Extratores para oráculos de teste de sistemas texto-fala utilizando recuperação de áudio baseada em conteúdo

Victor N. Gil[1], Rafael A. P. Oliveira[2], Márcio E. Delamaro[2], Fátima L. S. Nunes[1]

[1]Laboratório de Aplicações de Informática em Saúde (LApIS)
Escola de Artes, Ciências e Humanidades (EACH)
Universidade de São Paulo (USP)
CEP: 03828-000 – São Paulo, SP – Brasil

{victor.gil,fatima.nunes}@usp.br

[2]Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)
CEP: 13566-590 – São Carlos, SP – Brasil

{rpaes,delamaro}@icmc.usp.br

Abstract. *Automated tools for software testing have been developed for application in software with traditional outputs. Software with complex outputs, like images and sounds, remains as a challenge. This study applies content-based audio retrieval techniques for the development of feature extractors aiming to support the test of text-to-speech systems. The feature extractors are plug-ins of a framework to support the creation of test oracles for systems with complex outputs. They were validated with real systems and the results show that the approach is promising for automation of test oracles for systems with audio output.*

Resumo. *Ferramentas automatizadas para teste de sistemas computacionais têm sido desenvolvidas para aplicação em sistemas com saídas tradicionais. Sistemas com saídas complexas, como imagens e sons, ainda constituem um desafio. Este trabalho aplica técnicas de recuperação de áudio baseada em conteúdo no desenvolvimento de extratores de características visando a apoiar o teste de sistemas texto-fala. Os extratores desenvolvidos constituem complementos de um framework para apoiar a criação de oráculos de teste para sistemas com saídas complexas. Os extratores foram validados com sistemas reais e os resultados mostram que a abordagem é promissora para automatização de oráculos de teste para sistemas com saída sonora.*

1. Introdução

As atividades de Verificação, Validação e Teste (V,V&T) são consideradas fundamentais para a garantia da qualidade dos sistemas computacionais [Bertolino, 2007]. A indústria de software vem demonstrando interesse no aumento da qualidade dos produtos desenvolvidos, por motivos como a crescente competitividade do setor, o alto nível de criticidade que os sistemas automatizados adquiriram e os prejuízos econômicos proporcionados por software de baixa qualidade [Sommerville, 2004]. No entanto, a aplicação de métodos adequados de teste implica em um aumento considerável do custo de produção dos sistemas [Charette, 2005].

Como forma de minimizar esse problema, o teste automatizado tem se mostrado bastante útil na redução de custos e no aumento da eficiência das atividades de teste. Dentre os diversos benefícios, destacam-se a facilidade de reexecução dos casos de teste, a redução dos esforços humanos, o aumento da cobertura, dentre outros [Rafi et al., 2012]. No âmbito dos testes automatizados, a existência de estruturas denominadas oráculos de teste é um pressuposto fundamental [Baresi; Young, 2001]. Um oráculo representa um mecanismo que avalia a saída ou comportamento de um sistema, diante de determinadas entradas, classificando-os como corretos ou incorretos, de acordo com alguma especificação do funcionamento esperado [Baresi; Young, 2001]. Uma versão anterior de um sistema, uma especificação formal ou mesmo o conhecimento de um testador a respeito do funcionamento esperado de um sistema podem desempenhar funções de oráculos de teste.

A definição e a implementação de mecanismos de teste automatizados é uma tarefa complexa, e os desafios aumentam consideravelmente quando o programa a ser testado inclui saídas consideradas complexas, como imagens, sons ou interfaces gráficas [Oliveira et al., 2008]. Em compensação, mecanismos adequados de teste podem auxiliar na diminuição do tempo necessário para testar o software e aumentar a qualidade do teste, quando comparados com os testes manuais.

Sistemas de Síntese de Voz e Texto-Fala (TTS, do inglês, *Text-to-Speech*) são exemplos de sistemas que geram saídas complexas. Apesar do amplo emprego de tais sistemas, qualidade e precisão baixas ainda são comuns [Taylor, 2009]. Características como falta de naturalidade, problemas de entonação, pausa e pronúncia são comuns no áudio produzido por sistemas TTS [Taylor, 2009]. Um exemplo real em que a qualidade de tais sistemas é crucial é a sintetização de voz em dispositivos de GPS (*Global Positioning System*) em automóveis, no qual a qualidade do áudio é determinante para que o motorista compreenda as instruções com clareza.

As técnicas de avaliação de sistemas TTS buscam garantir a qualidade por meio da análise de aspectos como a inteligibilidade, a naturalidade e a compreensibilidade do texto convertido em fala [Klatt, 1987]. São comuns verificações manuais para validação dos sistemas TTS. Os sistemas são submetidos a ouvintes que avaliam subjetivamente as saídas [Oliveira, 2012], uma vez que alguns dos aspectos a serem analisados podem ser de difícil quantificação [Yu-Yun, 2011]. Dentre outros problemas de tal estratégia, destaca-se o "efeito de aprendizagem", que ocorre quando o testador cognitivamente se familiariza com a fala gerada pelo sistema, o que implica em uma melhoria nos resultados [Leite et al., 2014]. Além disso, há problemas relacionados a aspectos fisiológicos humanos, subjetividade, necessidade de domínio da língua escrita e falada e utilização de elevado número de recursos humanos [Leite et al., 2014]. Todos esses fatores implicam em aumento do custo e tempo do projeto de software. Fica clara a necessidade, portanto, da criação de mecanismos de teste automatizados para que tais sistemas possam ser sistematicamente testados e validados [Oliveira, 2012].

O objetivo deste artigo é apresentar o desenvolvimento de extratores de características a partir de sinais sonoros oriundos das saídas de sistemas TTS. A contribuição do trabalho reside, portanto, na implementação de técnicas de extração de características para sistemas de saída sonora e também na parametrização e adaptação desses extratores para uso em oráculos de teste para sistemas dessa natureza.

Adicionalmente, são apresentados estudos empíricos realizados para validar tais extratores.

Além desta seção introdutória, na Seção 2 deste artigo são apresentados os conceitos básicos necessários para a contextualização dos algoritmos implementados. Na Seção 3 são apresentados trabalhos relacionados às áreas abordadas no presente trabalho. Na Seção 4 é descrita a metodologia adotada neste trabalho e são apresentados os extratores de características desenvolvidos. Na Seção 5 são apresentados e discutidos os resultados obtidos usando-se saídas de sistemas TTS reais. Por fim, são apresentadas as conclusões do trabalho.

2. Conceitos

2.1 CBIR

De acordo com Bueno et al. (2002), Recuperação de Imagens Baseada em Conteúdo (CBIR, do inglês, *Content-Based Images Retrieval*) é uma técnica que busca, em um banco de dados, imagens semelhantes a uma imagem de referência, utilizando critérios previamente definidos. Tais sistemas possuem três componentes básicos: extratores (1), funções de similaridade (2) e estruturas de indexação (3).

Os extratores são estruturas responsáveis por obter características baseadas em diversos aspectos do objeto em análise, como forma, cor e textura em uma imagem. Cada característica é representada por um valor numérico resultante do processamento realizado pelo extrator. O conjunto de características obtidas por esses extratores forma um vetor de características.

As funções de similaridade medem a distância entre vetores de características, determinando a sua semelhança. Esses vetores são indexados para posterior recuperação das imagens armazenadas no banco de dados [Smeulders et al., 2000].

De acordo com Torres e Falcão (2006), um sistema de CBIR inicia-se com a execução de um conjunto de algoritmos extratores sobre duas imagens de entrada. É criado um vetor de características associado a cada uma das imagens, contendo os resultados dos extratores de características. Por fim, uma função de similaridade calcula e retorna um valor numérico que representa o grau de similaridade entre as imagens.

2.2 Processamento de áudio digital

O som é uma onda mecânica gerada pela oscilação de pressão em um meio, que a propaga [Tafner, 1996], podendo ser representado por uma função contínua da variação da amplitude da onda sonora em relação ao tempo, como mostrado na Figura 1(a) [Bosi; Goldberg, 2005]. A digitalização do som é o processo de conversão do sinal contínuo para discreto. Este processo é realizado por meio da amostragem do som, em que a amplitude instantânea da onda sonora é medida em intervalos precisos de tempo, como apresentado na Figura 1(b). Cada valor de amplitude obtido, denominado amostra, precisa ter uma precisão finita de *bits* para ser processado. O processo responsável por definir essa precisão é denominado quantização [Bosi; Goldberg, 2005].

A frequência de amostragem, outra propriedade básica do sinal de áudio digital, representa o número de amostras obtidas por segundo durante o processo de amostragem, e é medida em Hertz (Hz). O formato de amostra é o número de bits

utilizados para o armazenamento dos valores de amplitude obtidos na digitalização do som [Bosi; Goldberg, 2005]. Dentre os diversos formatos de arquivo para armazenamento digital de áudio, o formato WAVE (*Waveform Audio File Format*) é o mais comumente utilizado como opção de saída em sistemas de síntese de voz.

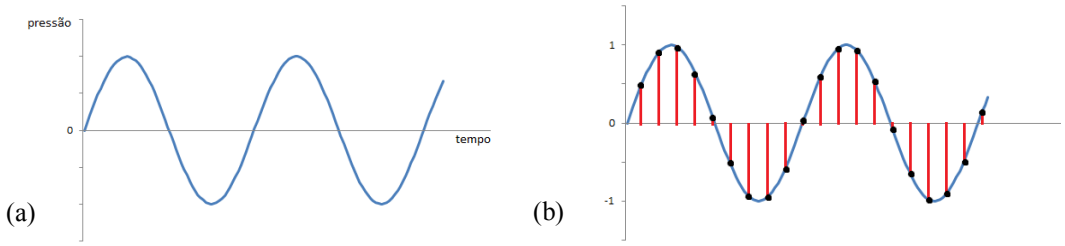

Figura 1. (a) Representação de uma onda sonora; (b) Representação do processo de digitalização, com amostras obtidas a intervalos regulares

2.3 CBAR

Os conceitos e técnicas utilizados para a extração de características de imagens podem ser estendidos para outros tipos de dados complexos, dentre eles os sinais de áudio. Nesse caso, tem-se a Recuperação de Áudio Baseada em Conteúdo (CBAR, do inglês, *Content-Based Audio Retrieval*). Nesse contexto, a extração de vetores de características de sons depende de técnicas de processamento e digitalização da onda sonora, além da análise do formato espectral do sinal [Barioni, 2006]. Na maioria dos casos de CBAR, as características são extraídas a partir da análise do formato espectral do sinal, o que é realizado por meio da aplicação de um método de análise de Tempo-Frequência [Barioni, 2006].

O formato espectral é a representação do sinal no domínio da amplitude da onda por frequência, e não no domínio da amplitude por tempo, como no caso representado na Figura 1(a) [Barioni, 2006]. Enquadram-se nos métodos de análise Tempo-Frequência a Transformada de Fourier de Curta Duração (STFT, do inglês, Short Time Fourier Transform) e os Coeficientes de Frequência Mel-Cepstrais (MFCC, do inglês, Mel Frequency Cepstral Coefficients), sendo que ambas as técnicas podem ser utilizadas no processamento e análise de fala [Barioni, 2006].

2.4 Extração de características de sinais sonoros

Métodos para a identificação de fonemas são muito utilizados por programas de reconhecimento de fala. Fonema é a menor unidade de som da fala humana [Callou; Leite, 1995]. De acordo com Bresolin (2003), também se pode caracterizar um fonema como a unidade sonora capaz de prover diferenças de significado entre as palavras.

Um fonema é representado graficamente por uma ou mais letras, ou seja, nem sempre há uma correspondência entre o número de fonemas e o número de letras em determinada palavra. Além disso, uma mesma letra pode representar diversos fonemas e um fonema pode ser representado por mais de uma letra [Dias, 2008].

Cada idioma possui o seu próprio conjunto de fonemas, sendo que na língua portuguesa existem 13 fonemas vocálicos, 19 consonantais e duas semivogais. O alfabeto fonético é um sistema criado para representar graficamente os fonemas, evitando assim os problemas de correspondência com as letras [Bresolin, 2003]. No

alfabeto fonético cada símbolo representa apenas um fonema, de forma que a escrita de uma palavra utilizando este alfabeto indica a pronúncia correta de todos os sons.

2.4.1 Formantes e Fonemas

Dentre os métodos utilizados para a identificação de fonemas está a análise de formantes. Formantes são as diferentes regiões no espectro do som, ou faixas de frequência, que apresentam picos de intensidade [Tafner, 1996]. Os formantes caracterizam e proveem significado à fala humana, permitindo ao aparelho auditivo, por exemplo, diferenciar as vogais. São nomeados F_1, F_2, F_3..., F_N, ordenados da menor para a maior frequência à qual estão associados. Na Figura 2 é exemplificado um gráfico com as dimensões de frequência e intensidade de uma onda sonora, destacando a localização dos três primeiros formantes. Existe um número infinito de formantes, porém os três primeiros são os mais importantes, não sendo comuns análises espectrográficas além de F_3 [Russo e Behlau, 1993].

Considerando sons produzidos pelo trato vocal humano, o primeiro formante está associado à elevação da língua no momento da produção do som, sendo que o aumento de F_1 é inversamente proporcional ao nível de constrição da passagem de ar. O segundo formante está associado à posição em que essa constrição ocorre, de forma que o avanço da posição de estreitamento aumenta o valor de F_2 [Miranda; Meireles, 2011].

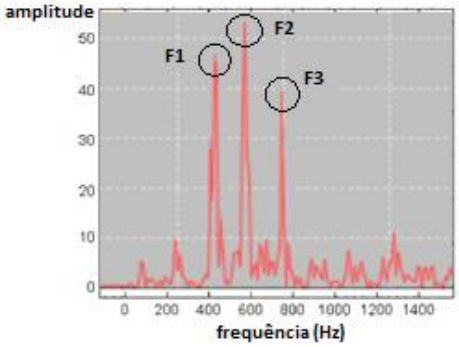

Figura 2. Gráfico de frequências de um sinal sonoro e seus formantes

As vogais faladas são caracterizadas pelas frequências de seus formantes. Para a identificação das vogais por meio da análise dos formantes, uma correspondência acústico-articulatória entre o aparelho vocálico e os formantes do som produzido permite identificar as vogais a partir dos valores dos dois primeiros formantes apenas, F_1 e F_2 [Miranda; Meireles, 2011].

Os formantes de um sinal sonoro podem ser obtidos por meio da aplicação do modelo matemático da Transformada de Fourier [Miranda; Meireles, 2011], que consiste em uma base matemática para conversão de funções periódicas em suas componentes de frequência [Bresolin, 2003]. A transição entre os domínios do tempo e frequência é dada pela Transformada Contínua de Fourier apresentada na Equação 1. Considerando sinais discretos, ou seja, sinais sonoros formados por um número N de amostras, a Transformada Discreta de Fourier (TDF), apresentada na Equação 2, deve ser utilizada.

$$X(f) = \int_{-\infty}^{+\infty} x(t).\, e^{-j2\pi ft}\, dt \qquad (1)$$

$$X[n] = \frac{1}{N} \sum_{k=0}^{N-1} x(k).\, e^{-j2\pi nk/N} \quad (2)$$

Além da análise de formantes, a extração de fonemas de sinais de fala pode ser feita usando o cálculo dos Coeficientes de Frequência Mel-Cepstrais (MFCC, do inglês *Mel-Frequency Cepstral Coefficients*). O processo de cálculo desses coeficientes gera um vetor numérico que contém informações relevantes do espectro do sinal sonoro para o reconhecimento de fala. [Coutinho; Oliveira, 2012]. Um dos fundamentos teóricos para a utilização desse método é que ele aproxima-se do comportamento do sistema auditivo humano ao identificar os fonemas por meio de uma escala não linear. Para o cálculo do MFCC, inicialmente o sinal sonoro, no domínio do tempo, é dividido em janelas denominadas *frames.* De maneira simplificada, o cálculo é realizado por meio da aplicação da TDF a cada *frame* e da conversão dos valores de frequência em Hertz para a escala Mel, por meio da Equação 3.

$$m = 1127\, log_e \left(1 + \frac{f}{700}\right) \quad (3)$$

A conversão é efetuada por meio de um número definido (parametrizável) de filtros triangulares na escala Mel, convenientemente espaçados, e então é calculada a energia de cada um desses filtros. O cálculo prossegue com aplicação do logaritmo dos valores obtidos e, então, é aplicada a Transformada Discreta de Cosseno (TDC), resultando então em um vetor de coeficientes utilizados para o reconhecimento dos fonemas [Terssetti, 2010].

3. Trabalhos relacionados

Foram levantados na literatura trabalhos relacionados às áreas de teste automatizado, oráculos de teste aplicados a saídas complexas e também técnicas de extração de características de sinais sonoros, apresentados a seguir.

3.1 Oráculos de teste para sistemas com saídas complexas

Em geral, os artigos encontrados apresentam características e tipos de oráculos como estratégias de teste automatizado. Dentre os trabalhos com esse tipo de abordagem podem ser destacados Baresi e Young (2001) e Hoffman (2001). Existem também trabalhos focados especificamente no teste de alguns tipos de saídas complexas, como GUIs (do inglês, *Grafical User Interface)* e aplicações Web, com destaque para os trabalhos de Memon et al. (2000) e Memon (2001). De acordo com Chang et al. (2010), o formato complexo da saída é o principal empecilho para estratégias de automatização do teste.

Considerando trabalhos acerca de oráculos de teste para sistemas com saídas gráficas podem ser destacadas as pesquisas realizadas por Oliveira (2012), Oliveira et al. (2008) e Delamaro et al. (2013), que se baseiam no uso de técnicas de CBIR para a criação de extratores de características e definição de oráculos de teste para esse tipo de sistema. Não foram encontrados trabalhos na literatura que abordem o uso oráculos de teste para a automatização do teste de sistemas com saídas sonoras.

3.2 Extração de características de sinais sonoros

Diversos estudos relacionados à CBAR e à extração de características de sinais sonoros estão disponíveis na literatura. Apesar disso, poucos trabalhos com foco específico no teste de sistemas de saída sonora podem ser encontrados.

O trabalho realizado por Mitrovic et al. (2010) apresenta uma revisão acerca da extração de características de sinais sonoros, propondo uma taxonomia para a organização dos extratores e uma extensa lista de trabalhos publicados, que aplicam extratores de características em diversos domínios.Na Tabela 1 é apresentado um resumo das técnicas de extração de características mais utilizadas de acordo com a literatura recente sobre o tema.

Tabela 1. Técnicas para extratores de características encontradas na literatura

Referência	Extrator	Descrição
Chaudhary e Hamid (2012)	Mel Frequency Cepstral Coefficients (MFCC)	Detalhamento do funcionamento da extração de características utilizando MFCC e transformadas de Fourier.
Esfandian, Razzazi e Behrad (2012)	Weighted K-means (WKM)	Aplicação da técnica WKM no domínio espectro-temporal. As matrizes de covariância são utilizadas como vetores de características.
Ghoraani e Krishnan (2011)	Time-Frequency Matrix (TFM)	Construção de uma TFM com o objetivo de extrair e classificar som ambiente.
Guihua et al. (2012)	Transformação Relativa	Aplicação de *Mel-Frequency Cepstral Coefficients* em segmentos combinados por transformação relativa.
Hyunsin, Takiguchi e Ariki (2008)	Independent Component Analysis (ICA)	Extrator que obtém informações de correlação entre fonemas, apresentando bom desempenho no reconhecimento de palavras isoladas.
Junchang, Yuanyuan e Jian (2011)	Kernel principal component analysis (KPCA)	Extrator robusto em relação à ruídos baseado em clusterização *fuzzy K-Means* com altas taxas de reconhecimento de fala.
Qiang, Liqing e Guangchuan (2011)	Gabor Analysis	Extrator robusto a ruídos baseado em *Gabor Fltering* e *Tensor Factorization* que explora características espectro-temporais localizadas para extração.
Seyedin e Ahadi (2008)	Minimum Variance Distortionless Response (MVDR)	Extrator robusto em relação a ruídos, baseado em *Discrete Cosine Transform*, apresentando melhores resultados comparando com o MFCC.
Xiang, Feng e Jingao (2008)	Discrete Wavelet Transform (DWT)	Evolução de baixo custo computacional da técnica MFCC, bem adaptado a condições de teste e treino distintas.
Xiaolan et al. (2011)	Gaussian Mixture Model	Extrator baseado em transformada de *Wavelet*, com decomposição de sinais e ênfase nas características da voz

4. Materiais e métodos

Tendo como objetivo principal o desenvolvimento de extratores de características de sinais sonoros obtidos a partir da execução de sistemas TTS, este trabalho busca aplicar os conceitos de CBIR e oráculos de teste de sistemas com saídas gráficas ao processamento de som.

Na fase de desenvolvimento foram implementados dois extratores de características que foram adaptados à arquitetura do *framework O-FIm* (*Oracle for Images*). Esse *framework* utiliza técnicas de CBIR para o apoio ao teste de sistemas que produzem

saídas gráficas [Delamaro et al., 2013]. Os extratores do presente trabalho implementam as interfaces padrão do *framework*, garantindo suas utilidades como *plugins* para a criação de oráculos de teste parametrizáveis.

4.1. *Framework O-FIm*

O *framework O-FIm*[1], desenvolvido na linguagem Java, visa a criar um ambiente flexível para a configuração de oráculos de teste para sistemas de saídas gráficas, que são chamados de oráculos gráficos [Delamaro et al., 2013]. A arquitetura do *framework O-FIm* (Figura 3) permite que um testador configure um descritor de oráculo por meio de um arquivo de configuração contendo os extratores, as funções de similaridade e outros parâmetros a serem utilizados nos testes. Um componente *parser* é responsável pelo reconhecimento dessas configurações e o núcleo inicializa as classes necessárias e realiza a comparação [Delamaro et al., 2013].

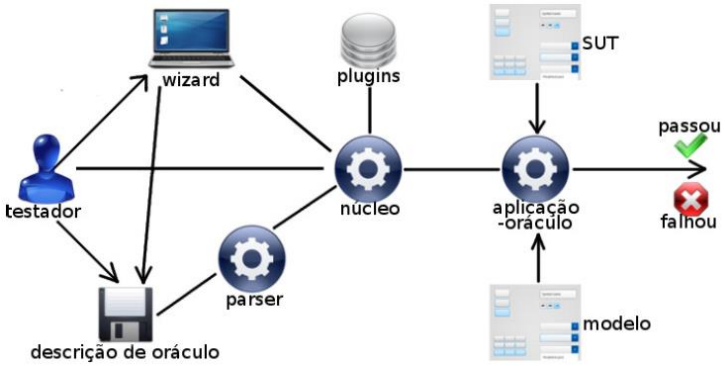

Figura 3. A Arquitetura do framework O-FIm [Delamaro et al., 2013]

Os extratores de características instalados podem receber parâmetros específicos para sua execução, e esses também devem ser informados no descritor do oráculo. Quando o oráculo de teste é executado pelo *framework* cada um dos extratores retorna como resultado um valor numérico que corresponde à característica extraída. O resultado de cada extrator é armazenado em uma posição do vetor de características criado. Esse vetor é obtido para as duas saídas complexas (imagens ou sons, por exemplo) que estão sendo comparadas, e o resultado final do oráculo é a distância entre estes dois vetores, calculada pela função de similaridade.

4.2. Extratores de Características

Para o desenvolvimento dos extratores foram utilizadas APIs (*Application Programming Interfaces*) para auxílio na leitura e manipulação de arquivos WAVE em linguagem Java. Dentre as funcionalidades de interesse destacam-se a leitura de amostras, a obtenção de dados básicos como duração, taxa de amostragem, canais, formato das amostras, dentre outras, e também componentes para reprodução dos arquivos. As bibliotecas JMF[2] e *Java Sound*[3] foram utilizadas em conjunto para atender essas necessidades.

---

[1] *Oracle for Images*: `http:/ http://ccsl.icmc.usp.br/pt-br/projects/o-fim-oracle-images`

[2] *Java Media Framework*: `http://www.oracle.com/technetwork/java/index-jsp-140239.html`

[3] *Java Sound API:* `http://www.oracle.com/technetwork/java/index-jsp-140234.html`

Além disso, a biblioteca CoMIRVA[1] (*Collection of Music Information Retrieval and Visualization Applications*) foi selecionada para utilização de sua implementação de alguns algoritmos de processamento de sinal necessários para os extratores desenvolvidos.

4.2.1. Implementação do extrator de vogais

O extrator de vogais desenvolvido tem como objetivo a identificação da presença e do instante em que ocorrem os sete fonemas vocálicos orais da língua portuguesa, representados na Tabela 2. Para o desenvolvimento do extrator foi projetado um algoritmo capaz de extrair os dois primeiros formantes do sinal sonoro e gerar como saída informações sobre a ocorrência dos fonemas vocálicos orais. O fluxo completo do extrator desenvolvido pode ser visualizado na Figura 4.

Tabela 2. Fonemas vocálicos orais

Fonema	Exemplo
/a/	**a**mor
/e/	**e**xtrator
/ɛ/	caf**é**
/i/	p**i**lha
/o/	**o**lho
/ɔ/	**ó**culos
/u/	**u**va

Inicialmente, o algoritmo obtém dados básicos sobre o arquivo WAVE recebido, como o método de codificação, o tamanho das amostras, a taxa de amostragem, dentre outras informações necessárias para a leitura das amostras e processamento do sinal. Em seguida, o algoritmo divide as amostras do arquivo de acordo com alguns critérios configuráveis, entre eles tempo, amplitude média, máxima e mínima, além da frequência da onda, considerando o sinal no domínio do tempo. Com isso, obtém-se grupos menores, com o objetivo de analisar o espectro da onda apenas para as amostras dentro de um mesmo grupo.

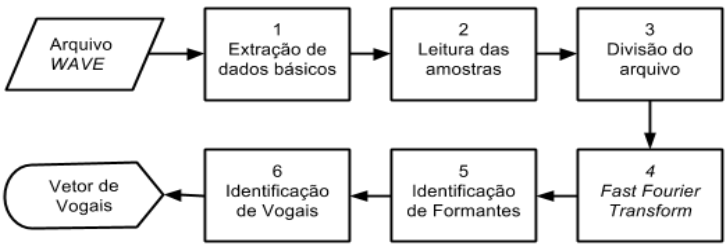

Figura 4. Diagrama de fluxo do processo de extração de vogais

Para realizar a transposição entre o domínio do tempo e o domínio da frequência em cada grupo gerado foi utilizado o algoritmo FFT (*Fast Fourier Transform*), disponibilizado na biblioteca CoMIRVA. Esse algoritmo é uma implementação otimizada do cálculo da TDF, e transforma o sinal obtido diretamente do arquivo *WAVE* em uma função no domínio da frequência e amplitude. Desse modo, é possível obter os valores da frequência do sinal nos picos de amplitude, identificando os formantes.

[1] *CoMIRVA Framework*: `http://www.cp.jku.at/comirva`

Esta etapa também possui diversos parâmetros que podem ser configurados pelo utilizador, de forma a definir como cada formante deve ser obtido. A busca de cada um dos formantes (F_1 e F_2) pode ser restringida a uma determinada faixa de frequências, assim como podem ser incluídas restrições entre os valores de F_1 e F_2, que são analisados em conjunto. Esse recurso é necessário para evitar que o algoritmo considere incorretamente alguns picos de intensidade como formantes.

Definidos os dois primeiros formantes do grupo, um algoritmo de identificação de vogal verifica qual vogal possui o par (F_1, F_2) mais próximo, considerando a distância euclidiana entre eles. Para efeitos de validação do extrator, os valores obtidos empiricamente por Miranda e Meireles (2011), para cada fonema vocálico oral, foram utilizados para a classificação do resultado da distância (Figura 5). O extrator permite também a configuração desses valores, que podem ser definidos de acordo com o tipo de áudio a ser analisado, considerando informações como idade e sexo do falante, além de particularidades do sistema de síntese de voz gerador do sinal. Nesta etapa outros parâmetros também podem ser definidos pelo usuário, como a distância mínima necessária para que um grupo possua uma vogal definida.

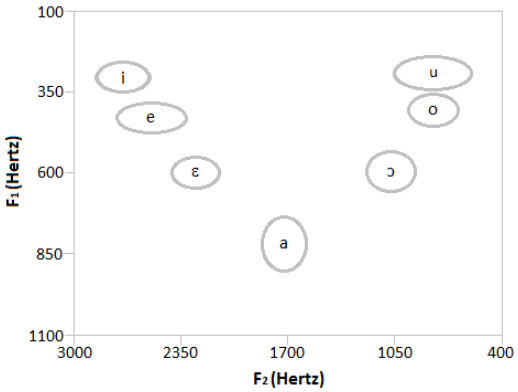

Figura 5. Os sete fonemas vocálicos orais da língua portuguesa em um plano F_1 x F_2 [Miranda; Meireles, 2011]

4.2.2 Implementação do extrator de fonemas

Considerando a grande quantidade e variabilidade entre os fonemas da língua portuguesa, a extração e identificação de cada um deles exige uma combinação de diversas técnicas e algoritmos distintos [Bresolin, 2003]. Durante a implementação, foi considerado como objetivo do extrator a identificação da presença e do momento em que ocorrem, no sinal sonoro analisado, os três fonemas mais comuns da língua portuguesa escrita.

De acordo com a pesquisa de Viaro e Guimarães-Filho (2003), a estrutura silábica mais comum do conjunto analisado é a denominada CV, que corresponde às sílabas formadas por um fonema consonantal seguido de um fonema vocálico, totalizando 60,6% do total de estruturas silábicas existentes. Dentre as sílabas da estrutura CV, os segmentos consonantais mais frequentes são /k/ (10%), /t/ (9,4%) e /m/ (8,4%).

De modo semelhante ao extrator de vogais (Figura 4), o algoritmo inicialmente divide o sinal sonoro, no domínio do tempo, em grupos de curta duração, de acordo com critérios como tempo, amplitude média, máxima e mínima e frequência da onda. Para a extração de fonemas os parâmetros para esta divisão são configurados de modo a gerar

um número maior de grupos, em comparação ao extrator de vogais. Isso é feito devido à duração da pronúncia dos fonemas ser, em geral, menor do que a duração das vogais, e também devido às diferenças de comportamento dos algoritmos face aos efeitos diferentes dos fonemas vocálicos e consonantais no espectro do som.

O algoritmo de extração do vetor MFCC, disponibilizado na biblioteca CoMIRVA, é então aplicado a cada grupo gerado, e o vetor resultante é comparado com valores pré-determinados por análises empíricas, realizadas por meio da aplicação do algoritmo a amostras dos três fonemas de interesse. O extrator permite também a configuração destes valores, que podem ser definidos de acordo com o tipo de áudio a ser analisado, considerando informações como idade e sexo do falante, além de particularidades do sistema de síntese de voz gerador do sinal.

4.3. Descritor dos extratores de vogais e fonemas

Os extratores de vogais e fonemas desenvolvidos possuem quatro parâmetros, que visam a definir qual vogal ou fonema deve ser analisado, qual a informação desejada pelo testador como resultado, a partir de qual ponto do arquivo o extrator deve avaliar e, por fim, um arquivo no padrão XML (do inglês, *eXtensible Markup Language*) que define configurações adicionais. A Tabela 3 apresenta detalhadamente os parâmetros.

Tabela 3. Parâmetros dos extratores de vogais e fonemas

Parâmetro	Descrição	Domínio
vowel / phoneme	letra que representa a vogal ou fonema a ser extraído	{A, E, I, O, U} {K, T, M}
return	tipo de informação desejada: quantidade ou posição	{Q, P}
interval	intervalo de tempo em milissegundos utilizado para divisão do sinal em posições	número inteiro
nth	enésimo intervalo de tempo a partir da qual o cálculo será realizado	{1, 2...N}
file	caminho completo do arquivo XML de configurações	caminho válido

Os parâmetros *vowel* e *phoneme* devem ser utilizados pelo criador do oráculo para definir qual vogal ou fonema, dentre as opções disponibilizadas pelos extratores, deve ser analisado. A Tabela 4 apresenta uma associação entre esses parâmetros e o fonema extraído.

Tabela 4. Fonemas extraídos de acordo com o parâmetro do extrator

Parâmetro	Fonema	Parâmetro	Fonema
A	/a/, /ɐ/	K	/k/
E	/e/, /ɛ/	T	/t/
I	/i/	M	/m/
O	/o/, /ɔ/		
U	/u/		

O parâmetro *return* foi criado com o objetivo de prover mais flexibilidade ao criador do oráculo. A aplicação da função de similaridade pelo *framework* baseia-se no retorno de apenas um valor numérico como resultado da execução de cada extrator. Assim, os extratores desenvolvidos utilizam o parâmetro *return* para permitir que sejam

avaliadas pelo oráculo tanto a informação da quantidade quanto o instante de tempo em que ocorrem as vogais ou fonemas no sinal.

De maneira mais detalhada, o comportamento dos extratores quando o parâmetro *return* é igual a '*Q*' é retornar a quantidade de fonemas associados ao parâmetro *vowel* ou *phoneme* que foram encontrados no sinal de áudio processado. Quando o parâmetro é igual a '*P'* os extratores retornam o resultado de uma função que relaciona a ocorrência do fonema com a posição do sinal em que ocorrem. A posição é determinada pela divisão do sinal em intervalos de tempo definidos por meio do parâmetro *interval*. Além disso, o parâmetro *nth* determina a partir de qual intervalo de tempo o extrator iniciará o cálculo. Assim, o valor 'e' retornado pelo extrator quando *return = Q* é determinado pela Equação 4, onde *'i'* representa a ocorrência do fonema, '*n'* representa o intervalo de tempo em que o fonema foi encontrado e '*N*' o número total de ocorrências do fonema.

$$e = \sum_{nth}^{N} i.n \quad (4)$$

Por fim, o parâmetro *file* deve conter o caminho completo de acesso ao arquivo XML de configurações adicionais do intervalo.

4.4. Validação dos extratores

O processo de validação empírica dos extratores foi realizado por meio da aplicação dos algoritmos desenvolvidos a dois sistemas TTS reais. Os experimentos foram divididos em duas etapas. A primeira objetivava avaliar independência do funcionamento dos algoritmos em relação ao sistema utilizado como teste e a segunda teve como objetivo simular um cenário real de testes, em que um dos sistemas representa o oráculo e o outro representa o sistema a ser testado, utilizando para isso as ferramentas de automatização de testes do *framework O-FIm*. Os dois sistemas utilizados neste estudo foram o *CPqD Texto-Fala (versão 3.3)*[1] e o *Google Translate Text-to-Speech*[2].

O conjunto de dados utilizado na validação foi composto por 100 palavras em língua portuguesa. A escolha das palavras não obedeceu nenhuma restrição. Os arquivos utilizados têm como conteúdo o som sintetizado a partir do texto de três notícias jornalísticas, cada uma abordando assuntos distintos. As palavras foram extraídas dos arquivos por meio de um software de edição de sons e foram gerados 100 novos arquivos no formato WAVE, cada um deles contendo apenas uma palavra.

Em seguida, o conjunto de textos foi submetido ao sistema *Google Text-to-Speech* e o mesmo processo de extração das palavras selecionadas foi realizado. Foram gerados mais 100 arquivos de som, também no formato WAVE.

O conjunto de palavras foi então analisado manualmente para verificação da ocorrência dos fonemas vocálicos e consonantais de interesse. A distribuição da ocorrência de cada um dos fonemas no conjunto de dados é apresentada na Tabela 5.

Esta primeira etapa da execução dos extratores buscou verificar a precisão da identificação dos fonemas em relação à análise manual. Para isso foram selecionadas três

[1] *CpQD Texto-Fala*: `http://www.cpqd.com.br/textofala`
[2] *Google Text-to-Speech:* `http://code.google.com/p/java-google-translate-text-to-speech`

ocorrências de cada um dos fonemas para calibrar os parâmetros de configuração dos extratores de acordo com as características de cada sistema TTS. Dentre as configurações realizadas estão a velocidade da fala, ajustes nos valores dos formantes das vogais e definição do vetor de coeficientes mel-cepstrais dos fonemas consonantais.

Tabela 5. Número de ocorrências dos fonemas no conjunto de dados

Fonema	Ocorrências totais	Ocorrências em palavras distintas
/a/, /ɐ/	73	55
/e/, /ɛ/	58	48
/i/	47	43
/o/, /ɔ/	46	41
/u/	25	25
/k/	19	19
/t/	36	33
/m/	15	15

Após a parametrização dos arquivos de configuração os extratores de características foram executados, recebendo como entrada os 100 arquivos de som de cada um dos dois sistemas avaliados. Foram mensurados os acertos em relação à presença ou ausência dos fonemas de interesse nos arquivos processados, além da verificação de múltiplas ocorrências em uma mesma palavra.

A segunda etapa de validação buscou parametrizar oráculos de teste no *framework O-FIm* visando a simular um cenário real de testes de um sistema TTS. Para isso, os arquivos gerados pelo sistema *"CPqD Texto-Fala"* representaram exemplos de saídas corretas, e o sistema *"Google Text-to-Speech"* representou o sistema TTS a ser testado. Tal decisão foi motivada pelo alto nível de qualidade dos sinais gerados pelo sistema "*CPqD*", utilizado em aplicações comerciais reais no setor bancário, telefônico e diversos outros.

Inicialmente os oráculos foram executados para os mesmos arquivos utilizados na etapa de validação anterior. O objetivo foi verificar, por meio do resultado fornecido pelo *framework*, a semelhança entre os arquivos, contendo as mesmas palavras, porém gerados por sistemas diferentes.

5. Resultados e discussões

5.1. Análise da precisão de identificação

Com o objetivo de verificar a precisão do Extrator de Vogais na identificação dos fonemas vocálicos orais, o seguinte descritor de oráculo foi criado (ver Seção 4.2):

```
extractor OralVowelExtractor {vowel = "a" return = "Q" interval = 100 nth = 0
config = "config_cpqd.xml"}
```

Com a execução do extrator, aplicado a todas as palavras do conjunto de dados, foi possível analisar a precisão da extração dos fonemas vocálicos associados à vogal "A" (Tabela 4). Após a coleta dos resultados, o parâmetro *"vowel"* foi alterado e o mesmo processo foi repetido para as demais vogais.

Os resultados obtidos foram comparados com a extração manual realizada sobre o conjunto de palavras. Para cada vogal, a Tabela 6 apresenta o percentual de palavras em que todas as ocorrências dessa determinada vogal foram identificadas corretamente pelo extrator. Os dados foram agrupados por sistema e pelo número de ocorrências da vogal nas palavras. Por exemplo, na primeira linha da Tabela 6 verifica-se que algoritmo apontou corretamente a ausência da vogal 'A' em 71% das palavras geradas pelo sistema "*CPqD*" que não possuíam nenhuma ocorrência dessa vogal. Considerando as palavras que possuíam exatamente uma ocorrência da vogal 'A', o extrator identificou corretamente essa ocorrência em 85% das palavras.

Tabela 6. Percentual de palavras com resultado correto

Parâmetro *Vowel*	**Número de Ocorrências da Vogal**					
	CPqD Texto-Fala			*Google TTS*		
	0	**1**	**> 1**	**0**	**1**	**> 1**
A	71%	85%	57%	60%	68%	64%
E	65%	76%	60%	73%	61%	50%
I	89%	72%	50%	74%	59%	25%
O	75%	69%	40%	61%	72%	40%
U	68%	68%	100%	93%	40%	100%
Média de acertos	74%	74%	61%	72%	60%	56%

Os resultados obtidos mostram, em primeira análise, dados semelhantes entre os dois sistemas analisados. Isso pode indicar que há independência do algoritmo em relação a cada sistema TTS, ou seja, que o extrator funciona de modo semelhante para os dois sistemas, apresentando sucesso ou insucesso em casos similares. Outra possível explicação seria a existência de eventuais semelhanças nos algoritmos dos dois sistemas. Testes futuros com outros sistemas TTS poderão comprovar tais hipóteses.

Também se pode extrair dos resultados uma tendência de queda na capacidade de identificação quando existe mais de uma ocorrência da mesma vogal em uma palavra. Isso pode ser explicado pelo fato de que, dada a probabilidade de erro na identificação de cada ocorrência, quanto mais ocorrências a palavra possuir, maior a probabilidade de ao menos uma delas não ser identificada. No caso da vogal 'U', em particular, isso não foi observado, o que pode ser explicado pelo número reduzido de palavras no conjunto de dados que possuem mais de uma ocorrência dessa vogal.

Uma análise aprofundada dos casos da identificação incorreta de vogais mostrou que um ponto do algoritmo que pode ser evoluído futuramente é a etapa de divisão do sinal sonoro. Conforme descrito na Seção 4.1.2, o algoritmo FFT é aplicado ao sinal sonoro dividido de acordo com o padrão da forma de onda no domínio do tempo. Quando a divisão não ocorre de maneira ideal, como no caso da quebra do trecho correspondente à pronúncia de uma mesma vogal, podem ocorrer desvios no cálculo dos formantes, o que reduz a precisão do algoritmo.

Outro ponto que contribui para a redução da precisão do extrator é a proximidade dos valores de formantes de alguns grupos de vogais (Figura 5). Tal problema afeta, principalmente, a capacidade de diferenciação entre as vogais 'O' e 'U' e entre 'I' e 'E'. Esse fato pode explicar o menor percentual de acertos entre esses grupos quando comparados com a vogal 'A', que possui valores de formantes relativamente distantes das demais. Uma possibilidade de melhoria futura é a utilização de outros formantes para auxiliar na identificação desses casos.

Além disso, a pronúncia dos fonemas vocálicos /o/ e /u/, muitas vezes, ocorre de maneira distinta dependendo da variação da língua portuguesa utilizada pelo sistema TTS. Considerando como exemplo a palavra "*zer**o**",* presente no conjunto de dados, foi verificado que o último fonema vocálico, no arquivo gerado pelo sistema *CPqD,* é na verdade o fonema /ʊ/, cuja pronúncia pode ser descrita como intermediária entre os fonemas /o/ e /u/. Situação semelhante ocorre com os fonemas /i/ e /e/, como por exemplo na palavra "*debat**e**".*

É importante destacar que tais diferenças de pronúncia podem ou não ser consideradas erros do sistema analisado. De maneira geral, isso dependerá do objetivo do testador e da pronúncia considerada correta para o sistema a ser testado. Caso as diferenças de pronúncia sejam vistas como situações normais, de acordo com as expectativas do testador, o algoritmo poderia ser parametrizado de modo a flexibilizar determinadas pronúncias.

A validação do extrator de fonemas foi realizada de modo análogo ao extrator anterior. O descritor de oráculos criado foi o seguinte:

```
extractor PhonemeExtractor {phoneme = "k" return = "Q" interval = 100 nth = 0
config = "config_fonema_cpqd.xml"}
```

Com a execução no *framework O-FIm* foi possível analisar a precisão da extração do fonema consonantal */k/*. Após a coleta dos resultados, o parâmetro *"phoneme"* foi alterado e o mesmo processo repetido para os fonemas */t/* e */m/*.

De modo semelhante à validação do extrator de vogais, os resultados obtidos foram comparados com a extração manual realizada, e os dados gerados foram consolidados de acordo com o número de ocorrências de cada fonema nas palavras. Os percentuais obtidos representam o total de palavras que tiveram o resultado do extrator igual ao resultado esperado pela análise manual. Na Tabela 7 são apresentados os resultados obtidos para os dois sistemas TTS avaliados.

Tabela 7. Percentual de palavras com resultado correto

Parâmetro *Phoneme*	**Número de Ocorrências do Fonema**			
	CPqD Texto-Fala		*Google TTS*	
	0	**1**	**0**	**1**
K	74%	53%	81%	63%
T	85%	48%	68%	52%
M	73%	47%	52%	20%
Média de acertos	77%	49%	67%	45%

Os resultados mostram, assim como ocorrido no extrator de vogais, independência do funcionamento em relação aos dois sistemas avaliados. Em contrapartida, é possível observar uma redução da capacidade de identificação em relação ao desempenho obtido pelo extrator de vogais.

A análise dos casos de erro mostrou o agravamento do problema de divisão do sinal no extrator de fonemas, principalmente devido ao menor intervalo de tempo, em geral, utilizado na pronúncia dos fonemas consonantais, quando comparados com os fonemas vocálicos. Além disso, na literatura sobre o tema foram encontradas diversas técnicas distintas utilizadas na identificação de fonemas, cada uma apresentando vantagens e desvantagens, o que sugere, como possibilidade de melhoria, a combinação

de outros métodos com o algoritmo de extração MFCC. A identificação de vogais por meio da análise de formantes, em oposição, é o método utilizado de maneira mais generalizada em outros trabalhos pesquisados.

5.2. Resultados da execução dos oráculos

Na segunda etapa de validação foram definidos e parametrizados no *framework O-FIm* dois oráculos de teste, utilizando cada um dos extratores de características desenvolvidos. Foi usada como função de similaridade dos oráculos a distância euclidiana entre os vetores de características de dois objetos comparados.

No descritor do primeiro oráculo foram criadas 15 entradas referentes à chamada do extrator de vogais. As cinco primeiras correspondem à chamada do oráculo para os cinco valores possíveis do parâmetro "*vowel",* fixando o parâmetro "*return"* com o valor "*Q*" (ver Seção 4.2). Mais cinco entradas semelhantes foram criadas, porém com a alteração do parâmetro "*return"* para o valor "*P*", visando a extrair as informações de posição, ou ordem de ocorrência, dos fonemas vocálicos nas palavras do conjunto de dados. Por fim, cinco novas entradas foram criadas mantendo-se o valor do parâmetro *"return",* porém alterando-se o parâmetro *"interval"* para metade do valor anterior. O objetivo dessa alteração foi evitar a influência de possíveis coincidências no cálculo do extrator quando *return = Q*, já que a fórmula de cálculo adotada possibilita retornos iguais para palavras distintas.

O descritor do segundo oráculo foi criado de maneira análoga, porém utilizando o extrator de fonemas e alternando o parâmetro *"phoneme"* entre os três fonemas consonantais de interesse. Com isso, o vetor de características gerado na execução dos dois oráculos de testes descritos possui um total de 15 posições, cada uma representando o retorno de uma das chamadas do extrator. Foram consideradas como saídas corretas (oráculos) os arquivos gerados pelo sistema *CPqD Texto-Fala.*

O retorno do oráculo de testes, quando aplicado a duas entradas, é um valor *booleano* indicando se estas são ou não semelhantes, com base em um limiar definido empiricamente. Na execução dos oráculos foram utilizados como entradas arquivos de som contendo as mesmas palavras, porém cada uma delas gerada por um dos sistemas. Assim, o resultado esperado seria a identificação da semelhança (valor de retorno positivo) para todos os arquivos.

O resultado obtido a partir deste experimento foi a identificação de 76% das palavras como semelhantes para o extrator de vogais e 63% para o extrator de fonemas. As demais palavras representam os casos de erro, ou seja, casos em que dois arquivos contendo a mesma palavra, gerada utilizando os dois sistemas, foram considerados diferentes pelo extrator.

A análise dos casos de erro mostrou que, em geral, a semelhança não foi verificada devido às imprecisões já apontadas relacionadas à divisão dos arquivos em partes correspondentes aos fonemas. Apesar disso, existem casos de divergências de pronúncia entre os dois sistemas, o que pode ser considerado, dependendo dos objetivos do testador, como a identificação de um possível erro no sistema que está sendo testado pelo oráculo. Pode-se observar também que os números obtidos são coerentes com a precisão de identificação dos extratores, obtida nos experimentos anteriores, indicando que melhorias nos algoritmos de identificação de vogais e fonemas pode, consequentemente, aumentar a precisão dos oráculos de teste que utilizem os extratores.

5.3. Flexibilidade dos extratores

Os extratores desenvolvidos possuem arquivos de configuração que permitem adaptá-los a saídas geradas por sistemas TTS diferentes dos avaliados. Dentre as características que podem ser facilmente configuradas estão a velocidade da fala e o volume. Além disso, os extratores podem ser configurados para identificar vozes masculinas e femininas, que se diferenciam em relação a seus formantes. Mesmo considerando outros idiomas dos quais o conjunto dos fonemas vocálicos e consonantais analisados faça parte, é possível que os algoritmos mantenham níveis de acerto semelhantes aos obtidos.

Dependendo das características da voz, pode ser necessário calibrar os parâmetros do extrator, o que pode ser feito por meio da aplicação dos algoritmos à saídas modelo, ou seja, exemplos de ocorrências de cada fonema de interesse. Para cada uma delas o algoritmo pode fornecer os valores de formantes e coeficientes mel-cepstrais identificados, para que então os arquivos de configuração sejam ajustados.

Algumas funções presentes nos algoritmos desenvolvidos, como as funções matemáticas, de análise de frequências e de divisão de palavras foram desenvolvidas de modo independente e podem ser reutilizadas por outros extratores, dependendo das características que se deseja avaliar. Apesar disso, o esforço necessário para o desenvolvimento de novos extratores dependerá da complexidade da técnica utilizada. Os extratores apresentados, assim com o *framework O-Fim*, foram desenvolvidos em linguagem Java. Os conceitos utilizados na implementação podem ser usadas para implementação usando outras tecnologias, desde que estejam disponíveis bibliotecas semelhantes às citadas no presente trabalho.

6. Considerações Finais

Este trabalho buscou desenvolver e validar algoritmos de extração de características de sinal sonoro e aplicá-los no contexto do teste de software, por meio da integração com o *framework O-FIm.* Para isso foram utilizados de maneira integrada conceitos de CBIR, CBAR e oráculos de teste, além da adaptação de métodos disponíveis na literatura, com o intuito de aproveitar e complementar as estratégias utilizadas para a extração de características de sinais.

Conforme apresentado, são grandes os desafios ao desenvolver métodos e ferramentas para o teste automatizado de sistemas com saídas complexas, como imagens e sons, e não existem muitos trabalhos focados na aplicação de oráculos de teste voltados a esse tipo de sistema. O *framework O-FIm,* já consolidado como ferramenta de apoio na definição de oráculos para sistemas com saída gráfica, vem sendo evoluído para tornar-se uma ferramenta que atenda também aos domínios de sinais e outras saídas complexas.

Nesse sentido, o presente trabalho integra-se a estes esforços para que, futuramente, um grande número de extratores esteja disponível para a definição de oráculos. Além disso, muitas das funções implementadas por estes dois extratores iniciais poderão ser reaproveitadas por extratores futuros, auxiliando assim no crescimento da ferramenta.

Os resultados obtidos mostram que os métodos utilizados são viáveis e podem cumprir com os objetos levantados. Porém, é de interesse que em próximos trabalhos os algoritmos sejam refinados por meio da utilização de novas técnicas que possam sanar os problemas encontrados, aumentado assim a precisão da identificação da fala.

Referências

Baresi, L.; Young, M. *Test Oracles*. Technical Report CIS-TR-01-02. University of Oregon - Deptartment of Computer and Information Science, Eugene/OR, USA, 2001. Disponível em: <http://ix.cs.uoregon.edu/~michal/pubs/oracles.html>. Acesso em 09 mar. 2013.

Barioni, M. C. N. *Operações de consulta por similaridade em grandes bases de dados complexos*. 2006. Tese (Doutorado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos/SP, 2006.

Bertolino, A. Software testing research: Achievements, challenges, dreams. In: *Future of Software Engineering (FOSE 2007)*, Minneapolis/MN, USA, 2007, p. 85-103.

Bosi, M; Goldberg, R. E. *Introduction to digital audio coding and standards*. Kluwer Academic Publishers, 458 p., 2002.

Bresolin, A. A. *Estudo do Reconhecimento de Voz para o Acionamento de Equipamentos Elétricos via Comandos em Português*. 2003. Dissertação (Mestrado em Automação Industrial) – Centro de Ciências Tecnológicas, Universidade do Estado de Santa Catarina, Santa Catarina, 2003.

Bueno, J. M.; Chino, F. J. T.; Traina, A. J. M.; Traina, C.; Marques, P. M. A. How to Add Content-based Image Retrieval Capability in a PACS. In: *Proceedings of the 15th IEEE International Conference on Computer Based Medical Systems (CBMS 2002)*, Maribor, Slovenia, 2002, p. 321-326.

Callou, D.; Leite, Y. *Iniciação à fonética e à fonologia*. 5.ed. Rio de Janeiro: Jorge Zahar, 1995.

Chang, T.; Yeh, T.; Miller, R. GUI testing using computer vision. In: *Proceedings of the 28th International Conference on Human Factors in Computing Systems (CHI 2010)*, Atlanta/GA, USA, 2010, p. 1535-1544.

Charette, R. N. Why software Fails. *Online, IEEE Spectrum*, 2005. Disponível em: <http:// http://spectrum.ieee.org/computing/software/why-software-fails>. Acesso em 09 mar. 2013.

Chaudhary, P.; Hamid, N.H.B. Amplitude independent feature extraction for effective speech retrieval. In: *Proceedings of the 2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC 2012),* Solan, Índia, 2012, p. 861-864.

Coutinho, V. A.; Oliveira, H. Um sistema para classificação automática de gêneros musicais. In: *Anais do XX Congresso de Iniciação Científica da Universidade Federal de Pernambuco (UFPE)*, Recife/PE, Brasil, 2012.

Delamaro, M. E.; Nunes, F. L. S.; Oliveira, R. A. P. Using concepts of content-based image retrieval to implement graphical testing oracles. *Software Testing, Verification and Reliability*, v. 23, n. 3, p. 171–198, 2013.

Dias, M. V. *Reconhecimento de vocábulos utilizando Redes Neurais*. 2008. Monografia (Bacharelado em Engenharia da Computação) – Faculdade de Tecnologias e Ciências Sociais Aplicadas, Centro Universitário de Brasília, Brasília/DF, 2008.

Esfandian, N.; Razzazi, F.; Behrad, A. A feature extraction method for speech recognition based on temporal tracking of clusters in spectro-temporal domain. In: *Proceedings of the 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*, Shiraz, Irã, 2012, p. 12-17.

Ghoraani, B.; Krishnan, S. Time–Frequency Matrix Feature Extraction and Classification of Environmental Audio Signals. *IEEE Transactions on Audio, Speech, and Language Processing*, v. 19, n. 7, p. 2197-2209, 2011.

Guihua, W.; Jian, T.; Lijun, J.; Jia, W. Audio feature extraction for classification using relative transformation. In: *Proceedings of the International Conference on Audio, Language and Image Processing (ICALIP 2012)*, Shanghai, China, 2012, p. 260-265.

Hoffman, D. Using Oracles in Test Automation. In: *Proceedings of the 19th Pacific Northwest Software Quality Conference (PNSQC 2001)*, Portland/OR, USA, 2001, p. 91-102.

Hyunsin, P.; Takiguchi, T.; Ariki, Y. Integration of Phoneme-Subspaces Using ICA for Speech Feature Extraction and Recognition. In: *Proceedings of the Hands-Free Speech Communication and Microphone Arrays (HSCMA 2008)*, Trento, Itália, 2008, p. 148,151.

Junchang, Z.; Yuanyuan, C.; Jian, Z. Speech feature extraction of KPCA based on kernel fuzzy K-means Clustering. In: *Proceedings of the International Conference on Computer Science and Service System (CSSS 2011)*, Nanjing, China, 2011, p. 756-759.

Klatt, D. Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America (JASA)*, v. 82, n. 3, p. 737-793, 1987.

Leite, H.; Carvalho, S.; Cinto, T.; Arantes, D. Avaliação de Vozes Artificiais: Inteligibilidade, Compreensibilidade e Naturalidade. In: *Anais do Computer on the Beach*, Florianópolis/SC, Brasil, 2014, p. 144-153.

Memon, A.; Pollack, M.; Soffa, M. Automated Test Oracles for GUIs. *ACM SIGSOFT Software Engineering Notes,* v. 25, n. 6, p. 30-39, 2000.

Memon, A. *A Comprehensive Framework for Testing Graphical User Interfaces.* Ph.d. thesis, University of Pittsburgh, Pittsburgh/PA, USA, 2001.

Miranda, I. I.; Meireles, A. Análise acústico-comparativa de vogais brasileiras com vogais norte-americanas. In: *Anais do I Congresso Nacional de Estudos Linguísticos*, Vitória/ES, Brasil, 2011.

Mitrovic, D.; Zeppelzauer, M.; Breiteneder, C. Features for Content-Based Audio Retrieval. In: *Advances in Computers – Improving the Web,* v. 78, p. 71-150, 2010.

Oliveira, R. A. P. *Apoio à automatização de oráculos de teste para programas com interfaces gráficas*. 2012. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos/SP, 2012.

Oliveira, R. A. P.; Delamaro, M. E.; Nunes, F. L. S. Estrutura para utilização de Recuperação de Imagem Baseada em Conteúdo em oráculos de teste de software com saída gráfica. In: *Anais do IV Workshop de Visão Computacional (WVC 2008),* Bauru/SP, Brasil, 2008, p. 7-12.

Qiang W.; Liqing Z.; Guangchuan S. Robust Multifactor Speech Feature Extraction Based on Gabor Analysis. *IEEE Transactions on Audio, Speech, and Language Processing*, v. 19, n. 4, p. 927-936, 2011.

Rafi, D.; Moses, K.; Petersen, K.; Mantyla, M. Benefits and limitations of automated software testing: Systematic literature review and practitioner survey. In: *Proceedings of the 7th International Workshop on Automation of Software Test (AST 2012)*, Zurique, Suiça, 2012, p. 36-42.

Russo, I.; Behlau, M.. *Percepção da fala: análise acústica do português brasileiro*. São Paulo: Lovise, 1993.

Seyedin, S.; Ahadi, M. Feature extraction based on DCT and MVDR spectral estimation for robust speech recognition. In: *Proceedings of the 9th International Conference on Signal Processing (ICSP 2008)*, Leipzig, Alemanha, 2008, p. 605-608.

Smeulders, A. W. M.; Worring, M.; Santini, S.; Gupta, A.; Jain, R. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, v. 22, n. 12, p. 1349-1380, 2000.

Sommerville, I. *Software Engineering.7th ed.* Harlow, UK: Addison Wesley (International Computer Science Series), 784 p., 2004.

Tafner, M. A. *Reconhecimento de palavras isoladas usando redes neurais artificiais*. 1996. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis/SC, 1996.

Taylor, P. *Text-to-speech synthesis*. Cambridge: Cambridge University Press, 626 p., 2009.

Terssetti, F. B. *Desenvolvimento de um identificador automático de sonoridade em fonemas plosivos e fricativos para auxílio na terapia fonoaudiológica de crianças*. 2010. Dissertação (Mestrado em Engenharia da Informação) - Universidade Federal do ABC, Santo André/SP, 2010.

Torres, R. da S.; Falcao, A. X. Content-based image retrieval: Theory and applications. *Revista de Informática: Teórica e Aplicada*, v. 13, n. 2, p. 165–189, 2006.

Viaro, M. E; Guimarães Filho, Z. O. Análise quantitativa da frequência dos fonemas e estruturas silábicas portuguesas. *Estudos Linguísticos*, v.36, p. 28-36, 2007.

Xiang, W.; Feng, T.; Jingao, L. An improved speech feature extraction algorithm using DWT. In: *Proceedings of the International Conference on Audio, Language and Image Processing 2008 (ICALIP 2008)*, Shanghai, China, 2008, p. 1086-1090.

Xiaolan, Z.; Zuguo, W.; Jiren, X.; Keren, W.; Jihai, N. Speech Signal Feature Extraction Based on Wavelet Transform. In: *Proceedings of the International Conference on Intelligent Computation and Bio-Medical Instrumentation (ICBMI 2011)*, Wuhan, China, 2011, p.179-182.

Yu-Yun, C.Evaluation of TTS systems in intelligibility and comprehension tasks. In: *Proceedings of the 23rd Conference on Computational Linguistics and Speech Processing,* Stroudsburg/PA, USA, 2011, p. 64–78.