Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Dissertação de Mestrado

Uma Arquitetura Multimodal para Recomendação Baseada em Conteúdo para TV Digital

Reudismam Rolim de Sousa

Campina Grande
Fevereiro - 2014

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Uma Arquitetura Multimodal para Recomendação Baseada em Conteúdo para TV Digital

Reudismam Rolim de Sousa

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Engenharia de Software para Computação Pervasiva

Hyggo Almeida / Angelo Perkusich
(Orientadores)

Campina Grande, Paraíba, Brasil

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

S725a Sousa, Reudismam Rolim de.

Uma arquitetura multimodal para recomendação baseada em conteúdo para tv digital / Reudismam Rolim de Sousa. – Campina Grande, 2014.

121 f. : il. Color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

"Orientação: Prof. Hyggo Almeida, Prof. Angelo Perkusich".

Referências.

1. Sistemas de Recomendação. 2. Multimodalidades. 3. Tv Digital. I. Almeida, Hyggo. II. Perkusich, Angelo. III. Título.

CDU 004.65(043)

"UMA ARQUITETURA MULTIMODAL PARA RECOMENDAÇÃO BASEADA EM CONTEÚDO PARA TV DIGITAL"

REUDISMAM ROLIM DE SOUSA

DISSERTAÇÃO APROVADA EM 24/02/2014

HYGGO OLIVEIRA DE ALMEIDA, D.Sc, UFCG
Orientador(a)

ANGELO PERKUSICH, D.Sc, UFCG
Orientador(a)

MARCOS RICARDO ALCÂNTARA MORAIS, D.Sc, UFCG
Examinador(a)

LEANDRO BALBY MARINHO, Dr., UFCG
Examinador(a)

CAMPINA GRANDE - PB

Resumo

Os provedores de conteúdo de TV Digital estão cada vez mais disseminados, com centenas de programas disponibilizados a cada dia. A sobrecarga de informação torna difícil para o usuário encontrar programas de interesse. Para ajudar o usuário, sistemas de recomendação (SRs) são abordagens populares. Contudo, aplicar SRs em alguns ambientes apresenta problemas, ou devido à falta de dados, ou porque os dados disponíveis são insuficientes para criar recomendações acuradas utilizando abordagens padrões. No domínio de TV Digital, a principal informação disponível é o guia eletrônico de programação (EPG). Os dados contidos no EPG são limitados, contendo somente dados textuais reduzidos, tornando difícil obter recomendações acuradas usando técnicas de recomendação padrões. Para resolver esse problema, neste trabalho é introduzida uma arquitetura que utiliza uma abordagem multimodal para recomendar programas de TV, combinando o texto do EPG e informações visuais. Um experimento foi realizado e demostrou que usando características multimodais a acurácia da recomendação pode ser elevada quando comparada com uma abordagem de recomendação padrão.

Abstract

Digital TV content providers are becoming widespread, with hundreds of programs available each day. The information overload makes difficult for the user to find programs of interest. To help the user, *recommender systems (RSs)* are a popular path. However, applying RSs to some environments is not an easy task, either due to the lack of data or because the data available is insufficient to create accurate recommendations using standard RS approaches. In the Digital TV domain, the main information available to make recommendations is the *Electronic Program Guide (EPG)*. The information available on EPG is limited, containing only reduced textual data, making difficult to get an accurate recommendation using standard techniques. To solve this problem, in this work we introduce an architecture that uses multimodal approach to recommend Digital TV programs, combining EPG text and visual information. We performed an experiment and demonstrated that using multimodal features the accuracy of the recommendation can be improved when compared with a recommender standard approach.

Agradecimentos

Primeiramente gostaria de agradecer a Deus por permanecer comigo nos momentos bons e ruins, segundo aos meus pais, Maria do Socorro Rolim de Sousa e Deusimar Antonio de Sousa e a minha esposa Luciana de Oliveira Silva. Também gostaria de agradecer aos colegas de mestrado, em especial Antonio Alexandre Moura Costa e Felipe Barbosa Araújo Ramos e Ricardo de Sousa Job. Aos colegas de Análise e Desenvolvimento de Sistemas (Campus Cajazeiras). Aos colegas do Laboratório de Sistemas Embarcados e Computação Pervasiva (Embedded). A todos que participaram da validação do trabalho. Aos professores orientadores Dr. Hyggo Almeida e Angelo Perkusich e com quem fiz alguma disciplina, Dr. Jacques Philippe Sauvé, Dr[a] Raquel Vigolvino Lopes, Dr. Leandro Balby Marinho, Dr[a] Joseana de Macêdo Fechine, Dr. Herman Martins Gomes e Dr. Antônio Berto Machado. Aos integrantes do grupo de leitura em Sistemas de Recomendação. E a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e a Coordenação de Pós-graduação em Informática (COPIN).

Conteúdo

Lista de Símbolos

SRs - *Sistemas de Recomendação*

DTV - *TV Digital*

RC - *Recomendação baseada em Conteúdo*

FC - *Filtragem Colaborativa*

EPG - *Guia Eletrônico de Programação*

SRI - *Sistemas de Recuperação de Informação*

FM - *Fatoração de Matrizes*

EQM - *Erro Quadrado Médio*

REQM - *Raiz do Erro Quadrado Médio*

EMA - *Erro Médio Absoluto*

EMAN - *Erro Médio Absoluto Normalizado*

DCG - *Ganho Cumulativo Descontado*

SIFT - *Scale Invariant Feature Transform*

DMC - *Detecção de Mudança de Câmera*

PSNR - *Razão Pico Sinal por Ruído*

SSIM - *Similaridade Estruturada*

MRF - *Markov Random Field*

SVH - *Sistema Visual Humano*

GIC - *Grafo de Interação entre Características*

TF-IDF - *Frequência do Termo - Inverso da Frequência nos Documentos*

k-NN - *Algoritmo dos Vizinhos mais Próximos*

DAO - *Objeto de Acesso aos Dados*

JWNL - *Java WordNet Library*

WS4J - *WordNet Similarity for Java*

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Sistemas de recomendação (SRs) são ferramentas de software e técnicas que fornecem sugestões de itens para os usuários [71]. Dentre suas funções, têm-se: aumentar a quantidade de itens vendidos, aumentar a satisfação do usuário e entender o que o usuário deseja. Item é um termo usado para denotar o que é recomendado para o usuário.

Os SRs estão em evidência pela demanda crescente por aplicações que capturem os interesses do usuário devido ao crescimento das opções de escolha [13, 18, 57, 68, 71] (i.e. filmes, livros, programas de TV, etc.), que dificultam a seleção dos itens de interesse. Por outro lado, empresas buscam entender o usuário para aplicarem seus recursos de maneira mais eficiente [39].

Na forma mais simples, recomendações personalizadas são oferecidas como uma lista ordenada de itens. Durante a ordenação, o sistema de recomendação tenta predizer quais os itens ou serviços mais relevantes baseado nos interesses do usuário [13, 18, 57, 68, 71]. Para realizar essa tarefa os SRs coletam as preferências do usuário, ou explicitamente, na forma de avaliação, ou implicitamente, mediante interpretação das ações do usuário [71].

Duas técnicas são largamente usadas em sistemas de recomendação: recomendação baseada em conteúdo e filtragem colaborativa [13, 57, 68, 71]. Na recomendação baseada em conteúdo o sistema indica itens usando os atributos dos itens similares aos que o usuário interagiu [54, 71]. Em contrapartida, a abordagem mais simples desenvolvida para filtragem colaborativa recomenda para o usuário itens que usuários com interesses similares gostaram no passado [50, 71]. A similaridade é calculada a partir das avaliações dos usuários para os itens [50]. Como o trabalho está voltado para recomendação baseada em conteúdo, as

discussões seguintes se referem a essa abordagem.

Para o desenvolvimento de sistemas de recomendação, técnicas de aprendizagem de máquina são tipicamente aplicadas [3,67,71]. Essas técnicas usam o conjunto de dados disponível para inferir as preferências do usuário pelos itens. Porém, para alguns ambientes a coleta dos dados pode não ser uma tarefa trivial, ou porque os meios não fornecem informações suficientes para a construção de uma representação realística do item ou pela inexistência dos meios de coleta de informação [54].

A indisponibilidade ou a pouca representatividade dos dados é uma das principais limitações da abordagem baseada em conteúdo [13, 18, 57, 68, 71, 76]. Segundo Lops et al. [54] nenhum sistema de recomendação baseado em conteúdo pode fornecer sugestões adequadas se não existem informações suficientes para discriminar itens de interesse para o usuário.

Uma área em que a recomendação baseada em conteúdo vem sendo largamente aplicada é a TV digital [4,5,12,27,41,52,60,65,67,76,87,88,91,92]. Com centenas de programas disponibilizados a cada dia, os usuários estão sentindo a necessidade de serviços personalizados que os auxiliem a encontrar os programas de interesse. Porém, a principal fonte de dados disponível para o desenvolvimento de serviços de personalização é o guia de programação EPG (do inglês *Eletronic Program Guide* - Figura 1.1)[1].

Um problema recorrente no ambiente de TV é o novo item. Itens novos surgem com frequência, dificultando a aplicação de uma abordagem colaborativa [85].

As informações contidas no EPG são apenas textuais [6, 76] e estudos demonstram que o uso de características multimodais (textos, vídeos, usuários, etc.) elevam a acurácia dos sistemas de recomendação [25, 56, 79, 89]. Uma vez que cada tipo de dado possui propriedades distintas que elevam a representatividade do item [7, 79]. Luo et al. [56] identificou que as características visuais são componentes críticos para recomendação baseada em vídeo. Porém, poucos estudos exploram características multimodais para recomendação de programas em TV digital. Pelo maior de nosso conhecimento, apenas Luo et al. [56] apresenta uma abordagem multimodal para recomendação de vídeos de notícias de TV. Porém, diferentemente de Luo et al., este trabalho é voltado para a recomendação de programas de TV em geral e não apenas uma categoria deles.

Os sistemas de recuperação de informação (SRI) [51], uma área correlata aos sistemas

[1]Imagem retirada livremente da internet site: `http://goo.gl/uhLgcN`

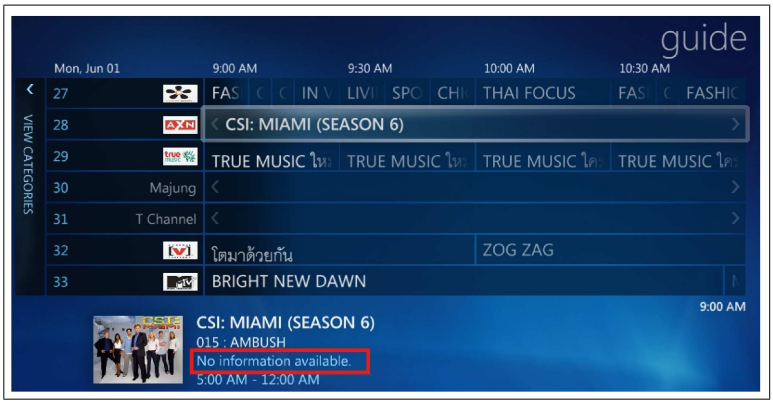

Figura 1.1: Ilustração do guia eletrônico de programação, alguns itens presentes no guia não possuem descritivos textuais, o que dificulta a aplicação de recomendadores baseados unicamente em texto.

de recomendação, estão usando a diversidade de informação presente em ambientes sociais multimídias como fonte de informações para construir modelos para seleção dos melhores itens [25, 80] (Figura 1.2 modificada da rede social Flickr [83]). No entanto, pesquisas no tocante à aplicação de multimodalidades em sistemas de recomendação necessitam de estudos mais amplos.

A diferença entre SRI e sistemas de recomendação é que a recuperação de informação busca encontrar itens relevantes com base em um único item, enquanto que os sistemas de recomendação calculam a relevância baseando-se no perfil do usuário, usualmente composto por um conjunto de itens.

1.1 Objetivo do Trabalho

Neste trabalho, tem-se como objetivo desenvolver uma arquitetura para recomendação baseada em conteúdo utilizando dados multimodais para TV Digital. O objetivo da arquitetura é encontrar programas que o usuário gostaria de assistir a partir dos programas assistidos por ele.

A multimodalidade é expressa em termos de características textuais e visuais. Para que a multimodalidade seja modelada, é necessário encontrar representações adequadas para as características utilizadas. Por isso, para a extração de características textuais são empregadas

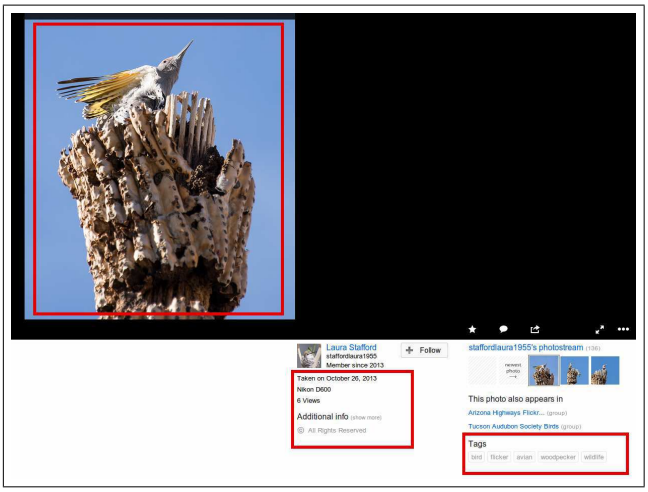

Figura 1.2: Ilustração de um objeto multimídia.

técnicas de mineração de texto. No entanto, para a extração de características visuais, como o vídeo é formado por um grande conjunto de quadros e a maioria deles são repetições, é necessária uma técnica para selecionar somente o conjunto mais relevante dos quadros (não repetitivos) do vídeo. A partir do conjunto de quadros selecionados, são extraídas as características visuais. O programa e o usuário são representados em termos das características multimodais extraídas. A abordagem de recomendação utiliza essas representações para gerar a lista recomendada. Por isso, para alcançar o objetivo do trabalho, alguns objetivos específicos foram identificados, são eles:

1. Investigação de técnicas para seleção do conjunto de quadros mais representativos do programa;

2. Emprego de técnicas de extração de características visuais nos quadros (imagens);

3. Caracterização do programa em termos de características textuais e visuais;

4. Recomendação de programas empregando dados multimodais.

A solução proposta no trabalho emprega a técnica de detecção de mudança de câmera para a seleção do conjunto de quadros mais representativos para o ambiente de TV e caracteriza o programa em termos de aspectos textuais e visuais, sendo estes representados por descritores de interesse e extraídos do conjunto de quadros identificado e aqueles da descrição do programa no EPG. A abordagem de recomendação empregada utiliza um grafo de

interação entre as diferentes modalidades para construir um modelo do item e do usuário usado para predição dos programas recomendados.

Outras características investigadas no trabalho incluem: determinação da eficiência da detecção de mudança de câmera para seleção do conjunto de quadros mais representativos e dois experimentos para detectar a capacidade das características visuais em representar os programas de TV. O primeiro investiga o uso de descritores visuais para classificar eventos (abraço, aperto de mão, toque cinco, etc.) em programas de TV. O segundo investiga o uso desses eventos para classificar programas.

Para validação do trabalho foi realizada uma pesquisa em que os participantes indicavam os programas assistidos por eles, e quantas vezes esses programas eram vistos no intervalo de uma semana. Esses dados foram transformados para refletir o *feedback* implícito do usuário para o programa. Essas informações foram utilizadas para recomendação e validação do trabalho, comparando a abordagem proposta com uma técnica de recomendação padrão. Diferentes abordagens de recomendação foram testadas e a que apresentou melhores resultados foi comparada com a abordagem multimodal. As recomendações foram validadas em termos de duas métricas: Precisão e DCG (do inglês *Discounted cumulative gain*) (Capítulo 4).

1.2 Relevância

O advento da Internet e da *World Wide Web* disponibilizou grandes quantidades de informações e serviços [3]. Embora seja desejável, a sobrecarga de informação dificulta a seleção do conteúdo de interesse [12]. A solução são os sistemas de recomendação, auxiliando o usuário a encontrar o conteúdo de acordo com suas preferências.

Com a expansão dos conteúdos televisivos, a TV digital também sofre com a sobrecarga de informação [5, 12, 27, 41, 60, 65, 67, 76, 91, 92]. Com centenas de programas disponíveis, os usuários necessitam encontrar programas de interesse. Dessa forma, aplicações que modelem o perfil do usuário e personalizem o acesso à informação são necessárias para que o usuário encontre itens que se adequem aos seus interesses [5, 12, 27, 41, 60, 65, 67, 76, 91, 92].

O emprego de multimodalidades é atrativo para aplicação em sistemas de TV personalizada, pois cada tipo de característica representa aspectos distintos e complementares do item [25, 26, 61, 62, 79] contribuindo assim para elevar a qualidade da recomendação, uma

vez que os itens são melhores representados.

O trabalho contribui para a área de sistemas de recomendação aplicados à TV digital, uma vez que as multimodalidades, embora relevantes, sejam pouco abordadas no ambiente. Como o trabalho utiliza características visuais, a pesquisa também é relevante para visão computacional aplicada, pois os trabalhos focam na detecção, classificação e reconhecimento de objetos, e a pesquisa aplica as características visuais para recomendação. A importância da aplicação de multimodalidades é apoiada na literatura, a qual ressalta os benefícios de trabalhos dessa natureza.

Por fim, a pesquisa possui relevância direta para avanços nas pesquisas do Laboratório de Sistemas Embarcados e Computação Pervasiva (Embedded) voltadas para o domínio de TV Digital.

1.3 Estrutura da Dissertação

A dissertação está organizada em 6 capítulos. No Capítulo 2 é apresentada a fundamentação teórica, introduzindo os sistemas de recomendação e as diferentes abordagens de recomendação, a mineração de imagem e vídeo e suas aplicações, o processamento de texto, o modelo de probabilidade usado no trabalho e as abordagens de aprendizado para dados multimodais.

No Capítulo 3 é apresentada a arquitetura proposta no trabalho, definindo técnicas para redução do conjunto de quadros do programa, abordagens para extração do conjunto de características visuais, técnicas para extração do conjunto de características textuais e modelos para representação do item e usuário para recomendação em termos de multimodalidades. No Capítulo 4 é realizada a validação da arquitetura, mostrando o uso da arquitetura para recomendação e um experimento comparando o modelo multimodal usado com um modelo de recomendação padrão. No Capítulo 5 são destacados os trabalhos relacionados, mostrando algumas arquiteturas usadas para recomendação em TV digital e alguns modelos multimodais que podem ser usados. No Capítulo 6 é apresentada a conclusão e os trabalhos futuros.

Além disso, no Apêndice A é descrito um experimento para determinar os melhores parâmetros para serem usados na detecção de mudança de câmera. No Apêndice B são descritos experimentos que abordam a capacidade dos descritores de características em representar o vídeo. No Apêndice C são descritas características em relação à arquitetura proposta, tais

como implantação, padrões arquiteturais, linguagens e ferramentas de suporte empregados. No Apêndice D são descritos os artefatos de software usados no protótipo instanciado na arquitetura. No Apêndice E pode ser visto o trabalho publicado como resultado da pesquisa. E por fim, no Apêndice F é apresentado o questionário usado na avaliação.

Capítulo 2

Fundamentação Teórica

2.1 Sistemas de Recomendação

Sistemas de recomendação são ferramentas de software e técnicas que fornecem sugestões de itens para os usuários [71]. Dentre as abordagens de recomendação se destacam: recomendação baseada em conteúdo, filtragem colaborativa e recomendação híbrida.

Sistemas de recomendação baseados em conteúdo tentam indicar itens similares aos que o usuário gostou no passado. O processo básico da recomendação baseada em conteúdo consiste em fazer correspondência entre os atributos do perfil do usuário e as características do item [13, 18, 57, 68, 71].

O paradigma de recomendação baseada em conteúdo analisa um conjunto de itens avaliados pelo usuário e constrói o perfil do usuário usando as características dos itens [13, 18, 57, 68, 71].

Uma arquitetura em alto nível do paradigma de recomendação baseada em conteúdo foi apresentada em [54](Figura 2.1). A arquitetura engloba três componentes principais:

- **Analisador de conteúdo** – este módulo pré-processa a informação. Quando a informação é não estruturada como texto e vídeo, é necessário extrair uma estrutura da informação. Os dados são analisados e suas características extraídas;
- **Gerenciador de perfis** – este módulo coleta informações sobre o usuário e constrói seu perfil;

- **Componente de filtragem** – este módulo usa o perfil do usuário para sugerir itens baseado na similaridade entre o perfil e as características dos itens. O resultado é uma lista de recomendações.

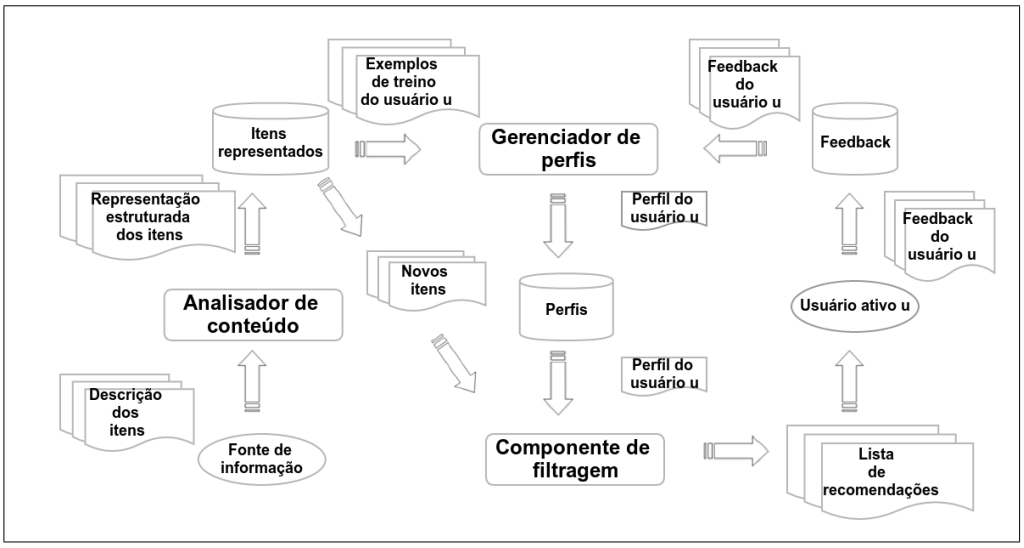

Figura 2.1: Ilustração da arquitetura padrão para recomendação baseada em conteúdo.

A filtragem colaborativa produz recomendação sem a necessidade de muita informação sobre usuários ou itens. Esse paradigma ganhou bastante popularidade depois da competição oferecida pela Netflix[1] com o modelo Fatoração de Matrizes (FM) [49].

A abordagem híbrida combina técnicas, aproveitando as vantagens e minimizando as desvantagens. Por exemplo, a filtragem colaborativa sofre do problema do novo item (ela não pode recomendar itens que não foram avaliados). Isso não limita a filtragem baseada em conteúdo, uma vez que a predição é feita usando os atributos do usuário e item [71].

2.1.1 Métricas de Avaliação

Para avaliar os vários algoritmos propostos na literatura, métricas específicas de validação são usadas. Exemplos destas métricas são aquelas baseadas em acurácia e técnicas diferenciadas de avaliação.

[1] `https://www.netflix.com/`

No processo de avaliação das abordagens de aprendizado de máquina e consequentemente dos sistemas de recomendação, os dados são divididos em duas amostras: treino e teste. A amostra de treino é utilizada para aprendizado dos parâmetros do modelo ou configurar o algoritmo utilizado no processo de análise (etapa denominada treino). Em contrapartida, a amostra de teste é utilizada para avaliar o modelo ou a configuração obtida na fase de treino, investigando a adequação das generalizações em relação aos dados ainda não vistos, ou seja, não utilizados durante o treino (etapa denominada teste) [2].

A acurácia mede, empiricamente, a proximidade dos itens na lista recomendada dos itens preferidos pelo usuário ou a distância entre a predição do sistema de recomendação da avaliação do usuário [42].

As métricas de avaliação baseadas em acurácia são as mais comuns [38, 42], dentre elas têm-se: métricas para avaliar a predição, a seleção de bons itens e a utilidade. A predição se refere a especular a nota do usuário para o item, enquanto que a seleção de bons itens se refere à escolha de um conjunto de itens que o usuário tem interesse; em contrapartida a utilidade avalia a lista recomendada pelos itens nela presentes e a ordem deles.

Para avaliação da predição de item, técnicas comuns são o Erro Quadrado Médio (EQM) [38](Equação 2.1),

$$EQM = \frac{\sum_{(u,i)}(r_{u,i} - \hat{r}_{u,i})^2}{N}, \tag{2.1}$$

onde u é o usuário e i é o item no conjunto de dados, $r_{u,i}$ é a avaliação do usuário, $\hat{r}_{u,i}$ é a predição feita e N é o total de itens avaliados no conjunto de teste. Ou variantes como a Raiz do Error Quadrado Médio (REQM) (Equação 2.2),

$$REQM = \sqrt{\frac{\sum_{(u,i)}(r_{u,i} - \hat{r}_{u,i})^2}{N}}, \tag{2.2}$$

o Erro Médio Absoluto (EMA) (Equação 2.3),

$$EMA = \frac{\sum_{(u,i)} |r_{u,i} - \hat{r}_{u,i}|}{N}, \tag{2.3}$$

ou o Erro Médio Absoluto Normalizado (EAMN) [38] (Equação 2.4).

$$EAMN = \sqrt{\frac{\sum_{(u,i)} |r_{u,i} - \hat{r}_{u,i}|}{N}}. \tag{2.4}$$

A diferença entre EMQ e EMA é que o primeiro penaliza mais erros maiores.

Em alguns ambientes o interesse é encontrar bons itens, nesse contexto, usualmente, consideram-se indicativos binários: o item é bom (1) ou não (0) [38]. As métricas de avaliação mais comuns para esse tipo de recomendação são Precisão (Equação 2.5) e *Recall* (Equação 2.2). A tarefa de prever bons itens é a mais comum em sistemas de recomendação e são aplicadas na indústria [38], como a Amazon[2] e Netflix.

$$P = \frac{\#positivos}{\#positivos + \#falsos\_positivos} \tag{2.5}$$

$$R = \frac{\#positivos}{(\#positivos + \#falsos\_negativos)} \tag{2.6}$$

$$FM = \frac{2 \times P \times R}{(P + R)} \tag{2.7}$$

A semântica dos componentes das equações é dada pela Tabela 2.1

Tabela 2.1: Matriz com a classificação dos possíveis resultados para a recomendação de um item para um usuário [38]

	Recomendados	**Não Recomendados**
Preferidos	Positivos	Falsos Positivos
Não Preferidos	Negativos	Falsos Negativos

Métricas foram desenvolvidas que combinam a Precisão e o *Recall* em uma única métrica. Dentre delas, F-Measure (Equação 2.7) e Área sobre a Curva [38, 42].

As métricas de avaliação para recomendação baseadas em utilidade (também denominada medida de *ranking*) consideram que o usuário percorrerá toda a lista recomendada para encontrar o item desejado [38], por isso os itens são penalizados pela ordem em que aparecem na lista.

Como exemplo de métricas baseadas em utilidade, tem-se DCG (do inglês *Discounted cumulative Gain* - Equação 2.8 [17]) e nDCG (do inglês *normalized Discounted Cumulative Gain*).

$$DCG@k = \sum_{u \epsilon U} \frac{2^{Rel_{ui}} - 1}{\log_2 i + 1} \tag{2.8}$$

[2] `http://www.amazon.com/`

Onde Rel_{ui} é o indicativo de relevância do usuário para o item na posição i.

Métricas diferenciadas de recomendação avaliam os itens recomendados em relação a aspectos além da acurácia. Como exemplos dessas técnicas, têm-se: métricas baseadas na diversidade dos itens recomendados, na quantidade de itens novos e na surpresa que os itens oferecem para o usuário. Como o interesse do trabalho é a acurácia, voltou-se a atenção para essas métricas.

2.2 Processamento de Imagem e Vídeo

Nesta seção é descrito o processamento de imagem e vídeo, mostrando o processo de extração de informação e suas aplicações, tais como reconhecimento, detecção e classificação de objetos.

A extração de informação é uma etapa do processo de aprendizagem de máquina que consiste em descrever a informação em um nível representativo [11]. O processo de extração de características da imagem é composto por três etapas: detecção de pontos de interesse, descrição de pontos de interesse e descoberta de correspondência entre pontos de interesse. A primeira etapa se refere ao processo de encontrar características salientes, pontos que atraem a atenção do sistema visual humano (SVH). A segunda se refere ao processo de descrever os pontos anteriormente extraídos de forma que eles sejam invariantes a deformações como rotação, iluminação e escala. A terceira se refere a encontrar correspondência entre descritores de interesse.

2.2.1 Representação de Imagem Digital

Uma imagem pode ser representada como uma função $f(x, y)$ no espaço bidimensional em que cada coordenada (x, y) se refere à intensidade do pixel na localização [29, 36]. Em se tratando de uma imagem colorida no sistema RGB, têm-se três canais, cada um representado por uma imagem no sistema de cores [29, 36]. Na Equação 2.9 pode ser vista uma representação da imagem com dimensões $M \times N$.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(0,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1) & \cdots & f(M-1,N-1) \end{bmatrix} \tag{2.9}$$

2.2.2 Detecção de Pontos de Interesse

Antes de entrar em detalhes, são descritas as operações que podem ser aplicados às imagens [29, 82], são elas:

- **Operação local (OL)** - consiste no processo em que o pixel na posição (x, y) da imagem de saída é afetado apenas pelo pixel (x, y) da imagem de entrada. A imagem de entrada é à matriz em que a operação é realizada, enquanto que a imagem de saída se refere à matriz resultante do processo;

- **Operação baseada em vizinhança (OBV)** - em que o pixel (x, y) da imagem de saída é afetada pelos vizinhos do ponto (x, y) da imagem de entrada (Figure 2.2 [29]). Como exemplo de operações baseadas em vizinhança, têm-se ajuste de contraste, de iluminação, suavização de imagens e a redução de ruídos.

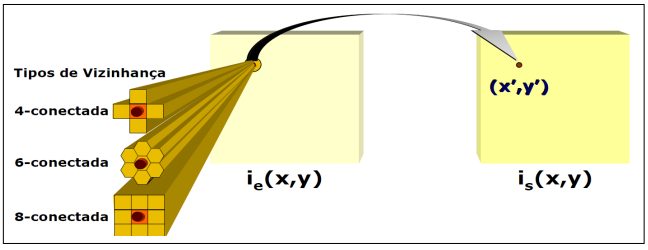

Figura 2.2: Ilustração de uma operação baseada em vizinhança sobre a imagem.

A detecção de pontos de interesse consiste na aplicação de operações baseadas em vizinhança na imagem de entrada. Efetuada pela aplicação de um filtro específico, uma matriz de dimensão reduzida $K \times K$, deslizado por cada pixel da imagem. Na matrizes 2.10 [53] e 2.11 [53] pode ser visto o filtro clássico de Sobel nas direções x e y.

$$G(x) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2.10}$$

$$G(y) = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{2.11}$$

O processo de deslizar o filtro sobre a imagem é denominado de convolução (Equação 2.12 [82]).

$$g(x,y) = \sum_{x,y} f(x,y) * h(x,y) \tag{2.12}$$

Onde, $g(x,y)$ é o pixel da imagem de saída, $f(x,y)$ corresponde ao pixel da imagem de entrada, $h(x,y)$ é o filtro e * representa a operação de convolução. Cada tipo de filtro possui características específicas, por exemplo detecção de bordas, linhas, círculos [82], etc.

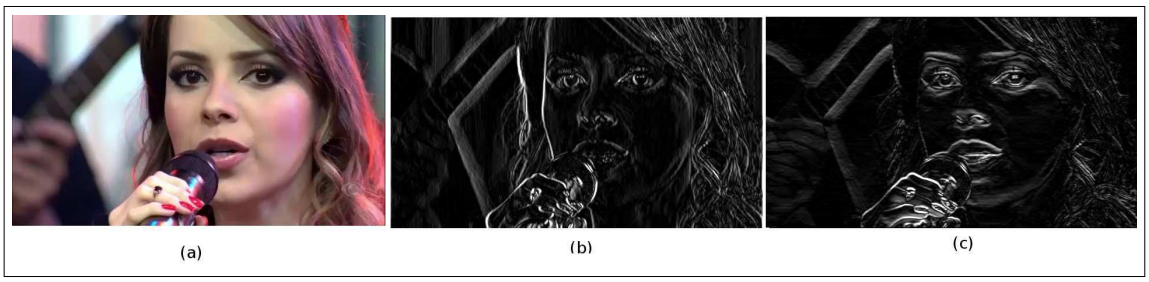

Figura 2.3: Ilustração de uma abordagem simples para extração de característica da imagem por meio de operadores gradientes pela convolução da imagem aplicando um filtro de Sobel. (a) imagem original, (b) gradiente da imagem na direção x e (c) gradiente da imagem na direção y. Note como as saliências são bem destacadas. O gradiente na direção x destaca pontos na horizontal e o gradiente na direção y na direção vertical.

Os operadores gradientes representam uma das operações mais básicas e importantes que podem ser aplicadas à imagem [15]. Um dos operadores mais usados é o filtro de Sobel [77] usado para detecção de bordas (Figura 2.3).

Diferentes das bordas, os pontos de interesse são localizações específicas na imagem, tais como picos de montanhas, padrões específicos na neve, esquina de uma construção e assim por diante (Figura 2.4 [82]) [82].

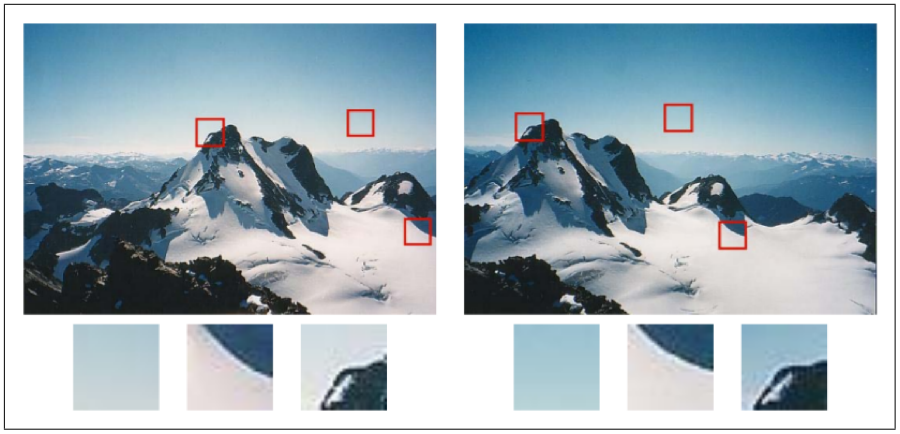

Figura 2.4: Ilustração dos pontos de interesse.

Várias abordagens para detecção de pontos de interesse foram propostas [82]. Uma das mais usadas é o *Scale Invariant Feature Transform* (SIFT [55]) [37].

Além dos pontos de interesse, outras estruturas podem ser usadas para representação da imagem de interesse, tais como bordas e linhas [82].

Detector de Pontos de Interesse SIFT

Os objetos do mundo real se apresentam das mais diversas formas e são sujeitos a deformações como mudança de escala, orientação e iluminação.

Por isso, os detectores e descritores de características necessitam ser invariantes a tais modificações [37, 82].

SIFT é um detector e descritor de característica bastante aplicado em problemas de visão computacional. Nesta seção, o SIFT é apresentado como detector de características, em seção posterior descreve-se esse método como descritor de características.

O detector é baseado em um modelo em cascata que reduz o tempo do processo. O processo é composto por um conjunto de etapas. A primeira etapa consiste na convolução da imagem com diferentes filtros Gaussianos $G(x, y, \alpha)$ (Equação 2.13 [55]) apenas variando o desvio padrão α. Nesse processo, a imagem é sucessivamente suavizada por um conjunto de

filtros Gaussianos incrementados por uma constante k. Em seguida, é computada a diferença entre duas suavizações sucessivas em um processo denominado Diferença de Gaussianas (do inglês *Difference of Gauss* (DOG)) (Equação 2.14 [82]). Por último, a resolução da imagem é reduzida por um fator de dois, selecionando o segundo pixel em cada linha e coluna e o processo se repete (Figura 2.5 [55]).

$$G(x, y, \alpha) = \frac{1}{2\pi\alpha^2} e^{-\frac{x^2+y^2}{2\alpha^2}} \tag{2.13}$$

$$D(x, y, \alpha) = G(x, y, k\alpha) - G(x, y, \alpha) \tag{2.14}$$

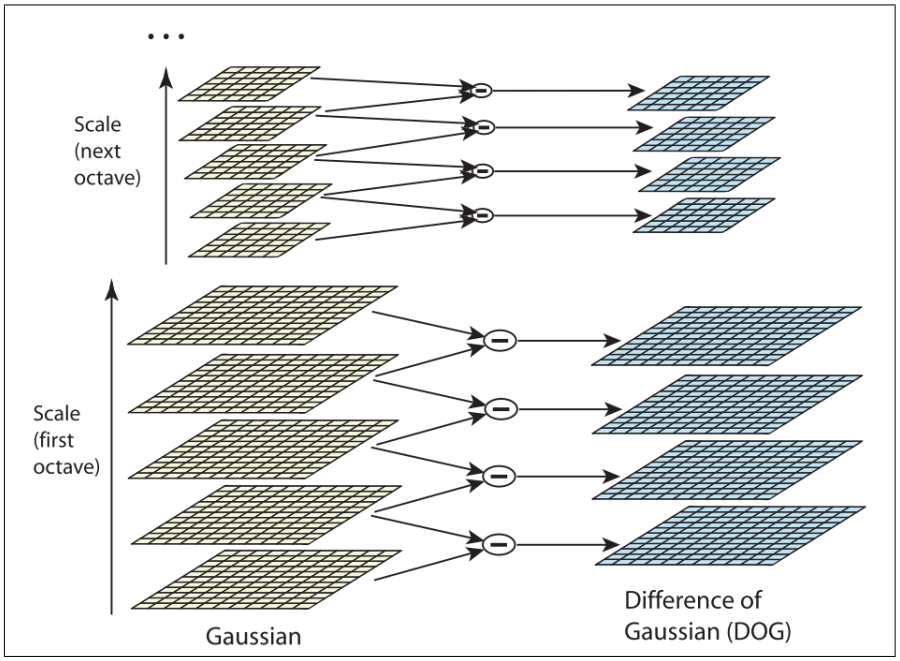

Figura 2.5: Ilustração do processo de detecção de pontos de interesse na abordagem SIFT.

O ponto de interesse é computado por meio da determinação do máximo e mínimo da diferença entre o pixel de interesse com os 26 vizinhos adjacentes, 9 do DoG acima do pixel, 9 do DoG abaixo do pixel e 8 no próprio DoG (Figura 2.6 [55]). O ponto de interesse é selecionado se é maior ou menor que todos os seus vizinhos.

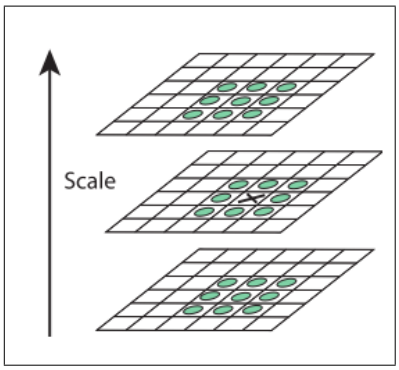

Figura 2.6: Ilustração da detecção de um ponto de interesse na abordagem SIFT.

2.2.3 Descritores de Características

Os descritores de características são representações dos pontos de interesse e são projetados para serem invariantes a mudanças como iluminação, escala, dentre outras [53, 82].

Como descreve Laganiere [53], descritores de características são usualmente vetores N-dimensionais que descrevem um ponto de interesse, idealmente de maneira que seja invariante à iluminação e pequenas deformações de perspectivas.

Dada uma imagem, o resultado de um descritor de característica é uma matriz que contém número de linhas correspondente aos pontos de interesse encontrados. Cada linha é um vetor N-dimensional, no caso do descritor SIFT N igual a 128. Esse vetor, usualmente, é caracterizado pelo padrão de intensidade ao redor do ponto de interesse. Quanto mais similares os pontos de interesse, mais próximos os vetores [53, 82].

Para demonstrar os descritores de características, a abordagem SIFT é exemplificada.

Descritor de Características SIFT

O descritor de característica SIFT é um vetor de dimensão 128 computado pelo gradiente da região ao redor do ponto de interesse. O processo de extração dos descritores de interesse é composto por um conjunto de etapas. Primeiro uma região de dimensão 16×16 é computada ao redor do ponto de interesse. Segundo o gradiente da região é computado e suavizado por uma janela Gaussiana (círculo da Figura 2.7 [55] à esquerda). A região é sumarizada em sub-regiões de dimensão 4×4 (parte direita da figura). A imagem mostra uma região 8×8 e sub-regiões de dimensão 2×2. Porém na abordagem real, o vetor possui respectivamente

dimensões 16×16 e 4×4.

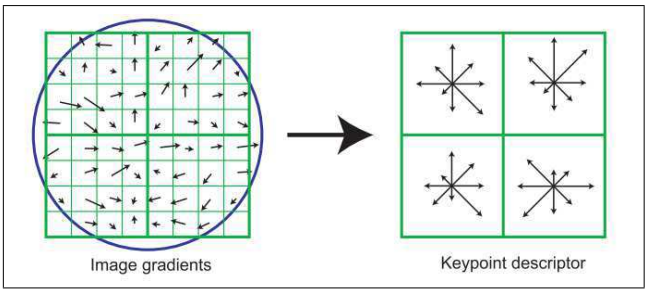

Figura 2.7: Ilustração do processo de extração de descritores de características na abordagem SIFT.

O descritor de interesse é computado pela magnitude e direção dos pontos de interesse nas sub-regiões. Cada região é sumarizada em um histograma de tamanho 8. Como se têm $4 \times 4 \times 8 = 128$, esse é o tamanho do descritor de características [37, 55].

2.2.4 Correspondências entre Descritores de Características

Após a extração dos descritores, a próxima etapa é estabelecer similaridade, também denominada de correspondência entre os descritores de características.

Szeliski [82] divide o processo de correspondência entre imagens em estratégias de similaridade e algoritmos de detecção de correspondência.

Assumindo que os descritores de características foram extraídos e que possuem uma representação vetorial N-dimensional, uma estratégia de similaridade simples é calcular a distância Euclidiana entre dois descritores de características e comparar o resultado com um limiar.

Após a definição da estratégia de similaridade, é necessário calcular quais pontos de interesse estão relacionados. Uma estratégia simples é por força bruta, comparando par a par todos os pontos de interesse (Figura 2.8 [82]).

2.2.5 Detecção de Mudanças de Câmera

A detecção de mudança de câmera consiste em determinar os limites de fim da captura realizada por uma câmera e o começo da captura de outra. Essa técnica produz bons resultados

Figura 2.8: Ilustração do processo de detecção de correspondência entre os pontos de interesse entre duas imagens, direita e esquerda, destacando os pontos de interesse correspondentes.

na detecção de padrões [90]. A mudança de câmera é detectada pelo cálculo da similaridade entre dois quadros consecutivos.

Sinal Ruído de Pico

Razão Pico Sinal por Ruído PSNR (do inglês *Peak Signal-to-Noise Ratio*) é uma abordagem simples e leve usada para computar a similaridade entre imagens [84]. Essa técnica é, usualmente, empregada como métrica para teste de qualidade em imagens e vídeo [70]. A PSNR é calculada dividindo o pico de sinal útil pelo Erro Quadrado Médio (EQM) da diferença entre dois quadros [70] (Equação 2.15).

$$EQM(i,j) = \frac{1}{MN}\sum_{x=0}^{M}\sum_{y=0}^{N}\Big(f(x,y) - g(x,y)\Big)^2 \tag{2.15}$$

$$PSNR = 10 \times \log_{10}\left(\frac{MAX_I^2}{EQM}\right) \tag{2.16}$$

Onde MAX_I é o valor máximo que o pixel pode assumir, usualmente 255 e EQM é a diferença pixel a pixel dos quadros $f(x,y)$ e $g(x,y)$ (Equação 2.16). Se a imagem for colorida o EQM é a média de cada canal R, G e B.

Similaridade Estruturada

O objetivo da técnica Similaridade Estruturada SSIM (do inglês *Structured Similarity*), assim como o PSNR é computar a similaridade entre imagens. Diferente da PSNR, a SSIM busca imitar o sistema visual humano [84].

A Similaridade Estruturada é baseada em três propriedades estatísticas da imagem: iluminação, contraste e estrutura. Esses componentes são combinados para computar a similaridade global [84].

A iluminação de uma imagem é calculada pela média da intensidade dos pixels da imagem (Equação 2.17).

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.17}$$

A iluminação entre imagens é uma função $l(x, y)$ de similaridade entre as imagens x e y, com intensidade média, respectivamente, μ_x e μ_y (Equação 2.18)

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{2.18}$$

Onde C_1 é adicionado à similaridade para evitar indefinição matemática quando $\mu_x^2 + \mu_y^2$ é próximo a zero [84].

Para o cálculo do contraste é usado o desvio padrão dos pixels da imagem (Equação 2.20).

A similaridade de contraste entre duas imagens x e y é dada pela função $c(x, y)$ (Equação 2.19).

$$c(x, y) = \frac{2\alpha_x\alpha_y + C_2}{\alpha_x^2 + \alpha_y^2 + C_2} \tag{2.19}$$

$$\alpha_x = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2} \tag{2.20}$$

A similaridade estrutural é calculada usando a normalização dos pixels da imagem $(x_{i,j} - \mu_x)/\alpha_x)$. A similaridade estrutural entre duas imagens x e y é uma função $s(x, y)$ (Equação 2.21) calculada mediante a correlação entre os pixels normalizados das imagens.

$$s(x, y) = \frac{\alpha_{xy} + C_3}{\alpha_x\alpha_y + C_3} \tag{2.21}$$

Onde α_{xy} é dado pela co-ocorrência entre os pixels (Equação 2.22).

$$\alpha_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{2.22}$$

Depois de computados os componentes do modelo (iluminação, contraste e estrutura), eles são combinados para calcular a similaridade entre as imagens x e y. A similaridade é efetuada por meio da multiplicação de cada componente parametrizado pelos fatores α, β e λ (Equação 2.23).

$$SSIM = [l(x,y)]^{\alpha} \times [c(x,y)]^{\beta} \times [s(x,y)]^{\lambda} \tag{2.23}$$

2.2.6 Abordagem para Classificação Bag of Keypoints

Umas das tarefas mais complexas da visão computacional é o entendimento e reconhecimento de imagens [82], pois os objetos podem aparecer em diferentes formas, tamanhos e posições tornando difícil o desenvolvimento de algoritmos que capturem essas variações [82].

No reconhecimento de imagens um desafio é a categorização dos objetos nelas contidos. Vários algoritmos foram desenvolvidos e um algoritmo simples que ganhou destaque foi a abordagem *Bag of Keypoints* [82]. A abordagem *Bag of Keypoints* é útil em várias atividades da visão computacional [24].

O processo de aquisição da *Bag of Keypoints* consiste nas seguintes etapas [24]:

1. **Detecção de pontos de interesse** – operação descrita na seção 2.2.2;
2. **Extração de descritores de interesse** – operação descrita na seção 2.2.3;
3. **Criação do vocabulário** - corresponde aos centróides (descritores de características) após a aplicação de uma técnica de agrupamento como o *K-means*. Analogicamente à recuperação de texto, cada ponto de interesse corresponde a uma palavra do vocabulário;
4. **Construção da *Bag of keypoints*** - corresponde ao histograma de centróides presentes na imagem.

5. **Processamento do ponto de interesse** – corresponde ao uso dos histogramas extraídos para as aplicações apropriadas, tais como detecção, reconhecimento, classificação, etc.

Estudos foram realizados e demonstraram bons resultados para a abordagem [24].

2.3 Processamento de Texto

Em sistemas de recuperação de informação e sistemas de recomendação baseada em conteúdo uma abordagem comum para representar o item é o uso de vetores de palavras chaves, ou *bag of words*, em que cada posição do vetor corresponde a um termo no vocabulário [10, 33]. Sendo a posição preenchida, usualmente, com a frequência da palavra no descritivo textual do item, ou o TF-IDF (do inglês *Term Frequency–Inverse Document Frequency*) do termo [54].

Outra abordagem é tratar o item como um grafo e calcular a similaridade entre eles por meio da probabilidade de interação entre as palavras [63]. Uma abordagem comum é mediante banco de dados ou dicionários [25, 46]. Um dicionário bastante empregado é o *WordNet* [34, 64].

2.3.1 WordNet

No *WordNet* várias classes de palavras estão inclusas, tais como verbos, substantivos, adjetivos, pronomes, e relações são estabelecidas entre os membros de cada classe. Nesse dicionário uma palavra é representada como uma dupla $w = (f, s)$, em que f é a cadeia de caracteres representativa do item e s os sentidos da palavra dentro do conjunto de significados [64]. Na Tabela 2.2 podem ser vistas estatísticas da quantidade de palavras e sentidos codificados no *WordNet* [64].

Tabela 2.2: Estatísticas do *WordNet*

Quantidade de Palavras	**Sentidos**	**Relações de Sentido** $w(f, s)$
118,000	90,000	166,000

Várias relações entre palavras estão codificadas no *WordNet* [64], tais como sinônimo, antônimo, etc.

2.3.2 *Stemming* e Lematização

Em algumas aplicações é necessária a redução da palavra a sua forma básica [59]. Isso decorre por questões gramaticais, pois a palavra pode assumir diferentes derivações dependendo do contexto onde é empregada; algumas recebem desinência de plural, outras permanecem no singular e assim por diante.

Dois processos podem ser utilizados para redução da palavra a sua forma básica, são eles: *stemming* e lematização [59]. *Stemming* refere-se a um conjunto de processos que removem os afixos das palavras, em contrapartida a lematização usa um vocabulário definido e efetua uma análise morfológica da palavra, reduzindo-a a sua forma primitiva - a maneira que ela é apresentada no dicionário-, também denominado lema [59].

Além dos dois processos vistos, outra abordagem utilizada quando se trabalha com texto é a remoção de palavras não significativas como preposição, artigo e conjunção. Pois, para que a palavra tenha um valor discriminativo é necessário que ela possua uma semântica associada.

2.4 Modelos Multimodais

Os modelos multimodais se referem a representações de alto nível com diferentes tipos de dados (texto, áudio e vídeo). Esses modelos são aplicados em várias atividades como a detecção de eventos, conceitos e análise visual.

A fusão de multimodalidades possuem benefícios, porém com a adição de custo e complexidade no processo de análise [7]. Atrey et al. [7] realizou uma pesquisa sistemática desses modelos em ambiente multimídia e destacou aspectos relacionados a sua construção, são eles:

- **Nível de fusão** - aspecto relacionado ao momento da fusão. Duas abordagens são possíveis: fusão em nível de características e fusão em nível de decisão. A fusão em nível de características ou fusão prévia engloba as diferentes modalidades em um modelo único para posteriormente calcular a relevância global. Por outro lado, a fusão em nível de decisão computa a relevância em relação as diferentes modalidades e, em seguida, computa a relevância global usando os resultados obtidos. Uma abordagem

híbrida também é possível;

- **Como fundir** - que abordagens são utilizadas para realizar a fusão. Dentre essas técnicas se destacam abordagens baseadas em regras, classificadores e métodos baseados na estimação de parâmetros;
- **Quando fundir** - momento em que a fusão deve ser realizada;
- **O que fundir** - níveis de características que devem ser fundidos, tais como texto, áudio e aspectos visuais.

2.5 Modelos de Markov e Distribuição de Gibbs

Há normalmente dois tipos de modelos de probabilidade em grafos, aqueles que operam em grafos diretos direcionados e aqueles que operam em grafos direcionados. Os modelos em grafos diretos correspondem aos modelos bayesianos, entretanto o trabalho não aplica esse tipo de rede; portanto, os mesmos não serão discutidos. Porém, para os leitores interessados consultem Kollen e Friedman [48].

Os modelos em grafos não direcionados formam uma rede markoviana (modelo de Markov ou *Markov Random Field*) em que os nós são variáveis aleatórias do problema e as arestas são as afinidades entre os nós. Por exemplo, considere a Tabela 2.3 apresentada em Kollen e Friedman [48], em que estudantes trabalham juntos para responder a uma atividade escolar. Para o exemplo, os nós são os estudantes e as arestas são as afinidades entre eles.

Tabela 2.3: Rede de Markov com os valores das interações entre os nós na rede para o exemplo da atividade escolar.

$\phi(A,B)$			$\phi(B,C)$			$\phi(C,D)$			$\phi(D,A)$		
a^0	b^0	30	b^0	c^0	100	c^0	d^0	1	d^0	a^0	100
a^0	b^1	5	b^0	c^1	1	c^0	d^1	100	d^0	a^1	1
a^1	b^0	1	b^1	c^0	1	c^1	d^0	100	d^1	a^0	1
a^1	b^1	10	b^1	c^1	100	c^1	d^1	1	d^1	a^1	100
(a)			(b)			(c)			(d)		

Como os relacionamentos são indiretos, é apropriado utilizar uma função $\phi(D)$ que descreva a afinidade entre os nós do grafo [48]. Essa função é uma parte importante do modelo e recebe o nome de função potencial definida sobre um fator da forma seguinte [48]:

> Seja D um conjunto de variáveis aleatórias, define-se um fator como sendo uma função em relação ao $Val(D) \in \mathbb{R}$. Um fator é positivo se todas as entradas são positivas. O conjunto de variáveis é denominado escopo do fator descrito por $Escopo[\phi]$.

Apesar de ser permitido um fator ser negativo, usualmente apenas os positivos são considerados.

Note pela Tabela 2.3(a) que o fator $\phi(A, B)$ possui afinidade maior quando ambos concordam, seja positiva ou negativamente. De maneira similar é modelada a interação entre os demais fatores $\phi(B, C)$, $\phi(C, D)$ e $\phi(D, A)$.

A distribuição conjunta de probabilidade $P(A, B, C, D)$, ou seja os parâmetros do modelo, é calculada pelo produto da probabilidade de interação entre os fatores [48] (Equação 2.24).

$$P(A, B, C, D) = \frac{1}{Z} \prod_{i \in \phi} \phi_i \tag{2.24}$$

Onde Z é denominado de função de partição e corresponde à soma dos produtos dos fatores (Equação 2.25).

$$Z = \sum_{i \in \phi} \prod_{i \in \phi} \phi_i \tag{2.25}$$

Para entender melhor considere a Tabela 2.4 [48] que contém o produto dos fatores individuais para o exemplo dado. Cada estudante possui duas possibilidades, interagir (0) ou não com o colega (1). Cada interação do grafo possui um peso associado, e o produto desses pesos corresponde à probabilidade do grafo. Por exemplo, considerando a primeira linha da Tabela 2.4 e os valores presentes na Tabela 2.3, tem-se uma configuração específica do grafo em que o estudante A não interage com o estudante B, $\phi(a^0, b^0) = 30$, o estudante B não interage com estudante C, $\phi(b^0, c^0) = 100$, o estudante C não interage com o estudante D, $\phi(c^0, d^0) = 1$, e o estudante D não interage com o estudante A, $\phi(d^0, a^0) = 100$. Multiplicando os fatores par a par, tem-se como resultado 300000 que define a probabilidade não

normalizada da configuração. Fazendo isso para todas as possibilidades e somando os resultados, obtém-se a função de partição. Dividindo o produto da probabilidade não normalizada pela função de partição, obtém-se a probabilidade normalizada dessa configuração específica do grafo.

Tabela 2.4: Produto das probabilidades individuais. Mostrando o cálculo da função de partição e as probabilidades não normalizadas. A distribuição de probabilidade é calculada dividindo a probabilidade não normalizada pela função de partição.

Fatores				**Probabilidade Não Normalizada**	**Probabilidade Normalizada**
a^0	b^0	c^0	d^0	300000	0.04
a^0	b^0	c^0	d^1	300000	0.04
a^0	b^0	c^1	d^0	300000	0.04
a^0	b^0	c^1	d^1	30	4.1×10^{-6}
a^0	b^1	c^0	d^0	500	6.9×10^{-5}
a^0	b^1	c^0	d^1	500	6.9×10^{-5}
a^0	b^1	c^1	d^0	5000000	0.69
a^0	b^1	c^1	d^1	500	6.9×10^{-5}
a^1	b^0	c^0	d^0	100	1.4×10^{-5}
a^1	b^0	c^0	d^1	1000000	0.14
a^1	b^0	c^1	d^0	100	1.4×10^{-5}
a^1	b^0	c^1	d^1	100	1.4×10^{-5}
a^1	b^1	c^0	d^0	10	1.4×10^{-6}
a^1	b^1	c^0	d^1	100000	0.014
a^1	b^1	c^1	d^0	100000	0.014
a^1	b^1	c^1	d^1	100000	0.014
				Z= 7201840	

A parametrização do modelo de Markov, como dito anteriormente, é descrita em termos da interação entre fatores no grafo, por isso a distribuição de probabilidade segue uma distribuição de Gibbs [48], como descrito a seguir:

> Uma distribuição $P\phi$ é dita de Gibbs parametrizada pelo conjunto de fatores $\Phi = \{\phi_1, \phi_2, \ldots, \phi_{|\Phi|}\}$ se é definida da forma seguinte:

$$P\Phi(X_1, X_2, \ldots, X_\Phi) = \frac{1}{Z}\overline{P}_{\,\Phi}(X_1, X_2, \ldots, X_n),$$

onde,

$\overline{P}_{\,\Phi}(X_1, X_2, \ldots, X_n) = \prod_{i\in\Phi} \phi(D_i)$

é uma medida não normalizada e

$Z = \sum_{i\in\phi} \overline{P}_{\,\Phi}(X_1, X_2, \ldots, X_n)$

é uma constante de normalização denominada função de partição.

Os fatores na distribuição de Gibbs correspondem aos cliques no grafo. Devido aos cliques serem subgrafos dos cliques máximos, é necessário apenas usar os cliques máximos. Isso reduz a quantidade de parâmetros que necessitam ser aprendidos no uso de uma distribuição de Gibbs [48].

Um clique em um grafo consiste no conjunto em que todos os vértices estão conectados entre si [23, 58, 81]. O clique maximal é o conjunto máximo de vértices conectados. O problema de encontrar os cliques no gráfico é um problema NP-completo [23].

2.6 Arquitetura da TV digital

A TV digital diferentemente da TV tradicional oferece ao usuário várias funcionalidades, dentre elas [8]:

- **Suporte a TV interativa** – a TV digital é bidirecional permitindo que o usuário interaja com o sistema;
- **Navegação temporal** - a TV digital permite navegação temporal sobre os programas como pausar, continuar e voltar;
- **Personalização** - a TV digital permite personalizar a experiência como espectador, oferecendo suporte a decidir o que assistir e quando assistir.

Na Figura 2.9 [8] pode ser vista a arquitetura usada em TV digital: O *data center* (também conhecido com *head end*), recebe conteúdos oriundos de várias fontes, dentre elas, portadores terrestres, satélite e a cabo. Uma vez recebido, um número de componentes de hardware é usado para entregar o conteúdo na rede. O *set-top-box* é um componente eletrônico que conecta rede e televisores. Ele é responsável por receber os pacotes e mostrar na

TV do usuário. O controle remoto fornece acesso a outros recursos do *set-top-box* dentre eles o guia de programação [8].

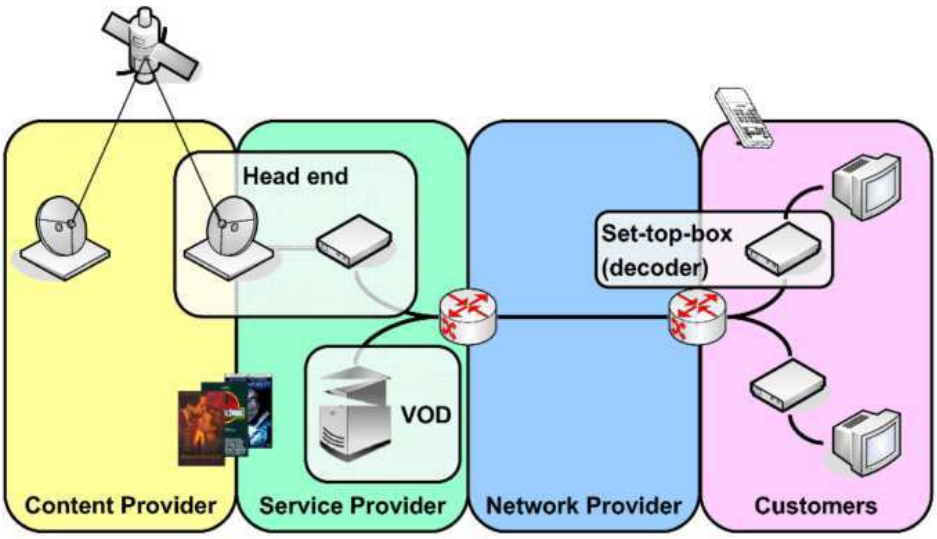

Figura 2.9: Ilustração da arquitetura para TV digital.

O EPG fornece várias informações sobre os programas, tais como título, sinopse, categorias e assim por diante [92].

Capítulo 3

Arquitetura Proposta

Neste capítulo é apresentada a arquitetura proposta no trabalho, definindo técnicas para redução do conjunto de quadros do programa, abordagens para extração do conjunto de características visuais, técnicas para extração do conjunto de características textuais e modelos para representação do item e usuário para recomendação em termos de multimodalidades.

3.1 Visão Geral da Solução

O objetivo da arquitetura é encontrar programas que o usuário gostaria de assistir a partir dos programas assistidos por ele (Figura 3.1).

Para gerar recomendações para um usuário específico (1), o sistema de recomendação coleta informações do conjunto de programas assistidos pelo usuário (2) e também do conjunto de programas não assistidos por ele (3). Cada programa é composto por duas fontes de informações: o EPG e o vídeo do programa. Essas informações são utilizadas pelo Sistema de recomendação baseado em conteúdo para gerar a lista de programas recomendados para o usuário (4). Foi assumido que o vídeo e o EPG de cada programa estão disponíveis para que o Sistema de recomendação baseado em conteúdo possa realizar o seu trabalho.

O Sistema de recomendação baseado em conteúdo usa um conjunto de componentes internos para gerar a lista de recomendações (Figura 3.2). O primeiro processo é realizado pelo componente Filtragem de dados (1) (seção 3.3) que extrai informações do vídeo e do EPG de cada programa. Como o programa é composto de duas modalidades, a extração de dados é realizada em duas etapas: a primeira coleta informações do vídeo de cada programa

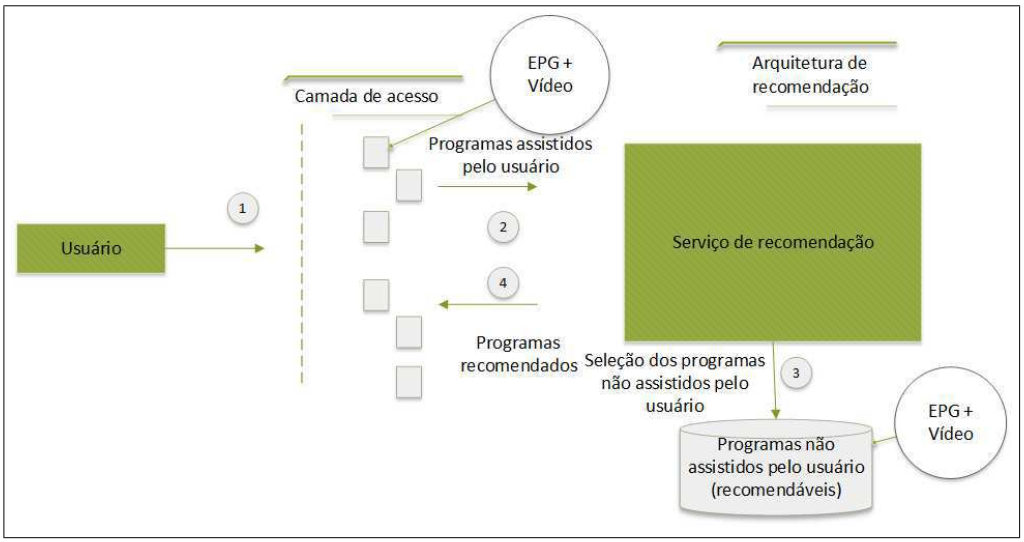

Figura 3.1: Representação simplificada do processo de recomendação objetivado no trabalho proposto. O sistema de recomendação usa o conjunto de programas assistidos pelo usuário e constrói um modelo de recomendação *offline* para gerar uma lista de programas que o telespectador gostaria de assistir. O programa corrente é adicionado ao perfil do usuário e um novo modelo é criado.

(1.1) (seção 3.3.1) e a segunda minera informações dos dados textuais de cada programa contido no EPG (1.2) (seção 3.3.2), tais como descrição e conjunto de categorias pertencente ao programa. As características visuais coletadas do vídeo correspondem aos descritores de interesse extraídos dos quadros mais relevantes do programa. A seleção dos quadros mais relevantes de cada programa é efetuada na etapa Redução de quadros (1.1.1). Dos quadros mais relevantes de cada programa são extraídos os descritores de interesse (1.1.2). As características textuais correspondem ao *stem* (raiz) da palavra e são extraídas da descrição de cada programa contido no EPG na etapa Extração de texto (1.2).

Com o conjunto de características multimodais (descritores de características e *stem*) de cada programa, a representação do programa é construída pelo componente Gerenciamento do programa (2) (seção 3.4). Após a construção da representação de cada programa, a representação do usuário é composta usando as representações multimodais dos programas assistidos por ele pelo componente Gerenciamento do usuário (3) (seção 3.5). O Recomendador baseado em conteúdo (4) (seção 3.6) utiliza a representação do programa e a representação do usuário para gerar a lista de itens recomendados para o usuário comparando a representação do usuário com a representação de cada programa. A lista de recomendação é retornada com o conjunto de programas recomendados mais similares à representação do usuário.

A arquitetura é composta pelos seguintes componentes:

- **Filtragem de dados (FD)** – o processo de extração de dados corresponde a identificar características úteis nos dados (por exemplo, a palavra "humor" é mais discriminativa do que o artigo "a"). Como a recomendação engloba mais de um tipo de dados, é necessária a extração para cada um deles. Trabalhando com vídeo e descrição, ao final desse processo dois conjuntos são gerados para cada programa no banco de dados, um contendo características textuais (*bag of words*) e o outro contendo características visuais (*bag of keypoints*);

- **Gerenciador de programa – (GP)** - na abordagem o programa é representado pelos dois conjuntos de dados extraídos: conjunto de características textuais e conjunto de características visuais;

- **Gerenciador de usuário – (GU)** - o perfil do usuário é basicamente a união dos conjuntos de características dos programas assistidos por ele;

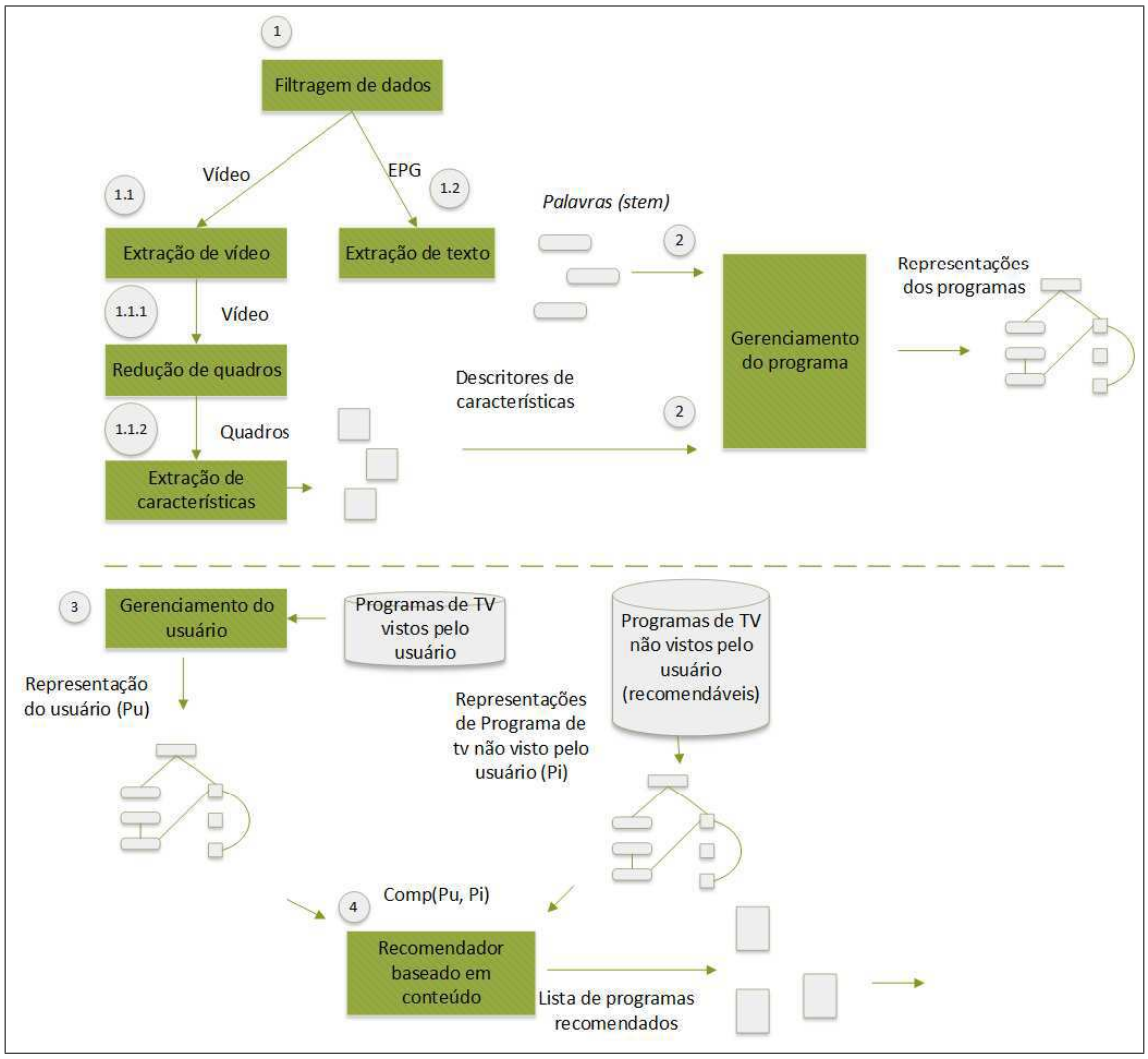

Figura 3.2: Ilustração de como ocorre o processo dentro do serviço de recomendação.

- **Recomendador baseado em conteúdo (RecBC)** - a recomendação baseada em conteúdo consiste em comparar o perfil do usuário com a representação do programa (processo também denominado de cálculo da similaridade entre conjuntos). A similaridade entre a representação do programa e o perfil do usuário é basicamente a intersecção entre o conjunto de características no perfil do usuário e o conjunto de características na representação do item.

3.2 Projeto da Arquitetura

O esboço em alto nível da arquitetura proposta pode ser visto na Figura 3.3. A arquitetura para sistemas de recomendação baseados em conteúdo inclui componentes para representação adequada do item, construção de perfis e estratégias de similaridade entre o conteúdo do item e o perfil do usuário [54] (seção 2.1). Além desses, a arquitetura proposta adiciona componentes para lidar com multimodalidades.

Nos trabalhos da literatura a coleta de dados no domínio de TV é feita no EPG e as características extraídas são textuais. Neste trabalho, além de uma fonte de dados textual há uma fonte de dados visual. Dessa forma, em alguns componentes, além dos módulos textuais é necessário acrescentar módulos visuais. O ganho com multimodalidades é que cada tipo de dado possui aspectos particulares de representação do item que não são capturados pelo uso de uma modalidade única. Como aponta a literatura e demonstrado neste estudo (Capítulo 4), o desempenho de usar ambos (texto e vídeo) é superior ao de usar apenas um tipo de dado [25, 26, 79]. Um exemplo dessa característica das multimodalidades foi apresentado em Yang et al. [89]: considerando um vídeo sobre uma praia, palavras relacionadas à praia podem ser “céu”, “areia”, “pessoa” e assim por diante. Essas palavras também podem estar relacionadas a vídeo irrelevante como deserto.

Os dados para extração de interesse correspondem a um conjunto de programas e da indicação do usuário do interesse pelo item (Tabela 3.1) da seguinte forma:

- **Usuário** - pessoa que avalia o programa;
- **Programa de TV**;
- **Avaliação do usuário para o item**.

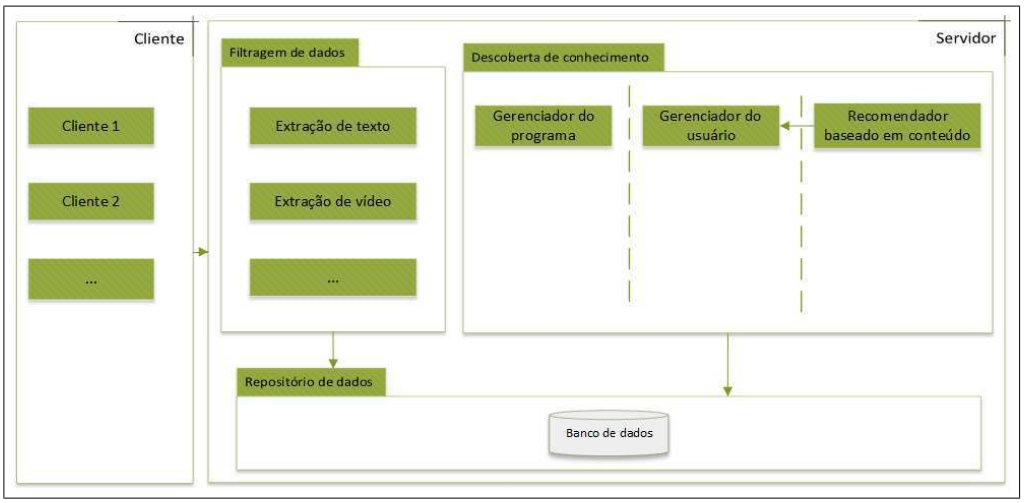

Figura 3.3: Ilustração da arquitetura para recomendação multimodal para TV Digital proposta no trabalho.

Tabela 3.1: Exemplificação dos dados utilizados para construção de recomendadores para TV Digital

Usuário/Item	**Programa 1**	**Programa 2**	...	**Programa N**
Usuário 1	5	?	...	2
Usuário 2	?	3	...	?
⋮	...	...	...	⋮
Usuário N	3	?	...	?

Os dados foram coletados usando questionários[1]. As pessoas indicaram quais programas assistiram e quantas vezes o fizeram no intervalo de uma semana. Os vídeos de cada programa foram coletados do *Vimeo* [43] e as descrições textuais são extraídas do EPG.

Ao final da pesquisa 42 pessoas responderam ao questionário e 95 programas de TV foram indicados, esses dados compõem a base de dados. As pessoas envolvidas no questionário são, majoritariamente, alunos do curso de Ciência da Computação. Na base estão inclusas pessoas de ambos os sexos com idades entre 20 e 35 anos.

Estatísticas sobre a base de dados utilizada com respeito ao número de programas no perfil dos usuários podem ser encontradas na Tabela 3.2.

[1]O formulário usado se encontra no link `http://goo.gl/0DHdop`

Tabela 3.2: Estatísticas sobre a base de dados utilizada.

Mínimo	Primeiro Quartil	Mediana	Média	3 Quartil	Máximo	Desvio Padrão
4.00	10.00	16.00	15.89	19.00	40.00	7.88

A abordagem de recomendação empregada é uma adaptação da abordagem de Cui et al. [25], porém qualquer abordagem de recomendação pode ser aplicada. Esse modelo foi escolhido por dois motivos. O primeiro é a possibilidade de recomendação tanto unimodal quanto multimodal. O segundo é que a abordagem foi aplicada com resultados positivos tanto para a recomendação quanto para recuperação de informação [25].

Antes de detalhar os componentes, é descrito o conceito de TV utilizado no trabalho. Os trabalhos anteriores, em geral, referem-se a IPTV - um ambiente em que o usuário pode assistir diferentes filmes e seriados - entretanto o conceito aplicado neste trabalho é mais abrangente incluindo programas com diferentes categorias, tais como jornal, seriados, jogos de futebol que possuem um conjunto de características visuais próprias que intuitivamente possui um valor representativo maior. Por exemplo, um programa jornalístico, usualmente, possui um conjunto característico de eventos, tais como entrevistas, documentários e reportagens, em contrapartida os programas esportivos trazem, torcida, jogadores e gol. Por uma análise dos histogramas de cada programa é possível discriminar as categorias de cada um deles como demonstra o experimento da seção B (Apêndice B).

A seguir são descritos os componentes da arquitetura em relação ao modelo usado.

3.3 Filtragem de Dados

Os trabalhos em TV usualmente empregam apenas características textuais no processo de Filtragem de dados [5, 12, 27, 41, 60, 65, 67, 76, 91, 92]. Como este trabalho usa multimodalidades (texto e vídeo) dois módulos são adicionados, um para extração de características textuais (Extração de texto) e um para extração de características visuais (Extração de vídeo) vistos na Figura 3.3.

Cada modalidade de dado é extraída de uma forma específica, por isso esta seção é dividida em duas partes: a primeira aborda a extração de características visuais a partir do vídeo do programa e a segunda a mineração de texto.

3.3.1 Extração de Vídeo

A extração de vídeo consiste em buscar uma representação visual para o item. Esse conjunto de características visuais, normalmente, é agrupado para construir o vocabulário de características visuais. As características visuais podem variar desde à intensidade dos pixels em uma região $K \times K$ da imagem [25], ou modelos mais abrangentes como as bordas e os pontos de interesse [26, 80] (seção 2.2.2).

Como o vídeo é formado com um grande volume de quadros, o custo computacional para processá-los é alto. Por isso, antes que as características visuais sejam extraídas, processos como a detecção de mudança de câmera podem ser aplicados para reduzir o número de quadros analisados. Dessa forma, o módulo de Extração de vídeo pode ser dividido em dois submódulos: Redutor de quadros e Extrator de características visuais (Figura 3.4).

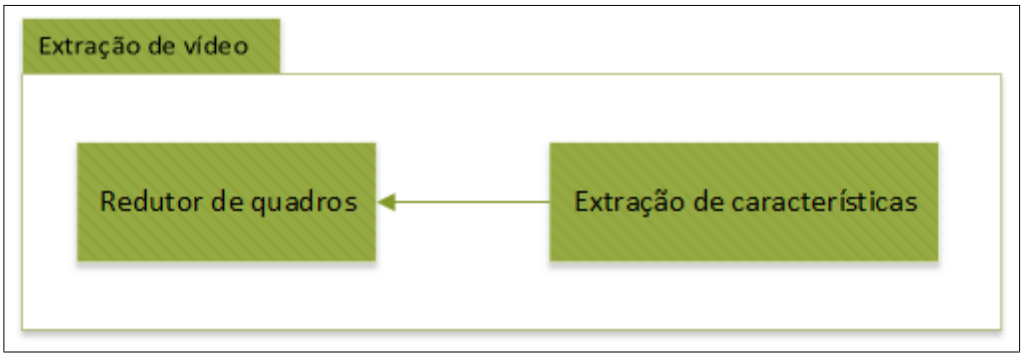

Figura 3.4: Ilustração do módulo de extração de vídeo.

O processo de interação entre os submódulos pode ser visto na Figura 3.5. Enquanto o vídeo é exibido, o quadro corrente é selecionado. Em seguida, o Redutor de quadros elege o quadro como significativo ou não dependendo das regras de negócio e requisitos da aplicação. Quando um quadro é identificado como significativo, o processo de extração de características se inicia. Caso o vídeo tenha chegado ao seu final, o processo termina, caso contrário outro quadro do vídeo é analisado.

A abordagem empregada no trabalho seleciona o conjunto de quadros representativos por meio da detecção de mudança de câmera (seção 2.2.5). A detecção de mudança de câmera consiste basicamente em comparar o quadro atual com o quadro seguinte. Se a diferença entre eles for maior que um limiar predefinido, uma mudança de câmera é detectada (Fi-

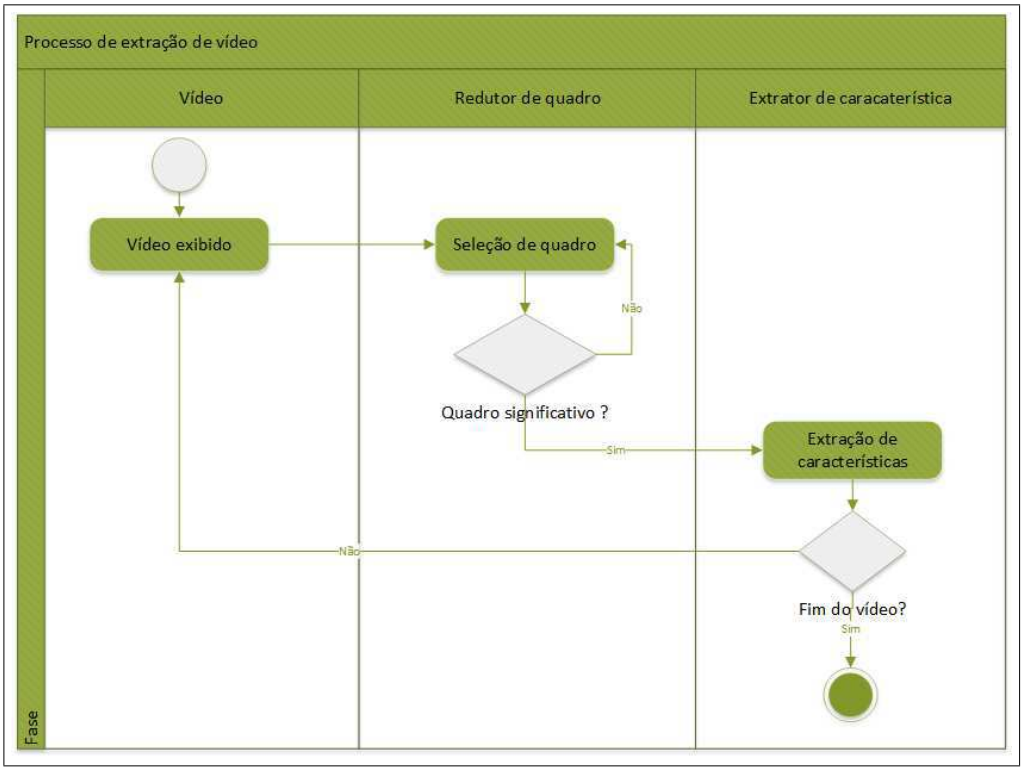

Figura 3.5: Ilustração do processo de extração de quadro no módulo de Extração de vídeo.

gura 3.6). A comparação entre dois quadros consecutivos é uma abordagem normalmente utilizada em visão computacional em processos como a detecção de movimento e determinação de fluxo ótico.

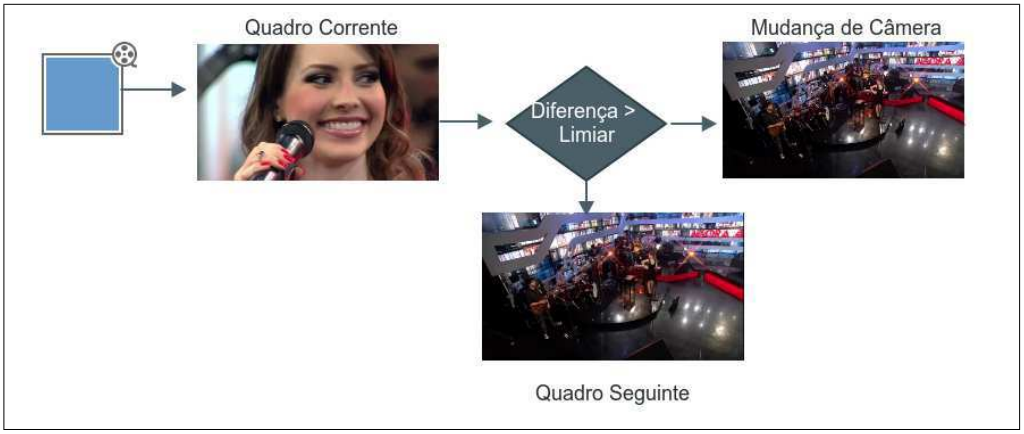

Figura 3.6: Ilustração da detecção de mudança de câmera.

Para a detecção de mudança de câmera duas técnicas foram aplicadas em conjunto, PSNR e SSIM (seção 2.2.5).

Um *tradeoff* existe entre o desempenho da detecção de mudança de câmera da técnica e tempo para processá-la. O PSNR é uma abordagem rápida e o SSIM uma abordagem acurada. Por isso, os dois algoritmos foram aplicados em conjunto em uma abordagem em cascata. O PSNR foi aplicado primeiro, por ser mais rápido, e o SSIM como desempate – quando a diferença entre os quadros é indicada como significativa pelo PSNR, o SSIM é usado para investigar diferenças mais sutis como descrito em [66]. Os limiares usados foram definidos por meio de um experimento (Apêndice A). São necessários quatro limiares, considerando vídeos coloridos, um para o PSNR e três para o SSIM (um para cada canal de cor). Por questões de simplicidade foi utilizado o mesmo limiar para cada canal de cor na técnica SSIM. Os limiares que mostraram melhores resultados no experimento estão presentes na Tabela 3.3.

Tabela 3.3: Limiares utilizados para detecção de mudanças de câmera aplicados no trabalho

PSRN	**SSIM**
22	65

O processo de extração de características visuais dos quadros identificados é realizado em cascata. Primeiro os pontos de interesse são extraídos das mudanças de câmeras usando a abordagem SIFT (seção 2.2.2). A partir dos pontos de interesse são extraídos descritores de características (seção 2.2.3), também usando a abordagem SIFT (seção 2.2.3). Esses descritores são usados para construir o vocabulário do programa aplicando uma abordagem de agrupamento. Esse processo em cascada corresponde à abordagem *Bag of Keypoints* (seção 2.2.6) em que cada descritor de características (*keypoint*) agrupado possui um índice no vocabulário de características visuais.

O processo de extração do vocabulário ocorre da forma seguinte: para cada programa, são extraídos os descritores de características. Os descritores são agrupados usando o *K-means* [11] (K = 1500) e os centróides formam o vocabulário. O vocabulário é salvo para uso posterior.

No Apêndice B podem ser vistos experimentos realizados para determinar a capacidade dos descritores de características para representar o programa.

3.3.2 Extração de Texto

O processo de extração de texto consiste em selecionar um conjunto representativo dos termos para compor o programa. Três abordagens são, usualmente, empregadas na extração de texto. São elas, *stemming*, lematização e remoção de palavras indesejadas (seção 2.3). No trabalho foram empregadas as abordagens *stemming* e remoção de palavras indesejadas. Aquela corresponde à remoção dos afixos da palavra permanecendo apenas a raiz e esta consiste em remover palavras sem semântica definida como artigos, preposições e conjunções.

O texto é extraído da forma seguinte: primeiro o descritivo textual do programa é capturado do EPG; em seguida são removidos os sinais de pontuação; logo após são removidas as palavras indesejadas; em seguida o descritivo é decomposto em *tokens*; logo após os *tokens* são reduzidos a sua forma básica. O resultado é um conjunto de palavras (*tokens*) que são as características textuais do programa. Esse é o processo geralmente usado para recuperação de informação usando processamento de linguagem natural [59].

3.4 Gerenciamento do Programa

Uma abordagem em geral usada para representação do item em técnicas de aprendizado de máquina é o espaço de vetores de palavras [26]. Como no trabalho é utilizado multimodalidades, o programa é representado em termos dos diferentes tipos de dados. Como dito anteriormente, o ganho com multimodalidades é que cada uma possui aspectos próprios para representar o item que se complementam.

Em uma representação multimodal um programa pode ser visto como a agregação de diferentes tipos de dados (Figura 3.7) [89]. No trabalho foi empregado textual e visual, porém a quantidade de tipos pode ser elevada, por exemplo adicionando informações dos usuários (idade, gênero) e áudio.

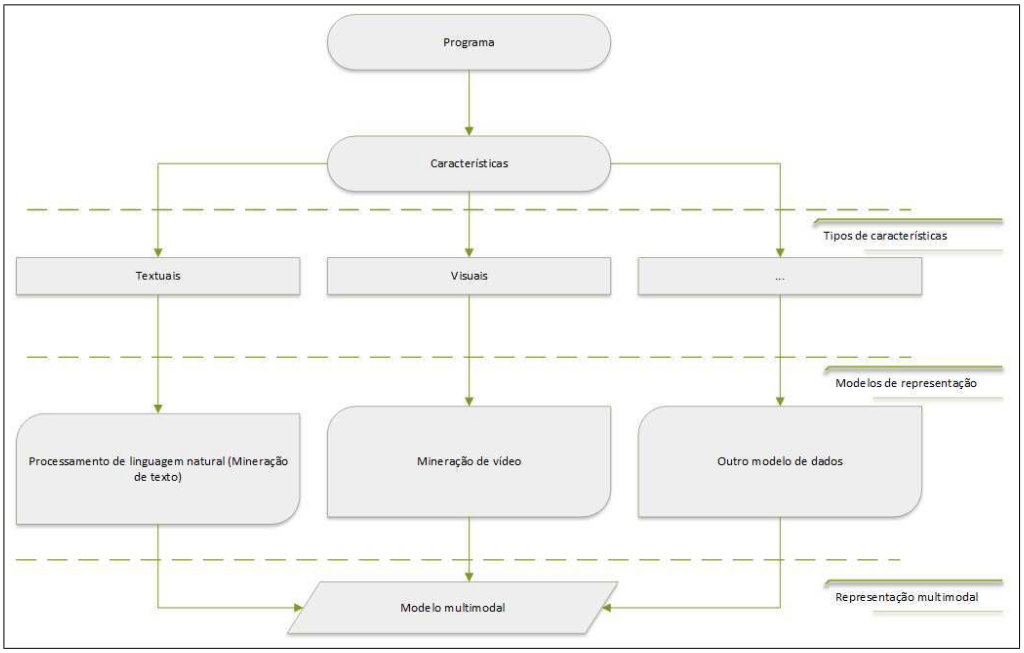

Figura 3.7: Representação do programa em termos de multimodalidade empregada no trabalho, nela o programa é divido em aspectos visuais e textuais. Esses são fundidos para compor a representação do programa. Outras modalidades podem ser inseridas dependendo do objetivo almejado.

No tocante a multimodalidades existem duas abordagens [7, 25] (seção 2.4):

- **Fusão tardia (FT)** – constrói modelos de relevância para cada espaço de caracterís-

ticas (textual, visual, etc.) separadamente, em seguida agrega os diferentes resultados para os diferentes espaços de características em um modelo único. Trata-se da abordagem mais empregada nos trabalhos anteriores [7];

- **Fusão prévia – (FP)** – constrói um modelo que unifica as diferentes modalidades em um único espaço de características e calcula a relevância usando esse modelo. Trata-se da abordagem que vem se mostrando relevante no campo de recomendação e recuperação de informação [25] e que é usada no trabalho.

Diferente de representar o item no espaço de vetores de palavras, a abordagem empregada no trabalho utiliza um modelo de grafo em uma abordagem de fusão prévia. A vantagem de empregar esse modelo é a captura da correlação relacionada às diferentes características. Características essas que podem assumir diferentes tipos de dados.

A abordagem utiliza o modelo probabilístico *Markov Random Field* (MRF). O modelo MRF assume que os nós são independentes dados os vizinhos e é representado em uma estrutura de grafos não direcionados. Mais informações sobre o modelo podem ser encontradas em [25, 47, 48, 63].

Os itens são modelados em um Grafo de Interação entre Características (GIC). Nessa estrutura os programas são representados por $P =< T, V >$, onde T são as características textuais, V são as características visuais. Todo o conjunto de característica é ligado a um vértice raiz "virtual" que representa o programa (Figura 3.8).

Como identificado por [3–5, 65, 76, 92], as categorias dos programas influenciam a acurácia da recomendação, por isso elas foram adicionadas à estrutura GIC adicionando uma aresta entre cada tipo de característica e categoria do programa.

Existem dois tipos de correlação entre as características dos itens, intracorrelação e intercorrelação:

- **Intracorrelação** – é a medida de correlação entre características do mesmo tipo, exemplo texto-texto, visual-visual;

- **Intercorrelação** - é a medida de correlação entre itens de tipos de características distintos, exemplo texto-visual.

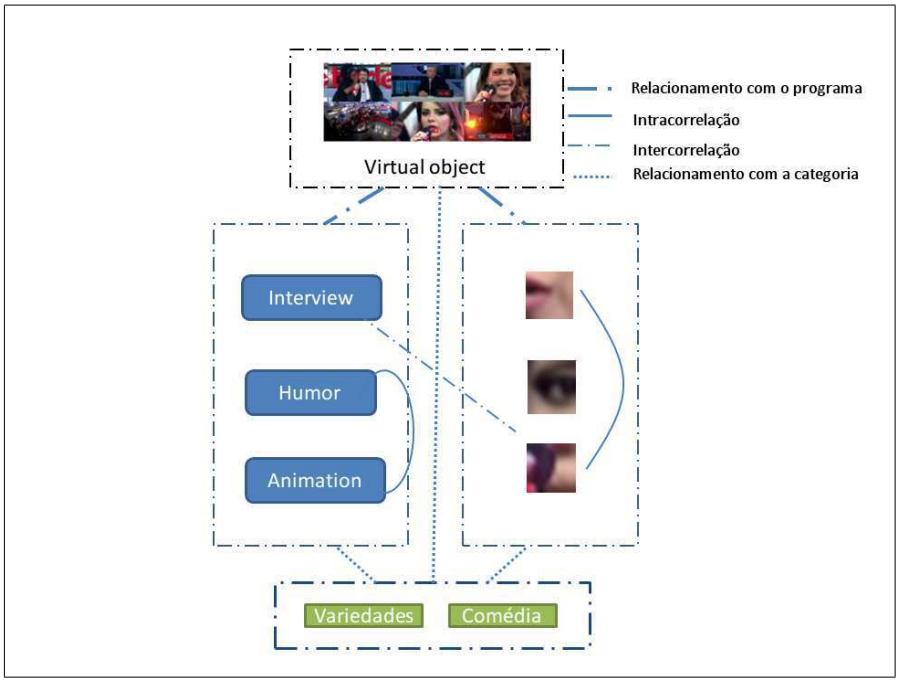

Figura 3.8: Representação do programa em um Grafo de Interação entre Características. Os nós representam as modalidades empregadas e as arestas as correlações entre as características.

A medida de intracorrelação entre as características textuais é dada pela semântica entre características fornecida pelo *WordNet* [25, 46]. Dentre as várias funções de cálculo semântico codificadas no *WordNet*, no trabalho foi empregada a abordagem *Wu & Palmer* [86] que produziu os melhores resultados nos dados.

A intracorrelação entre características visuais é realizada em um conjunto de etapas baseado na abordagem *Bag of Keypoints* (seção 2.2.6). São elas, extração de pontos de interesse do conjunto de imagens, construção de descritores de características, agrupamento do conjunto de descritores de características para representar o vocabulário.

Para cada programa é construído um vetor v, onde cada posição do vetor corresponde a frequência do descritivo visual (centróide do vocabulário) no programa de TV. A intracorrelação visual corresponde à correlação de *Pearson* (Equação 3.1 [74]) entre os descritores de características representados pelos centróides do vocabulário.

$$Corr(v_1, v_2) = \frac{\sum_{i \in |P|}(v_1 - \overline{v_1})(v_2 - \overline{v_2})}{\sqrt{\sum_{i \in |P|}(v_1 - \overline{v_1})^2 \sum_{i \in |P|}(v_2 - \overline{v_2})^2}} \tag{3.1}$$

Onde v_1 e v_2 são dois centróides do vocabulário.

A intercorrelação entre as características textuais e visuais também é calculada usando a correlação de *Pearson* usando a abordagem anterior.

Um programa possui um histograma de características visuais (Figura 3.9, mostrando as primeiras 10 posições do vetor) que apresenta grande parte dos descritores visuais. Sendo caro construir a representação do programa com todos eles, por isso apenas os descritores de características mais frequentes são selecionados para compor o programa.

Uma aresta é adicionada entre as características se a correlação entre elas é maior que um limiar. Os limiares utilizados no trabalho podem ser vistos na Tabela 3.4.

Tabela 3.4: Limiares utilizados para capturar a interação entre características

Limiar Textual-Textual	**Limiar Visual-Visual**	**Limiar Textual-Visual**
0.2	0.8	16.00

Ao final dos dois processos (intracorrelação e intercorrelação), tem-se um grafo não direcionado G que é a representação final do programa.

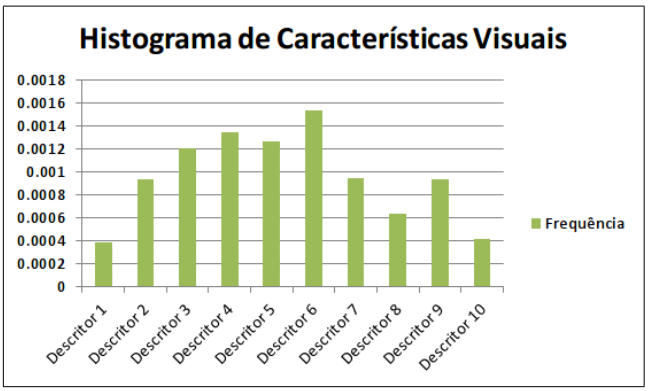

Figura 3.9: Ilustração do histograma de características visuais de um programa.

3.5 Gerenciamento do Usuário

O perfil do usuário é construído pela agregação dos programas vistos por ele $P_u = \{p_{u1}, p_{u2}, \ldots, p_{u|P_u|}\}$. Para cada programa no perfil do usuário é construído um grafo não direcionado G e o usuário é representado pelo conjunto $G_u = \{G_{u1}, G_{u2}, \ldots, G_{u|G_u|}\}$.

Por exemplo, se o usuário assistiu aos programas P_1, P_2 e P_3, o seu perfil será formado por o conjunto de cliques máximos extraídos dos programas P_1, P_2 e P_3.

As características entre diferentes programas no perfil do usuário não são conectadas, isso reduz o número de cliques extraídos. Para extração dos cliques máximos foi aplicado o algoritmo *Bron-Kerbosch* [16].

3.6 Recomendador Baseado em Conteúdo

O recomendador utiliza as GICs dos programas não vistos pelo usuário e o perfil dele para gerar a lista de recomendação. O objetivo é o desenvolvimento de uma função baseada em utilidade (seção 2.1.1) para encontrar os itens mais relevantes dado o perfil do usuário.

A abordagem empregada utiliza o modelo de Markov para calcular a probabilidade do programa aparecer junto com o perfil do usuário $P(G, G_u)$. Essa probabilidade é calculada sobre os cliques máximos do perfil do usuário G_u e programa G. Visualizando o perfil do usuário como um conjunto de cliques máximos $C = \{c_1, c_2, \ldots c_{|c(G_u)|}\}$, a probabilidade anterior pode ser definida como (Equação 3.3),

$$P(G, G_u) = P(G, c_1, c_2, \ldots c_{|c(G_u)|}) \tag{3.2}$$

Que define a probabilidade do programa G e do conjunto de características nos programas vistos pelo usuário (definidas sobre os cliques máximos $< c_1, c_2, \ldots c_{|c(G_u)|} >$ no perfil do usuário) aparecerem juntos.

Essa distribuição de probabilidade é igual a (Equação 3.3):

$$P(G, c_1, c_2, \ldots c_{|c(G_u)|}) = P(G_u)P(G|G_u) \tag{3.3}$$

Como $P(G_u)$ é constante e o interesse é encontrar um valor que indique o quão próximo está o programa do perfil do usuário, basta calcular a probabilidade não normalizada $P(G|c_1, c_2, \ldots c_{|c(G_u)|})$.

Essa probabilidade em um modelo markoviano segue uma distribuição de Gibbs (Equação 3.4). Nesse modelo, a probabilidade conjunta dos eventos corresponde ao produto da correlação entre as características (Equação 3.4).

$$P(G, c_1, c_2, \ldots c_{|c(G_u)|}) = \frac{1}{Z} \prod_{c \in c(G_u)} \varphi(c; \Lambda) \tag{3.4}$$

Onde Z é denominado de função de partição e normaliza a distribuição de probabilidade e $\varphi(c; \Lambda) = e^{\lambda_c f(c)}$ é uma função potencial sobre os cliques máximos do perfil do usuário G_u e do programa G. Para cada clique, a função potencial retorna um valor que define a estimação da probabilidade dos nós internos do clique aparecerem juntos [25]. Como o interesse é um valor que seja utilizado para indicar a posição do programa na lista de recomendação, pode-se retirar o logaritmo da distribuição e remover a função de partição (um valor constante). Isso reduz o tempo de processamento, uma vez que a função de partição é o componente mais caro para ser computado (Equação 3.5).

$$\begin{aligned} P(G, c_1, c_2, \ldots c_{|c(G_u)|}) &= \tfrac{1}{Z} \textstyle\prod_{c \in c(G_u)} \varphi(c; \Lambda) \\ &\propto \textstyle\sum_{c \in c_{|c(G_u)|}} \lambda_c f(c) \end{aligned} \tag{3.5}$$

Dessa forma, a probabilidade é definida como a soma da função potencial parametrizada por λ_c. Nesse modelo, os parâmetros são definidos sobre o tamanho dos cliques na função potencial.

A função potencial utilizada no trabalho foi a definida em Cui et al. [25] (Equação 3.6). Essa função é dividida em duas partes: a primeira estabelece a probabilidade do clique máximo dos programas no perfil do usuário aparecer no programa G mediante a frequência aparente do clique c no programa G. A segunda define a correlação entre as características individuais do clique máximo no perfil do usuário com as características individuais nos cliques do programa G. Pois é comum em ambientes multimídia que as características presentes em um clique estejam relacionadas com características de outros cliques [25].

$$\lambda_c f(c) \quad = \lambda_c \left(\alpha \frac{freq(c,G)}{|c(G)|} + (1-\alpha) \frac{\sum_{f_i \in c} \sum_{f_j \in |c(G)-c|} Corr(f_i, f_j)}{(|c|-1) \times (|\{c(G)-c\}|)} \right) \tag{3.6}$$

Os parâmetros do modelo λ_c foram aprendidos usando uma abordagem de maximização de função de utilidade (do inglês *Learning to Ranking*). Dentre os vários métodos disponíveis, no trabalho foi utilizada a abordagem *ListNet* [21], pois produziu os melhores resultados nos dados.

Para aprender os melhores parâmetros usando a abordagem *ListNet*, a seguinte metodologia foi utilizada. Primeiro, foram extraídos todos os cliques máximos para cada programa; em seguida foi construído um vetor vc, onde cada posição do vetor corresponde à frequência do tamanho do clique para cada programa. Depois desse processo, cada programa foi passado como uma pesquisa no banco de dados e foi retornado o conjunto de programas mais semelhantes à pesquisa. Para cada programa no conjunto retornado foi atribuído um *ranking*.Três valores de *ranking* são possíveis, 3 se o item é completamente relevante para a pesquisa, 2 se o item está entre relevante e irrelevante e 1 se o item é completamente irrelevante. A relevância foi medida em termo da intersecção entre categorias dos programas. Se o item possui todas as categorias do programa pesquisado, então sua relevância é 3, se o item possui metade das categorias do programa então sua relevância é 2, caso contrário sua relevância é 1. No domínio de TV os programas possuem apenas duas categorias cada.

A abordagem *ListNet* recebe como parâmetro uma matriz em que a primeira coluna é o *ranking* para uma pesquisa específica e as demais colunas representam o vetor de caracterís-

ticas do item. Dessa forma, a matriz para aprendizado dos parâmetros foi aprendida usando uma matriz como descrito anteriormente, a primeira linha é a relevância do programa e as demais linhas são a frequência do tamanho do clique no programa retornado da pesquisa.

O hiperparâmetro específico α foi definido por tentativa e erro usando a abordagem descrita anteriormente; realizando uma pesquisa por um programa e analisando os programas retornados. O hiperparâmetro é alterado baseado no número de programas cujas categorias possuem intersecção com o programa pesquisado. Quanto maior a intersecção entre as categorias, maior é a relevância do programa retornado. O valor de α usado no trabalho e que apresentou melhores resultados foi 0.9.

No Apêndice C podem ser vistas outras características em relação à arquitetura proposta, tais como implantação, padrões arquiteturais, linguagens e ferramentas de suporte empregados.

3.7 Considerações finais do Capítulo

Este capítulo apresentou uma arquitetura para recomendação baseada em conteúdo utilizando multimodalidades. A multimodalidade foi expressa em termos de características textuais e visuais. Para a extração de características visuais foi apresentada uma abordagem baseada na detecção de mudanças de câmera que selecionou os quadros mais relevantes do programa. Do conjunto de quadros dos programas foram extraídos descritores de interesse que foram utilizados junto com características textuais extraídas EPG para construir a representação do programa. A partir do modelo de cada programa assistido pelo usuário foi gerada a representação do usuário. Uma abordagem baseada na interação entre as diferentes características foi utilizada para gerar uma lista de recomendação para os usuários comparando o modelo do programa com o modelo do usuário. O próximo capítulo aborda um experimento realizado para validação da arquitetura.

Capítulo 4

Validação

4.1 Experimento

O objetivo do experimento é: comparação de diferentes modelos de recomendação unimodal e multimodal com respeito à acurácia do ponto de vista do usuário no contexto de TV digital.

A métrica acurácia foi utilizada, pois a recomendação de programas que estão de acordo com as preferências do usuário de forma acurada é o objetivo maior da recomendação de programas de TV, dado que a acurácia da recomendação influencia a percepção do usuário sobre as recomendações diretamente [22].

O experimento investigou o efeito das características multimodais em relação à acurácia dos sistemas de recomendação.

Neste experimento, pretende-se responder a seguinte pergunta: a acurácia dos sistemas de recomendação para TV Digital aumenta quando se emprega características multimodais?

As métricas de validação são o DCG@5 e Precisão@5 da recomendação para cada usuário. Os dados são numéricos de natureza quantitativa, classificados na escala de razão.

Os modelos correspondem ao multimodal, um textual e um visual. Além desses modelos, uma abordagem padrão de recomendação foi utilizada como *baseline*. Para o modelo padrão de recomendação os dados das avaliações dos usuários foram convertidos para o intervalo entre um e cinco, pois existe uma diferença entre as quantidades de exibições dos programas. Certos programas são exibidos, usualmente, uma vez por semana como os jogos de futebol, enquanto que outros, geralmente são transmitidos durante toda a semana como as novelas. Dessa forma, é preciso converter esses dados para um indicativo representativo

do interesse do usuário pelo programa. Para isso, a seguinte heurística foi utilizada: dividir a quantidade semanal indicada pelo usuário pelo número de vezes que o programa é exibido semanalmente. Para mapear o intervalo entre um e cinco, o resultado é multiplicado por 5 (Equação 4.1).

$$AI_{up} = \left\lceil \frac{A_{up}}{E_p} \times 5 \right\rceil \tag{4.1}$$

Onde AI_{up} é a avaliação implícita do usuário u para o programa p, A_{up} é a indicação do usuário da quantidade de vezes que assiste ao programa semanalmente e E_p é a quantidade de vezes que o programa é exibido no intervalo de uma semana. Para as abordagens multimodais essa heurística não foi empregada, dessa forma apenas a avaliação unária (assistiu ou não ao programa) é necessária.

O conjunto de etapas seguidas no experimento é descrito abaixo:

1. Geração de recomendações aleatórias;
2. Geração de lista de recomendação para cada usuário;
3. Cálculo das métricas Precisão e DCG;
4. Verificação de normalidade dos dados;
5. Estimação de intervalos de confiança;
6. Realização de testes estatísticos.

O resultado do experimento pode ser visto nas Figuras 4.1 e 4.2. Visualmente, a abordagem baseada em multimodalidades é mais acurada do que a abordagem usado como *baseline* em termos de DCG@5 e Precisão@5. Perceba que os resultados foram removidos para abordagem baseada em texto (Figura 4.1). Devido ao limitado número de termos na descrição dos itens presentes no EPG do programa e da quantidade de itens assistidos pelo usuário não foi possível gerar lista de recomendações com cinco itens para alguns usuários, e a abordagem DCG é influenciada pelo número de itens na lista. Na métrica DCG@5 foram aceitas recomendações com até 4 itens, em contrapartida na métrica Precisão@5 foram aceitas recomendações com até 3 itens.

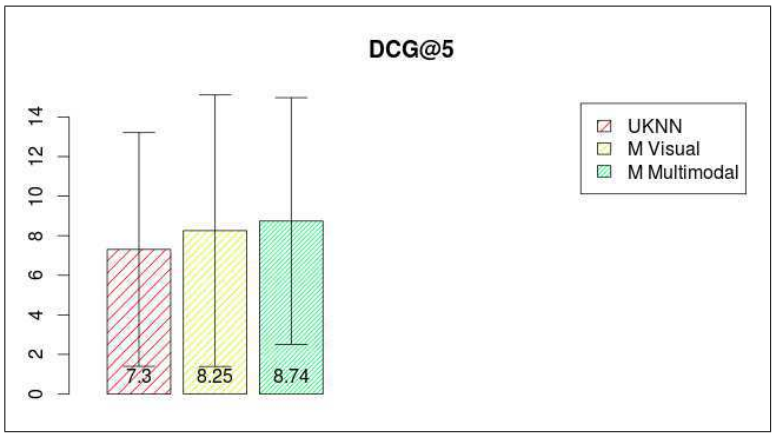

Figura 4.1: Resultado para a métrica DCG.

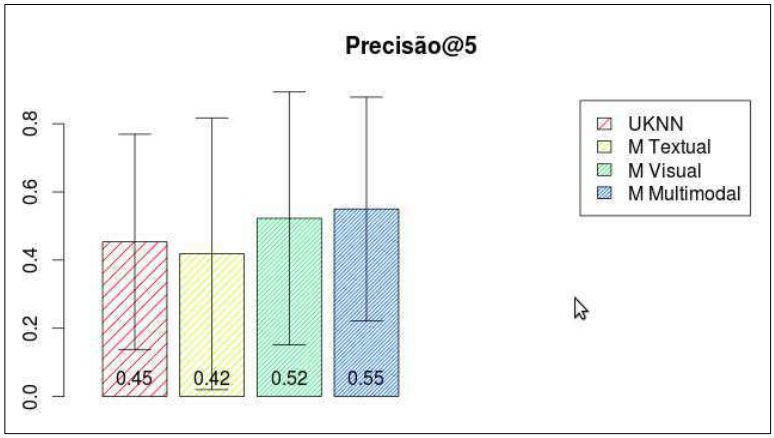

Figura 4.2: Resultado para a métrica Precisão.

O experimento foi conduzido comparando a abordagem baseada em multimodalidades com a abordagem *baseline*.

Para selecionar o modelo para ser comparado com a abordagem multimodal, diferentes abordagens de recomendação foram testadas, dentre elas Fatoração de Matrizes e algoritmo dos vizinhos mais próximos baseado no usuário (UKNN) com respeito ao Erro Quadrado Médio. Nessa seleção 75 % dos dados dos usuários foram usados para treino e 25 % para teste e a abordagem que apresentou melhores resultados foi o algoritmo dos vizinhos mais próximos baseado no usuário (Tabela 4.1), por isso essa abordagem foi utilizada como *baseline*.

Para cada usuário foi calculado o Erro Quadrado Médio e os resultados podem ser vistos na Tabela 4.1. Nesse procedimento 75 % dos dados dos usuários foram utilizados para treino e 25 % para teste.

Tabela 4.1: Resultado da avaliação do experimento em termos da métrica Erro Quadrado Médio para seleção do método de recomendação para ser comparado com a abordagem proposta

#	Média	Desvio Padrão	Intervalo de Confiança (95%)
Fatoração de Matrizes	1.0162	0.060	[0.993, 1.038]
UKNN	0.846	0.160	[0.786, 0.906]

Para cada abordagem de recomendação (modelo de Markov baseada em texto, modelo de Markov baseado em características visuais, modelo multimodal e vizinhos mais próximos baseado no usuário) foi gerada uma lista de recomendações para cada usuário e enviada por e-mail. Dada a lista de programas apresentada para cada abordagem de recomendação, o usuário deveria julgar o programa com uma nota de 0 a 3, onde 0 implica irrelevante, 3 completamente relevante e 1 ou 2 algo entre relevante e irrelevante. 30 participantes no total responderam aos e-mails e suas respostas foram usadas na validação. Um exemplo de um questionário enviado para um usuário específico pode ser visto no Apêndice F.

Esta abordagem de validação centrada no usuário (questionando o usuário) foi utilizada por três motivos: o primeiro é devido ao fato da escassez de dados para tirar-se conclusões estatísticas; o segundo as métricas de validação empregadas em sistemas de recomendação

somente extraem parte da representatividade, pois existem dados apenas sobre os itens que o usuário viu; e o terceiro porque se trabalha com *feedback* implícito, então não se têm informações negativas, por exemplo, sabe-se que o usuário interagiu com os itens A e B, mas não existe informação sobre o item C que o usuário não interagiu.

A abordagem Precisão considera apenas indicativos positivos e negativos de interesse (ou seja, o usuário gostou ou não do item (seção 2.1)). Para converter o valor de interesse indicado pelo usuário para um valor requerido pela métrica foram considerados valores 3 e 2 como positivos e 1 e 0 como negativos.

Como a abordagem multimodal, visualmente, apresenta melhores resultados ela foi usada para validação e foram geradas duas hipóteses alternativas:
$< H_1 >$ – O DCG para a abordagem multimodal (M Multimodal) é maior que a abordagem dos vizinhos mais próximos baseado no usuário (UKNN).

$$\mu_{MMultimodal} > \mu_{UKNN} \tag{4.2}$$

$< H_1 >$ – A Precisão para a abordagem multimodal é maior que a abordagem dos vizinhos mais próximos baseado no usuário.

$$\mu_{MMultimodal} > \mu_{UKNN} \tag{4.3}$$

Testes de normalidade foram conduzidos usando o teste *Shapiro-Wilk* e foi verificado que algumas abordagens não são normalmente distribuídas usando p-valor = 90 $\%$. Por isso, o teste não paramétrico *Wilcoxon* foi utilizado na validação. Após rodar os testes, para H_1 foi obtido um p-valor = 0.107, dessa forma se conclui com 89.93 $\%$ de confiança que a abordagem multimodal apresenta melhores resultados em termos de DCG, e para H_2 foi obtido um p-valor = 0.097, dessa forma se conclui com 90 $\%$ de confiança que a abordagem baseada em multimodalidades apresenta melhores resultados em termos de Precisão.

4.2 Ameaças à Validade

Nesta seção, destacam-se ameaças à validade do experimento (fatores que podem influenciar os resultados obtidos). Sendo categorizadas da maneira seguinte: ameaça à validade interna, externa e de construção.

Ameaças à validade interna, usualmente, estão relacionadas ao poder dos métodos estatísticos empregados no experimento. No experimento, tem-se como ameaça à validade interna o tamanho reduzido da amostra; os métodos estatísticos aplicados são influenciados pelo tamanho da amostra, no entanto como aponta a literatura uma amostra de tamanho 30 é suficiente para extrair informações dos dados [14].

As ameaças à validade externa se referem ao ambiente em que o experimento foi executado e a capacidade das abordagens em se adaptarem a outros domínios. No trabalho, tem-se como ameaça à validade externa a quantidade de programas disponíveis para recomendação. Na literatura existem várias abordagens de recomendação colaborativa que produzem resultados significativos quando a quantidade de dados aumenta, tornando os dados esparsos como a Fatoração de Matrizes e o k-NN baseado em itens. Dessa forma, estudos adicionais precisam ser realizados quando o número de itens e usuários aumenta. No entanto, o foco do trabalho é a recomendação baseadas em conteúdo, sendo uma abordagem colaborativa usada apenas como *baseline*. Além disso, o modelo empregado neste trabalho foi utilizado em outros ambientes (mídias sociais) e demonstrou resultados significativos com bases de dados maiores.

Quanto às ameaças à validade de construção, referem-se aos instrumentos utilizados para construir a qualidade da recomendação. Como ameaça à validade de construção, têm-se as métricas de validação empregadas. As métricas dependem do objetivo almejado. O experimento foi avaliado em termo de DCG e Precisão, porém outras métricas são possíveis como a Novidade e a Surpresa. No entanto, o interesse é medir a qualidade da recomendação em termos de acurácia, sendo as abordagens empregadas adequadas.

Capítulo 5

Revisão da Literatura

Neste capítulo são destacados os trabalhos relacionados, mostrando algumas arquiteturas usadas para recomendação em TV digital e alguns modelos multimodais que podem ser usados.

5.1 Arquiteturas para TV Digital

5.1.1 PersonalTVware: Uma Proposta de Arquitetura para Suporte a Personalização Ciente de Contexto de Programas de TV

Uma proposta de arquitetura para personalização de serviços foi desenvolvida em [27] (Figura 5.1). A arquitetura objetiva oferecer suporte para o desenvolvimento de aplicações para personalização de serviços usando o contexto. Os dados contextuais usados possuem as seguintes dimensões: quem (identidade), quando (tempo), onde (localização), o que (atividade) e como (uma maneira de identificar como os elementos do contexto são coletados).

A arquitetura é composta por dois subsistemas: Dispositivo do usuário e Provedor de serviços. O subsistema Dispositivo do usuário pode ser desenvolvido no *set-top-box*, computador portátil, ou qualquer dispositivo móvel com um *middleware* para TV digital instalado. A comunicação entre os subsistemas é bidirecional mediante uma interface de serviço web.

Os módulos contidos no subsistema Dispositivo do usuário são descritos abaixo:

- **Gerenciador de recomendação** – é a interface entre o subsistema Dispositivo do usuário e os outros módulos do subsistema. É responsável pelo gerenciamento do processo

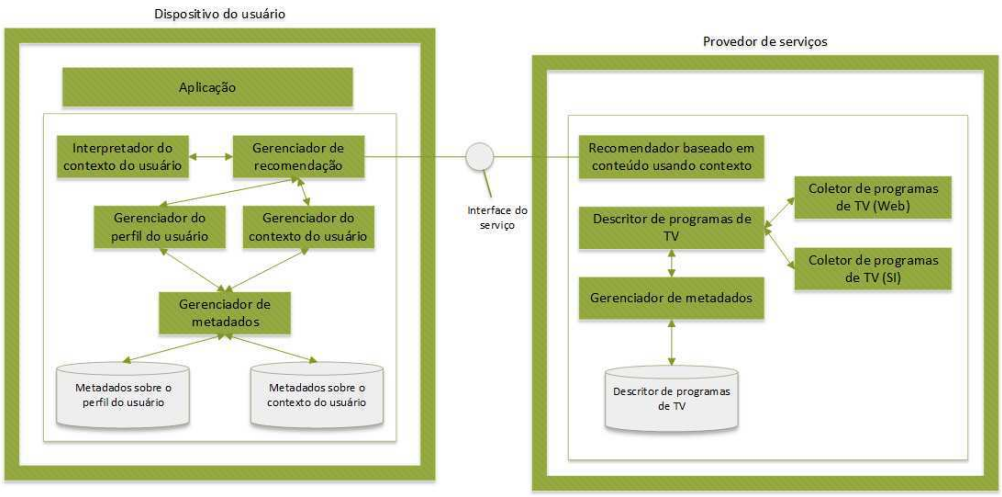

Figura 5.1: Arquitetura utilizada no trabalho: PersonalTVware: Uma Proposta de Arquitetura para Suporte a Personalização Ciente de Contexto de Programas de TV

de recomendação;

- **Gerenciador do contexto do usuário** - é responsável pelo acesso e aquisição de forma implícita de informações contextuais atuais e passadas;

- **Gerenciador do perfil do usuário** – é responsável pelo acesso e aquisição de informações que constitui o perfil do usuário, coletadas de forma explícita. Por meio desse componente, o usuário é capaz de especificar dados como informações pessoais (nome, idade, gênero, etc.) e preferências (programas de TV, atores, categorias, etc.);

- **Interpretador do contexto do usuário** - é responsável por inferir preferências implícitas por canais e programas de TV por meio das informações contextuais atuais e passadas disponíveis no módulo Gerenciador de Contexto do Usuário.

Os módulos contidos no subsistema Provedor de serviço são descritos abaixo:

- **Filtragem baseada em conteúdo usando o contexto** - é disponibilizado no subsistema Provedor de serviços devido às limitações de recursos do subsistema Dispositivos do usuário. Esse módulo é responsável pela filtragem dos programas de TV que poderão ser relevantes para usuário considerando o contexto. A filtragem usa informações contextuais como (dia, tempo), perfil do usuário, inferências implícitas geradas

pelo Interpretador de contexto do usuário e a descrição do conteúdo dos programas de TV;

- **Gerenciador de descrições do programa** - é responsável pela consulta e inserção de informações dos programas de TV;
- **Gerenciador de metadados** - prove suporte para outros módulos na arquitetura, e é responsável pela recuperação, armazenamento e validação de metadados;
- **Coletor de programas de TV** (WEB e SI) - é utilizado para capturar informações relativas aos programas de TV por meio de recursos externos com a WEB e SI (Serviço de informação). O módulo WEB permite ao administrador atualizar as informações sobre os programas e o módulo SI permite ao sistema recuperar informações sobre o programa. Desta forma, ambos os módulos são responsáveis pela atualização dos metadados dos programas de TV.

O processo de execução da recomendação acontece da seguinte forma: dado que o usuário explicitamente definiu o perfil, o módulo Gerenciador de recomendação recebe uma requisição para recuperar uma lista de recomendações; capturando informações contextuais do usuário, como a identificação, localização, dia e horário da interação e tipo de dispositivo utilizado. As informações contextuais do módulo Gerenciador de contexto e Gerenciador de perfil do usuário são coletadas pelo Gerenciador de recomendação e passadas para o Interpretador do contexto do usuário que infere preferências implícitas para o programa. O Gerenciador de recomendação recebe as preferências implícitas inferidas pelo Interpretador de contexto, coleta as informações definidas no perfil do usuário e repassa para o módulo Filtragem baseada em conteúdo. O módulo Filtragem baseada em conteúdo realiza a filtragem dos programas de TV por meio da comparação de informações contextuais, perfil do usuário e preferências inferidas e informações sobre os programas coletadas pelo módulo Gerenciador de descrição do programa. A lista de recomendação é gerada e repassada ao Gerenciador de recomendação.

A vantagem da arquitetura é que ela é escalável. A divisão da arquitetura em dois subsistemas possibilita que uma carga maior de processamento seja inserida no lado Provedor de serviço e funcionalidades que demandem menos recursos sejam implantadas no lado Dispositivo do usuário.

5.1.2 Um Sistema de Recomendação para um Provedor de Serviços de IPTV: Um Ambiente de Produção em Larga-Escala

Em [8] foi apresentada uma arquitetura para personalização de serviços em IPTV (Figura 5.2).

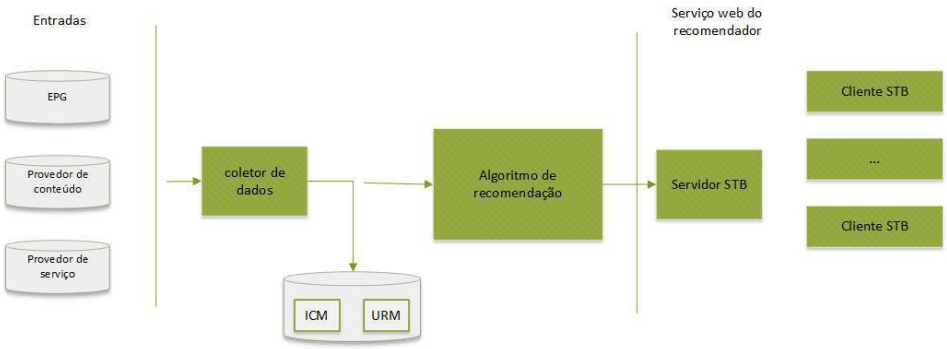

Figura 5.2: Arquitetura utilizada no trabalho: Um Sistema de Recomendação para um Provedor de Serviços de IPTV: Um Ambiente de Produção em Larga-Escala

Os módulos da arquitetura são descritos a seguir:

- **Coletor de dados** - pré-processa os dados e gera entradas para o sistema de recomendação. Esse módulo recebe dados de várias fontes EPG, provedor de conteúdo e provedor de serviços. As informações coletadas são estruturadas em duas matrizes: matriz do conteúdo do item (ICM – do inglês *item-content matrix*) e matriz de avaliações do usuário (URM – do inglês *user-rating matrix*). A matriz ICM descreve as principais características (*metadados*) dos itens, enquanto que as informações armazenadas em URM representam as avaliações dos usuários em relação aos itens.

- **Algoritmos de recomendação** - composto por três sistemas de recomendação: uma abordagem baseada em conteúdo, uma filtragem colaborativa baseada no item e uma filtragem colaborativa baseada no usuário.

Para respeitar restrições de recursos, o sistema de recomendação usa uma abordagem baseada em modelo. O processo é dividido em duas etapas (Figura 5.3):

- **Processamento em *batch*** - cria um modelo dos dados de entrada. Esta etapa é realizada nos horários vagos;

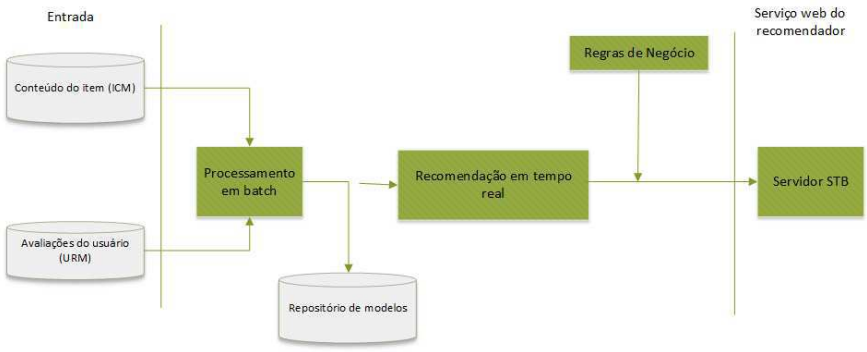

Figura 5.3: Estágios do processo de recomendação utilizados no trabalho: Um Sistema de Recomendação para um Provedor de Serviços de IPTV: Um Ambiente de Produção em Larga-Escala

- **Recomendações em tempo real** - usa o modelo gerado na etapa anterior para produzir recomendações em tempo real.

O repositório de modelo - faz as duas etapas serem assíncronas: enquanto o processo Recomendações em tempo real está sendo executado outro modelo está sendo construído.

A vantagem da arquitetura é o tempo de resposta devido à divisão lógica do processamento em etapas.

5.1.3 Recomendação Personalizada de Programas de TV

Ardissono et al. [3] apresentou uma arquitetura para TV digital que engloba diferentes tipos de informações sobre o usuário e itens que ele deseja interagir, tais como:

- Preferências explícitas por categorias indicadas pelo usuário, tais como filmes, documentários, etc;
- Informações estereotipadas sobre as categorias de preferência do usuário baseadas no número de programas assistidos de cada categoria;
- Informações sociais demográficas, tais como idade, ocupação, etc;
- Informações sobre o dia a dia do usuário como interesses e passatempos;

- Informações à priori sobre o estereótipo do comportamento de visualizações do usuário.

A arquitetura pode ser vista na Figura 5.4.

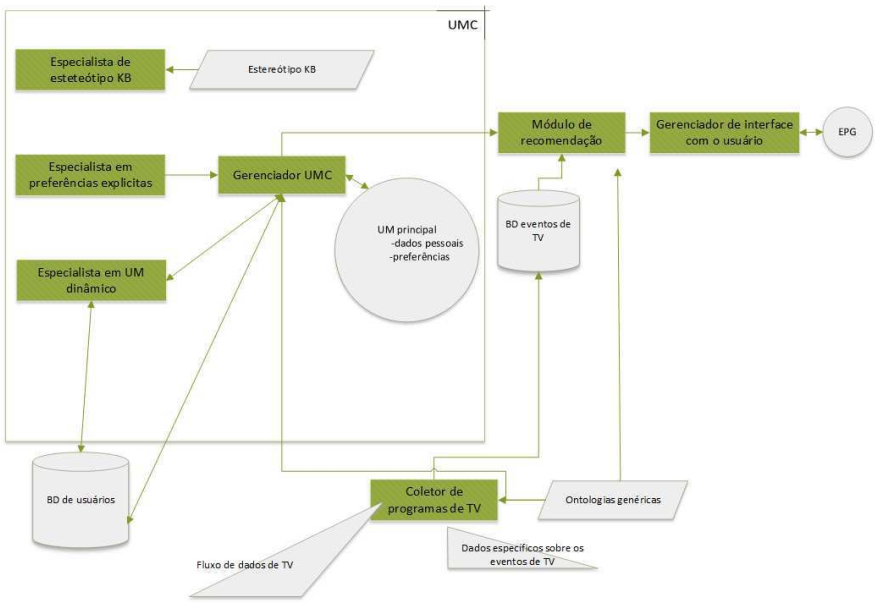

Figura 5.4: Arquitetura utilizada no trabalho: Recomendação Personalizada de Programas de TV

A arquitetura foi designada para rodar no lado cliente (*set-to-box*), porém por questões de demonstração foi implantada em um computador de mesa.

Essa arquitetura foi aplicada também nos trabalhos [4] e [5].

5.1.4 Sistema de TV Personalizado

Zimmerman [92] apresentou uma arquitetura baseada em dados implícitos e explícitos dos hábitos de visualização do usuário. A arquitetura é composta por diferentes módulos que coletam esses dados e fornecem recomendação (Figura 5.5).

A arquitetura foi designada para rodar no lado cliente da TV (*set-to-box*). Os dados implícitos são coletados analisando o histórico de visualização do usuário. Foi assumido que o histórico do usuário é disponível quando a arquitetura é implantada, porém para demonstração os dados foram coletados mediante diário de anotações. Os usuários anotavam

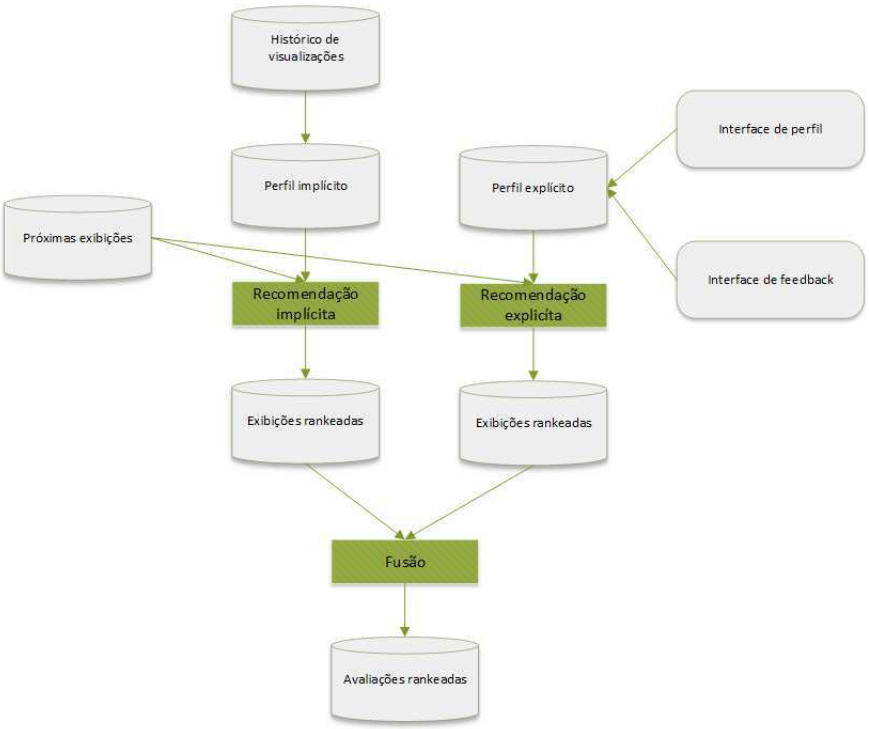

Figura 5.5: Arquitetura utilizada no trabalho: Sistema de TV Personalizado

informações sobre os programas assistidos e no intervalo de um mês os dados eram coletados.

Para a coleta de dados explícitos foi desenvolvido um sistema especial em que o usuário poderia avaliar um conjunto de programas com notas entre 1 a 7. Módulos especiais para cada tipo de dado computam a relevância em separado e os resultados dos módulos para cada modalidade são fundidos para gerar a lista de recomendação.

5.1.5 Guia Eletrônico Personalizado de Televisão

Smyth et al. [76] apresentou uma arquitetura para TV personalizada que emprega uma abordagem híbrida (Figura 5.6), combinando recomendação baseada em conteúdo e recomendação colaborativa. Essa técnica de recomendação é empregada por a abordagem baseada em conteúdo, embora apresente resultados significativos; na falta de informações (descritivo textual significativo para extração de informação), o desempenho da recomendação é reduzido.

A arquitetura inclui várias unidades funcionais e banco de dados. O banco de dados de

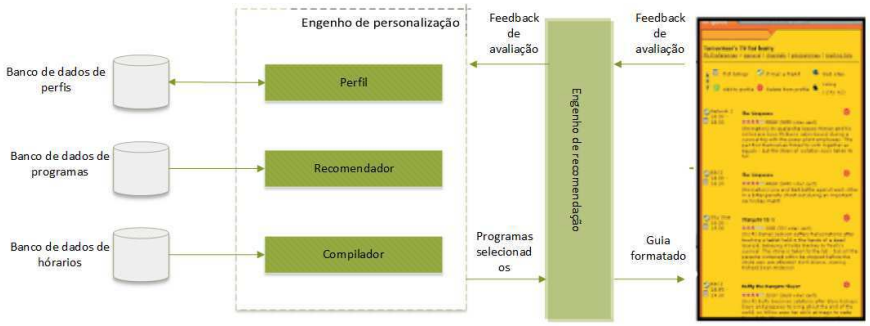

Figura 5.6: Arquitetura utilizada no trabalho: Guia Eletrônico Personalizado de Televisão

perfis armazena perfis individuais que codificam as preferências dos usuários para os programas de TV, listando canais, horário preferido de interação, programas e gêneros preferidos e assim por diante.

O Banco de dados de programas armazena a descrição textual do programa. Cada entrada descreve características específicas do programa, tais como título, diretor, apresentador, o país de origem e a linguagem. Essas informações são indispensáveis para a recomendação baseada em conteúdo.

O Banco de dados de horários armazena informações, tais como data de início, data de fim e informações sobre o programa em questão.

O Gerenciador de perfis é responsável por manter atualizadas todas as informações de perfis. O recomendador é o núcleo da arquitetura e é responsável por selecionar o perfil do usuário e encontrar uma lista de programas que ele tenha interesse. A arquitetura foi designada para trabalhar com diferentes tipos de interface. O componente Compilador é responsável por traduzir as recomendações para as diferentes formas de acesso (web, móvel, TV).

5.1.6 Televisão Personalizada Interativa: Dos Guias aos Programas

Sullivan et al. [67] apresentou uma abordagem para recomendação baseada nas preferências encontradas no perfil de interesse dos usuários. A abordagem usa uma técnica de recomendação híbrida que combina resultados de uma abordagem de recomendação baseada em conteúdo e colaborativa. As recomendações são apresentadas como um guia eletrônico de programação (Figura 5.7).

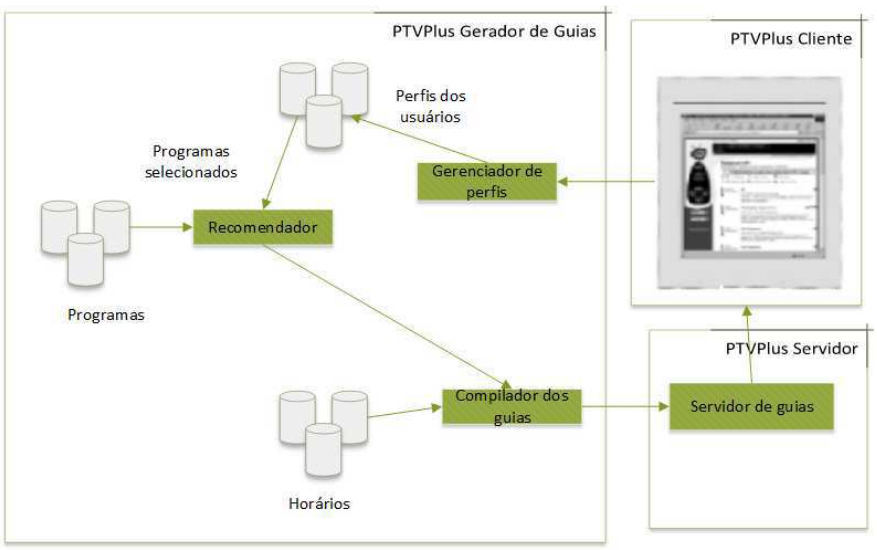

Figura 5.7: Arquitetura utilizada no trabalho: Televisão Personalizada Interativa: Dos Guias aos Programas

As informações foram coletadas através de um ambiente em que o usuário é capaz de assistir aos programas e gravar a sua interação com o ambiente.

5.1.7 Um Método de Aprendizado para Personalização de TV Futura

Shi et al. [75] apresentou uma abordagem para televisão personalizada usando metadados extraídos do ambiente *TV-Anytime*. Segundo Shi et al. as informações presentes no EPG são limitadas, incluindo somente informações básicas sobre o programa (canal, título, data de começo e data de fim), para encontrar as preferências do usuário.

O sistema é composto de dois componentes principais: um servidor e uma interface agente (Figura 5.8). O servidor inclui o Banco de dados dos perfis dos usuários o Agente de filtragem e recomendação e um Agente de perfil. O Banco de dados de perfis contém o perfil de múltiplos usuários. O módulo de Filtragem e recomendação coleta e faz a correspondência entre o perfil do usuário e os programas baseado em metadados. O Agente de perfis cria e atualiza o perfil do usuário baseado em informações coletadas da interface com o usuário.

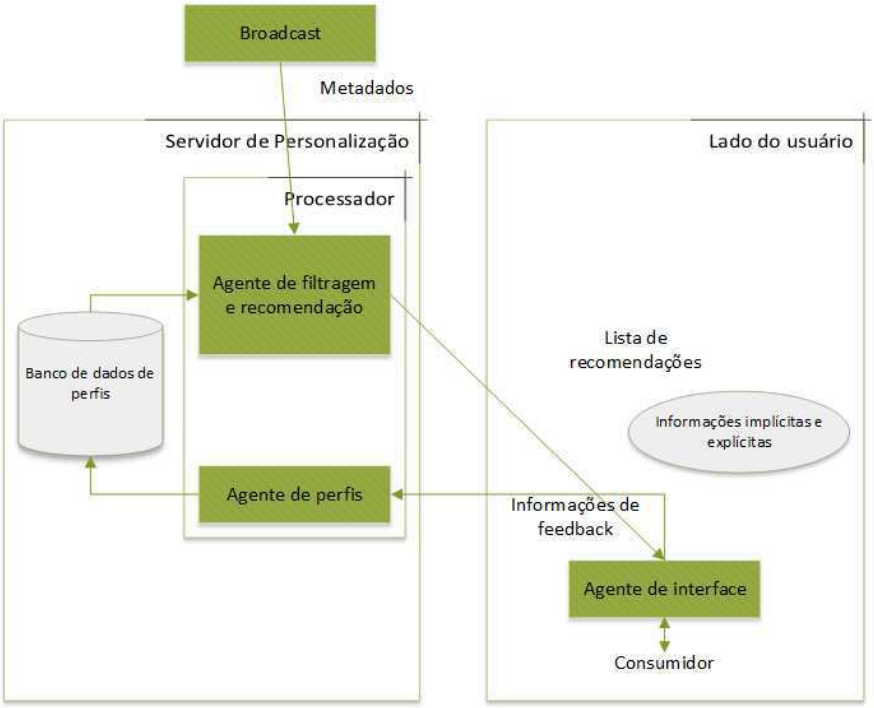

Figura 5.8: Arquitetura utilizada no trabalho: Um Método de Aprendizado para Personalização de TV Futura

5.1.8 Outros Trabalhos em Personalização de Serviços

Masthoff et al. [60] aplicou uma abordagem de recomendação para indicar itens baseada nos interesses de grupos de usuário, diferente dos trabalhos na área que focam na recomendação para um usuário único. O autor identifica que o hábito de assistir TV é um evento social que em geral é feito em grupo com familiares ou amigos. Hara et al. [41] estudou o uso de estereótipos para classificar o usuário baseado nos programas que eles assistem. Hara et al. acredita que embora outros trabalhos apliquem informações demográficas como gênero, idade, localização, etc.; nada pode descrever melhor o usuário do que os programas que ele assiste. Musto et al. [65] pesquisou o uso de texto para personalização em ambiente de TV. Musto et al. utiliza metadados da Wikipédia[1] para enriquecer o conteúdo textual associado com o programa.

Para uma visão mais abrangente sobre os trabalhos em TV, veja o trabalho de Chang et al. [22] que realizou uma revisão da literatura sobre o tema.

[1] `http://www.wikipedia.org/`

Apesar de as abordagens citadas se mostrarem relevantes, nenhuma delas faz uso de multimodalidades para recomendação em TV. O uso de multimodalidades mostrou resultados significativos em ambientes multimídia, e é investigado no trabalho.

5.2 Modelos Multimodais

Nesta seção são apresentados alguns modelos multimodais usados na literatura.

5.2.1 Fusão de Múltiplas Características para Aplicações de Mídia Social

Cui et al. [25] apresentou uma abordagem para recuperação de informações e sistemas de recomendação de itens em ambientes sociais multimídia. Itens no contexto do trabalho se refere a postagens da rede social *Flickr* [83]. Nesse ambiente o usuário, usualmente, realiza uma pesquisa inserindo termos chaves associados com os itens de interesse, tais como cantor favorito, categoria da postagem ou termos relacionados à postagem.

Para encontrar itens relevantes para a pesquisa a similaridade entre os itens é usada. Para isso, são extraídas características dos usuários ou itens, tais como pessoas que consumiram o item, o conjunto de descritivos textuais associado ao item como comentário, título da postagem e descrição do item, assim como características visuais. Diferente das abordagens anteriores, Cui et al. [25] explorou o uso de multimodalidades (usuário, texto e vídeo) para construir o modelo.

O autor utilizou um modelo probabilístico baseado em grafos em que as características correspondem aos vértices do grafo e as arestas são as afinidades entre eles, medida por meio da correlação entre as características. Dois tipos de correlação são possíveis, intracorrelação (entre características da mesma modalidade) e intercorrelação (entre características de diferentes modalidades). A correlação entre características textuais é medida por meio do *WordNet* e a similaridade entre características visuais e intercorrelação é medida mediante a co-ocorrência dos termos e a correlação entre os usuários é baseada nos grupos aos quais o usuário pertence.

O autor realizou experimentos e concluiu que o uso de multimodalidades eleva à acurácia

dos sistemas de recomendação em ambientes multimídia.

5.2.2 Aprendizado Multimodal com Máquinas de Boltzmann

Srivastava et al. [80] empregou um modelo de *Deep Learning* para recuperação de informação multimodal, assim como [25] a rede social *Flickr* [83] foi utilizada.

Deep Learning é um tipo de rede neural baseado em grafos que tenta imitar a organização neural humana [9].

A vantagem da aplicação de *Deep Learning* é que ela trabalha com dados de diferentes propriedades estatísticas [80], uma importante característica, pois, em geral cada modalidade possui propriedades estatísticas distintas (Figura 5.9 (adaptada de [72])). Por exemplo texto, usualmente, possui uma representação esparsa em que o vocabulário de palavras consiste de todas as palavras do domínio, porém o conjunto de termos presente em uma postagem de uma rede social possui um subconjunto reduzido dos termos do vocabulário, enquanto que imagens e vídeos são representados de forma densa em que uma imagem possui muitas das características presentes no vocabulário de evidências visuais [80].

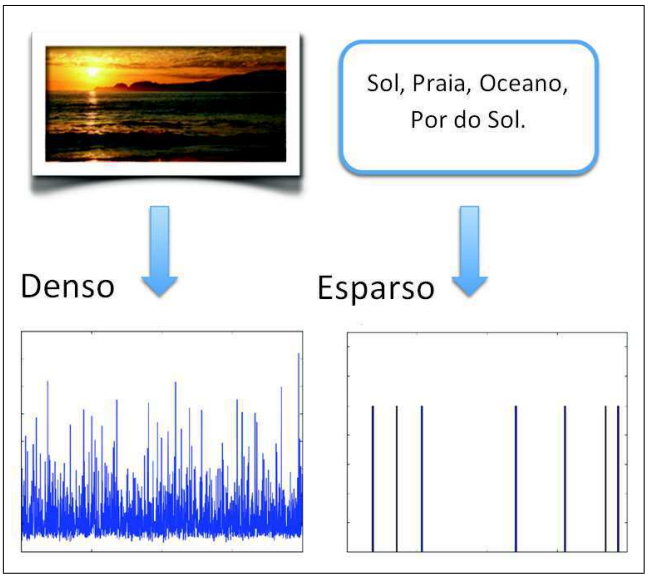

Figura 5.9: Representação dos dados. Imagens são representadas de forma densa e textos são representados de forma esparsa.

Outra vantagem da *Deep Learning* é que uma abordagem de aprendizado de máquina

não supervisionada, uma característica importante, pois em alguns ambientes não existem dados rotulados suficientes.

5.2.3 Recomendação de Vídeos Online Baseado na Fusão de Multimodalidades e *Feedback* Relevantes

Yang et al. [89] apresentou uma abordagem para recomendação multimodal baseada em fusão tardia que emprega tipos de dados textuais, visuais e áudio (Figura 5.10).

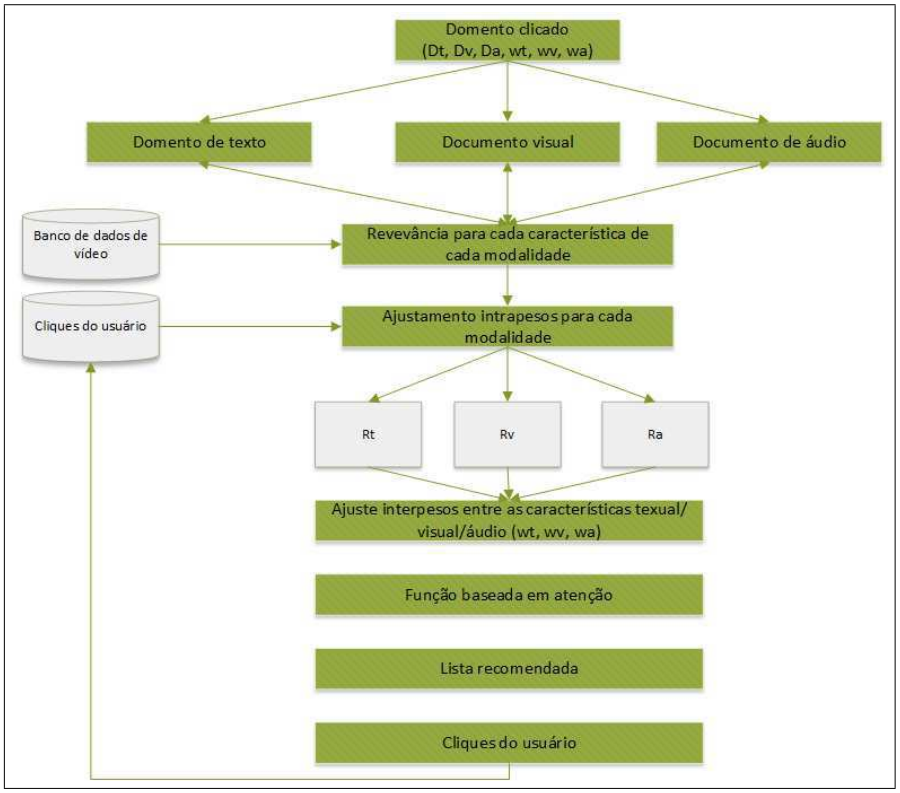

Figura 5.10: Modalidades utilizadas no trabalho: Recomendação de Vídeos Online Baseada na Fusão de Multimodalidades e *Feedback* Relevantes.

A modalidade textual é extraída do texto ao redor do vídeo, legenda e também categoria do vídeo. As características visuais usadas no trabalho são o histograma de cores do vídeo, a intensidade de movimento e frequência de quadros (quadros por segundo). A modalidade de áudio é capturada baseada no desvio padrão da velocidade do áudio (música ou fala). O peso atribuído e alterado para a relevância de cada modalidade é baseado no *feedback* do usuário.

O conjunto de características extraído funciona como um arcabouço e foi empregado também nos trabalhos [61] e [62].

5.2.4 Recomendação Personalizada de Vídeos de Notícia

Luo et al. [56] apresentou uma abordagem para recomendação de notícias personalizadas utilizando as modalidades texto, áudio e vídeo (Figura 5.11).

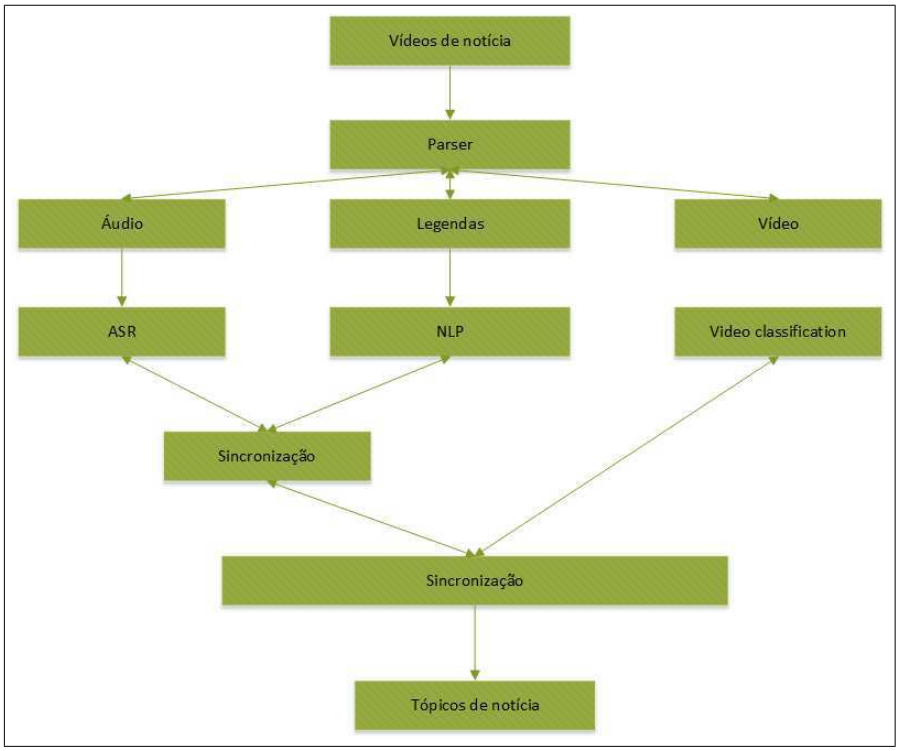

Figura 5.11: Modalidades utilizadas no trabalho: Recomendação Personalizada de Vídeos de Notícia

A recomendação é baseada na extração do tópico do vídeo. A extração é realizada em um processo utilizando características multimodais: primeiro, detecção automática de fala (*automatic speech recognition* (ASR)), processamento de linguagem natural (*natural language processing* (NLP)) e classificação semântica de vídeo. Essas modalidades são processadas em paralelo para determinar as palavras chaves para descrição do tópico da notícia de áudio e legendas e detectar o conceito do vídeo para o canal. Segundo, o áudio é sincronizado com a legenda e em seguida o vídeo é sincronizado com ambos. Finalmente, a extração do tópico

da notícia é realizado pela fusão das diferentes modalidades.

5.2.5 Enriquecimento de Classificadores de Vídeo para Classificação de Vídeos Web

Cui et al. [26] apresentou uma abordagem para classificação de vídeos baseada no enriquecimento do descritivo textual usando características do vídeo como cor, texturas e bordas. Diferente das abordagens anteriores, o conteúdo visual não é extraído em tempo de classificação.

A abordagem adiciona as informações do conteúdo visual para enriquecer a semântica entre palavras. Embora a abordagem empregue texto e vídeo para classificação de vídeos web, a técnica é diferente das abordagens de fusão tradicionais por dois aspectos: o conteúdo visual dos dados de treino é apenas utilizado para enriquecer a semântica do texto e não é utilizado em tempo de classificação (Figura 5.12). O tempo de execução da abordagem em relação à técnica que emprega texto é o mesmo, no entanto a abordagem possui melhor acurácia que as abordagens tradicionais por empregar diferentes modalidades, texto e vídeo. Sendo assim, essa abordagem reduz um grande problema da abordagem que utiliza apenas texto, o problema da esparsidade.

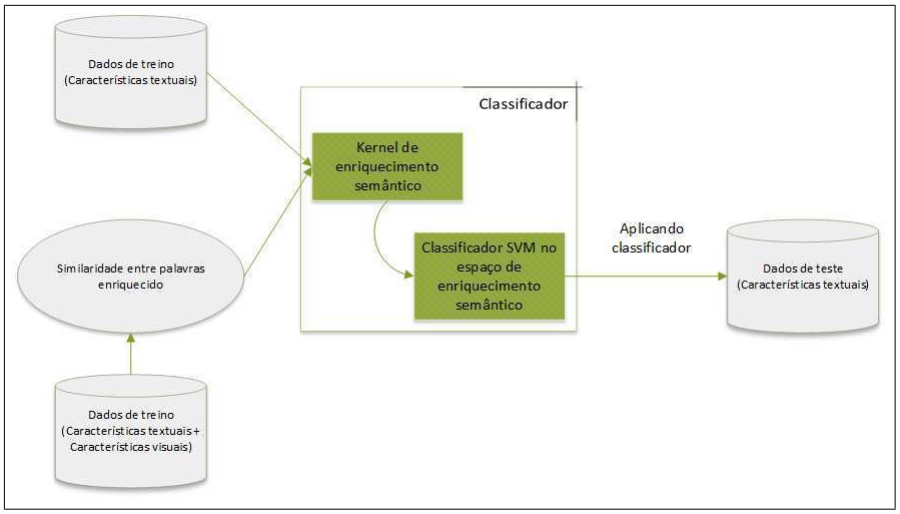

Figura 5.12: Modalidades utilizadas no trabalho: Enriquecimento de Classificadores de Vídeo para Classificação de Vídeos Web.

Para mais informações sobre os trabalhos com multimodalidades Atrey [7] realizou uma revisão da literatura abordando o tema.

Capítulo 6

Considerações Finais

No trabalho foi apresentada uma arquitetura com suporte a uni e multimodalidades para recomendação baseada em conteúdo para TV digital. Diferentes modelos de recomendação uni e multimodal foram testados experimentalmente e se demostrou que usando características multimodais a acurácia da recomendação pode ser elevada quando comparada com uma abordagem de recomendação padrão.

As contribuições do trabalho são descritas a seguir:

- Uma arquitetura para recomendação multimodal em programas de TV;
- Uma abordagem para recomendação multimodal em TV digital usada no trabalho;
- Uma base de dados com usuários e itens para recomendação em programas de TV;
- Uma base de dados com um conjunto de eventos em programas de TV para aplicações em visão computacional.

Em trabalhos futuros, pretende-se:

- **Aumentar a base de dados** – a base de dados utilizada para construir o perfil do usuário e a representação do programa é comporta por 42 usuários e 95 programas, dessa forma mais dados podem ser capturados para a construção mais realística de modelos de recomendação.
- **Aumentar o número de sujeitos utilizados na validação** – para investigar a efetividade da solução proposta no trabalho foi usada uma amostra de tamanho 30, dessa forma uma amostra maior pode ser utilizada para se obter maior poder estatístico.

- **Testar e comparar outros modelos de recomendação multimodais** – além da abordagem aplicada no trabalho, outros modelos de recomendação multimodal podem ser utilizados, em especial uma abordagem que está apresentado resultados positivos na recomendação multimodal são as máquinas de *Boltzmann*. Dessa forma estudos adicionais podem ser realizados nesse tocante, assim como comparar os diferentes modelos.

- **Utilizar metadados** – outra forma de enriquecer o EPG é pelo uso de metadados. Neste trabalho foi utilizada uma abordagem baseada em multimodalidades, porém o uso de metadados pode ser utilizado em conjunto com as diferentes modalidades.

- **Melhorar a função potencial** – outras funções potenciais podem ser desenvolvidas além da utilizada no trabalho;

- **Melhorar o modelo de recomendação usando a abordagem GIC** – no trabalho, uma abordagem simples, tratar o perfil do usuário como a agregação dos GIC foi usada, porém outras estratégias podem ser aplicadas.

- **Acrescentar mais modalidades ao modelo usado** – no trabalho foi aplicado texto e o vídeo como fonte de dados, no entanto outros tipos de dados podem ser acrescentados ao modelo como os usuários.

Bibliografia

[1] Jgrapht. Disponível em:`http://jgrapht.org/`, acessado em: 10 de Janeiro de 2014.

[2] Xavier Amatriain, Alejandro Jaimes, Nuria Oliver, and Josep M. Pujol. Data mining methods for recommender systems. In *Recommender Systems Handbook*, pages 39–71. 2011.

[3] L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Chiarotto, A. Difino, and B. Negro. Personalized recommendation of tv programs. In *In LNAI n. 2829. AI*IA 2003: Advances in Artificial Intelligence*, pages 474–486. Springer Verlag, 2003.

[4] L. Ardissono, F. Portis, P. Torasso, F. Bellifemine, A. Chiarotto, and A. Difino. Architecture of a system for the generation of personalized electronic program guides. In *Proc. UM'01 Workshop on Personalization in Future TV*, pages 1 – 8, 2001.

[5] Liliana Ardissono, Cristina Gena, Pietro Torasso, Fabio Bellifemine, Angelo Difino, and Barbara Negro. User modeling and recommendation techniques for personalized electronic program guides. In *Personalized Digital Television – Targeting Programs to Individual Viewers, volume 6 of Human-Computer Interaction Series, chapter 1*, pages 3–26. Kluwer Academic Publishers, 2004.

[6] Liliana Ardissono, Alfred Kobsa, and Mark T. Maybury. *Personalized Digital Television: Targeting Programs to Individual Viewers*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[7] Pradeep K. Atrey, M. Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. Multimodal fusion for multimedia analysis: A survey. *Multimedia Systems*, 16(6):345–379, 2010.

[8] Riccardo Bambini, Paolo Cremonesi, and Roberto Turrin. A recommender system for an iptv service provider: a real large-scale production environment. In *Recommender Systems Handbook*, pages 299–331. 2011.

[9] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.

[10] Michael W. Berry and Malu Castellanos. *Survey of Text Mining II: Clustering, Classification, and Retrieval*. 1 edition.

[11] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[12] M. Bjelica and A. Peric. Adaptive feedback schemes for personalized content retrieval. *Consumer Electronics, IEEE Transactions on*, 57(3):1251 –1257, august 2011.

[13] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46(0):109 – 132, 2013.

[14] Sarah Boslaugh and Paul Andrew Watters. *Statistics in a nutshell - a desktop quick reference.* O'Reilly, 2008.

[15] Dr. Gary Rost Bradski and Adrian Kaehler. *Learning Opencv, 1st Edition*. O'Reilly Media, Inc., first edition, 2008.

[16] Coen Bron and Joep Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, September 1973.

[17] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 89–96, New York, NY, USA, 2005. ACM.

[18] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November 2002.

[19] Frank Buschmann, Kevlin Henney, and Douglas C. Schmidt. *Pattern-Oriented Software Architecture, Volume 4: A Pattern Language for Distributed Computing*. Wiley, Chichester, UK, 2007.

[20] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-oriented Software Architecture: A System of Patterns*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[21] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 129–136, New York, NY, USA, 2007. ACM.

[22] Na Chang, Mhd Irvan, and Takao Terano. A tv program recommender framework. *Procedia Computer Science*, 22(0):561 – 570, 2013. 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013.

[23] James Cheng, Yiping Ke, Ada Wai-Chee Fu, Jeffrey Xu Yu, and Linhong Zhu. Finding maximal cliques in massive networks. *ACM Trans. Database Syst.*, 36(4):21:1–21:34, December 2011.

[24] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[25] Bin Cui, Anthony K.H. Tung, Ce Zhang, and Zhe Zhao. Multiple feature fusion for social media applications. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 435–446, New York, NY, USA, 2010. ACM.

[26] Bin Cui, Ce Zhang, and Gao Cong. Content-enriched classifier for web video classification. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 619–626, New York, NY, USA, 2010. ACM.

[27] Fabio Santos da Silva, Luís Gustavo Pacola Alves, and Graça Bressan. PersonalTVware: A proposal of architecture to support the context-aware personalized recommendation of TV programs. In *Proceedings of the seventh european conference on European interactive television conference*, pages 1–4, New York, NY, USA, 2009. ACM.

[28] Howe Daniel. Ritawn. Disponível em: `http://rednoise.org/rita/wordnet/documentation/index.htm/`, acessado em: 02 Fevereiro de 2014.

[29] José Estáquio Rangel de Queiroz and Herman Martins Gomes. Introdução ao processamento digital de imagens. *RITA*, 13(2):11–42, 2006.

[30] Harvey M. Deitel and Paul J. Deitel. *Java How to Program (6th Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2004.

[31] Paul J. Deitel. *C++ How to Program. P.J. Deitel, H.M. Deitel*. Pearson Education, 7th edition, 2010.

[32] Ramez Elmasri and Shamkant Navathe. *Fundamentals of Database Systems*. Addison-Wesley Publishing Company, USA, 6th edition, 2010.

[33] Ronen Feldman and James Sanger. *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, New York, NY, USA, 2006.

[34] Christiane Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[35] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Mymedialite: A free recommender system library. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 305–308, New York, NY, USA, 2011. ACM.

[36] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[37] Kristen Grauman and Bastian Leibe. *Visual Object Recognition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.

[38] Asela Gunawardana and Guy Shani. A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res.*, 10:2935–2962, December 2009.

[39] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. Social media recommendation based on people and tags. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 194–201, New York, NY, USA, 2010. ACM.

[40] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[41] Yumiko Hara, Yumiko Tomomune, and Maki Shigemori. Categorization of japanese tv viewers based on program genres they watch. *User Modeling and User-Adapted Interaction*, 14(1):87–117, February 2004.

[42] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[43] Vimeo Inc. Vimeo, your videos belong here. Disponível em: `http://vimeo.com/`, acessado em: 02 Fevereiro de 2014.

[44] Mecenas Ivan. *Java 2 - Fundamentos Swing e JDBC*. Alta Books, 1th edition, 2005.

[45] Didion John. Jwnl (java wordnet library). Disponível em: `http://sourceforge.net/projects/jwordnet/`, acessado em: 02 Fevereiro de 2014.

[46] Dhiraj Joshi, Ritendra Datta, Ziming Zhuang, W. P. Weiss, Marc Friedenberg, Jia Li, and James Ze Wang. Paragrab: A comprehensive architecture for web image management and multimodal querying. In Umeshwar Dayal, Kyu-Young Whang, David B. Lomet, Gustavo Alonso, Guy M. Lohman, Martin L. Kersten, Sang Kyun Cha, and Young-Kuk Kim, editors, *VLDB*, pages 1163–1166. ACM, 2006.

[47] Ross Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. AMS, 1980.

[48] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[49] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[50] Yehuda Koren and Robert M. Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. 2011.

[51] Gerald Kowalski and Mark T. Maybury. *Information Storage and Retrieval Systems: Theory and Implementation*. Kluwer Academic Publishers, Norwell, MA, USA, 2nd edition, 2000.

[52] Jaroslav Kuchař and Tomáš Kliegr. Gain: Web service for user tracking and preference learning - a smart tv use case. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 467–468, New York, NY, USA, 2013. ACM.

[53] Robert Laganière. *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing, May 2011.

[54] Pasquale Lops, Marco Gemmis, and Giovanni Semeraro. Content-based Recommender Systems: State of the Art and Trends Recommender Systems Handbook. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, chapter 3, pages 73–105. Springer US, Boston, MA, 2011.

[55] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[56] Hangzai Luo, Jianping Fan, and Daniel A. Keim. Personalized news video recommendation. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pages 1001–1002, New York, NY, USA, 2008. ACM.

[57] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics Reports*, 519(1):1 – 49, 2012. Recommender Systems.

[58] Kazuhisa Makino and Takeaki Uno. New algorithms for enumerating all maximal cliques. In Torben Hagerup and Jyrki Katajainen, editors, *SWAT*, volume 3111 of *Lecture Notes in Computer Science*, pages 260–272. Springer, 2004.

[59] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[60] Judith Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, February 2004.

[61] Tao Mei, Bo Yang, Xian S. Hua, and Shipeng Li. Contextual Video Recommendation by Multimodal Relevance and User Feedback. *ACM Trans. Inf. Syst.*, 29, April 2011.

[62] Tao Mei, Bo Yang, Xian-Sheng Hua, Linjun Yang, Shi-Qiang Yang, and Shipeng Li. Videoreach: An online video recommendation system. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 767–768, New York, NY, USA, 2007. ACM.

[63] Donald Metzler and W. Bruce Croft. A markov random field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 472–479, New York, NY, USA, 2005. ACM.

[64] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[65] Cataldo Musto, Fedelucio Narducci, Pasquale Lops, Giovanni Semeraro, Marco de Gemmis, Mauro Barbieri, Jan H. M. Korst, Verus Pronk, and Ramon Clout. Enhanced semantic tv-show representation for personalized electronic program guides. In Judith Masthoff, Bamshad Mobasher, Michel C. Desmarais, and Roger Nkambou, editors, *UMAP*, volume 7379 of *Lecture Notes in Computer Science*, pages 188–199. Springer, 2012.

[66] Docs OpenCV. Video input with opencv and similarity measurement. Disponível em: `http://docs.opencv.org/doc/tutorials/highgui/`

video-input-psnr-ssim/video-input-psnr-ssim.html/, acessado em: 02 Feveiro de 2014.

[67] D. O'Sullivan, B. Smyth, D. Wilson, K. McDonald, and A. Smeaton. Interactive Television Personalization. chapter 4, pages 73–91. Kluwer, 2004.

[68] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11):10059 – 10072, 2012.

[69] Alonso Patron-Perez. Tv human interaction dataset. Disponível em: http://www.robots.ox.ac.uk/~alonso/tv_human_interactions.html/, acessado em: 02 Fevereiro de 2014.

[70] Valentina Pullano, Alessandro Vanelli-Coralli, and Giovanni E. Corazza. PSNR evaluation and alignment recovery for mobile satellite video broadcasting. *2012 6th Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC)*, pages 176–181, September 2012.

[71] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to Recommender Systems Handbook. In Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, chapter 1, pages 1–35. Springer, Boston, MA, 2011.

[72] Salakhutdinov Ruslan. Multimodal learning with deep boltzmann machines. Disponível em: http://videolectures.net/nips2012_salakhutdinov_multimodal_learning/, acessado em: 02 Fevereiro de 2014.

[73] Salakhutdinov Ruslan. The stanford natural language processing group. Disponível em: http://nlp.stanford.edu/software/corenlp.shtml/, acessado em: 02 Fevereiro de 2014.

[74] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[75] Xiaowei Shi and Jin Hua. An adaptive preference learning method for future personalized tv. In *Integration of Knowledge Intensive Multi-Agent Systems, 2005. International Conference on*, pages 260–264, 2005.

[76] Barry Smyth and Paul Cotter. Case-Studies on the Evolution of the Personalized Electronic Program Guide. pages 53–71. 2004.

[77] I. Sobel and G. Feldman. A 3x3 Isotropic Gradient Operator for Image Processing. Never published but presented at a talk at the Stanford Artificial Project, 1968.

[78] Ian Sommerville. *Software Engineering*. Addison-Wesley, Harlow, England, 9 edition, 2010.

[79] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep boltzmann machines. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2231–2239. 2012.

[80] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep boltzmann machines. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *NIPS*, pages 2231–2239, 2012.

[81] Volker Stix. Finding all maximal cliques in dynamic graphs. *Comput. Optim. Appl.*, 27(2):173–186, February 2004.

[82] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[83] Flick Team. Metadados de programação de tv. Disponível em: `http://www.flickr.com/`, acessado em: 02 Fevereiro de 2014.

[84] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[85] Christian Wartena, Wout Slakhorst, Martin Wibbels, Zeno Gantner, Christoph Freudenthaler, Chris Newell, and Lars Schmidt-Thieme. Keyword-based tv program recommendation. In *Proceedings of the 9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems (ITWP'11)*, volume 756 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

[86] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[87] Yu Xin and Harald Steck. Multi-value probabilistic matrix factorization for ip-tv recommendations. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 221–228, New York, NY, USA, 2011. ACM.

[88] Mengxi Xu, Shlomo Berkovsky, Sebastien Ardon, Sipat Triukose, Anirban Mahanti, and Irena Koprinska. Catch-up tv recommendations: Show old favourites and find new ones. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 285–294, New York, NY, USA, 2013. ACM.

[89] Bo Yang, Tao Mei, Xian-Sheng Hua, Linjun Yang, Shi-Qiang Yang, and Mingjing Li. Online video recommendation based on multimodal fusion and relevance feedback. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pages 73–80, New York, NY, USA, 2007. ACM.

[90] Jinhui Yuan, Huiyi Wang, Lan Xiao, Wujie Zheng, Jianmin Li, Fuzong Lin, and Bo Zhang. A formal study of shot boundary detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(2):168–186, 2007.

[91] Hongguang Zhang and Shibao Zheng. Personalized tv program recommendation based on tv-anytime metadata. In *Consumer Electronics, 2005. (ISCE 2005). Proceedings of the Ninth International Symposium on*, pages 242 – 246, june 2005.

[92] J Zimmerman, K Kurapati, A Buczak, D Schaffer, S Gutta, and J Martino. TV personalization system - Design of a TV show recommender engine and interface.

In Ardissono, L and Kobsa, A and Maybury, M, editor, *PERSONALIZED DIGITAL TELEVISION: TARGETING PROGRAMS TO INDIVIDUAL VIEWERS*, HUMAN-COMPUTER INTERACTION SERIES, pages 27–51, PO BOX 17, 3300 AA DORDRECHT, NETHERLANDS, 2004. SPRINGER. 3rd Conference on User Modeling, Johnstown, PA, JUN 23, 2003.

Apêndice A

Determinando Os Parâmetros na Detecção de Mudança de Câmera

Um experimento simples foi desenvolvido para investigar a redução de dimensionalidade de quadros no processamento de vídeo das técnicas PSNR e SSIM.

Como não se dispõe de uma base de dados com informações sobre mudanças de câmera, o experimento foi realizado por meio de um estudo de caso, ou seja apenas observando os sujeitos e inferindo os melhores parâmetros para a seleção de mudanças de câmera.

O objetivo do experimento é identificar quais os melhores parâmetros para as técnicas PSNR e SSIM.

No experimento os dois algoritmos foram aplicados em conjunto. O PSNR foi aplicado primeiro por ser mais rápido e o SSIM como desempate como descrito em [66]. São necessários quatro limiares, considerando vídeos coloridos, um para o PSNR e três para o SSIM (um para cada canal de cor). Por questões de simplicidade foi utilizado o mesmo limiar para cada canal de cor na técnica SSIM. Limiares foram testados por tentativa e erro e os que mostraram melhores resultados estão presentes na Tabela A.1.

Tabela A.1: Limiares utilizados na detecção de mudanças de câmera para a abordagem do trabalho

PSRN	**SSIM**
22	65

As variáveis independentes do experimento são os limiares do PSRN e SSIM e a variável

resposta é o indicativo do resultado, 1 se a mudança de câmera foi detectada corretamente, 0 caso contrário.

Os sujeitos do experimento são os programas de TV. Para cada programa é aplicado o algoritmo de detecção de mudança de câmera e observado o resultado. Varia-se o limiar das técnicas quando o resultado obtido não é satisfatório.

Uma característica encontrada durante essa etapa é que cada tipo de programa jornal, futebol, entrevista, etc. possui limiares próprios. Na solução desenvolvida, um único limiar foi compartilhado por todos os programas. No futuro, pode-se construir um algoritmo que encontre limiares adaptativos para cada tipo de programa.

Por uma análise subjetiva dos resultados, a junção dos algoritmos possui bom desempenho na detecção de mudança de câmera. Também foi verificado que apenas 1% dos quadros é suficiente para descrever um programa.

Empiricamente, existe um problema quando se trabalha com programas com bastante movimento em cenas consecutivas como esporte e inícios de programas de TV com grande variação de iluminação.

Apêndice B

Descritores de Características Visuais na Representação do Programa

Foram desenvolvidos experimentos no sentido de identificar aspectos relacionados ao uso de características visuais para recomendação. Os experimentos abordam a capacidade dos descritores de características em representar o vídeo.

Experimento 1

O experimento foi realizado com o objetivo de identificar a acurácia do uso de descritores de características para detecção de eventos em vídeo. Os resultados do experimento são importantes, pois fornecem indicativos da capacidade deles para representar programas de TV, próximo experimento.

Como não se tem uma base de dados voltada para o ambiente de TV, os dados foram coletados de uma base que contém interações humanas em séries de TV disponíveis em [69]. Os dados são compostos por quatro tipos de interações - aperto de mão, toque cinco, abraço e beijo - (Figura B.1 [69]). O banco de dados é composto de 300 vídeos coletados de mais de 20 programas de TV. Cada quadro é rotulado com o evento ou o indicativo de não ocorrência.

Para cada categoria de evento foi construído um classificador SVM usando a técnica *Bag of Keypoints* e a abordagem um-versus-todos (do inglês *one-vs-all*). O resultado do experimento pode ser visto na Tabela B.1.

Figura B.1: Ilustração dos eventos utilizados para classificar os programas de TV em categorias.

Tabela B.1: Matriz de confusão para a classificação de eventos usando descritores de características visuais

#	Aperto de mão	Toque cinco	Abraço	Beijo
Aperto de mão	0.935	0	0.015	0.048
Toque cinco	0	0.931	0.017	0.05
Abraço	0	0	0.953	0.046
Beijo	0	0	0.017	0.982

Experimento 2

A pergunta que se busca responder é: os descritores extraídos dos programas são úteis para descrever programas de TV?

Intuitivamente, são, pois cada programa possui um conjunto de eventos característicos. Por exemplo, um jogo de futebol possui jogadores, campo e torcida, assim como os demais programas possuem um conjunto de características próprias.

Para cada programa é montado um histograma que corresponde à frequência de cada um dos eventos no vídeo (Figura B.2). Os histogramas foram usados para classificar programas de acordo com eventos neles contidos.

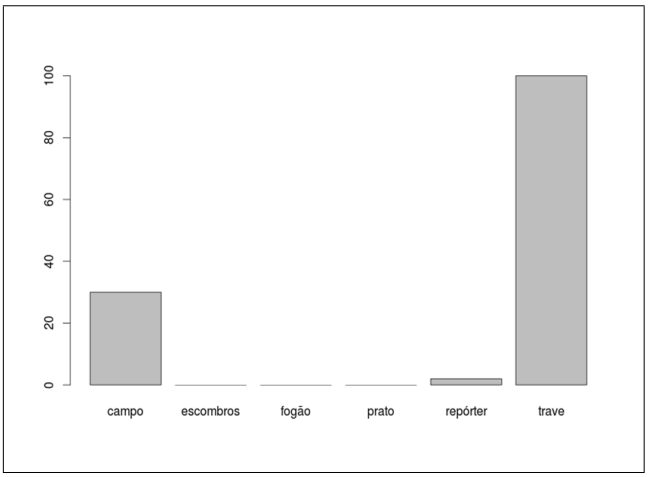

Figura B.2: Ilustração de um jogo de futebol por meio de um histograma dos eventos nele contido.

A base de dados utilizada no experimento foi construída manualmente (durante um pe-

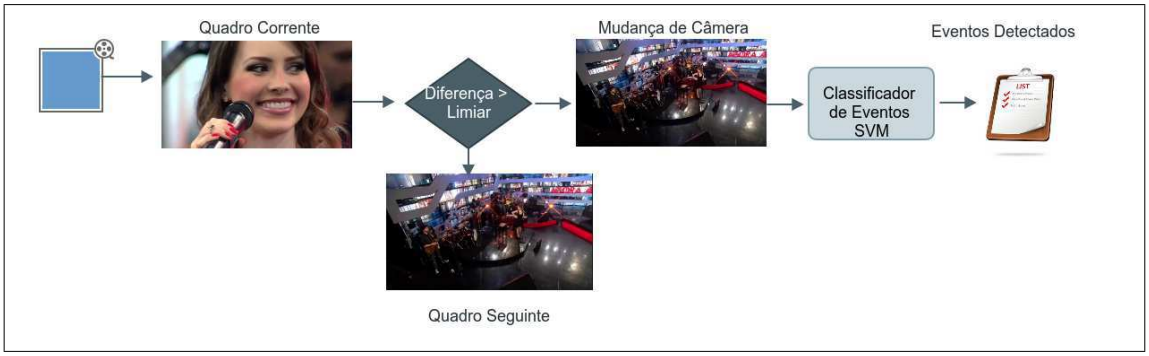

Figura B.3: Esquematização da detecção de mudança de câmera.

ríodo de três meses) e contém apenas um conjunto reduzido de imagens para cada evento e um conjunto reduzido de eventos. Por isso, foram construídos classificadores apenas para os eventos disponíveis, são eles:

- Presença de apresentadores de jornal;
- Presença de utensílio culinário (panela);
- Presença de torcida;
- Presença de repórteres;
- Presença de entrevista.

Na Tabela B.2 podem ser vistos os eventos coletados e o total de imagens para cada evento.

Tabela B.2: Eventos e quantidade de imagens para a classificação de eventos usando características visuais em ambientes de TV Digital

Eventos	**Imagens Disponíveis**
Presença de Apresentadores de Jornal	751
Presença de utensílio culinário	1605
Presença de Torcida	1078
Presença de Repórteres	1479
Presença de Entrevista	1050

Tabela B.3: Exemplificação dos eventos utilizados na classificação de programas

Apresentadores de jornal	Panela	Torcida	Repórteres	Entrevista
				
				

Exemplos dos eventos utilizados podem ser vistos na Tabela B.3.

Assim como os eventos, apenas um conjunto reduzido das categorias de programas de TV foi utilizado na classificação de programas baseada nos eventos, são elas:

- Culinária;
- Jornal;
- Futebol.

Para classificar os programas foi usado o Weka [40]. Diferentes classificadores foram aplicados e os resultados podem ser vistos na Tabela B.

Os dados foram validados em termos da métrica Precisão que mede a porcentagem de acerto do classificador.

Tabela B.4: Resultado da aplicação de diferentes classificadores para categorização de programas usando o histograma de eventos neles contidos

Classificadores utilizados	Precisão
k-NN (K = 1)	0.828
Árvore de decisão	0.942
Redes bayesianas	0.942
Redes neurais	0.942

Os resultados demonstram que o uso dos eventos em TV é significativo para classificação de programas, consequentemente são úteis na recomendação de programas de TV.

Apesar de relevantes, seu uso nem sempre é possível, por exemplo no contexto de TV digital é necessária a montagem da base de dados, dada a inexistência de uma fonte de dados disponível com as informações.

Para realização do experimento anteriormente descrito foram selecionados um conjunto de vídeos do ambiente social multimídia *Vimeo* [43] e um conjunto de imagens disponíveis na Web que foram usados para construir os classificadores.

O uso de eventos é limitado a um conjunto de eventos pré-estabelecidos. Dessa forma, um novo classificador deve ser criado para cada evento que surge. Por isso, as pesquisas em ambientes multimídias trabalham com abstrações dos itens como os descritores de características. No trabalho, essas abstrações foram utilizadas no modelo de recomendação.

Apêndice C

Artefatos Arquiteturais

A arquitetura foi organizada em três camadas logicamente conectadas, mas com responsabilidades distintas (Figura C.1).

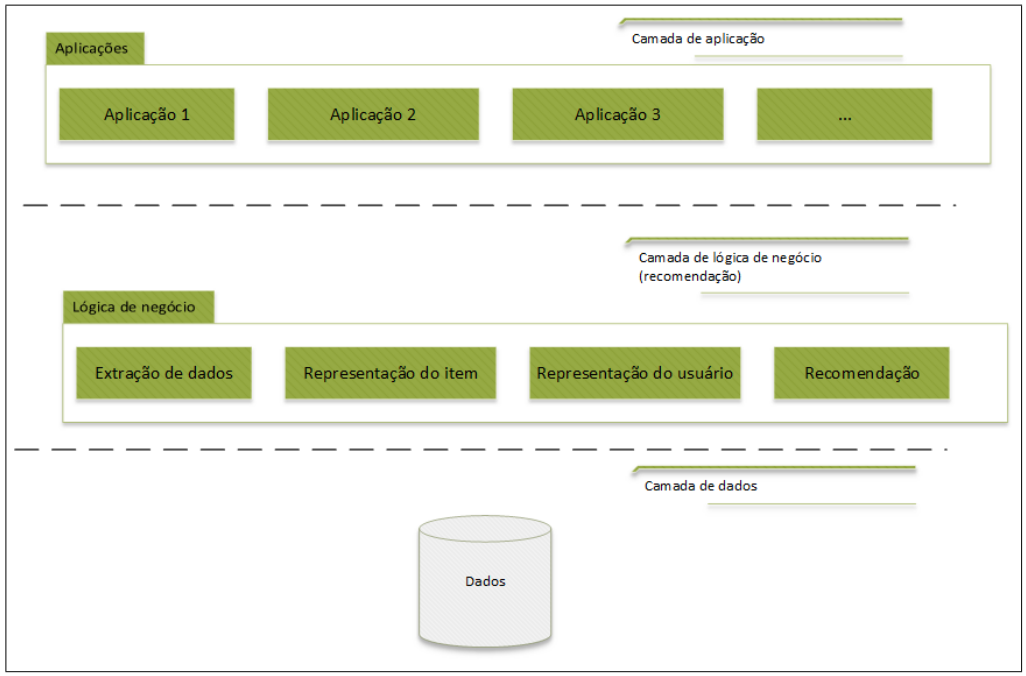

Figura C.1: Ilustração da divisão da arquitetura em camadas.

A divisão da arquitetura em camada favorece a divisão de responsabilidade entre os componentes, dessa forma permitindo o reuso de software. Esta estrutura segue o padrão arquitetural Camadas [20]. O padrão Camadas ajuda a estruturar aplicações que podem ser decompostas em grupos de subtarefas em que cada grupo de tarefa oferece um nível particular

de abstração [20]. Um resumo das principais características do padrão Camadas pode ser encontrado na Tabela C.1.

Tabela C.1: Resumo do padrão Camadas

Descrição	O padrão Camadas ajuda a estruturar aplicações que podem ser decompostas em grupos de subtarefas em que cada grupo de tarefa oferece um nível particular de abstração
Quando usar	Em sistemas grandes que precisam ser decompostos.
Vantagens	Reusabilidade – se uma camada individual oferece um bom nível de abstração e possui uma interface bem definida, a camada pode ser reusada em múltiplos contextos. Outros benefícios incluem suporte à padronização, as dependências são locais e as camadas podem ser trocadas
Desvantagens	Propagação de mudanças e trabalho desnecessário – se uma camada oferece serviços duplicados isso acarreta perca de desempenho.

A camada de lógica de negócio fornece serviços de recomendação de programas para aplicações de TV interessadas. Essa estrutura é dividida basicamente em quatro componentes, Extração de dados, Representação do item, Representação do usuário e Recomendação. A Extração de dados é o componente responsável pela coleta de características do conteúdo do programa que são usadas para realizar a recomendação. Os dados extraídos podem assumir diferentes tipos, tais como textuais, visuais e assim por diante. O componente Representação do item utiliza os dados extraídos para construir um modelo do programa.

O componente Perfil do usuário utiliza as informações sobre os programas que o usuário interagiu para construir o perfil dele.

O componente Recomendação utiliza o perfil do usuário para encontrar os programas mais relevantes. Diferentes abordagens de recomendação podem ser empregadas.

A camada de dados armazena as informações necessárias para a realização da recomendação (Figura C.2). Esta camada é modelada em um banco de dados relacional [32].

No modelo relacional empregado o usuário possui um nome e pode assistir a quantos programas de TV desejar. Um programa possui id, descrição e um conjunto de características. Uma característica contém um tipo de dado específico (i.e. texto, vídeo, etc), um id e

um valor que corresponde à frequência do termo no programa. Uma descrição das entidades pode ser vista na Tabela C.3. A estrutura de organização da camada de dados segue o padrão arquitetural Objeto de Acesso aos Dados [19] (do inglês *Data Access Object* - DAO). Um resumo como as principais características do padrão pode ser visto na Tabela C.2.

Tabela C.2: Resumo do padrão Objecto de Acesso aos Dados

Descrição	Introduz uma camada entre a aplicação (lógica de negócio) e o banco de dados.
Quando usar	Quando se deseja desacoplar a aplicação do banco de dados.
Vantagens	A estratégia de acesso ao banco de dados pode ser alterada.

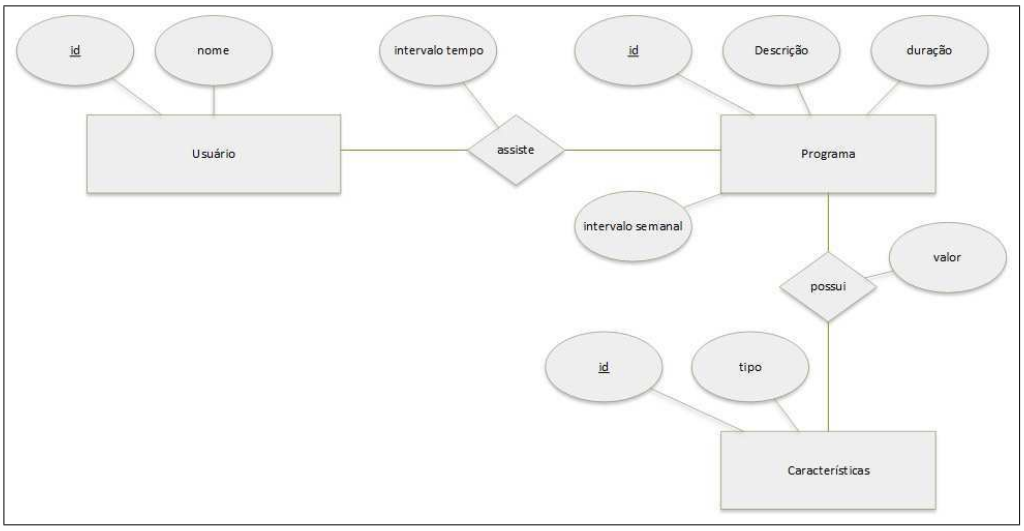

Figura C.2: Diagrama entidade relacionamento simplificado.

Na camada de aplicação estão os diferentes aplicativos (clientes) que desejam utilizar o serviço de recomendação.

A interação entre as camadas ocorre da forma seguinte: a camada de aplicação requisita serviços de recomendação à camada de lógica de negócio que, por sua vez utiliza a camada de dados para atender a requisição (Figura C.3).

Tabela C.3: Dicionário de dados

Usuário: pessoa que deseja receber recomendação.
Programa: programa de TV.
Característica: característica do programa. Cada característica possui um tipo de dado e valor que corresponde a sua frequência no programa. Como exemplo de uma característica do tipo textual, tem-se "jornalismo" e como de uma característica visual, tem-se 1000 que correspondente ao id de um descritivo visual específico no vocabulário.

C.1 Implantação

A arquitetura é implantada no lado servidor oferecendo serviços de recomendação para diferentes aplicativos de TV (Figura C.4). Dessa forma, a arquitetura funciona como uma infraestrutura de software usada para prover serviços de recomendação. Como os componentes da arquitetura estão no lado servidor eles podem requisitar serviços mediante a chamada de método.

Nessa estrutura, aplicativos de TV como exemplo fictício: "Diga-me, o que assistir em seguida" podem usar o serviço de recomendação.

C.2 Organização Geral da Arquitetura

A organização geral da arquitetura pode ser vista na Figura C.5.

A arquitetura foi dividida em dois subsistemas, Filtragem de dados e Descoberta de conhecimento.

Essa estrutura de organização segue dois padrões arquiteturais, Repositório [78] e Cliente-Servidor [78]. O padrão Repositório é utilizado em aplicações em que o conjunto de componentes necessitem compartilhar dados [78]. Esse padrão é ideal em domínios em que os dados são gerados por um componente e consumidos por outros; dessa forma não necessitando da troca de dados entre eles [78]. Para o emprego de modelos multimodais que utilizem tipo de dado que demande muito recurso computacional para ser computado como extração de vídeo essa estrutura é aplicável. Sendo assim, modelos são computados *offline* e armazenados para posterior uso. Sommeville [78] oferece um resumo com as principais

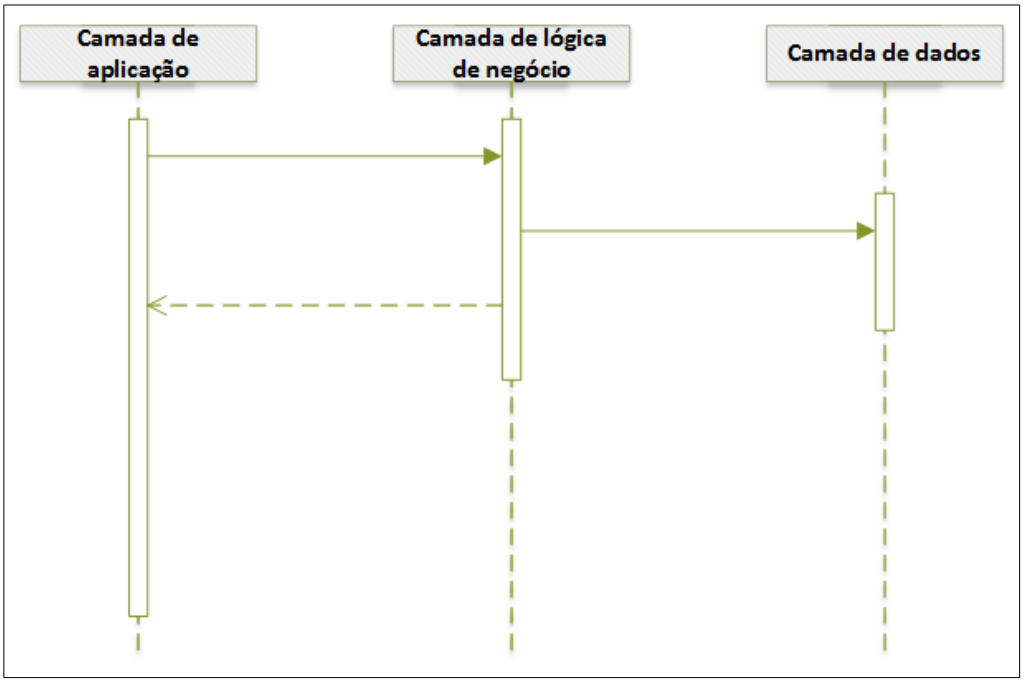

Figura C.3: Ilustração da interação entre as camadas.

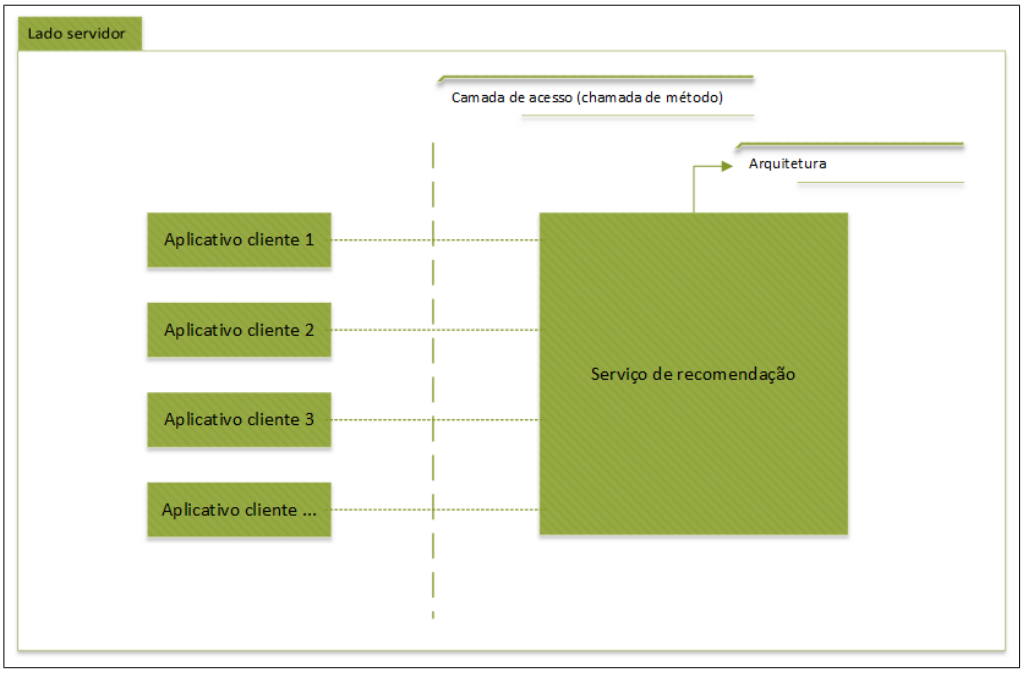

Figura C.4: Ilustração da implantação arquitetura.

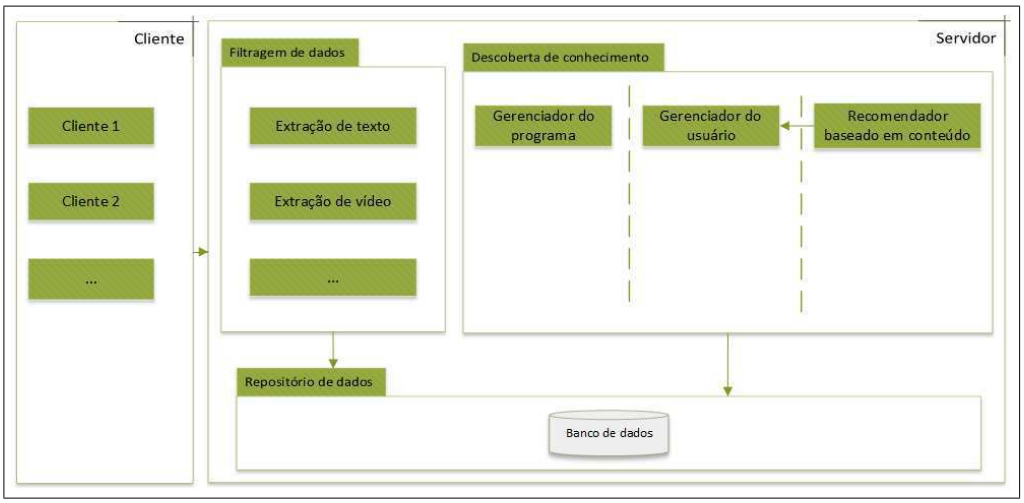

Figura C.5: Ilustração da organização geral da arquitetura.

características desse padrão (Tabela C.4).

Tabela C.4: Resumo do padrão Repositório

Descrição	Todos os dados dos sistemas são gerenciados por um repositório central que é acessível por todos os componentes. Os componentes não interagem diretamente, somente através do repositório.
Quando usar	Esse padrão é usado quando se tem um sistema em que um grande volume de informação é gerado e deve ser armazenado por um período longo.
Vantagens	Os componentes são independentes - eles não precisam saber da existência de outros componentes. As mudanças feitas em um componente são propagadas por todo o sistema.
Desvantagens	O repositório é um ponto único de acesso, dessa forma uma falha afeta todo o sistema.

No padrão Cliente-Servidor as funcionalidades do sistema são organizadas como serviços e são disponibilizadas em servidores. Os clientes são usuários desses serviços e acessam os servidores para fazerem uso deles [78]. Essa estrutura organizacional foi utilizada para diminuir a carga no lado cliente. Dessa forma, operações que demandam maior poder computacional; como os serviços de recomendação são oferecidas como serviços no lado servidor.

Sommeville [78] fornece um resumo da arquitetura Cliente-Servidor (Tabela C.5).

Tabela C.5: Resumo do padrão Cliente-Servidor

Descrição	Na arquitetura Cliente-Servidor as funcionalidades do sistema são organizadas como serviços. Em que cada serviço é entregue por um servidor em separado. Os clientes são usuário dos serviços e os acessam para fazerem uso deles.
Quando usar	Usado quando as funcionalidades precisam ser acessadas de um grande número de localizações. Devido à capacidade de replicação, pode ser usado quando a carga no servidor é variável.
Vantagens	Os serviços podem ser distribuídos na rede. Funcionalidades em geral podem ser acessadas por todos os clientes que não precisam implementar todos os serviços.
Desvantagens	O servidor é o ponto central de falha, podendo ser alvo de ataques. O desempenho é imprevisível, pois depende da disponibilidade de rede.

C.3 Interação entre os componentes

A interação entre os componentes da arquitetura ocorre da forma seguinte: os dados são extraídos para a modalidade específica e são armazenados no repositório central. O gerenciador de programas usa os dados extraídos para construir o modelo do programa de forma *offline*. O Gerenciador de perfil constrói o perfil do usuário baseado nos programas assistidos por ele. O Recomendador baseado em conteúdo utiliza o perfil do usuário e a representação do programa para gerar a lista de recomendação (Figura C.6).

C.4 Filtragem de dados

O subsistema Filtragem de dados é responsável pela extração de dados do conteúdo do item. Esse subsistema é composto por diferentes módulos, cada um responsável pela extração das características de interesse para o tipo de dado particular. Na Figura C.5 pode ser visto dois

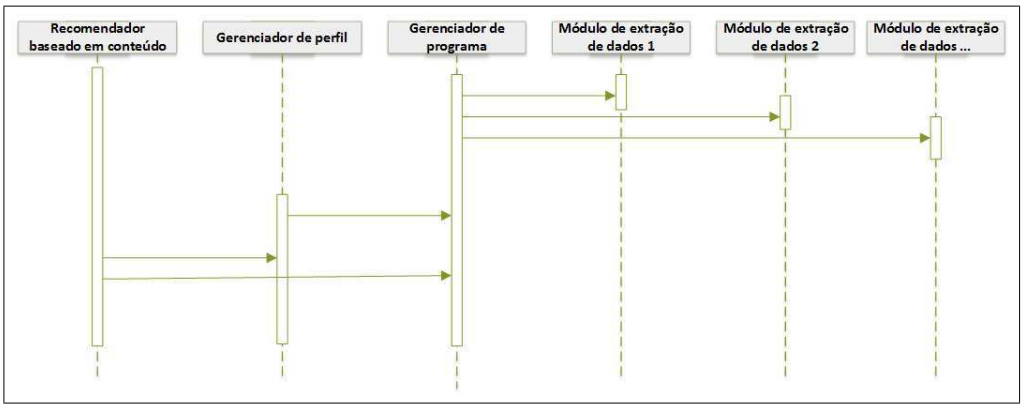

Figura C.6: Ilustração da interação entre os módulos do sistema. A figura mostra a interação direta entre os módulos, porém a interação é realizada através do repositório.

tipos de módulos, Extração de texto e Extração de vídeo, porém a adição dos módulos fica a cargo do desenvolvedor.

C.4.1 Módulo de Extração de texto

A extração de texto consiste em encontrar termos que sejam relevantes para detecção de padrão salientes como discriminar os itens que o usuário gosta. Em algumas aplicações é necessária a redução da palavra a sua forma básica [59]. Isso decorre por questões gramáticas, pois a palavra pode assumir diferentes derivações dependendo do contexto onde é empregada; algumas recebem desinência de plural, outras permanecem no singular e assim por diante.

Dois processos podem ser utilizados para redução da palavra a sua forma básica, são eles *stemming* e lematização [59]. *Stemming* se refere a um conjunto de processos que removem os afixos das palavras, em contrapartida a lematização usa um vocabulário definido e efetua uma análise morfológica da palavra, reduzindo-a a sua forma primitiva - a maneira que ela é apresentada no dicionário, também denominado lema [59].

Para análise dos processos linguísticos tanto para *stemming* quanto lematização, ferramentas, usualmente são empregadas [59]. Dentre elas, JWNL [45], Rita.WordNet [28], Stanford CoreNLP [73] e assim por diante. Cada idioma oferece diferentes regras para formação morfológica das palavras, dessa forma são necessárias ferramentas específicas para

cada língua.

Além dos dois processos vistos, outra abordagem utilizada quando se trabalha com extração de texto é a remoção de palavras não significativas como preposição, artigo e conjunção. Pois, para que a palavra tenha um valor discriminativo é necessário que ela possua uma semântica associada.

C.4.2 Módulo de Extração de vídeo

A extração de vídeo consiste em buscar uma representação visual para o item, na arquitetura proposta, programas de TV. Esse conjunto de características visuais, normalmente, é agrupado para construir o vocabulário de características visuais. As características visuais podem variar desde a intensidade de pixels em uma região $K \times K$ da imagem [25] ou modelos mais abrangentes como as bordas e os pontos de interesse [26, 80] (seção 2.2.2).

Como o vídeo é formado com um grande volume de quadros, o custo computacional para processá-los é alto. Por isso, antes que as características visuais sejam extraídas, processos como a detecção de mudança de câmera podem ser aplicados para reduzir o número de quadros que necessitam ser analisados. Dessa forma, o módulo de Extração de vídeo pode ser dividido em dois submódulos, Redutor de quadros e Extrator de características visuais (Figura C.7).

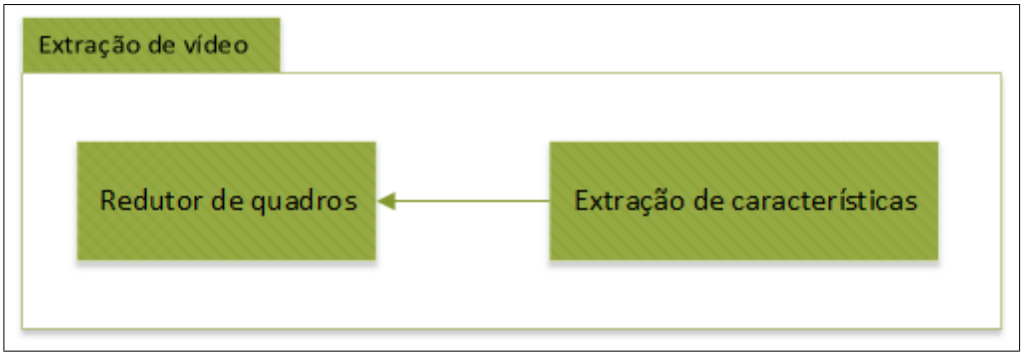

Figura C.7: Ilustração do módulo de extração de vídeo.

O processo de interação entre os submódulos pode ser visto na Figura C.8. Enquanto o vídeo é exibido, o quadro corrente é selecionado. Em seguida o Redutor de quadros elege o quadro como significativo ou não dependendo das regras de negócio e requisitos da aplica-

ção. Quando um quadro é identificado como significativo, o processo de extração de características se inicia. Caso o vídeo tenha chegado ao seu final, o processo termina, caso contrário outro quadro do vídeo é analisado.

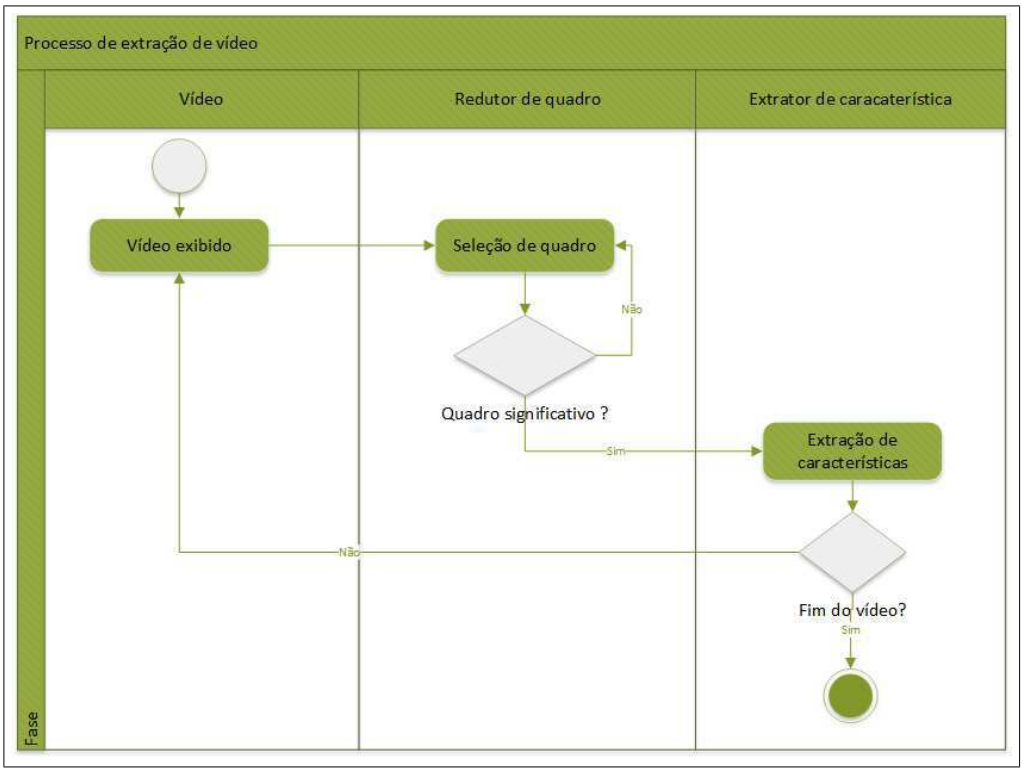

Figura C.8: Ilustração do processo de extração de quadro no módulo de Extração de vídeo.

C.5 Subsistema Descoberta de Conhecimento

A camada Descoberta de conhecimento é composta por três componentes, Gerenciador do programa, Gerenciador de perfil e Recomendador baseado em conteúdo.

C.5.1 Componente Gerenciador do programa

Na arquitetura proposta um programa pode ser visto como a agregação de diferentes modalidades (Figura C.9).

Para uma visão mais detalhada sobre modelos multimodais em sistemas de recomendação veja a seção 2.4.

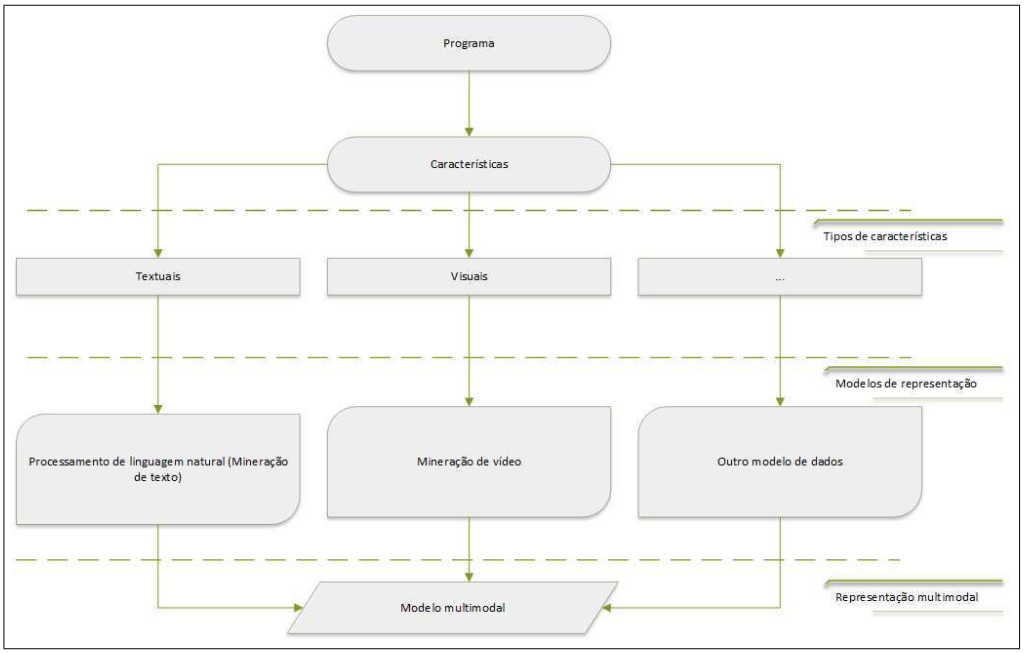

Figura C.9: Ilustração da representação do programa como um conjunto de características multimodais. Nessa representação, um programa é modelado usando diferentes tipos de dados que são posteriormente fundidos para compor o modelo de programa.

C.5.2 Componente Gerenciador de perfil

O componente Gerenciador de perfil utiliza o conjunto de programas assistidos pelo usuário para construir um modelo para distinguir os programas que o usuário possui interesse. Em geral, em sistemas de recomendação baseado em conteúdo a representação do usuário é construída como uma agregação dos programas em seu perfil. Essa representação é utilizada para calcular a similaridade entre o usuário e os demais programas não assistidos por ele. Para isso, diferentes abordagens podem ser utilizadas, tais como modelar a frequência do termo para o usuário como a soma da presença dos termos, a média do termo nos programas em seu perfil ou a combinação linear da frequência nos programas assistidos pelo usuário. Uma abordagem bastante utilizada na literatura [25, 39] é a combinação linear ponderada dos programas baseado na *timestamp* da exibição; em que programas menos recentes são penalizados.

C.5.3 Componente Recomendação baseada em conteúdo

O recomendador baseado em conteúdo utiliza o perfil do usuário e a representação do item para gerar uma lista de recomendação. No ambiente multimodal diferentes abordagens foram utilizadas para propósitos diversos, tal como classificação e recuperação de informação e foram organizados por Atrey [7], podendo ser vistas na Figura C.10. Esses modelos podem ser adaptados para o domínio de recomendação.

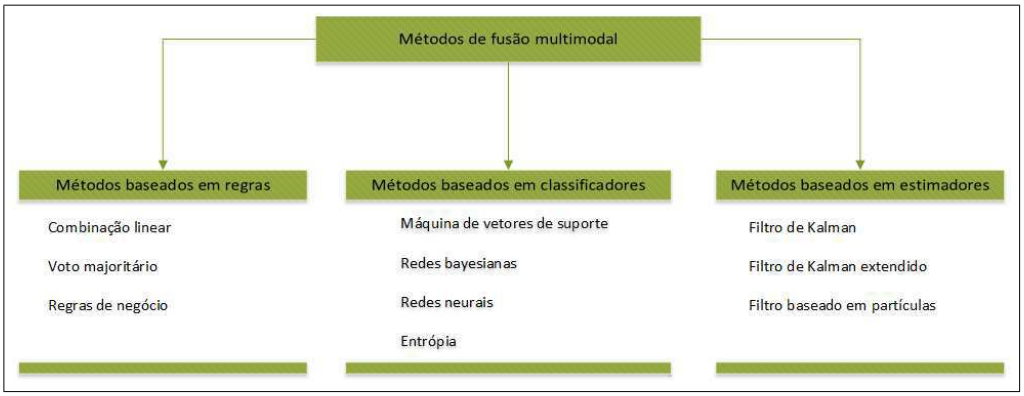

Figura C.10: Diferentes modelos de aprendizado de máquina que foram utilizados na literatura para propósitos diversos, desde a classificação e recuperação de informação.

C.6 Linguagens e Tecnologias de Desenvolvimento

Para suportar o desenvolvimento do projeto, ferramentas e linguagens de programação específicas foram utilizadas.

As linguagens utilizadas para o desenvolvimento da arquitetura são a linguagem de programação C++ [31] e Java [30]. A linguagem C++ foi utilizada para trabalhar com processamento de vídeo, por a biblioteca utilizada ser implementada nessa linguagem. A linguagem Java foi utilizada nos demais componentes da arquitetura. Para o desenvolvimento da arquitetura e da validação um conjunto de ferramentas foi utilizado, são elas:

- **OpenCV** - uma biblioteca para processamento de visão computacional com um grande número de algoritmos utilizados na área. A biblioteca possui versões para várias linguagens. No trabalho foi utilizada a versão 2.4.6 para C++;
- **JDBC** – uma biblioteca Java para comunicação com o banco de dados [44]
- **Weka** - uma biblioteca para aprendizagem de máquina que inclui um grande número de algoritmos utilizados na área. A biblioteca pode ser adaptada para várias linguagens. No trabalho foi utilizada a linguagem Java;
- **R** - Uma plataforma e linguagem para análise de dados. A plataforma foi utilizada no trabalho para aquisição de intervalos de confiança, testes e cálculo de correlação;
- **MyMediaLite** - uma biblioteca que apresenta vários algoritmos de recomendação [35].
- **JGraphT** – é uma biblioteca que disponibiliza um conjunto de objetos e algoritmos para se trabalhar com grafos [1].
- **JWNL** - JWNL (*do inglês Java WordNet Library*) é uma API para acesso ao banco de dados relacional do *Wordnet*. *WordNet* é uma ferramenta muito usada para processamento de linguagem natural. JWNL permite o desenvolvimento de aplicações na área.
- **WJ4J** - WJ4J (*WordNet Similarity for Java*) provê uma API para acesso a vários algoritmos de relação semântica (similaridade) codificados no WordNet.

C.7 Requisitos funcionais e não funcionais

No trabalho, requisitos funcionais se referem a encontrar programas de interesse para o usuário e os requisitos não funcionais estão relacionados ao desempenho medido em termos da acurácia da recomendação.

Apêndice D

Artefatos

Neste capítulo são descritos os artefatos de software usados na abordagem proposta no trabalho.

As camadas podem ser vistas na Figura D.1.

Figura D.1: Ilustração dos componentes de software usados na abordagem proposta no trabalho.

Cada camada é descrita da forma seguinte:

1. ***Similarity*** - contém componentes para calcular a similaridade entre dois GICs;
2. ***Graph*** - possui componente para construção dos GICs;
3. ***Recommender*** - contém diferentes abordagens de recomendação utilizando os GICs;
4. ***Video*** - contém artefatos para extração de características dos vídeos;

5. **Dao** - possui componentes para comunicação com o banco de dados;
6. ***Potential.function*** - contém componentes para o cálculo da função potencial;
7. ***Probability*** - possui componentes para cálculo de probabilidade utilizando os GICs;
8. ***Correlation*** - contém componentes para cálculo da correlação entre características;
9. **Nlp** - possui componentes para processamento de linguagem natural;
10. ***Object*** - contém os objetos utilizados;
11. ***Rank*** - possui componentes para aprendizado dos parâmetros do modelo;
12. ***Wup*** - contém componentes para cálculo de similaridade entre palavras utilizando a abordagem *Wu & Palmer [86]*.

D.1 Camadas

A seguir são descritas as camadas.

D.1.1 *Similarity*

Os componentes presentes na camada *Similarity* podem ser vistos na Figura D.2.

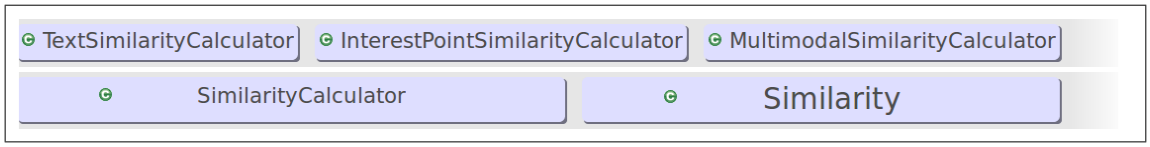

Figura D.2: Ilustração dos componentes da camada *Similarity*.

A camada *Similarity* inclui componentes para cálculo de similaridade entre GICs textuais, baseadas em pontos de interesse e multimodal, assim como um componente para representar a similaridade.

Os componentes também podem ser vistos no diagrama de classes da Figura D.3.

Os componentes presentes são os desenvolvidos para a instanciação da arquitetura, são eles similaridade entre texto, pontos de interesse e abordagem multimodal. Mais componentes podem ser adicionados a cargo do desenvolvedor.

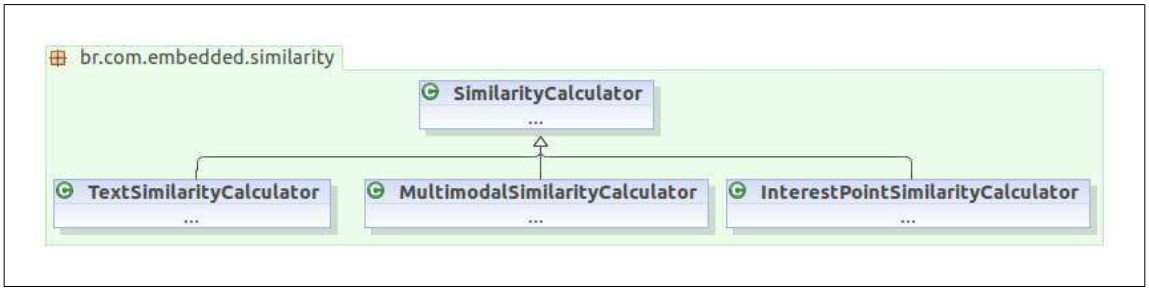

Figura D.3: Diagrama de classes da camada *Similarity*.

D.1.2 *Graph*

Os componentes presentes na camada *Graph* podem ser vistos na Figura D.4.

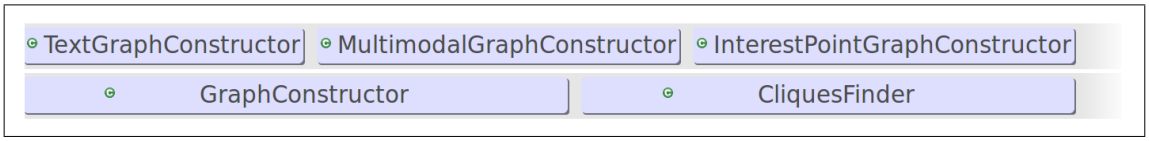

Figura D.4: Ilustração dos componentes da camada *Graph*.

A camada *Graph* inclui componentes para a criação de GICs textuais, baseadas em pontos de interesse e multimodais.

Os componentes também podem ser vistos no diagrama de classes da Figura D.5.

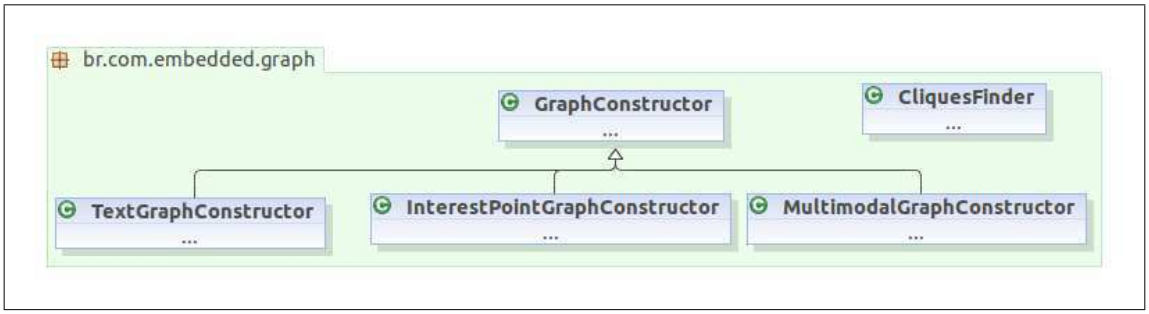

Figura D.5: Diagrama de classes da camada *Graph*.

Os componentes presentes são os desenvolvidos para a instanciação da arquitetura, são eles construtor de grafos baseado em texto, pontos de interesse e abordagem multimodal. Mais componentes podem ser adicionados a cargo do desenvolvedor. Essa camada também

possui componentes para a extração de cliques máximos do grafo.

D.1.3 *Recommender*

Os componentes presentes na camada *Recommender* podem ser vistos na Figura D.6.

Figura D.6: Ilustração dos componentes da camada *Recommender*.

A camada *Recommender* inclui componentes para recomendação usando as GICs textuais, baseadas em pontos de interesse e multimodais.

Os componentes também podem ser vistos no diagrama de classes da Figura D.7.

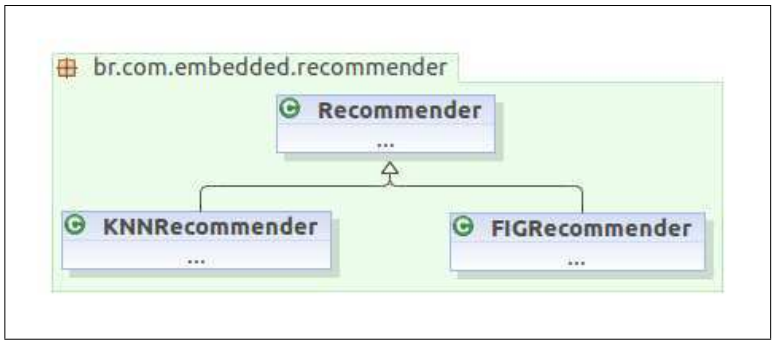

Figura D.7: Diagrama de classes da camada *Recommender*.

Os componentes presentes são os desenvolvidos para a instanciação da arquitetura. Foram desenvolvidos dois tipos de recomendadores: baseado em GICs (em inglês *Feature Interaction Graph* – FIG), descrito no trabalho e um aplicando a abordagem k-NN.

D.1.4 *Video*

Os componentes presentes na camada *Video* podem ser vistos na Figura D.8.

A camada *Video* inclui componentes para extração dos quadros chaves e gerenciamento de pontos de interesse.

Os componentes também podem ser vistos no diagrama de classes da Figura D.9.

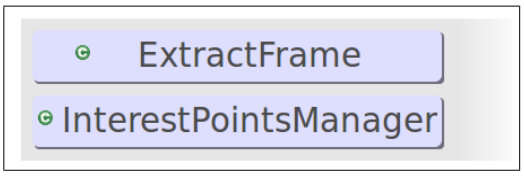

Figura D.8: Ilustração dos componentes da camada *Video*.

Figura D.9: Diagrama da camada *Video*.

D.1.5 Dao

Os componentes presentes na camada Dao podem ser vistos na Figura D.10.

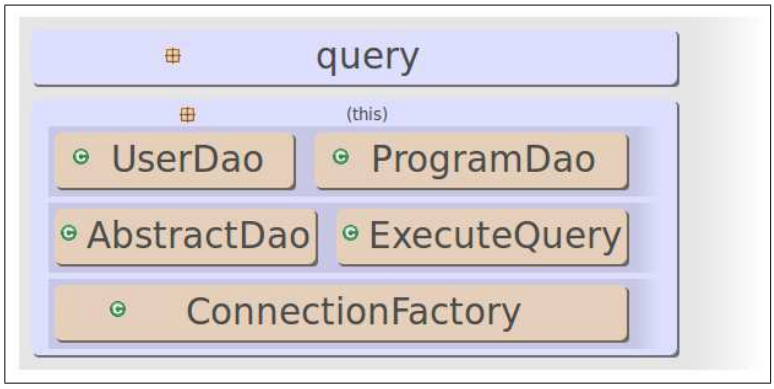

Figura D.10: Ilustração dos componentes da camada Dao.

A camada Dao inclui componentes para conexão com o banco de dados, mecanismos para seleção de entidades do banco de dados específicas, usuário e programa e um componente para consulta no banco.

Os componentes também podem ser vistos no diagrama de classes da Figura D.11.

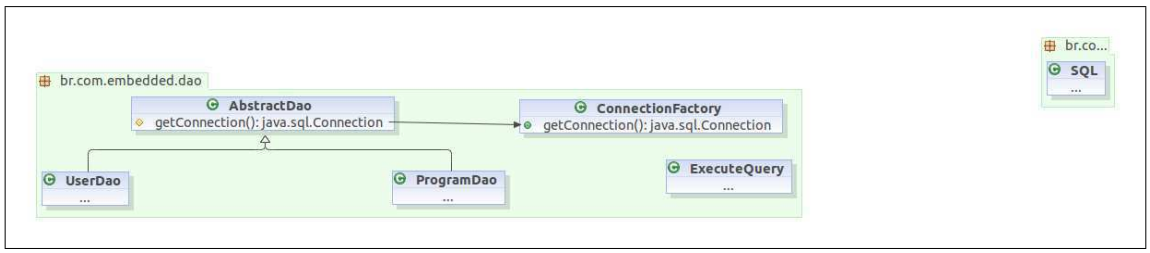

Figura D.11: Diagrama de classes da camada Dao.

D.1.6 *Potential.function*

Os componentes presentes na camada *Potential.function* podem ser vistos na Figura D.12.

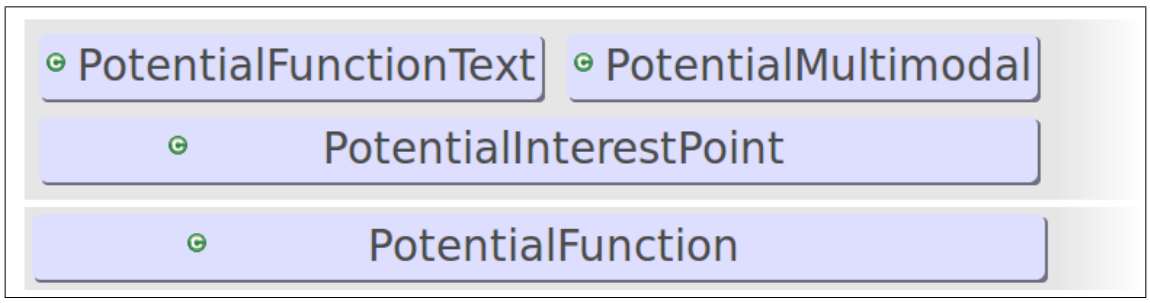

Figura D.12: Ilustração dos componentes da camada *Potential.function*.

A camada *Potential.function* inclui componentes para o cálculo da função potencial para texto, pontos de interesse e multimodal. Os componentes também podem ser vistos no diagrama de classes da Figura D.13.

D.1.7 *Probability*

Os componentes presentes na camada *Probability* podem ser vistos na Figura D.14.

A camada *Probability* inclui componentes para o cálculo da probabilidade entre as GICs.

Os componentes também podem ser vistos no diagrama de classes da Figura D.15.

Os componentes presentes são os desenvolvidos para a instanciação da arquitetura, são eles construtor de GICs baseados em texto, pontos de interesse e multimodal. Mais componentes podem ser adicionados a cargo do desenvolvedor.

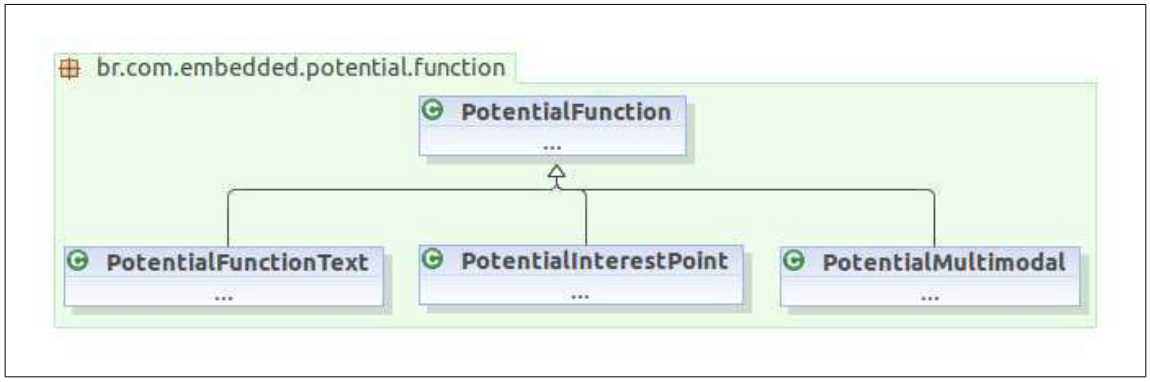

Figura D.13: Diagrama de classes da camada *Potential.function*.

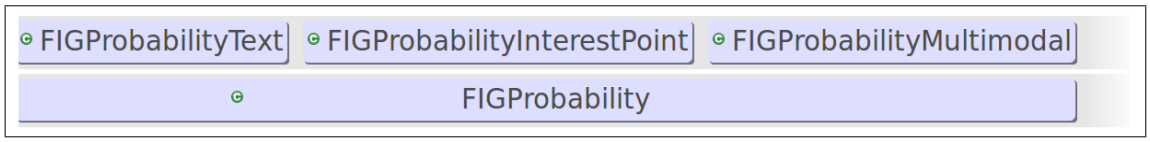

Figura D.14: Ilustração dos componentes da camada *Probability*.

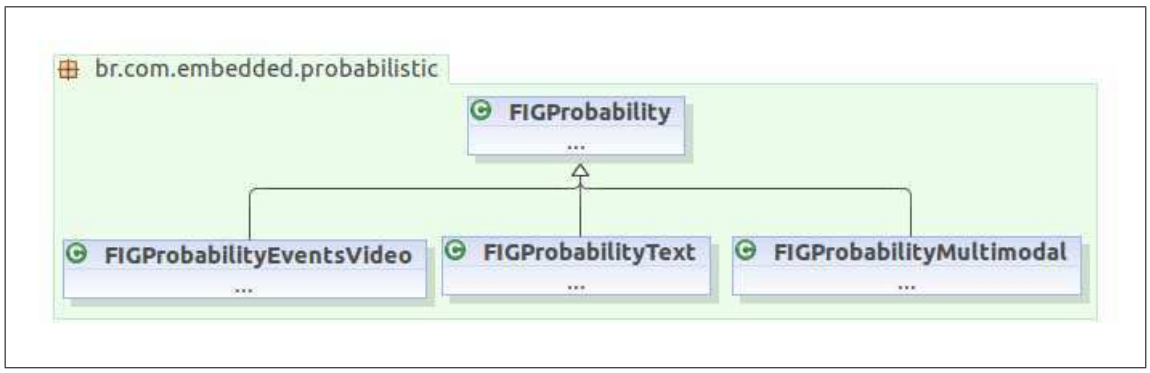

Figura D.15: Diagrama de classes da camada *Probability*.

D.1.8 *Correlation*

Os componentes presentes na camada *Correlation* podem ser vistos na Figura D.16.

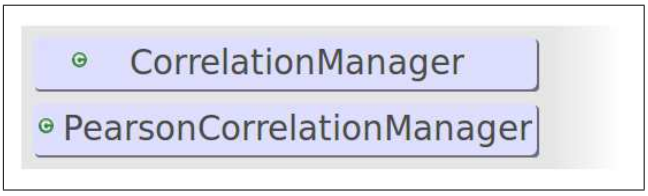

Figura D.16: Ilustração dos componentes da camada *Correlation*.

A camada *Correlation* inclui componentes para o cálculo da correlação entre as características e gerenciador da correlação.

Os componentes também podem ser vistos no diagrama de classes da Figura D.17.

Figura D.17: Diagrama de classes da camada *Correlation*.

D.1.9 Nlp

Os componentes presentes na camada Nlp podem ser vistos na Figura D.18.

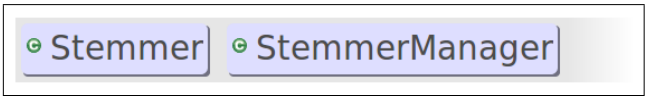

Figura D.18: Ilustração dos componentes da camada Nlp.

A camada Nlp inclui componentes para a realização de *stemming*.

Os componentes também podem ser vistos no diagrama de classes da Figura D.19.

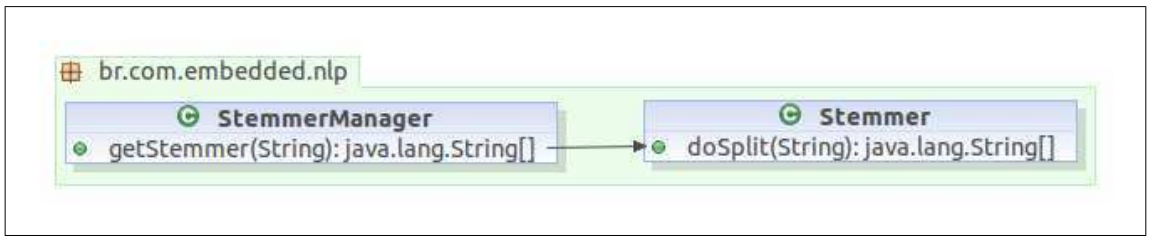

Figura D.19: Diagrama de classes da camada Nlp.

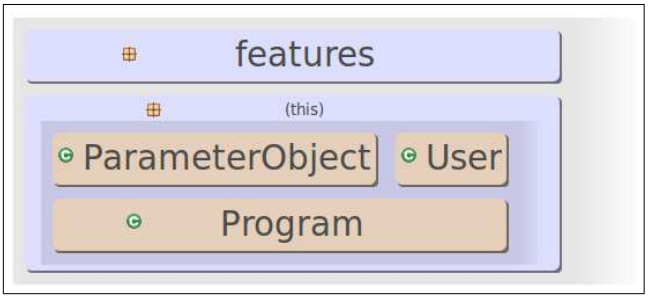

Figura D.20: Ilustração dos componentes da camada *Object*.

D.1.10 *Object*

Os componentes presentes na camada *Object* podem ser vistos na Figura D.20.

A camada *Object* inclui componentes para representar os objetos do domínio.

Os componentes também podem ser vistos no diagrama de classes da Figura D.21.

Figura D.21: Diagrama de classes da camada *Object*.

D.1.11 *Rank*

Os componentes presentes na camada *Rank* podem ser vistos na Figura D.22.

A camada *Rank* inclui componentes para a seleção dos parâmetros da função potencial.

Figura D.22: Ilustração dos componentes da camada *Rank*.

Os componentes também podem ser vistos no diagrama de classes da Figura D.23.

Figura D.23: Diagrama da camada *Rank*.

D.1.12 Wup

Os componentes presentes na camada Wup podem ser vistos na Figura D.24.

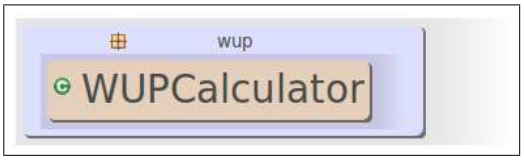

Figura D.24: Ilustração dos componentes da camada Wup.

A camada Wup inclui componentes para a computação da similaridade entre palavras usando a métrica *Wu & Palmer* [86].

Os componentes também podem ser vistos no diagrama de classes da Figura D.25.

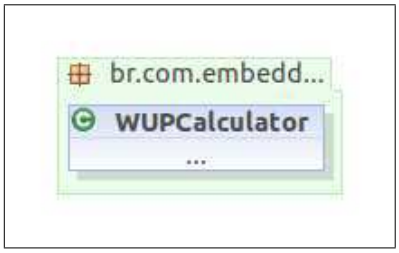

Figura D.25: Diagrama de classes da camada Wup.

Apêndice E

Artigos aceitos

A Recommendation Approach for Digital TV Systems based on Multimodal Features

Reudismam Rolim
Federal University of Campina Grande
Campina Grande, Brazil
reudismam@copin.ufcg.edu.br

Felipe Barbosa
Federal University of Campina Grande
Campina Grande, Brazil
feliperamos@copin.ufcg.edu.br

Alexandre Costa
Federal University of Campina Grande
Campina Grande, Brazil
antonioalexandre@copin.ufcg.edu.br

Giovanni Calheiros
Federal University of Campina Grande
Campina Grande, Brazil
gac@copin.ufcg.edu.br

Hyggo Almeida
Federal University of Campina Grande
Campina Grande, Brazil
hyggo@dsc.ufcg.edu.br

Angelo Perkusich
Federal University of Campina Grande
Campina Grande, Brazil
perkusic@dee.ufcg.edu.br

Aldenor Martins*
Federal University of Campina Grande
Campina Grande, Brazil
aldenor@dee.ufcg.edu.br

ABSTRACT

Digital TV content providers are becoming widespread, with hundreds of programs available each day. The information overload makes difficult for the user to find programs of interest. To help the user, *Recommendation Systems (RS)* are a popular path. However, applying RS to some environments is not easy, either due to the lack or insufficiency of data to create accurate recommendations. In Digital TV domain, the main information available to make recommendations is the *Electronic Program Guide (EPG)* that is limited, containing only reduced textual data, making difficult to get an accurate recommendation using standard techniques. In this work we introduce a multimodal approach to recommend Digital TV programs, combining EPG text and visual information. We experimentally demonstrated that using multimodal features improved accuracy when compared with RS standard approaches.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms

Performance, Theory, Experimentation

Keywords

Digital TV, Recommendation System, Information Retrieval

1. INTRODUCTION

In the Internet era, access to information is becoming ubiquitous. A huge amount of items is available (e.g. books, blogs, TV programs), and users need find those that best fit their interests [7]. Recommender Systems (RS) are the most popular and successful way to help users make better choices. RS are software systems that recommend new items users may like, based on their profile, such as web pages visited or programs watched [9].

Recommender Systems researches have been approached in three major dimensions: content-based, collaborative and hybrid. Content-based recommendation suggests new items based on attributes which the user previously enjoyed. For example, in the case of Digital TV, new items could be a program category, actors, among others. Collaborative recommendations suggest new items based on what other users enjoyed. A hybrid recommendation merges the two approaches.

However, the application of Recommender Systems to some domains is not an easy task, either due to the lack of data or because the data available is insufficient to create accurate recommendations using standard approaches. In the Digital TV environment, a domain that suffers from the lack of data, the main source to make recommendations is offered by the Electronic Programming Guide (EPG), providing solely reduced textual data.

Multimedia environment researches deal with this problem using multimodal features, besides the textual information, other features, such as the visual content (the video

*Aldenor Falcão Martins is a PhD student at COPELE - Programa de Pós-Graduação em Engenharia Elétrica, UFCG, Campina Grande, Brazil.

SAC 2014, March 24-28, 2014, Gyeongju, Republic of Korea

http://dx.doi.org/10.1145/2554850.2555154.

itself) and collaborative content (the set of users related in some way to each other), are used. Also, multimedia research usually involves social environments, where relationships between users, such as, user connections and friendships, already exist. We do not have access to social media information, therefore, in this work we use only textual and visual content.

For multimodal recommendation two approaches are used, late and early fusion [3]. Late fusion is based on extracting the relevance related to each modality (textual and visual information), and then joining them to get the global item relevance. Early fusion joins all feature modalities in a single model, usually applying graphs, and uses this model to get the global relevance. Researches show that at most cases early fusion outperforms late fusion [3]. Considering the advances in applying multimodal features in multimedia retrieval, we aim to apply an early fusion multimodal model to recommend programs on TV domain. We prove experimentally that using multimodal features improves the accuracy compared with a Recommender Systems standard approach.

2. RECOMMENDATION SYSTEM BY MULTIMODAL FEATURES

This section presents the recommendation problem definition, as well as the proposed solution for recommendation in Digital TV domain.

2.1 Notation and Problem Formulation

The problem of program recommendation is described in Definition 1.

Definition 1: Given a set of programs P_u watched by the user u, the task of program recommendation is expressed as finding a list of the most relevant programs to the user u.

This work uses the approach presented in Cui.et.al [3] that is based on a Markov Random Field (MRF) graphical model. The MRF model is described in an undirected graph where the nodes are the problem variables and the edges are the correlation between them and the probability distribution is calculated by the affinity between variables [5].

In this work the problem variables are the program features, which should have different modalities, here, textual and visual. In this way, we can represent a program p in a multimodal graph G, consisting of textual $T < t_1, t_2, \ldots, t_{|T|} >$ and visual $V < v_1, v_2, \ldots, v_{|V|} >$. Others modalities can be added as well.

In this graph structure we can have two types of correlation, intra-type, between the same type, and inter-type, between different modalities.

The textual intra-type correlation is computed using WordNet [8, 4] applying the Wu and Palmer (WuP) similarity.

The visual intra-type correlation is computed in a Bags of Keypoints approach, consisting of a set of steps: first, extract a set of relevant images from video, second extract interest point descriptions the images, third cluster the obtained interest point descriptions, and forth calculate the correlation between the cluster centroids.

Relevant images are extracted from video using short boundary detection. The short boundary detection is computed by the difference between two consecutive images, in this work we apply two techniques in sequence, Peak Signal-to-Noise Ratio (PSRN) and second Structural Similarity (SSIM). A boundary is detected if the difference between two consecutive images in video is bigger than a defined threshold. The SSIM is used only when PSRN fail, it reduces the processing time since PSRN is faster.

The interest point descriptors are extracted from relevant images found in previous step, using Scale-Invariant Feature Transform (SIFT) interest point detector and descriptor.

All interest point descriptors are passed to K-Means cluster with $K = 1500$, and the centroids are extracted. Each program p will have a histogram of centroids, and we calculate the similarity between them using Pearson Correlation. The program visual nodes are constructed converting the program to a binary representation by setting centroids frequencies in a program with value bigger than a defined threshold to 1 and 0 otherwise. The centroid is a visual feature of a specific program if it is set to 1.

To compute the inter-type correlation we constructed a vector for each program, where each position corresponds to how many occurrence of the feature is present on the program, and calculate the Pearson correlation between the features of program vectors.

To incorporate TV properties in the graph we use the categories programs present on EPG. At the end, each feature is linked to each category and program and we have the program graph G for each program.

2.2 Program Recommendation

The user profile G_u is an aggregation of all program multimodal graph G the user has watched $G_u = \{G_{u1}, G_{u2}, \ldots, G_{u|P_u|}\}$.

Program recommendation is calculated by the similarity between program graph G and user profile G_u, this is calculated by the probability of features of the user profile appear together with G features.

This probability is computed by summing up a potential function $\phi(c)$ defined as $\lambda_c f(c)$ over the cliques c of user profile G_u and program G, this is a unnormalized measure that describes the ranking of program p to user u. Where λ_c is the model parameter and $f(c)$ is a affinity function. In this work we use Cui.et.al [3] potential function (Equation 1).

$$\phi(c) \quad = \lambda_c \left(\alpha \frac{freq(c,G)}{|c(G)|} + (1-\alpha) \frac{\sum_{f_i \epsilon c} \sum_{f_j \epsilon |c(G)-c|}}{(|c|-1)\times|\{c(G)-c\}} \right) \tag{1}$$

Where $freq(c, G)$ is the frequency of cliques in G, and $c(G)$ is the G maximal clique set.

The parameters λ_c of MRF model is computed using a learning to ranking approach. Among different methods, in this work we apply ListNet [2], which produces the best results on our data.

3. RESULTS AND DISCUSSION

The results of our experiments can be seen in Figure 1. Visually, the Multimodal MRF approach outperforms the standard approach (UserKNN) in terms of Discounted cumulative gain (DCG@5) and Precision (P@5). We removed the text-based approach from DCG@5 because it returns a recommendation list less than five items for some users.

We compared a standard recommendation approach with MRF recommendation model. The dataset was constructed through a survey where participants indicated the programs viewed and how many times the interaction occurs in a week

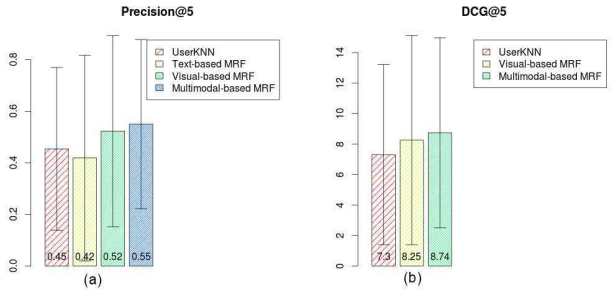

Figure 1: Precision and DCG Results

interval. The users are standard viewers with age between 20 and 35 years old of male and female gender. The items are regular Brazilian Digital TV programs presented between 2012 and 2013 and have different categories including Journal, Novel, Series, Soccer Match, etc.

The final dataset consisted of 42 users and 95 programs. The standard approach (we compared Matrix Factorization, UserKNN) that presented best results in this dataset was *User k-Nearest Neighbours (kNN)*, therefore we compared MRF approach with it.

We generated a top-n recommendation list for each approach (UserKNN, text-based MRF, visual-based MRF, and Multimodal-based MRF) and user in round-robin order, and emailed survey participants to judge each recommended program in the top-n list with a value between 0 and 3, where 0 means *irrelevant*, 3 means *completely relevant* and, 1 or 2, intermediary. 30 participants answered the email and we used their responses in validation.

We use two metrics in the experiment, DCG@5 and P@5, an approach in validation.

We tested MRF approaches and Multimodal MRF presented the best result in terms of both DCG and Precision metrics. Therefore, we compared the Multimodal MRF with UserKNN approach on two alternative hypotheses.

$H1_1$: The DCG for Multimodal MRF is greater than UserKNN.

$$\mu_{MultimodalMRF} > \mu_{UserKNN} \tag{2}$$

$H2_1$: The Precision for Multimodal MRF is greater than UserKNN.

$$\mu_{MultimodalMRF} > \mu_{UserKNN} \tag{3}$$

Normal tests were conducted using *Shapiro-Wilk test*, and some approaches are not normally distributed both for DCG@5 and P@5 with 90% confidence, therefore we applied *Wilcoxon test*.

For $H1_1$, we obtained a p-value equals to 0.107, so we concluded with 89.93% confidence that Multimodal MRF presents best DCG@5, and for $H2_1$ p-value equals to 0.097, therefore with 90% confidence we accepted $H2_1$ and concluded that Multimodal MRF presents best P@5.

4. RELATED WORK

Several studies have been developed in TV recommendation, and a special edition on this was published in [1], however few studies applied multimodal approach [3, 7, 6] for TV recommendation.

In this work we are interested in multimodal approach, some of them was: Cui.et.al [3] that applied an early fusion approach based on textual and visual features to recommend multimedia objects; Mei.et.al [7] explored a late fusion model to recommend videos based on user feedback and Luo.et.al [6] that applied an approach that recommends TV news program.

5. CONCLUSION AND FUTURE WORK

In this work we applied a multimodal MRF approach to recommend programs for Digital TV domain. We compared this model with a standard recommendation technique and concluded that Multimodal MRF outperforms the standard approach in terms of DCG@5 and P@5.

In future works we plan to create a similarity metric that computes the correlation within TV domain, improve the potential function including implicit user feedback, investigate others multimodal models, enlarge the dataset and the sample size used for statistical test.

6. ACKNOWLEDGMENTS

Our thanks to COPELE, CAPES and ENVISION for support this work.

7. REFERENCES

[1] L. Ardissono, A. Kobsa, and M. T. Maybury. *Personalized Digital Television: Targeting Programs to Individual Viewers*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[2] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 129–136, New York, NY, USA, 2007. ACM.

[3] B. Cui, A. K. Tung, C. Zhang, and Z. Zhao. Multiple feature fusion for social media applications. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 435–446, New York, NY, USA, 2010. ACM.

[4] C. Fellbaum, editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

[5] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[6] H. Luo, J. Fan, and D. A. Keim. Personalized news video recommendation. In *Proceedings of the 16th ACM international conference on Multimedia*, MM '08, pages 1001–1002, New York, NY, USA, 2008. ACM.

[7] T. Mei, B. Yang, X.-S. Hua, and S. Li. Contextual video recommendation by multimodal relevance and user feedback. *ACM Trans. Inf. Syst.*, 29(2):10:1–10:24, Apr. 2011.

[8] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[9] F. Ricci, L. Rokach, and B. Shapira. Introduction to Recommender Systems Handbook. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, chapter 1, pages 1–35. Springer US, Boston, MA, 2011.

Apêndice F

Questionário usado na avaliação

Dada a lista de programas apresentada para cada abordagem de recomendação, julgue o programa com uma nota de 0 à 3, onde 0 implica IRRELEVANTE, 3 COMPLETAMENTE RELEVANTE e 1 ou 2 algo entre RELEVANTE e IRRELEVANTE.

ABORDAGEM 1	JULGAMENTO	ABORDAGEM 2	JULGAMENTO	ABORDAGEM 3	JULGAMENTO	ABORDAGEM 4	JULGAMENTO
DOMINGO ESPETACULAR	1	CSI	3	HOW I MET YOUR MOTHER	3	HOW I MET YOUR MOTHER	3
A PRACA E NOSSA	0	THE BIG BANG THEORY	3	THE BIG BANG THEORY	3	FRIENDS	3
TELA QUENTE	3	HANNIBAL	2	FRIENDS	3	THE BIG BANG THEORY	3
CQC	3			EU, A PATROA E AS CRIANÇAS	3	CÂMERA RECORD	2
AUTO ESPORTE	3			THE MENTALIST	3	SMALLVILLE	2