2. Conceitos, técnicas e trabalhos relacionados

Durante a fase de desenvolvimento desta pesquisa não foram observados trabalhos similares a recomendação de segundo nível. Contudo, destacamos neste capítulo dois trabalhos que apresentaram aspectos parecidos com a nossa proposta. Ainda, neste capítulo, vamos apresentar os principais conceitos e técnicas utilizados no decorrer da dissertação.

2.1. Conceitos e Técnicas

2.1.1. Vector Space Model

O *Vector space model* representa documentos textuais na forma de vetores de termos, onde cada termo é composto de um par *(palavra, peso)*.

Segundo Manning et al. [12], os vetores mantém todos os termos da coleção de documentos, e não somente os termos que ocorrem no documento em si. Assim, os termos que estão na coleção dos documentos, mas não estão no documento que está sendo transformado em vetor, recebem o peso 0. Os demais termos recebem um peso que significa um grau de importância e que pode ser concebido de diversas formas. A forma mais comum de atribuição de pesos é o *TF/IDF* que mede a relação entre a frequência de um termo dentro do documento *(TF)* pelo inverso da frequência do termo em todos os documentos da coleção *(IDF)*.

Os vetores associados aos documentos são então representados em um espaço euclidiano, onde cada termo do vetor representa uma dimensão.

Desta forma, a similaridade entre dois vetores de termos pode ser conhecida através da fórmula de distância dos cossenos, que mede o ângulo formado entre os dois vetores de termos.

A distância dos cossenos é o resultado do produto escalar dos vetores dividido pelo produto das suas magnitudes:

$$similarity = \cos(\emptyset) = \frac{A * B}{||A||||B||} = \frac{\sum_{i=1}^{n} Ai * Bi}{\sqrt{\sum_{i=1}^{n}(Ai)^2} * \sqrt{\sum_{i=1}^{n}(Bi)^2}}$$

Função 1 – Fórmula de distância dos cossenos

O fator de similaridade entre dois vetores estará entre 0 e 1, desde que os pesos não sejam negativos.

2.1.2. PageRank

PageRank é um algoritmo de análise de rede que visa dar pesos numéricos a cada nó da rede baseado nas suas ligações com outros nós para identificar a popularidade entre eles. O algoritmo, quando aplicado na Web, utiliza os hiperlinks entre as páginas para computar o *pagerank* entre elas. O peso numérico atribuído a uma determinada página é chamado de *pagerank da página*.

O algoritmo original de *pagerank* foi desenvolvido pelos fundadores do Google, Larry Page e Sergey Brin [25], enquanto desenvolviam sua ferramenta de busca na universidade de Stanford em 1998.

O algoritmo estabelece um peso inicial para cada elemento pertencente à rede. Quando um elemento se conecta a outro elemento da rede, este cede um percentual do seu *pagerank* para o elemento conectado, que é a razão de seu próprio *pagerank* pelo total de links que saem do elemento. Assim, imaginemos a seguinte situação:

Sejam A,B,C e D páginas na web com *pageranks* iniciais de 0.25. Suponha que A, B e C se conectam apenas a D. Temos que o *pagerank* de D é 0.75:

$$PR(D) = \frac{PR(B)}{S(B)} + \frac{PR(C)}{S(C)} + \frac{PR(A)}{S(A)}$$

Função 2 – Exemplo de pagerank

2.1.3. Classificador bayesiano ingênuo

Um *classificador bayesiano ingênuo* é um classificador probabilístico baseado na aplicação do teorema de Bayes. Para melhor entender o classificador

bayesiano ingênuo, vamos antes relembrar o conceito de classificação.

Classificação é uma das técnicas existentes em aprendizado de máquina, que segundo Jacob Perkins [7] é definida como a tarefa de atribuir um determinado rótulo a um dado texto de entrada de acordo com o reconhecimento de um padrão existente. Ou seja, o classificador aprende um padrão e com isso rotula as instâncias que correspondem a esse padrão.

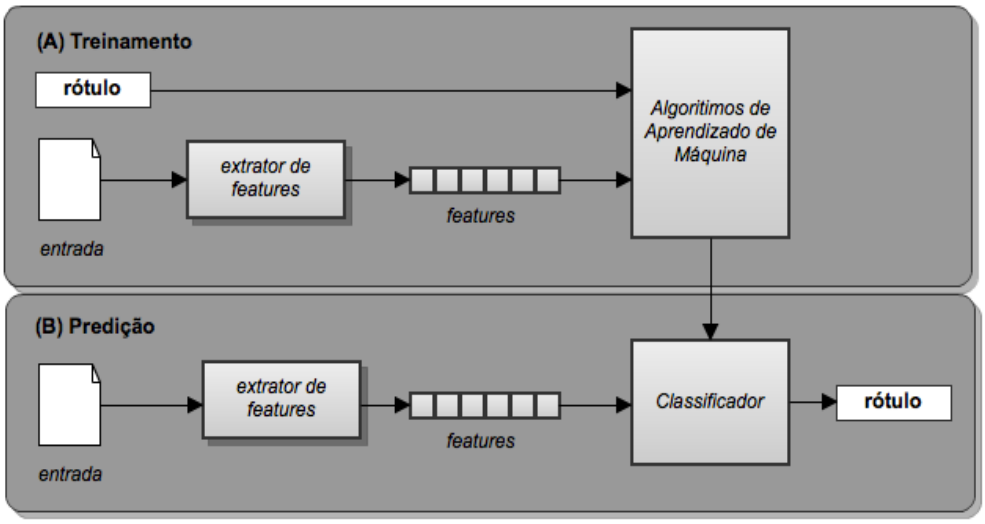

Figura 1 – Modelo do classificador

O aprendizado pode ser supervisionado ou não supervisionado. Para este trabalho, vamos nos limitar ao aprendizado supervisionado, onde o classificador utiliza um corpus de treinamento que o ajuda a inferir padrões para serem usados mais adiante no processo de classificação.

Os padrões são encontrados pelo classificador com o auxílio de *features* que são passadas para ele durante o aprendizado supervisionado. As *features* são características que possibilitam ao classificador o entendimento do corpus. Por exemplo, se tomarmos um classificador que reconhece a probabilidade de uma fruta ser uma maça, podemos definir como *features* o peso da fruta, a cor e o seu formato. Estas *features* são facilmente identificadas por serem comuns a outras frutas, mas variam de uma para outra fruta.

De forma simples, um classificador bayesiano ingênuo é um classificador probabilístico que parte do pressuposto que a presença ou ausência de uma *feature* de uma classe não esta relacionada à presença ou à ausência de outras *features* da

mesma classe, portanto, o termo *ingênuo*.

2.1.4. Part-of-speech Tagging

Segundo Perkins [7], *part-of-speech tagging* (*POS-tagger*) é o processo de converter uma sentença no formato de uma lista de palavras para uma lista de tuplas, onde cada tupla é composta de uma palavra e uma *tag***.** Neste caso, a *tag* refere-se à classe gramatical da palavra E é denominada *POS-tag*.

A dificuldade nesta tarefa se dá porque as palavras podem ter mais de um *POS-tag* possível e, com isso, é indispensável o entendimento do contexto onde a palavra foi empregada para que haja a desambiguação da *POS-tag*.

Na literatura encontramos o trabalho de Kepler e Finger [24], que apresentam a implementação de um *POS-tagger* para língua portuguesa com acurácia de 95,51%.

2.2. Trabalhos relacionados

2.2.1. Sistemas de recomendação

Sistemas de recomendação são softwares especializados em apresentar "opções" para serem usadas por seus usuários [5,6,18]. Esses sistemas auxiliam pessoas que não possuem muita experiência ou competência para pesquisar dados acerca de um determinado assunto.

Em linhas gerais, os sistemas de recomendação procuram oferecer as melhores opções de resposta para as necessidades dos usuários num processo de tomada de decisão. Os sistemas de recomendação podem oferecer sugestões em diferentes domínios como, por exemplo: que item comprar em uma loja virtual, que artigo ler em um site de notícias ou mesmo que restaurante visitar em uma cidade.

No sistema de recomendação do *IMDB* (*Internet Movie Database*), quando o usuário seleciona um filme do catálogo para ler a respeito, o sistema apresenta também uma lista de sugestões de filmes relacionados para o usuário. Assim como o *IMDB*, o Web site de compras *Amazon* apresenta, para cada produto selecionado pelo usuário, uma lista de recomendações de outros produtos que possam servir para o usuário dentro de uma mesma compra.

Sistemas de recomendação são usualmente personalizados com base em características individuais ou coletivas. Todavia, os sistemas de recomendação não personalizados também têm o seu espaço. Sistemas de recomendação não personalizados são mais simples de implementar e geralmente são usados para recomendações mais gerais como, por exemplo, a lista dos "10 mais" de um determinado assunto ou tema.

Sistemas de recomendação personalizados tentam antecipar as necessidades do usuário analisando informações do seu perfil e levando em consideração as restrições de domínio para então recomendar sugestões. Ainda, no Web site de compras da *Amazon*, uma vez realizada uma compra, as informações são armazenadas no perfil do usuário de modo que, para as compras futuras, o usuário receba como recomendação não apenas itens relacionados pela categoria em que se encaixam, mas também por que estão relacionados com a última compra feita por ele.

As próximas seções discutem técnicas utilizadas para descobrir as preferências do usuário baseadas no livro de Francesco Ricci [5].

Sistemas de recomendação baseados em conteúdo

Sistemas de recomendação baseados em conteúdo tentam recomendar opções que são similares a algum item que o usuário já selecionou no passado usando como *features*[1] informações extraídas do produto/conteúdo que o usuário consultou. Por exemplo, se um usuário leu um artigo sobre política em um portal de notícias, este portal poderia sugerir em consultas futuras artigos relacionados a política.

Segundo Yuanhaua et al. [1] os sistemas de recomendação baseados em conteúdo junto com os baseados em filtragem colaborativa são os mais difundidos e utilizados entre os sistemas de recomendação.

Sistemas de recomendação baseados em filtragem colaborativa

Sistemas de recomendação baseados em filtragem colaborativa tentam recomendar ao usuário opções que foram utilizadas por usuários com os mesmos

[1] Termos de uso consagrado, como este, serão mantidos no original em todo o texto.

interesses. Por exemplo, se um grupo de usuários seleciona o produto X e em seguida selecionam o produto Y, o sistema de recomendação entende que usuários que acessam o produto X também acessam o produto Y e passa a recomendar o produto Y toda vez que algum novo usuário se interessar pelo produto X.

Sistemas de recomendação baseados em nichos demográficos

Este tipo de sistema de recomendação baseia-se no perfil demográfico do usuário para recomendar opções. Usuários de uma determinada idade recebem recomendações diferentes de usuários de outras idades. Os nichos demográficos podem ser: idade, sexo, língua, etc.

Sistemas de recomendação baseados em conhecimento

Sistemas de recomendação baseados em conhecimento tentam recomendar ao usuário opções baseando-se no conhecimento específico do domínio do sistema. Este modelo também é conhecido como um modelo baseado em caso onde o problema é a analise das necessidades do usuário e a solução é o conjunto de opções a serem recomendadas. O principio de recomendação está, portanto, baseado na similaridade de uma solução para um dado problema.

Sistemas de recomendação com esta técnica tendem a trabalhar melhor que os outros no início, porém, se não são acompanhados de um componente de aprendizagem, tornam-se ineficientes.

Sistemas de recomendação baseados na comunidade

Este tipo de sistema de recomendação baseia-se nas relações que o usuário possui com sua rede de amigos para realizar as recomendações. Neste modelo, acredita-se que as recomendações feitas por pessoas ligadas ao usuário tendem a ser mais efetivas. Esta abordagem tem se tornado bastante atrativa, tendo em vista o grande crescimento das redes de relacionamento.

Sistemas de recomendação híbridos

Sistemas de recomendação híbridos procuram combinar diversas técnicas de

forma a compensar as deficiências de cada técnica específica.

2.2.2. Ferramentas para extração de entidades nomeadas

Ferramentas para extração de entidades nomeadas tipicamente recebem um texto como entrada e devolvem as entidades que fossem encontradas. A seguir, resumimos quatro serviços estudados durante o trabalho.

Yahoo Term Extraction

O serviço do Yahoo de extração de termos [26] permite a seus usuários o acesso a uma API para análise de textos que fornece ao final uma lista de palavras ou frases relevantes, em inglês, extraídas do texto submetido. O serviço pode ser acessado através do protocolo REST e responde os dados de saída em formatos XML e Json.

O serviço é gratuito. Porém, possui limite de requisições diárias estipulado em 5000 requisições/dia.

O serviço pode ser utilizado mediante o cadastramento e obtenção da chave de acesso.

NLTK

É um serviço Web [27] que funciona sob o protocolo REST para mineração de texto e processamento de linguagem natural. A API foi concebida com base nas premissas do *NLTK cookbook* e não tem fins comerciais, de modo que possui limites tanto para o número de requisições (1000 requisições diárias) quanto para o tamanho do texto enviado (10000 caracteres). O formato de saída pode ser em XML ou Json.

Ltasks

Também é um serviço Web [28] que funciona sob o protocolo REST e que apresenta várias possibilidades de extração de informação do texto entre elas o reconhecimento de entidades nomeadas em língua portuguesa. Para utilização on-line é necessária a utilização de uma chave de acesso que é obtida através de um cadastro no site.

Zemanta

É uma ferramenta [29] concebida para geração de conteúdo relacionado para blogs. Contudo, seus idealizadores provêem uma API REST que permite a extração de entidades contextualizadas ao texto submetido. Para fazer uso do serviço é necessário um cadastro e a obtenção de uma chave de acesso. O serviço é, em princípio, independente de idioma.

É importante ressaltar que os serviços utilizados neste trabalho não são os únicos disponíveis no mercado. Podemos citar por exemplo o serviço F-EXT disponibilizado pelo LEARN na PUC RIO que apresenta uma api para extração de entidades nomeadas em português. Este serviço pode ser acessado em http://www.learn.inf.puc-rio.br/. A não utilização deste serviço no grupo de testes se deu apenas pelo desconhecimento do mesmo durante a etapa de desenvolvimento do trabalho.

Comparação entre as ferramentas

A Tabela 1 compara as principais características das ferramentas analisadas.

Tabela 1 – Comparativo de extratores de entidades

Características	**ltasks**	**Yahoo**	**NLTK**	**ZEMANTA**
Tem suporte ao idioma português	sim	não	sim	sim
Linguagem de desenvolvimento	Java	-	Python	-
Limite de acesso diário	-	5000	1000	1000
Tamanho máximo do texto	-	-	10000	10000
Possui código aberto	não	não	não	não

Dentre as ferramentas observadas, o serviço **Ltasks** foi o que apresentou o melhor aproveitamento na extração de entidades. Porém, a ausência de um código aberto para aprimoramento do algoritmo, a limitação de acessos ao serviço e o tempo gasto em cada requisição são pontos desfavoráveis ao uso destas ferramentas.

2.2.3. Outras Ferramentas

Projeto Pure

O projeto **PURE** [15] é um sistema para recomendação de artigos médicos, contidos na base de dados **PubMed**, que utiliza o princípio de recomendação baseado na filtragem de conteúdo. A base de dados **PubMed** mantém um grande acervo de artigos de biologia e medicina com um volume diário de atualização da ordem de centenas de artigos.

O sistema **PURE** pode ser entendido pelo fluxo de operações a seguir:

1. O usuário acessa o sistema para informar os artigos do seu interesse. Estes artigos são armazenados na base de dados do **PURE**.
2. Um sistema de aprendizado de máquina é aplicado para extrair as preferências do usuário com base nos seus artigos de interesse.
3. O sistema **PURE** consulta a base de dados **PubMed** para baixar os novos artigos publicados.
4. Os artigos baixados da **PubMed** são ordenados com base no modelo treinado com as preferências do usuário.
5. Os artigos são então apresentados para o usuário.

Os módulos do sistema **PURE** são descritos a seguir.

Interface para registro de artigos do usuário

Para utilizar o sistema, o usuário precisa registrar os seus artigos de interesse na base de dados do **PURE**. Para esta atividade o usuário acessa a interface Web do sistema e seleciona os artigos de sua preferência em uma listagem. Os artigos selecionados são então gravados no perfil do usuário e armazenados no banco de dados do **PURE**.

O usuário tem a permissão de adicionar novos arquivos e alterar sua lista de interesse.

Treinamento do modelo probabilístico

Os artigos de interesse do usuário são utilizados para a concepção de um

modelo probabilístico que procura identificar as preferências do usuário para novos artigos. Esse modelo é dividido em duas etapas, descrita nas próximas seções.

Seleção de palavras e atribuição de peso

Nesta etapa, o sistema trata os artigos do **PubMed** como um vetor de palavras ordenadas por peso. Estas palavras são obtidas a partir da eliminação de palavras irrelevantes para o sistema classificadas como *stopwords*.

As *stopwords* são obtidas por duas estratégias distintas. A primeira consiste na geração do *DF* (*document frequency* - número de documentos onde a palavra aparece) e *TF-IDF* (ver Seção 2.2) das palavras oriundas de uma porção aleatória de artigos da base de dados do **PubMed**. As palavras com alto *DF* ou com baixo *TF-IDF* são consideradas *stopwords*. A segunda estratégia consiste em considerar como *stopwords* as palavras que respeitam as seguintes regras: 1) palavras com menos de 3 letras; 2) palavras sem caracteres alfabéticos; 3) palavras que aparecem no *Journal of Business Research* de janeiro de 2005 a 2006.

Após a eliminação das *stopwords* é dado um peso para cada palavra restante do documento. Este peso é obtido pela verificação da distribuição da palavra pelo documento (*TF* – ver Seção 2.2).

Etapa de Geração do modelo probabilístico

As palavras selecionadas na etapa anterior são usadas para treinar um classificador probabilístico que será usado para gerar uma métrica de recomendação para os novos artigos. A função utilizada para computar o grau de recomendação de um artigo é dada pela fórmula abaixo:

Sendo d um artigo, z a variável correspondente ao cluster dos artigos de interesse do usuário, s um campo existente na estrutura do artigo, por exemplo:o campo título, e w uma palavra do artigo, temos:

$$P(d) = \sum_z P(d,z) = \sum_z P(z) \prod_{s \in d} \prod_{w \in s} Ps(w|z)$$

Função 3 – Função de recomendação do **PURE**

Em seguida os autores treinam os parâmetros de probabilidade *P(z)* e *Ps*(*w*|*z*) a partir dos artigos preferidos do usuário, utilizando o algoritmo de Maximização de Expectativa (EM).

Recuperação diária de novos artigos da base PubMed

O sistema **PURE** diariamente executa uma operação de recuperação dos novos artigos publicados na base de dados do **PubMed**. Os novos artigos são armazenados na base de dados do sistema para serem classificados de acordo com as preferências de cada usuário.

Recomendação de artigos

Para cada artigo recuperado, são extraídas palavras que são usadas como base para geração do critério de ordenação, que é dado de acordo com a função de recomendação definida anteriormente. Como forma de ajustar possíveis desvios do algoritmo de recomendação, os autores apresentam um *score* adicional para cada artigo. O *Z-score* é obtido através do agrupamento dos artigos em conjuntos de artigos com o mesmo número de palavras. O *Z-score* de um artigo é dado pela formula:

$$Z = \frac{P(d) - \mu}{\sigma}$$

Função 4 – Cálculo do Z-score no **PURE**

Deste modo, é separado o mínimo e o máximo grau de recomendação do grupo. Os artigos com o maior *Z-score* são então recomendados para o usuário.

Query by Document

Neste trabalho, Yang et al. [10] descrevem uma técnica de recuperação de conteúdo relacionado utilizando informações existentes no texto para consulta na base de dados.

A primeira idéia consiste em extrair do texto as *frases substantivas* que possam ser relevantes para pesquisa por conteúdo relacionado. Em seguida, estas frases podem ser substituídas ou melhoradas através do uso de fontes externas, no

caso, a Wikipédia.

Para melhorar as frases com o uso do Wikipédia, um grafo de conceitos da Wikipédia é utilizado, onde os nós também representam frases substantivas. Após recuperar as frase presentes no texto, o grafo é percorrido, substituindo-se as frase substantivas encontradas no texto por outras.

Para extrair as frases substantivas do texto, um *POS-tagger* considera todas as frases cujo padrão de formação respeita o conceito de frase substantiva, definido no trabalho. A Figura 2 mostra os padrões de frases substantivas definidos.

Padrão	**Instância**
S	Nintendo
AS	global warning
SS	Apple computer
AAS	declarative approximate selection
SSS	computer science departament
ARAS	efficient and effective algorithm
ASSS	Junior United States Senator
SSSS	Microsoft Host Integration Server
...	...
SSSSS	United States President George Bush

A – Adjetivo, R – Artigo, S – Substantivo

Tabela 2 – Exemplo de frases substantivas

A partir deste ponto, as frases são ordenadas de acordo com um *score* que é dado por dois mecanismos distintos. O primeiro mecanismo utiliza o *TF/IDF* dos termos na frase para atribuir um *score*, enquanto que o segundo computa o *score* baseado nas informações mutuas dos termos da frase.

Para validar o experimento, os autores utilizaram o serviço **Mechanical Turk** da *Amazon* para avaliar a qualidade dos documentos recomendados com uso das frases substantivas. Basicamente, eles informavam o texto e os primeiros 5 documentos que foram recuperados com uso das frases substantivas para os usuários, pedindo para que eles avaliassem se os documentos retornados eram

relacionados ao texto ou não.