* em combinação com as métricas da Teoria de Informação (SEBASTIANI, 2002), tais como ganho de informação (*information gain*) ou *chi-quadrado*. Assim, as consultas multidimensionais ou cubos de dados fornecidos às aplicações podem consumir o conjunto de textos a partir de termos que reflitam sua importância para cada categoria do conjunto. As aplicações poderão escolher qual ou quais métricas utilizar para consumir os dados via interface ou *API*.

- **Armazenamento unificado de termos:** no modelo atual, permite-se o reconhecimento de termos compostos e termos simples, por exemplo *Hillary Clinton* e *Clinton*. Um trabalho interessante a ser desenvolvido é o reconhecimento destes termos, e até mesmo outros como *Hillary*, como sendo apenas um termo.

# Referências

ABELLO, A. et al. Using semantic web technologies for exploratory OLAP: a survey. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 27, n. 2, p. 571–588, 2015. Citado 4 vezes nas páginas 17, 51, 52 e 57.

AGARWAL, S.; SINGHAL, A. Autonomous ontology population from dbpedia based on context sensitive entity recognition. In: *Proceedings of the 4th International Joint Conference on Advances in Engineering and Technology*. Delhi Region, INDIA: [s.n.], 2013. p. 580–589. Citado 2 vezes nas páginas 53 e 55.

ATKINSON, M. et al. Online-monitoring of security-related events. In: *22nd International Conference on Computational Linguistics: Demonstration Papers*. [S.l.]: Association for Computational Linguistics, 2008. (COLING '08), p. 145–148. Citado na página 26.

BAHRAMI, M.; SINGHAL, M.; ZHUANG, Z. A cloud-based web crawler architecture. In: *Proceedings of 18th International Conference on Intelligence in Next Generation Networks*. Paris, França: [s.n.], 2015. p. 216–223. Citado na página 52.

BAI, Y. et al. Kwb: An automated quick news system for chinese readers. In: *Proceedings of the 8th SIGHAN Workshop on Chinese Language Processing*. Beijing, China: [s.n.], 2015. p. 110. Citado 2 vezes nas páginas 53 e 55.

BELIGA, S. Keyword extraction: a review of methods and approaches. *University of Rijeka, Department of Informatics, Rijeka*, Citeseer, 2014. Citado na página 42.

BIJALWAN, V. et al. KNN based machine learning approach for text and document mining. *International Journal of Database Theory and Application*, v. 7, n. 1, p. 61–70, 2014. Citado na página 44.

CRAMMER, K. et al. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, v. 7, n. Mar, p. 551–585, 2006. Citado na página 46.

DEHANKAR, S. K.; WAGH, K.; CHATUR, P. Web Page Classification Using Apriori Algorithm and Naïve Bayes Classifier. *International Journal of Advanced Research in Computer and Communication Engineering*, v. 3, n. 4, 2015. Citado 2 vezes nas páginas 54 e 55.

DEHNE, F. et al. Scalable real-time OLAP on cloud architectures. *Journal of Parallel and Distributed Computing*, Elsevier, v. 79, p. 31–41, 2015. Citado na página 51.

DEMsAR, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, JMLR.org, v. 7, p. 1–30, dez. 2006. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=1248547.1248548>. Citado na página 81.

DUDA, R.; HART, P. *Bayes Decision Theory*. [S.l.]: John Wiley & Sons, 1973. 10–43 p. Citado na página 44.

FACELI, K. et al. Inteligência artificial: Uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*, v. 2, p. 192, 2011. Citado na página 39.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Advances in knowledge discovery and data mining. In: . [S.l.]: American Association for Artificial Intelligence, 1996. cap. From Data Mining to Knowledge Discovery: An Overview, p. 1–34. Citado na página 25.

FIELDING, R. Representational state transfer. *Architectural Styles and the Design of Netowork-based Software Architecture*, p. 76–85, 2000. Citado na página 70.

FREUND, Y.; SCHAPIRE, R.; ABE, N. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, Japanese Society for Artificial Intelligence, v. 14, n. 771-780, p. 1612, 1999. Citado na página 45.

GARCIA, A. E. B.; TANAKA, A. K.; BAIÃO, F. A. Uma solução de bi em tempo real para o ambiente de computação em nuvem. In: *Anais do II Workshop de Teses e Dissertações em Sistemas de Informação*. Goiânia, Brasil: [s.n.], 2015. Citado 2 vezes nas páginas 51 e 82.

GOKER, A.; DAVIES, J. *Information retrieval: Searching in the 21st century*. [S.l.]: John Wiley & Sons, 2009. Citado na página 40.

GOLDSCHMIDT, R.; PASSOS, E. Data mining: um guia prático, conceitos, técnicas, ferramentas, orientações e aplicações. *Rio de Janeiro: Campus*, 2005. Citado na página 33.

GONG, J.-P.; SONG, J. Research on the performance of segmentation of text classification based on CNICC. In: *Proceedings of 15th International Conference on Computer and Information Science*. Okayama, Japan: [s.n.], 2016. p. 1–3. Citado na página 54.

HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques*. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 0123814790, 9780123814791. Citado 2 vezes nas páginas 31 e 37.

IGLESIAS, J. A. et al. Web news mining in an evolving framework. *Information Fusion*, v. 28, p. 90–98, 2015. ISSN 1566-2535. Citado 2 vezes nas páginas 53 e 55.

INMON, W. H. *Building the data warehouse*. New York, Estados Unidos: John Wiley & Sons, 2005. Citado 2 vezes nas páginas 31 e 34.

JENSEN, C. S.; PEDERSEN, T. B.; THOMSEN, C. Multidimensional databases and data warehousing. *Synthesis Lectures on Data Management*, Morgan & Claypool Publishers, v. 2, n. 1, p. 1–111, 2010. Citado na página 33.

JONNALAGEDDA, N. et al. Incorporating popularity in a personalized news recommender system. *PeerJ Computer Science*, v. 2, p. e63, 2016. Citado 3 vezes nas páginas 39, 53 e 55.

KIMBALL, R.; ROSS, M. *The data warehouse toolkit: the complete guide to dimensional modeling*. New York, Estados Unidos: John Wiley, 2011. Citado 5 vezes nas páginas 17, 31, 32, 33 e 34.

LESKOVEC, J.; RAJARAMAN, A.; ULLMAN, J. D. *Mining of massive datasets*. [S.l.]: Cambridge university press, 2014. Citado na página 40.

LI, L. et al. Scene: a scalable two-stage personalized news recommendation system. In: *Proceedings of the 34th International Conference on Research and development in Information Retrieval*. [S.l.: s.n.], 2011. p. 125–134. Citado na página 39.

LI, Z. et al. Fast text categorization using concise semantic analysis. *Pattern Recognition Letters*, v. 32, n. 3, p. 441 – 448, 2011. ISSN 0167-8655. Citado na página 40.

LIN, C. et al. Personalized news recommendation via implicit social experts. *Information Sciences*, Elsevier, v. 254, p. 1–18, 2014. Citado na página 39.

LIPARAS, D. et al. News articles classification using random forests and weighted multimodal features. In: *Proceedings of Information Retrieval Facility Conference*. [S.l.: s.n.], 2014. p. 63–75. Citado 2 vezes nas páginas 54 e 55.

LOSARWAR, V.; JOSHI, D. M. Data preprocessing in web usage mining. In: *Proceedings of International Conference on Artificial Intelligence and Embedded Systems*. New Asia, Singapura: [s.n.], 2012. p. 15–16. Citado na página 41.

MANGAL, S. B.; GOYAL, V. Text News Classification System using Naïve Bayes Classifier. *International Journal of Engineering Sciences*, v. 13, n. 12, p. 209–213, 2014. Citado 2 vezes nas páginas 54 e 55.

MANSMANN, S. et al. Discovering OLAP dimensions in semi-structured data. *Information Systems*, v. 44, p. 120 – 133, 2014. ISSN 0306-4379. Citado 5 vezes nas páginas 25, 49, 50, 55 e 62.

MCCALLUM, A.; NIGAM, K. et al. A comparison of event models for naïve bayes text classification. In: AAAI. *Proceedings of Workshop on Learning for Text Categorization*. Wisconsin, Estados Unidos, 1998. v. 752, p. 41–48. Citado na página 44.

MITTERMAYER, M.-A.; KNOLMAYER, G. F. Newscats: A news categorization and trading system. In: IEEE. *Proceedings of 6th International Conference on Data Mining*. Hong Kong, China, 2006. p. 1002–1007. Citado na página 79.

MOHIUDDIN, U.; AHMED, H.; ISMAIL, M. Newsd: A realtime news classification engine for web streaming data. In: ATLANTIS PRESS. *International Conference on Recent Advances in Computer Systems*. Baqaa, Arábia Saudita, 2015. Citado 2 vezes nas páginas 54 e 55.

MORAES, T. P. B. de et al. "Seeing is believing". Opinião pública, enquadramento midiático do NYT e Política de audiência. *Revista Andina de Estudios Políticos*, v. 6, n. 1, p. 121–141, 2016. Citado na página 26.

NAGABHUSHANA, S. *Data Warehousing, OLAP and Data Mining*. New Delhi, India: New Age International, 2006. Citado na página 32.

NGUYEN, K. M.; NGUYEN, T.-H.; HUYNH, X. H. Automated translation between restful/json and sparql messages for accessing semantic data. In: IEEE. *Electronics, Information, and Communications (ICEIC), 2016 International Conference on*. [S.l.], 2016. p. 1–4. Citado na página 70.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, n. Oct, p. 2825–2830, 2011. Citado na página 79.

PETROV, S.; DAS, D.; MCDONALD, R. A universal part-of-speech tagset. *Google Research Lab*, 2011. Citado na página 41.

PRASAD, K.; RAMAKRISHNA, S. Text analytics to data warehousing. *International Journal on Computer Science and Engineering*, Engg Journals Publications, v. 2, n. 6, 2010. Citado 2 vezes nas páginas 49 e 55.

REHMAN, N. U. et al. Building a data warehouse for twitter stream exploration. In: ACM/IEEE. *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*. Istanbul, Turquia, 2012. p. 1341–1348. Citado 2 vezes nas páginas 49 e 55.

RENSO, C.; RONCATO, A.; TRASARTI, R. Mob-warehouse: A semantic approach for mobility analysis with a trajectory data warehouse. In: SPRINGER. *International Conference on Conceptual Modeling*. Hong Kong, China, 2014. v. 8697, p. 127. Citado 2 vezes nas páginas 50 e 55.

ROCCHIO, J. J. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, Prentice Hall, 1971. Citado na página 45.

ROSE, S. et al. Automatic keyword extraction from individual documents. *Text Mining*, p. 1–20, 2010. Citado 2 vezes nas páginas 42 e 43.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. Citado na página 45.

SANTOS, R. J.; BERNARDINO, J. Real-time data warehouse loading methodology. In: ACM. *Proceedings of the 2008 international symposium on Database engineering & applications*. Coimbra, Portugal, 2008. p. 49–58. Citado na página 51.

SEBASTIANI, F. Machine learning in automated text categorization. *ACM Computing Surveys*, ACM, v. 34, n. 1, p. 1–47, 2002. ISSN 0360-0300. Citado na página 93.

SENN, S.; RICHARDSON, W. The first t-test. *Statistics in medicine*, Wiley Online Library, v. 13, n. 8, p. 785–803, 1994. Citado na página 80.

SILVA, T. S. et al. Um ambiente integrador de notícias de governo. *VII Simpósio Brasileiro de Sistemas de Informação, Salvador, Bahia, Brasil*, p. 373–383, 2011. Citado 2 vezes nas páginas 52 e 55.

SMOOT, P. D. O. IPTC - International Press Telecommunications Council. 2003. Citado 3 vezes nas páginas 27, 63 e 76.

TAO, F. et al. Eventcube: multi-dimensional search and mining of structured and text data. In: ACM. *Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining*. Chicago, Estados Unidos, 2013. p. 1494–1497. Citado 2 vezes nas páginas 50 e 55.

VAISMAN, A.; ZIMÁNYI, E. Data warehouses: Next challenges. In: *Business Intelligence*. [S.l.]: Springer, 2012. p. 1–26. Citado 4 vezes nas páginas 51, 83, 84 e 90.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995. Citado 2 vezes nas páginas 45 e 46.

VICTOR, S.; REX, M. M. X. Analytical implementation of web structure mining using data analysis in educational domain. *International Journal of Applied Engineering Research*, v. 11, n. 4, p. 2552–2556, 2016. Citado 4 vezes nas páginas 25, 50, 53 e 55.

WALKER, S. H.; DUNCAN, D. B. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, Oxford University Press, v. 54, n. 1-2, p. 167–179, 1967. Citado na página 44.

XIONG, H.; WU, J.; CHEN, J. K-means clustering versus validation measures: A data-distribution perspective. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, v. 39, p. 318–331, 2009. Citado na página 79.

XUE, D.; SUN, M. Raising high-degree overlapped character bigrams into trigrams for dimensionality reduction in chinese text categorization. In: SPRINGER. *International Conference on Intelligent Text Processing and Computational Linguistics*. [S.l.], 2004. p. 584–595. Citado na página 40.

ZHIQIANG, H.; WEI, S.; GUIXIAN, X. Research on tibetan news sites' web crawler and search engine. *International Conference on Logistics Engineering, Management and Computer Science*, Shenyang, China, 2015. Citado 2 vezes nas páginas 52 e 55.

ZINKEVICH, M. Online convex programming and generalized infinitesimal gradient ascent. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. [S.l.: s.n.], 2003. p. 928–936. Citado na página 46.