UNIVERSIDADE DE CAXIAS DO SUL

CRISTIAN KEIL DE ABREU

RECONHECIMENTO AUTOMÁTICO DE LÍNGUA BASEADA EM TOKENS

CAXIAS DO SUL

2013

UNIVERSIDADE DE CAXIAS DO SUL

CRISTIAN KEIL DE ABREU

RECONHECIMENTO AUTOMÁTICO DE LÍNGUA BASEADA EM TOKENS

Trabalho de Conclusão – TCC de Bacharel em Ciência da Computação pela Universidade de Caxias do Sul.
Área de concentração: Ciência da Computação
Orientador: Prof. Dr. André Gustavo Adami

CAXIAS DO SUL

2013

UNIVERSIDADE DE CAXIAS DO SUL

CRISTIAN KEIL DE ABREU

RECONHECIMENTO AUTOMÁTICO DE LÍNGUA BASEADA EM TOKENS

Trabalho de Conclusão – TCC de Bacharel em Ciência da Computação pela Universidade de Caxias do Sul.
Área de concentração: Ciência da Computação
Orientador: Prof. Dr. André Gustavo Adami

Banca Examinadora:

Prof. Dr. André Gustavo Adami
Universidade de Caxias do Sul – UCS

Prof. Dr. André Luis Martinotto
Universidade de Caxias do Sul – UCS

Profa. Dra. Adriana Miorelli Adami
Universidade de Caxias do Sul – UCS

AGRADECIMENTOS

Gostaria de agradecer a todos que participaram desta jornada até o final do TCC. Primeiro gostaria de agradecer a Deus por toda a ajuda dada, por ter guiado o meu caminho até aqui e nunca ter desistido de mim. Gostaria de agradecer aos meus pais, Juarez e Maria, e aos familiares mais próximos e minha namorada, Denuza, que conviveram comigo e sempre me apoiaram. Ao prof. André Adami por suas orientações prestadas, pelo seu tempo disponibilizado para o TCC. Sem sombra de dúvidas foram muito importantes para este trabalho. Gostaria de agradecer também a todos (amigos, familiares, irmãos, pessoal do trabalho) que de alguma forma me ajudaram a chegar até aqui. Não vou citar nomes para não correr o risco de esquecer ninguém. Um muito obrigado a todos.

“*No mundo há muitas línguas diferentes, mas cada uma faz sentido. Porém, se eu não entendo a língua na qual alguém está falando comigo, então quem fala essa língua é estrangeiro para mim, e eu sou um estrangeiro para ele.*”

I Coríntios 14:10 – 11.

RESUMO

Um sistema de reconhecimento automático de língua tem por objetivo identificar a língua que está sendo falada por um locutor. O reconhecimento da língua pode reduzir o problema de comunicação entre pessoas de diferentes partes do mundo. Este problema ocorre quando a comunicação envolve vários idiomas. Uma comunicação só pode ocorrer se ambas as partes conseguirem se entender. Sistemas de reconhecimento de língua podem ser aplicados na tradução automática de língua ou em serviços de atendimento ao cliente. Este tipo de tecnologia pode ser aplicado para a qualificação do atendimento de empresas e prestadores de serviços aos seus clientes de outros países. Este trabalho tem por objetivo propor um sistema automático de reconhecimento da língua baseado em tokens para servir de auxílio a esses tipos de tecnologias. O sistema desenvolvido explora informações acústicas e prosódicas para detectar a língua. Para avaliar o desempenho do sistema, foi utilizado o paradigma de avaliação do NIST 2003. Além disso, para montar os tokenizadores acústicos foram utilizados as bases de dados do OGI-TS e o OGI 22 *Languages*. Dentre os melhores resultados apresentados, o tokenizador prosódico obteve resultados inferiores a 25%. Aplicando a fusão entre características prosódicas e acústicas, o desempenho ficou próximo de 22%.

Palavras-chaves: Sistema de Reconhecimento automático de Língua. Tokens.

ABSTRACT

A system for automatic language recognition aims to identify the language being spoken by a speaker. The language recognition can reduce the problem of communication between people from different parts of the world. This problem occurs when the communication involves multiple languages. Communication can only occur if both parties are able to understand each other. Language recognition systems can be applied in automatic language translation services or customer service. This type of technology can be applied for the assistance rendered to enterprises and service providers to their customers in other countries. This work aims to propose an automatic system for language automatic recognition based on tokens to serve as aid to these types of technologies. The developed system explores acoustic and prosodic information to detect the language. To evaluate the system performance, the evaluation paradigm NIST 2003 was used. In addition, to mount the acoustic tokenizers databases OGI-TS and OGI 22 Languages were used. Among the best results, prosodic tokenizer got lower results at 25%. Applying the fusion between prosodic and acoustic features, performance was close to 22%.

Keywords: Automatic Language Recognition System, Tokens.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

Atualmente, a globalização é um fenômeno que une pessoas de diferentes partes do mundo (WONG, 2004). Porém, com o aumento da globalização também surge o problema de comunicação, principalmente devido aos diferentes idiomas falados em todo o mundo. Para que exista uma comunicação efetiva é necessário que todos os participantes da comunicação se entendam. O reconhecimento da língua pode oferecer um meio para facilitar o reconhecimento do meio de comunicação.

O reconhecimento automática da língua possui uma ampla possibilidade de aplicações em serviços para múltiplas línguas. Por exemplo, em um serviço de atendimento ao cliente que presta serviços de informações em vários idiomas, uma pessoa que fala outra língua liga para obter uma informação específica em seu idioma. O sistema deve receber a ligação do indivíduo, identificar o idioma dele, consultar um cadastro para identificar operadores capacitados ao determinado idioma, e redirecionar a ligação para o atendente selecionado pelo sistema.

Outra aplicação para facilitar a comunicação entre pessoas que falam diferentes idiomas é a tradução automática da fala. Atualmente algumas empresas, como a *Raytheon BBN Technologies*[1] e a *SpeechGear*[2], possuem produtos que realizam a tradução automática, porém em um número limitado de línguas. Estes produtos possuem o mesmo processo para a tradução da língua, transformam a fala em texto, traduzem o texto para a outra língua e depois transformam o texto em fala.

O ser humano possui uma grande capacidade para reconhecer a língua falada por uma pessoa (por exemplo, Inglês, Espanhol, Português). Segundo Navratil (2006, p. 234), “o ser humano é o modelo definitivo para um sistema de reconhecimento automático de língua”. Isto leva em conta que o ser humano pode reconhecer uma língua a partir de uma curta amostra de áudio de uma língua familiar a ele. Mesmo para uma língua que não lhe seja familiar, o ser humano pode associar a outra língua que lhe seja mais familiar através de alguma semelhança encontrada, por exemplo, que “soa como Japonês”.

O uso de um sistema de reconhecimento automático da língua inclui alguns benefícios como a redução de custos e o menor tempo de treinamento de funcionários

[1] http://www.bbn.com/technology/speech/speech_to_speech_translation
[2] http://www.speechgear.info/products/interact

associado ao sistema, além de possibilitar o melhor atendimento ao cliente (AMBIKAIRAJAH et al, 2011). Por exemplo, ao invés de treinar várias pessoas para reconhecer corretamente um conjunto de línguas, o sistema de reconhecimento de língua seria treinado uma única vez e depois executado em diversos computadores simultaneamente.

Apesar dos avanços na área de reconhecimento da língua, aplicações que utilizem este tipo de tecnologia ainda não estão disponíveis em todas as empresas e prestadoras de serviço, tornando assim muito difícil a comunicação entre pessoas de diferentes línguas. Por isso é muito importante prover recursos que facilitem o reconhecimento da língua que está sendo falada no momento para que as empresas e as prestadoras de serviço possam atender adequadamente os cidadãos estrangeiros.

1.1 OBJETIVOS

O objetivo principal deste trabalho de conclusão de curso é desenvolver um sistema de reconhecimento de língua baseado em tokens para decisão. O uso de tokens permite que novas fontes de informação contidas na voz sejam exploradas e agregadas a outros sistemas, com o intuito de melhorar o desempenho de reconhecimento destes sistemas. Na proposta deste trabalho, o sistema desenvolvido utiliza como fonte de dados as informações prosódicas e acústicas, ambas sendo representadas por tokens.

1.2 ESTRUTURA DO TRABALHO DE CONCLUSÃO DE CURSO

Este trabalho está organizado da seguinte forma:

- **Capítulo 2:** traz uma visão geral sobre sistemas de reconhecimento automático de língua, avaliação de desempenho, bases de áudio e paradigmas de avaliação.
- **Capítulo 3:** traz uma visão geral sobre sistemas de reconhecimento automático da língua que utilizam tokens, além das técnicas aplicadas para realizar a fusão de resultados obtidos através de diferentes classificadores.
- **Capítulo 4:** descreve a proposta de solução para o problema levantado neste trabalho de conclusão de curso. Também apresenta os resultados obtidos pelo sistema utilizando os diferentes tokenizadores aplicados.

2. RECONHECIMENTO AUTOMÁTICO DA LÍNGUA

O objetivo deste capítulo é revisar o reconhecimento automático da língua e suas características. A Seção 2.1 descreve como um sistema de reconhecimento automático da língua pode ser classificado. A Seção 2.2 descreve as informações que podem ser utilizadas para o reconhecimento da língua. A Seção 2.3 apresenta a arquitetura geral e as fases que um sistema de reconhecimento de língua possui. A Seção 2.4 apresenta como são medidos os desempenhos dos sistemas de reconhecimento. A Seção 2.5 descreve algumas bases de áudio que fornecem dados para validação para o reconhecimento da língua. A Seção 2.6 apresenta o paradigma de avaliação para sistemas de reconhecimento de línguas. A Seção 2.7 apresenta o desempenho de outros sistemas de reconhecimento de línguas que utilizam informações acústicas como fonte de dados.

O reconhecimento automático da língua tem por objetivo reconhecer um idioma de um locutor desconhecido utilizando uma amostra de áudio relativamente curta (entre 3 e 50 segundos) (ROUAS, 2005). Um sistema ideal de reconhecimento automático da língua deve possuir algumas características (AMBIKAIRAJAH et al, 2011):

- não pode ser tendencioso para reconhecer uma língua ou um grupo de línguas;
- o tempo de computação deve ser curto;
- não pode ter o desempenho prejudicado ao aumentar a quantidade de línguas ou diminuir o tempo de duração da expressão de teste;
- deve ser robusto em relação a locutores, variações de canais e outros ruídos.

2.1 CLASSIFICAÇÃO DOS SISTEMAS

Os sistemas de reconhecimento automático da língua podem ser classificados conforme as características a seguir:

- **Identificação ou detecção (verificação):** é o modo de operação do sistema (REYNOLDS e CAMPBELL, 2008). O objetivo na identificação da língua é determinar qual a língua falada em um determinado segmento de áudio a partir de um conjunto de línguas conhecidas. Neste modo de operação, a amostra da língua que está sendo testada precisa ser comparada com todas as línguas existentes no sistema. Apenas após todas estas comparações é que o sistema pode tomar uma

decisão para a amostra (identificar a língua ou informar que não pertence ao grupo). Os sistemas de identificação podem trabalhar em dois modos de operação relacionados ao grupo de línguas conhecidas: fechado ou aberto (REYNOLDS e CAMPBELL, 2008; ROSENBERG, BIMBOT e PARTHASARATHY, 2008). No modo de grupo fechado, sabe-se que a língua desconhecida da amostra pertence a um grupo conhecido de línguas. No modo de grupo aberto, a língua desconhecida da amostra pode ou não ser parte de um grupo de línguas conhecidas pelo sistema. O objetivo na detecção da língua é determinar se uma língua está sendo usada ou não em um segmento da sentença. Neste tipo de sistema, a amostra da língua é comparada apenas com a língua na qual a amostra alega-se ter sido falada. Para tomar a decisão são dadas ao sistema duas hipóteses. A primeira hipótese envolve o modelo da língua alvo que está sendo falada (isto é, a língua alegada para a amostra é a língua falada), enquanto a segunda hipótese envolve os modelos de outras línguas (isto é, a língua alegada não é a língua falada na amostra). A razão entre as verossimilhanças das duas hipóteses é comparada a um valor definido como limiar. Após esta comparação o sistema pode tomar a decisão de aceitar ou rejeitar a primeira hipótese. O modo de detecção pode ser aplicado em situações onde se aplica o modo de identificação para conjuntos abertos.

- **Dependente ou independente de texto:** é baseada no texto que é falado pelo locutor (REYNOLDS e CAMPBELL, 2008). No modo dependente de texto, o sistema possui um texto pré-definido que deve ser falado pelo locutor. Este tipo de informação pode representar uma vantagem, permitindo assim obter resultados melhores, caso sejam aplicadas as metodologias adequadas, como por exemplo, que levem em consideração os recursos léxicos. No modo independente de texto, o sistema não possui um texto pré-definido para ser falado pelo locutor, permitindo assim que as amostras utilizadas não sejam limitadas pelo vocabulário.
- **Tipo de característica (espectrais ou tokens):** as características espectrais são representadas pela extração de medidas do espectro da voz contida no sinal de entrada (SHEN et al, 2008). As medidas extraídas do sinal de voz estão relacionadas às propriedades acústicas da voz. Para representar as características espectrais, as informações são extraídas de pequenos segmentos de áudio de tamanho fixo. Os tokens são a transformação do sinal de entrada para uma representação utilizando símbolos. Eles podem ser baseados em segmentação

fonética ou em segmentação baseada em dados. Para a extração de tokens não existe um tamanho fixo para as janelas e cada token pode representar um tamanho diferente no sinal de fala.

O sistema de reconhecimento automático de língua desenvolvido pode ser classificado de acordo com os critérios estabelecidos acima. O sistema pode ser descrito como um sistema de detecção, independente de texto e baseado em tokens.

2.2 INFORMAÇÕES UTILIZADAS NO RECONHECIMENTO

Cada língua possui uma série de características que permite discriminá-la entre diversas línguas. Tais características são usadas pelas pessoas para diferenciar uma língua das outras (WANG, 2008).

As informações que podem ser exploradas para o reconhecimento da língua podem ser agrupadas nos seguintes níveis, como mostra a Figura 1:

- **Acústicas**: nível inicial da análise da produção de fala e o mais próximo do material original da fala (WANG, 2008). A partir das informações acústicas é possível obter informações fonotáticas e de nível de palavras através do uso de codificadores (WONG, 2004). A informação acústica é definida pela frequência, tempo e intensidade da fala. Para a extração de informações acústicas, as técnicas de parametrização mais usadas são coeficientes preditivos lineares (em Inglês *Linear Predictive Coefficients*, LPC), os coeficientes cepstrais de frequência mel (em Inglês *Mel Frequency Cepstral Coefficients*, MFCC), Coeficientes preditivos lineares perceptivos (em Inglês *Perceptual Linear Predictive Coefficients*, PLP) (AMBIKAIRAJAH et al, 2011). Sua representação é feita através da sequência de vetores de características e cada vetor representa um período específico de tempo (RONG, 2012).
- **Fonéticas:** são referentes às propriedades físicas dos sons da fala e da voz humana. Os fonemas são um grupo finito de sons significativos que podem ser produzidos fisicamente (WONG, 2004). Cada língua não utiliza o grupo completo de fonemas e sim apenas uma parte deste conjunto. Isto significa que os fonemas podem ser encontrados em um grupo de línguas, porém em outros grupos não. Por exemplo, enquanto o Holandês e o Alemão possuem o fonema

/x/, no Francês e no Italiano não o têm (WANG, 2008). A frequência com que os fonemas ocorrem também muda de uma língua para outra. Por exemplo, fonemas nasais ocorrem mais no Mandarim do que no Inglês.

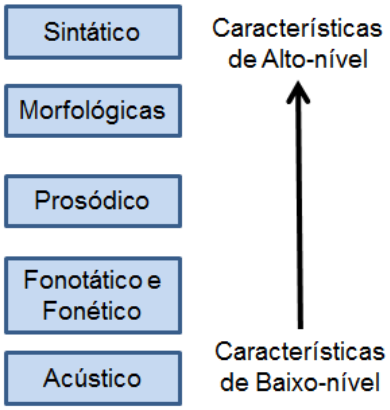

Figura 1. Níveis de Informações da língua.

- **Fonotáticas:** são referentes à distribuição e as combinações dos fonemas em uma língua. Entre as línguas existe uma grande variação nas regras fonotáticas, isto é, diferentes línguas podem ter diferentes regras para a construção das sequências fonéticas (RONG, 2012). Estas regras geram restrições fonéticas que podem variar de língua para língua. Por exemplo, o Inglês possui restrições mais flexíveis relacionados a múltiplas consoantes consecutivas se comparado com o Japonês. No Alemão existe a combinação entre os fonemas /i/ e /ç/, enquanto no Inglês esta sequência não existe (NAVRATIL, 2006). Atualmente, vários sistemas de reconhecimento automático de língua se beneficiam do uso deste tipo de informação, pois ela carrega mais informações que permitem discriminar a língua que os próprios fonemas.
- **Prosódicas:** referem-se à estrutura acústica que se estende em vários segmentos (RONG, 2012). Nesta estrutura são encontrados elementos que variam de uma língua para outra como: ritmo (duração dos fonemas, velocidade da fala), entonação (variação de contorno do tom) e estresse (acento tônico). Estas variações podem ser observadas nas características da prosódia durante a fala. Alguns fonemas são encontrados em várias línguas, porém a sua duração varia de acordo com a língua (WANG, 2008). A entonação pode gerar diferentes interpretações da sentença dependendo da língua, como no Inglês, onde o tom difere em uma frase em forma de pergunta em relação à mesma frase em forma afirmativa. O padrão de estresse dentro de uma palavra pode variar de uma língua

para outra. No Francês o estresse é encontrado no final da palavra, enquanto no Húngaro a primeira sílaba é a tônica.

- **Morfológicas:** referem-se à estrutura das palavras (WANG, 2008). Palavras são as menores unidades da sintaxe e na maioria das línguas elas podem estar relacionadas a outras palavras baseada em regras morfológicas. Diferentes línguas possuem seus próprios grupos de vocabulários e sua própria maneira de formar as palavras. A morfologia pode fornecer pistas para identificar uma língua (HARPER e MAXWELL, 2008). Por exemplo, podem-se utilizar sufixos comuns para discriminar algumas línguas românicas (-ment em Francês, -miento em Espanhol, -mento em Português e –mente em Italiano). Com o conhecimento morfológico das línguas candidatas, seu uso poderia ser associado em partes específicas da amostra quando está sendo discriminado entre duas línguas.
- **Sintáticas:** referem-se às regras de uma língua que dizem como as palavras se unem para formar uma sentença (WONG, 2004). Os padrões para a sintaxe variam de uma língua para outra. Mesmo palavras compartilhadas em idiomas diferentes podem possuir diferentes contextos sintáticos. Por exemplo, a palavra "bin" existe no Alemão e no Inglês, porém os grupos de palavras que antecedem e seguem as palavras são diferentes.

Quando estes tipos de informações são comparados, podem-se observar alguns aspectos importantes. As informações acústicas são mais fáceis de extrair do que informações de mais alto nível, porém as variações de locutores e de canais presentes podem tornar os dados voláteis, por isso são consideradas informações de mais baixo nível (AMBIKAIRAJAH et al, 2011). As informações morfológicas e sintáticas da língua são mais confiáveis e possuem informações mais discriminantes e por isso são consideradas informações de mais alto nível. Porém, para utilizar estes tipos de informação é necessário o uso de um grande de vocabulário de fala, que é dependente da língua e pode ser difícil de obter.

2.3 ARQUITETURA GERAL DO SISTEMA

Um sistema de reconhecimento automático de língua é composto por três módulos: extração de características, classificação e decisão (AMBIKAIRAJAH et al, 2011; RONG, 2012), como é mostrado na Figura 2. Da amostra de voz são extraídas as características que

refletem a informação da língua falada. Um classificador é aplicado às informações obtidas na extração de características para compará-las com os modelos de línguas já existentes. Este processo é executado para encontrar as similaridades entre os modelos e as características extraídas. Com os dados de similaridades calculados para cada amostra, o sistema pode então fazer a sua decisão e identificar a língua da amostra.

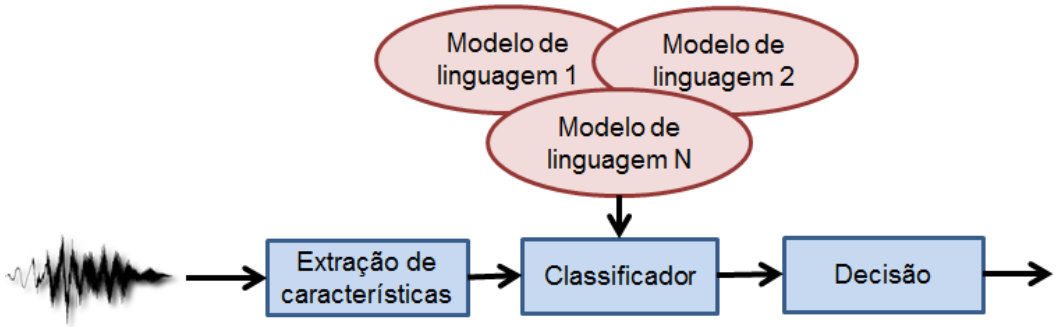

Figura 2. Processo genérico de reconhecimento da língua.

2.3.1 Extração de características

O objetivo da extração de características é produzir uma representação compacta e eficiente da fala, mantendo as informações mais relevantes e excluindo as informações redundantes (AMBIKAIRAJAH et al, 2011). A representação dos dados deve fornecer informações que permitam encontrar as diferenças que possam caracterizar cada língua. Para Wang (2008, p. 32), "a extração de características deveria utilizar apenas as informações discriminativas entre as línguas e também capaz de representar as diferentes informações". Alguns métodos para extração de características acústicas são:

- **Coeficientes Cepstrais de Frequência Mel**: são um dos mais populares bancos de filtros baseados em métodos de parametrização utilizados em aplicações de processamento de sinal (WANG, 2008). O modelo da representação Cepstrais da Frequência Mel é motivado por fatores perceptuais, tornando assim capaz de capturar informações relevantes da percepção auditiva. O uso da escala mel faz com que o método se aproxime da resolução de frequência não linear do ouvido humano (DAVIS e MERMELSTEIN, 1980). O banco de filtros usa um vetor de filtros de banda para analisar a fala através de diferentes frequências de comprimento de banda, correspondente a índices da frequência mel e para obter os coeficientes cepstrais de frequência mel é aplicada a transformada de Fourier

(WANG, 2008). Alguns estudos foram realizados utilizando o MFCC como, por exemplo, as pesquisas de CAMPBELL et al (2006) e Harwart e Hasegawa-Johnson (2010).

- **Codificação Preditiva Linear e Coeficientes Cepstrais de Predição Linear (em Inglês *Linear Predictive Cepstral Coefficients,* LPCC):** para Makhoul (1975, p.561) "o sinal é modelado como uma combinação linear dos valores do passado e valores do presente e do passado de uma entrada hipotética para um sistema onde a saída é um dado sinal". O LPC é baseado no modelo de produção da fala em que as características do trato vocal podem ser modeladas por um filtro de todos polos (em Inglês *all-pole filter*) (MAKHOUL, 1975). As características espectrais de um som em particular podem ser determinadas pelos parâmetros do filtro. O modelo linear preditivo define as frequências formantes, isto é, os picos do envelope formante. Já o LPCC, é descrito por Wong (2004, p. 50) como "o LPC representado no domínio cepstral". Alguns estudos foram realizados utilizando o LPC e o LPCC como, por exemplo, as pesquisas de Campbell et al. (2006) e Wong (2004).
- **Coeficientes Preditivos Lineares Perceptivos:** utiliza três características psicofísicas (relacionamento entre as sensações subjetivas e os estímulos físicos) da audição combinando o banco de filtros e a análise preditiva linear: resolução espectral de banda-crítica (similar ao MFCC, porém calculado usando a escala Bark), a curva de igualdade sonora (modela em diferentes frequências a sensibilidade da audição do ser humano) e a lei da potência de intensidade (relaciona a intensidade do som e a sonoridade percebida) (HERMANSKY, 1990). Para obter os coeficientes de PLP são aplicadas durante o processo equações que representam características auditivas (a curva de mascaramento de banda crítica, a curva de igualdade sonora e a lei da potência de intensidade sonora). Alguns estudos foram realizados utilizando o PLP como, por exemplo, a pesquisa de Wong (2006).
- **Delta Cepstra e Delta-Delta Cepstra:** representam as características dinâmicas da fala (FURUI, 1980). A união das características estáticas (como as extraídas pelos métodos de MFCC, LPC ou LPCC) e características dinâmicas (informações relacionadas à evolução do sinal) da fala são utilizadas para melhorar o desempenho do sistema de reconhecimento de voz.

- **Delta Cepstral Deslocada (em Inglês *Shifted Delta Cepstral*, SDC):** agrega ao Delta Cepstral, que consegue apenas visualizar os dados do momento, uma amostra futura do delta cepstral do sinal de fala (TORRES-CARRASQUILLO et al, 2002). Com isso detalhes menores dos aspectos temporais da língua não serão perdidos, se comparado apenas ao aumento da janela do Delta Cepstral, compensando assim as limitações da derivação de curta duração das características cepstrais. O uso de informação temporal adicional foi inspirado pelo sucesso obtido em pesquisas fonéticas que tem como base a tokenização em múltiplos janelamentos. Alguns estudos foram realizados utilizando o SDC como, por exemplo, a pesquisa do próprio Torres-Carrasquillo et al. (2002).
- **Frequência Modulada (em Inglês *Frequency Modulation*, FM):** é baseado no modelo AM-FM, onde é utilizada para armazenar as modulações que são produzidas pela oscilação do fluxo de ar nas paredes do trato vocal (THIRUVARAN, AMBIKAIRAJAH E EPPS, 2008). Este modelo trata cada ressonância do trato vocal como um sinal de AM-FM e a soma de todas as ressonâncias como os modelos de fala (AMBIKAIRAJAH et al, 2011).

Em sistemas de reconhecimento de língua que utilizem as informações acústicas, é necessário que além de extração das características da fala, que os sistemas também sejam robustos em relação aos ruídos (AMBIKAIRAJAH et al, 2011). Para isso são aplicadas técnicas de normalização às características para reduzir os efeitos de ruídos prejudiciais (como distorção nos canais) ao sistema de reconhecimento. Algumas destas técnicas são:

- **Subtração da Média Cepstral (em Inglês *Cepstral Mean Subtraction*, CMS):** busca minimizar os efeitos nos canais resultantes da transmissão da fala nas linhas telefônicas (AMBIKAIRAJAH et al, 2011). Isto é possível removendo qualquer distorção espectral fixa. O CMS é usado após a extração de características cepstrais (por exemplo, MFCC, LPCC ou PLP), quando a média do vetor características é subtraída de todo o vetor de características. Porém com o uso do CMS também é perdido a média da informação de configuração do trato vocal pertencente à língua (PELECANOS e SRIDHARAN, 2001).
- ***Feature Warping*:** tem por objetivo construir uma representação mais robusta de cada distribuição das características cepstrais (PELECANOS e SRIDHARAN, 2001). Este altera os valores das características em distribuições pré-determinadas em intervalos de tempos específicos. Esta distorção reduz os

problemas de ruídos adicionais e de incompatibilidades de canais.

As técnicas de extração das informações prosódicas podem ser utilizadas para extrair as seguintes características:

- **Entonação:** utiliza as variações de contorno da frequência fundamental, que é a frequência da oscilação das pregas vocais, para representar características específicas (como padrões fonológicos) para cada língua (MARY e YEGNANARAYANA, 2008; ROUAS, 2005).
- **Ritmo:** utiliza a sucessão silábica na fala para representar características rítmicas de uma língua. Para detectar as mudanças nas pregas vocais e a excitação características são analisados os intervalos vocálicos e a duração dos intervalos consonantais (ROUAS, 2005). Isto possibilita caracterizar uma língua em: estressada temporizada (intervalo regular em sílabas estressadas), silábica temporizada (intervalo regular em as sílabas) e mora temporizada (sucessiva mora, peso silábico, é quase equivalente em relação à duração).
- **Estresse:** todas as línguas possuem algumas sílabas que são perceptivelmente mais fortes que as demais (MARY e YEGNANARAYANA, 2008). Estas sílabas são conhecidas como sílabas tônicas. O modo como este estresse aparece na fala é dependente de cada língua. Isto fornece uma informação importante para O reconhecimento da língua. Para detectar estas sílabas são verificadas a energia, a frequência fundamental e a duração do fonema.

O sistema de reconhecimento desenvolvido utiliza informações acústicas e prosódicas para o reconhecimento da língua. O método utilizado para extrair as informações acústicas é o MFCC e as características prosódicas extraídas são a entonação, o ritmo e o estresse.

2.3.2 Classificação

Segundo Huang, Acero e Hon (2001), um classificador pode ser considerado como um dispositivo para particionar as características em regiões de decisão. Os classificadores são utilizados para estimar as similaridades entre os modelos existentes no sistema e as amostras de voz fornecidas para o sistema (AMBIKAIRAJAH et al, 2011). Os modelos das línguas conhecidas pelo sistema são construídos com características das línguas faladas, onde cada modelo possui as características de fala de apenas uma língua (WANG, 2008). Entre

estes classificadores temos:

- **Modelo de Misturas Gaussianas (em Inglês *Gaussian Mixture Models*, GMM):** baseia-se no fato de que qualquer distribuição de probabilidade continua pode ser modelada por uma combinação linear da distribuição gaussiana (AMBIKAIRAJAH et al, 2011). Neste caso, o GMM pode representar as classes acústicas de uma língua (REYNOLDS, QUATIERI e DUNN, 2000). Ele é muito utilizado para sistemas que o reconhecimento da língua não dependente de texto. Alguns estudos foram realizados utilizando o GMM como, por exemplo, as pesquisas de CAMPBELL et al. (2006) e Torres-Carrasquillo et al. (2002).
- **Cadeias Ocultas de Markov (em Inglês *Hidden Markov Models*, HMM):** possui a capacidade de representar as amostras de dados de uma série de tempos discretos (HUANG, ACERO e HON, 2001). O HMM parte do princípio que os dados são caracterizados como um processo aleatório e os parâmetros do processo estocástico podem ser estimados em uma estrutura precisa e bem definida, as cadeias de Markov. As cadeias de Markov introduzem um processo não-determinístico, que gera símbolos de observação de saída em qualquer dado estado. O HMM é mais indicado do que o GMM em aplicações onde o sistema possui um conhecimento prévio do texto e há a coincidência do que foi dito durante o treinamento e os testes (ROSENBERG, BIMBOT e PARTHASARATHY, 2008). Alguns estudos foram realizados utilizando o HMM como, por exemplo, a pesquisa de Wong e Siu (2004).
- **Máquinas de Vetor de Suporte (em Inglês *Support Vector Machines*, SVM):** tem por objetivo construir um hiperplano como superfície de decisão onde a margem de separação entre os exemplos positivos e negativos seja máxima (HAYKIN, 2001; MANNING, RAGHAVAN e SHÜTZE, 2009). O SVM utiliza a estratégia conhecida como “um contra todos”, isto é, que a língua falada em uma amostra será comparada com as demais línguas (SHEN et al, 2008). Se o resultado da comparação for positivo, então elas são a mesma língua. Caso o resultado seja negativo, então elas não pertencem à mesma língua. Alguns estudos foram realizados utilizando o SVM como, por exemplo, a pesquisa de CAMPBELL et al. (2006).
- **Quantização Vetorial (em Inglês *Vector Quantization*, VQ):** utiliza o conceito de quantização (aproximação de sinais de amplitude contínua) conjunta de vários

valores de sinal ou de parâmetros (HUANG, ACERO e HON, 2001). No reconhecimento de língua, o VQ é utilizado para construir um conjunto de amostras representativas da língua agrupando os dados da extração de características (ROSENBERG, BIMBOT e PARTHASARATHY, 2008). Cada agrupamento de características é definido pelo ponto central (média dos dados do agrupamento) e pela distância do ponto central (variância entre os dados do agrupamento).

- **Redes Neurais Artificiais (em Inglês *Artificial Neural Networks*, ANN):** representa mapeamentos não-lineares de diversas entradas para uma série saídas (BISHOP, 1995). A ANN atende a uma grande quantidade de restrições, realizando a avaliação paralela de muitas pistas e com restrições inter-relacionadas (HUANG, ACERO e HON, 2001). Um exemplo disso é a sequência de fonemas ou palavras que são usados para representar uma língua. Alguns estudos foram realizados utilizando a ANN como, por exemplo, as pesquisas de Li, Ma e Lee (2007) e Gauvain, Messaoudi e Schwenk (2004).

2.3.3 Decisão

A decisão a ser tomada depende de qual tarefa está sendo realizada por um sistema de reconhecimento de língua (ROSENBERG, BIMBOT e PARTHASARATHY, 2008). Na tarefa de identificação, uma amostra de voz é comparada ao conjunto de línguas presentes no sistema. O sistema retorna a língua cujo modelo apresenta a maior similaridade de pertencer à amostra de voz. Caso não seja encontrado um modelo com similaridade, o retorno informado ao sistema é sem correspondência (em Inglês *no-match*). Este último caso ocorre apenas quando existe a possibilidade da língua da amostra da voz não pertencer ao grupo de línguas existentes no sistema. Na tarefa de detecção da língua, a amostra é comparada apenas à língua que se diz pertencer à amostra e retorna se há a possibilidade ou não do modelo desta língua pertencer à amostra.

Um fator relevante entre estas duas tarefas é o desempenho computacional relacionado a elas. Na detecção, o tamanho do conjunto de línguas não interfere no desempenho computacional do sistema, pois é verificado apenas a amostra e a língua especificada. Porém, na tarefa de identificação, o aumento da quantidade de línguas contidas no conjunto aumenta a quantidade de comparações com as amostras e com isso o desempenho

computacional do sistema é afetado.

2.3.4 Fases do reconhecimento da língua

Um sistema de reconhecimento de língua pode ser dividido em duas fases: treinamento e reconhecimento (ZISSMAN e BERKLING, 2001; WONG, 2004; RONG, 2012), como ilustra a Figura 3. O treinamento tem por objetivo estimar um modelo para representar cada língua. Estes modelos são gerados a partir das informações específicas extraídas de amostras de fala de cada língua.

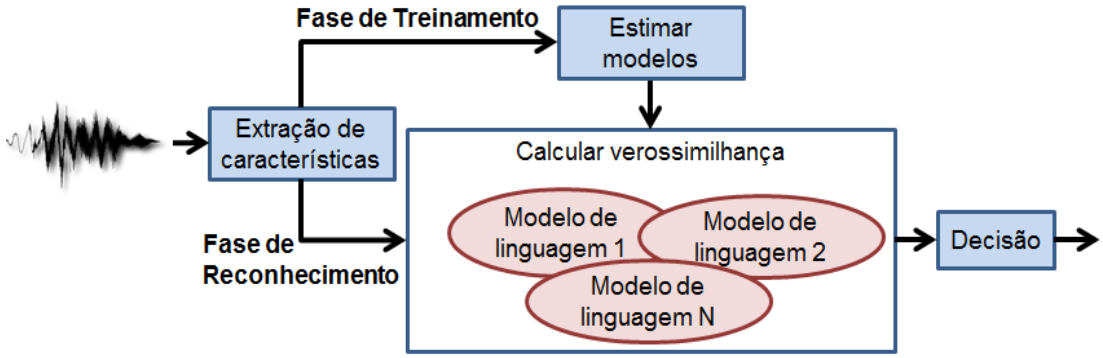

Figura 3. Representação das fases de um sistema de reconhecimento de língua.

O reconhecimento tem por objetivo tomar uma decisão a respeito da amostra informada. Na identificação da língua, isto acontece comparando as características extraídas da amostra com os modelos das línguas. Na detecção da língua, isto acontece comparando as características extraídas a um único modelo ao qual a amostra diz pertencer.

2.4 MEDIDAS DE DESEMPENHO

A medida de desempenho é a métrica utilizada para quantificar a eficiência de uma operação (BOURNE, 2003). A medida de desempenho não pode ser feita isoladamente, mas deve ser relevante dentro do referencial onde a eficiência da ação pode ser julgada. Um passo relevante na medida de desempenho é saber o que medir e saber como medir, tornando assim parte integrante do desenvolvimento de um projeto. As medidas de desempenho podem ser agrupadas conforme a necessidade da avaliação a ser realizada (FERRI, HERNÁNDEZ-ORALLO e MODROIU, 2009). Quando a medida é utilizada para minimizar o número de erros, sendo comum em muitas aplicações diretas de classificadores, são utilizadas as medidas

de desempenho de identificação, como por exemplo, a acurácia e a precisão. Quando a medida é utilizada para avaliar a confiabilidade são utilizadas as medidas de desempenho de detecção, como por exemplo, Taxa de Erro Igual (em Inglês *equal error rate*, EER) e a Função de Custo de Detecção (em Inglês *Detection Cost Function*, DCF). Neste caso o objetivo não é só medir quando ocorre a falha, mas relacionar a uma classe de erro em uma probabilidade alta ou baixa.

Uma forma comum para representar os resultados possíveis para um sistema é a matriz de confusão (SOKOLOVA e LAPALME, 2009). Uma representação para a matriz de confusão está na Tabela 1, onde a matriz mostra os resultados possíveis comparando os resultados reais e os preditos pelo sistema. Porém, pode ocorrer uma divergência entre o resultado predito e o resultado real. Estas divergências são conhecidas como erros de falso alarme e erro de não-detecção (ou detecção perdida) (RONG, 2012). Na identificação de língua, o falso alarme (em Inglês *false alarm*), equivalente ao falso positivo, ocorre quando uma amostra que pertence a uma língua X é reconhecida como pertencente à outra língua. A não-detecção (em Inglês *miss detection*), equivalente ao falso negativo, ocorre quando uma amostra pertence a uma língua X, porém não é reconhecida como tal. Estas informações são utilizadas nas medidas de desempenho de identificação e de detecção de língua

Tabela 1. Matriz de Confusão.

		Resultado predito	
		Positivo	**Negativo**
Resultado Real	**Positivo**	Verdadeiro Positivo (VP)	Identificação: Falso Negativo (FN) Detecção: Não-detecção
	Negativo	Identificação: Falso Positivo (FP) Detecção: Falso Alarme	Verdadeiro Negativo (VN)

2.4.1 Identificação

Com base na matriz de confusão da Tabela 1, o desempenho de sistemas que utilizam a identificação da língua pode ser avaliado através das seguintes medidas:

- **Acurácia:** é o grau de previsões corretas de um modelo (FERRI, HERNÁNDEZ-ORALLO e MODROIU), medida através de:

$$Acurácia = \frac{VP + VN}{VP + FN + FP + VN}$$

- **Precisão:** é o grau de previsões corretas em respostas informadas como verdadeiras (SOKOLOVA e LAPALME, 2009), medidas através de:

$$Precisão = \frac{VP}{VP + FP}$$

- **Recall ou Sensitividade:** é o grau de previsões corretas em que a resposta esperada fosse verdadeira (SOKOLOVA e LAPALME, 2009), medidas através de:

$$Recall = \frac{VP}{VP + FN}$$

- **Especificidade:** é o grau de previsões corretas em respostas informadas como falsas (SOKOLOVA e LAPALME, 2009), medidas através de:

$$Especificidade = \frac{VN}{VN + FN}$$

2.4.2 Detecção

A teoria de detecção de sinal é utilizada para analisar dados provenientes de experimentos onde a tarefa é categorizar resultados ambíguos (ABDI, 2007). O objetivo da teoria de detecção de sinal é estimar os valores correspondentes aos parâmetros de resultado real e de resultado obtido a partir dos dados de experimentos. Para tomar a decisão o sistema compara o resultado obtido ao limiar definido (REYNOLDS e CAMPBELL, 2008). Para retornar verdadeiro o resultado deve ser superior ou igual ao limiar, para retornar falso o resultado deve ser inferior ao limiar.

Os erros de falso alarme e de não-detecção são um dos meios pelo qual um sistema de reconhecimento da língua pode ser medido. Entre as medidas de desempenho que utilizam esses erros temos os seguintes métodos:

- **Taxa de Erro Igual (em Inglês *Equal Error Rate*, EER):** no EER a probabilidade de ocorrer um falso alarme é equivalente à probabilidade de ocorrer uma não-detecção (RONG, 2012). Este tipo de medida de desempenho é

utilizado para que os erros sejam medidos com a mesma equivalência.

- **Função de Custo de Detecção (em Inglês *Detection Cost Function*, DCF):** o DCF é a medida de desempenho adotada pelo Instituto Nacional de Padrões e Tecnologia (*National Institute of Standart and Technology*, NIST), que é o órgão do governo dos Estados Unidos análogo ao Inmetro, para avaliar sistemas de reconhecimento automático da língua, de locutor e de fala (NIST 2003). O DCF representa a probabilidade de ocorrer os erros de reconhecimento (a não-deteccção, P_{FN}, e o falso alarme, P_{FP}), de acordo com a representação de modelo de custo:

  $$C(L_T, L_N) = C_{FN} * P_{Target} * P_{FN}(L_T) + C_{FP} * (1 - P_{Target}) * P_{FP}(L_T, L_N),$$

  onde L_T é a língua alvo e L_N é a língua impostora, P_{Target} é a probabilidade *a-priori* de uma língua alvo, C_{FN} e C_{FP} são os custo da detecção de erros. Para as competições do NIST de reconhecimento da língua de 2003 foram definidos os valores de alguns parâmetros: $C_{FN} = 1$, $C_{FP = 1}$ e $P_{Target} = 0.5$. $C(L_T, L_N)$ representa o custo esperado para tomar uma decisão de detecção. Quanto menor o valor para representar este limite, melhor é a avaliação de desempenho de reconhecimento da língua do sistema.

O sistema de reconhecimento desenvolvido utilizou o EER para medir o desempenho do sistema. A acurácia foi utilizada apenas para detalhar os resultados obtidos pelo sistema na matriz de confusão para cada um dos tokenizadores.

2.5 BASES DE ÁUDIO

Diversas instituições, como o *Oregon Graduate Institute,* o *LDC (Linguist Data Consortium)* e o NIST coletam bases de fala para desenvolver e comparar os sistemas de processamento de fala. Cada uma dessas bases possuem diferentes características como: o número de línguas disponíveis, o tempo de gravação de áudio disponível, a quantidade de locutores, entre outras. Entre as bases de dados utilizadas para sistemas de reconhecimento de língua tem-se:

- ***OGI 22 Language Corpus*:** contém gravações telefônicas em 21 línguas: Árabe, Cantonês, Tcheco, Farsi (Irã), Alemão, Hindi (Índia), Húngaro, Japonês, Coreano, Malaio, Mandarim, Italiano, Polonês, Português, Russo, Espanhol, Sueco, Suaíli

(Quênia, Tanzânia e Uganda), Tâmil (Índia, Cingapura, Indonésia), Vietnamita e Inglês. Esta base possui pelo menos 300 locutores por língua, num total aproximado de 50000 arquivos, contendo mais de 98 horas de gravações telefônicas (LANDER et al, 1995).

- ***OGI Multi-Language Telephone Speech Corpus* (OGI-TS):** contém gravações telefônicas em 11 línguas: Farsi, Francês, Alemão, Hindi, Japonês, Coreano, Mandarim, Espanhol, Tâmil, Vietnamita e Inglês. A OGI-TS possui 2052 locutores, num total de 12152 arquivos, contendo 38,5 horas de gravações telefônicas (MUTHUSAMY, 1993).
- ***LDC CALLFRIEND Telephone Speech Corpus:*** é composta de um agrupamento de 15 coleções de gravações de diferentes línguas: Inglês Americano, Inglês Americano (Sul dos Estados Unidos), Francês, Árabe, Farsi, Alemão, Hindi, Japonês, Coreano, Mandarim Continental, Mandarim de Taiwan, Espanhol, Espanhol Caribenho, Tâmil e Vietnamita. Cada uma dessas bases contêm 60 conversas telefônicas em que o tempo de conversação varia entre 5 e 30 minutos (CANAVAN e ZIPPERLEN, 1996).
- ***LDC CALLHOME Corpus:*** é composta de um agrupamento de 6 coleções de gravações de diferentes línguas: Inglês Americano, Árabe, Mandarim, Alemão, Japonês e Espanhol. Cada uma dessas bases contêm 120 conversas telefônicas com duração de até 30 minutos para cada ligação e as conversas possuem transcrições (CANAVAN, GRAFF e ZIPPERLEN, 1997).
- ***Fisher Corpus*:** é composta de gravações telefônicas em Inglês, Mandarim e Árabe. Esta base de dados contêm 16454 conversas telefônicas com a média de duração de 10 minutos, num total de 2742 horas de gravações telefônicas (CIERI, MILLER e WALKER, 2004).
- ***Mixer Corpus:*** é composta de gravações telefônicas em Inglês, Árabe, Mandarim, Russo e Espanhol. Esta base de dados contêm 1402 locutores, num total de 7627 conversas telefônicas, onde 775 conversas são em Árabe, 5082 em Inglês, 502 em Mandarim, 523 em Russo e 744 em Espanhol (CIERI et al, 2004).

2.6 PARADIGMAS DE AVALIAÇÃO

O paradigma de avaliação tem por objetivo estabelecer um conjunto de tarefas para que se seja possível avaliar e comparar o desempenho de tais sistemas. Um dos paradigmas mais utilizados para o reconhecimento de línguas é o do *NIST Language Recognition Evaluation* (NIST LRE). Este paradigma tem por objetivo explorar novas ideias para o reconhecimento de língua, desenvolver tecnologia que incorpore estas ideias e medir o desempenho desta tecnologia (NIST 2009, 2009). As avaliações disponibilizadas são:

- ***1996 NIST Language Recognition*:** contém gravações telefônicas da base LDC CALLFRIEND Corpus nas 15 línguas. Essa avaliação possui 3600 arquivos de dados em 15 horas de gravações para treinamento e 4500 arquivos de dados em 18 horas de gravações para testes agrupados de acordo com a duração aproximada (3, 10 e 30 segundos) (NIST 1996, 1996).
- ***2003 NIST Language Recognition*:** contém gravações telefônicas da base LDC CALLFRIEND Corpus em 12 línguas: Inglês Americano, Árabe, Farsi, Francês, Hindi, Japonês, Coreano, Mandarim, Alemão, Espanhol, Vietnamita e Tâmil. Essa avaliação possui 7990 arquivos de dados em 30 horas de gravações para treinamento (arquivos vindos da base NIST 1996 provenientes das áreas de treinamento e de teste) e 3840 arquivos de dados em 17 horas de gravações para testes agrupados de acordo com a duração aproximada (3, 10 e 30 segundos) (NIST 2003, 2003).
- ***2005 NIST Language Recognition*:** contém gravações telefônicas da base LDC CALLFRIEND Corpus em 9 línguas: Inglês Americano, Inglês Indiano, Hindi, Japonês, Coreano, Mandarim Continental, Mandarim Taiwan, Espanhol e Tâmil. Essa avaliação possui arquivos de áudio agrupados de acordo com a duração aproximada (3, 10 e 30 segundos) não excedendo o total de 12000 arquivos (NIST 2005, 2005).
- ***2007 NIST Language Recognition*:** contém gravações telefônicas de bases do LDC CALLFRIEND, do LDC CALLHOME, coleções Mixer e Fisher num total de 26 línguas: Árabe, Bengali (Índia), Cantonês (China), Chinês, Min (China), Wu (China), Inglês, Inglês Americano, Inglês Indiano, Farsi, Alemão, Hindustani (Índia), Hindi, Coreano, Japonês, Mandarim, Mandarim Continental, Mandarim

Taiwan, Russo, Espanhol, Espanhol Caribenho, Espanhol não Caribenho, Tâmil, Tailandês, Urdu (Índia) e Vietnamita. Essa avaliação possui arquivos de áudio agrupados de acordo com a duração aproximada (3, 10 e 30 segundos) não excedendo o total de 12000 arquivos (NIST 2007, 2007).

- ***2009 NIST Language Recognition*:** contém gravações telefônicas de bases do LDC CALLFRIEND, do LDC CALLHOME, coleções Mixer e Fisher num total de 23 línguas: Amárico (Etiópia), Bósnio, Cantonês, Criolo (Haiti), Croata, Dari (Afeganistão), Inglês Americano, Inglês Indiano, Farsi, Francês, Georgiano, Hausa (Nigéria), Hindi, Coreano, Mandarim, Pachto (Afeganistão), Português, Russo, Espanhol, Turco, Ucraniano, Urdu e Vietnamita. Essa avaliação possui arquivos de áudio agrupados de acordo com a duração aproximada (3, 10 e 30 segundos) não excedendo a quantidade de 50000 arquivos (NIST 2009, 2009).

2.7 DESEMPENHO DOS MÉTODOS ACÚSTICOS

As pesquisas em busca de melhorias na área de reconhecimento automático da língua tem proporcionado a utilização de diferentes recursos, como diferentes métodos de classificação ou extração de características. Com a combinação de diferentes técnicas aplicadas no reconhecimento da língua e o uso de diferentes configurações de parametrização, pode-se observar os vários resultados obtidos em todo este tempo de pesquisa. Alguns resultados obtidos por vários pesquisadores no reconhecimento automático de língua são apresentados na Tabela 2. Campbell et al. (2006) utilizaram a base de dados do paradigma de avaliação do NIST de 2003 para o treinamento dos modelos e para o reconhecimento. Para criar cada modelo de língua foram utilizadas 20 conversas com aproximadamente 30 minutos de duração. O desempenho apresentado é para a tarefa de 30 segundos (as amostras de áudio utilizadas para o reconhecimento têm 30 segundos de duração). Noor e Aronowits (2006) utilizaram a base de dados do NIST 1996 para o treinamento e do NIST 2003 para o reconhecimento. Para avaliar o desempenho foram utilizados 1280 amostras de áudios, sendo no mínimo 80 amostras por língua. Harwart e Hasegawa-Johnson (2010) utilizaram a base de LDC Callfriend para o treinamento e para o reconhecimento da língua, sendo que o silêncio contido nas conversas telefônicas foi removida. Para criar os modelos das línguas foram utilizados 2,5 minutos iniciais de 2/3 das conversas telefônicas, enquanto do 1/3 restante

foram utilizados 30 segundos iniciais para o reconhecimento da língua. Torres-Carrasquillo et al. (2002) utilizaram a base de LDC Callfriend para o treinamento e para o reconhecimento, utilizando para cada uma das fases respectivamente uma partição de desenvolvimento com 20 conversas telefônicas por língua e uma partição de avaliação contendo 1492 amostras de áudio.

Tabela 2. Desempenho de Sistemas de Reconhecimento de Língua

Sistema	**Base**	**Desempenho**
MFCC-SDC/GMM (CAMPBELL et al, 2006).	NIST LRE 2003	6.1% EER
MFCC-SDC/SVM (CAMPBELL et al, 2006).	NIST LRE 2003	4.8% EER
SDC/GMM (NOOR e ARONOWITS, 2006)	NIST LRE 1996 + NIST LRE 2003	7.4% EER
MFCC/GMM (HARWART e HASEGAWA-JOHNSON, 2010).	CALLFRIEND	24.9% EER
SDC/GMM (TORRES-CARRASQUILLO et al, 2002).	CALLFRIEND	8.78% EER

3. RECONHECIMENTO AUTOMÁTICO DA LÍNGUA BASEADA EM TOKENS

Este capítulo faz uma revisão sobre o uso de tokens no reconhecimento automático de língua. A Seção 3.1 descreve a tokenização e os métodos para a geração de tokens. A Seção 3.2 apresenta alguns métodos de classificação baseados em tokens. A Seção 3.3 apresenta o desempenho de outros sistemas que utilizam a tokenização como fonte de dados. A Seção 3.4 explica sobre a fusão de sistemas e os desempenhos obtidos por outros sistemas.

O reconhecimento automático da língua baseada em tokens tem por objetivo utilizar uma representação discreta do sinal de fala (token) para representar as informações desejadas. Ao contrário da representação das informações espectrais, onde a janela de onde os dados são extraídos é fixa, os tokens não representam um tamanho fixo no sinal de fala. Os tokens podem ser utilizados para representar informações acústicas, fonéticas, fonotáticas e prosódicas da fala. Por exemplo, o uso de tokens na representação de fonemas pode representar as dependências fonotáticas de uma língua (NAVRATIL, 2006). As dependências fonotáticas são as leis que regem a distribuição dos fonemas dentro de uma mesma língua e que permitem distinguir as línguas. Os tokens também podem ser utilizados para representar os aspectos temporais da prosódia. Por exemplo, Rouas (2005) utiliza três rótulos (U - subida, D - descida, # - silêncio) para representar os dados extraídos de frequência fundamental e os dados da energia. Também são utilizados os tokens para representar as informações acústicas, como mostra a pesquisa de Torres-Carrasquillo, Reynolds e Deller (2002).

3.1 TOKENIZAÇÃO

A tokenização é o processo de converter o sinal de fala em uma sequência de símbolos que representam algum tipo de informação da fala. O uso da tokenização é representado pela Figura 4.

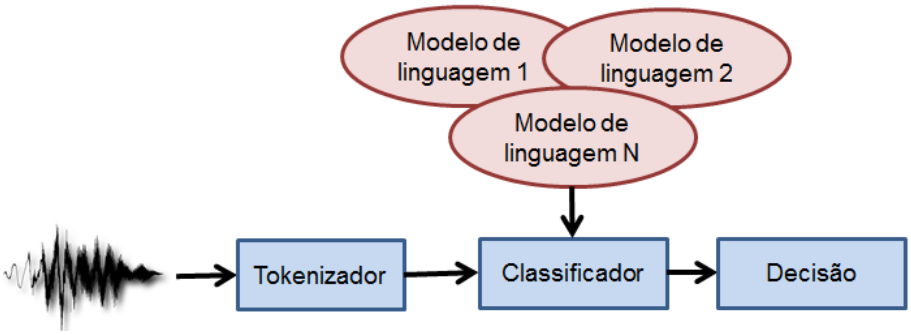

Figura 4. Sistema de reconhecimento com Tokenizador.

Entre as técnicas utilizadas para a tokenização, podem ser utilizados os seguintes métodos:

- **Reconhecimento de fonema seguido pela modelagem da língua (em Inglês *Phoneme Recognition followed by Language Modeling*, PRLM):** utiliza o reconhecimento de fonemas de uma língua independente seguido pelo módulo de modelagem dependente da língua (RONG, 2012). Em outras palavras, o sistema que utiliza o PRLM possui apenas um tokenizador de fonemas e vários modelos das diferentes línguas. Também é possível utilizar em paralelo vários sistemas PRLM. Esta arquitetura é chamada de Reconhecimento Paralelo de Fonema Seguido pela Modelagem da Língua (em Inglês *Parallel Phoneme Recognition followed by Language Modeling*, PPRLM), onde cada sistema PRLM treina uma língua diferente (AMBIKAIRAJAH et al, 2011). Porém, para o processo de reconhecimento da própria língua não é necessário que ela possua um tokenizador (SHEN et al, 2008). Os resultados gerados então são combinados de forma linear ou aplicando classificadores não-lineares. Segundo WONG (2004), o ideal é que existisse um tokenizador para cada língua, porém a falta de transcrições fonéticas para a maioria das línguas é um grande empecilho para alcançar este objetivo. Alguns estudos foram realizados utilizando o PPRLM, como por exemplo, as pesquisas de Gauvain, Messaoudi e Schwenk (2004) e Siniscalchi et al. (2012).

- **Tokenizador Fonético Orientado ao Alvo (em Inglês *Target-Oriented Phone Tokenizer*, TOPT):** consiste em otimizar o conjunto fonético reconfigurando o tokenizador para um alvo fonético específico, isto é, um subconjunto fonético pode discriminar melhor a língua alvo (RONG, 2012). Por exemplo, dado um reconhecedor fonético em Inglês, pode-se criar um tokenizador fonético Árabe

orientado ao Inglês ou um tokenizador Mandarim orientado ao Inglês, onde as estatísticas fonotáticas dessas línguas possam ser diferentes que o do Inglês. O TOPT utiliza apenas um subconjunto de fonemas, o que limita a cobertura fonética, e conforme o número de fonemas discriminativos selecionados, o desempenho pode ser afetado.

- **Modelos de Língua Consciente do Alvo (em Inglês *Target-Aware Language Models*, TALM):** busca derivar múltiplos tokenizadores fonéticos a partir de um reconhecedor fonético existente (RONG, 2012). Cada tokenizador compartilha um conjunto de modelos acústicos, porém cada tokenizador está relacionado a um único modelo de língua. Estes fonemas são obtidos da sequência fonética derivada do reconhecedor fonético original.

- **Modelagem das dinâmicas prosódicas:** é utilizada para capturar os aspectos temporais da prosódia (ADAMI et al, 2003). A modelagem dinâmica pode ser utilizada para representar se o estado do contorno de uma determinada característica prosódica (por exemplo: frequência fundamental, energia) está em queda, em subida ou sem voz para a região definida. Por exemplo, se representarmos a frequência fundamental e a energia com os tokens + (subida), - (queda) e uv (sem voz), podemos ter as seguintes combinações: ++, +-, -+, --, uv. Alguns estudos foram realizados utilizando a modelagem das dinâmicas prosódicas, como por exemplo, as pesquisas de Adami et al. (2003) e Rouas (2005).

- **Tokenizador GMM (em Inglês GMM Tokenizer):** o GMM é utilizado para construir um dicionário acústico da língua de treinamento (TORRES-CARRASQUILLO, REYNOLDS e DELLER, 2002). O Tokenizador GMM funciona de modo similar ao PPRLM. Porém, comparado ao PPRLM, o Tokenizador GMM não necessita de uma transcrição fonética ou ortográfica da fala, porém os resultados obtidos tendem a ser piores (TORRES-CARRASQUILLO et al, 2002). Alguns estudos foram realizados utilizando o Tokenizador GMM, como por exemplo, a pesquisa de Torres-Carrasquillo, Reynolds e Deller (2002).

3.2 CLASSIFICAÇÃO

Uma restrição dos sistemas baseados em tokens é que nem todos os classificadores podem ser utilizados devido à característica de que os tokens são dados categóricos. A maioria dos algoritmos de reconhecimento de padrões utiliza uma medida de similaridade (por exemplo, produto interno) ou dissimilaridade (por exemplo, distância) entre os padrões, já que são definidos para variáveis contínuas. Assim, isto acaba restringindo a aplicação de diversos métodos de classificação em dados categóricos. Apesar das restrições impostas pelos dados categóricos, alguns classificadores baseados em dados não numéricos podem ser utilizados:

- **N-grama:** é utilizado para estimar as probabilidades de todas as N sequências de tokens possíveis de serem encontradas. Quando aplicado na representação de fonemas, ele pode ser utilizado para capturar as informações fonotáticas para cada língua (SHEN et al, 2008). Aplicações com n-gramas na ordem de dois ou três (bigramas e trigramas) são os mais utilizados. Aplicações que utilizem n-gramas de ordens maiores são mais difíceis de serem encontrados, pois aumenta exponencialmente a quantidade de combinações fonéticas possíveis. Alguns estudos foram realizados utilizando n-gramas, como por exemplo, as pesquisas de Siniscalchi et al. (2012) e Wong (2004).

- **Árvores de Decisão Binária (em Inglês *Binary Decision Trees*):** sua representação consiste de nós terminais e não terminais (NAVRATIL, 2006). Cada nó não terminal representa uma pergunta que leva a escolha de um dos dois nós filhos. No final do processo de decisão da árvore, nos nós terminais, o método retorna a probabilidade de um determinado token estar associado a um determinado conjunto de tokens. Alguns estudos foram realizados utilizando as árvores de decisão binárias, como por exemplo, a pesquisa do próprio Navratil (2006).

- **Cadeias Ocultas de Markov Discreto (em Inglês *Discrete Hidden Markov Models*, DHMM):** um dos objetivos da DHMM no reconhecimento da língua é capturar os padrões de erro do tokenizador contra os fonemas de diferentes línguas (WONG e SIU, 2004). Para capturar as distribuições de erro, Wong e Siu (2004) utilizaram a DHMM de um estado para representar diferentes fonemas no

grupo de fonemas de cada língua. Alguns estudos foram realizados utilizando as DHMM, como por exemplo, a pesquisa dos próprios Wong e Siu (2004).

Uma outra solução para os dados categóricos é a sua transformação em dados numéricos através da utilização de vetores de propriedades (Duda et al, 2012). Um vetor com dimensionalidade *p* representa um conjunto de *p* propriedades, o qual pode ser utilizado como um dado contínuo. CAMPBELL et al (2004) utilizou as probabilidades estatísticas das sequências obtidas de tokens com N-Gramas para criar um vetor de propriedades. Estes vetores então foram processados através de Máquinas de Vetor de Suporte.

3.3 DESEMPENHO DA TOKENIZAÇÃO

A tokenização é mais um recurso importante que os pesquisadores têm utilizado para melhorar os sistemas de reconhecimento de língua. Alguns resultados obtidos explorando o recurso da tokenização por tais pesquisas no reconhecimento automático de língua são apresentados na Tabela 3. Adami (2004) em sua pesquisa utilizou para o treinamento e ao reconhecimento da língua o paradigma de avaliação do NIST 2003. O desempenho apresentado é referente à detecção de língua da tarefa de 30 segundos. Ng et al. (2010) utilizou para o treinamento e ao reconhecimento da língua o paradigma de avaliação do NIST 2009. Os desempenhos apresentados são referentes à detecção da língua de 10635 arquivos da tarefa de 30 segundos. Torres-Carrasquillo et al. (2002) utilizaram a base de LDC Callfriend para o treinamento e para o reconhecimento. Ao total foram utilizados 12 tokenizadores, cada um deles seguidos de 12 modelos da língua. Para o treinamento foi utilizado 20 conversas telefônicas por língua e para o reconhecimento foi utilizada 1492 amostras de áudio. O método de extração de características acústicas utilizado foi o SDC. Torres-Carrasquillo, Reynolds e Deller (2012) fizeram um teste similar ao anterior, porém ao invés de utilizar o SDC para a extração de características, utilizaram o MFCC. Siniscalchi et al. (2012) utilizaram para o treinamento a base OGI-TS e para o reconhecimento o paradigma NIST 2003. Foram utilizados 6 tokenizadores: Inglês, Alemão, Hindi, Japonês, Mandarim e Espanhol. Para criar os modelos de cada língua, foram utilizados os arquivos de áudio denominados de ‘histórias’ da base OGI-TS. Para a detecção foi utilizada a tarefa de amostras com 30 segundos de duração do NIST 2003. Gauvain, Messaoudi e Schwenk (2004) utilizaram as bases Callhome para o treinamento e do NIST 2003 para o reconhecimento.

Foram utilizados 3 tokenizadores: Inglês americano, Espanhol e Árabe. Para o treinamento foi utilizado 20 conversas telefônicas de 30 minutos para cada língua. Para a detecção foi utilizada 1280 amostras da tarefa de 30 segundos de duração do NIST 2003.

Tabela 3: Desempenho de Sistemas de Reconhecimento de Língua que utilizam tokens.

Sistema	**Base**	**Desempenho**
PROSÓDIA (Frequência Fundamental e intensidade) / N-GRAMA (ADAMI, 2004).	NIST LRE 2003	22.6% EER
PROSÓDIA (Duração) (NG et al, 2010).	NIST LRE 2009	23.24% EER
PROSÓDIA (Frequência Fundamental) (NG et al, 2010).	NIST LRE 2009	23.08% EER
PROSÓDIA (Intensidade) (NG et al, 2010).	NIST LRE 2009	20.58% EER
TOKENIZADOR GMM + SDC (TORRES-CARRASQUILLO et al, 2002).	CALLFRIEND	8.78% EER
TOKENIZADOR GMM + MFCC (TORRES-CARRASQUILLO, REYNOLDS e DELLER 2002)	CALLFRIEND	36.3% EER
PPRLM (TORRES-CARRASQUILLO et al, 2002).	CALLFRIEND	7.84% EER
PPRLM (SINISCALCHI et al, 2012).	OGI-TS + NIST LRE 2003	6.9% EER
PPRLM/N-GRAMA (GAUVAIN, MESSAOUDI e SCHWENK, 2004).	CALLHOME + NIST LRE 2003	6.8% EER
REDES NEURAIS (*multi-layer perceptron*) (GAUVAIN, MESSAOUDI e SCHWENK, 2004).	CALLHOME + NIST LRE 2003	4.0% EER

3.4 FUSÃO DE SISTEMAS

As técnicas de fusão são utilizadas para combinar os escores de múltiplos sistemas em paralelo para aumentar o desempenho de um sistema (SHEN et al, 2008). Isto ocorre partindo do princípio que os diferentes métodos de classificação contemplam diferentes características da fala, como representado na Figura 5. Com isso é possível transformar as diferentes características exploradas em informações complementares (WONG, 2004).

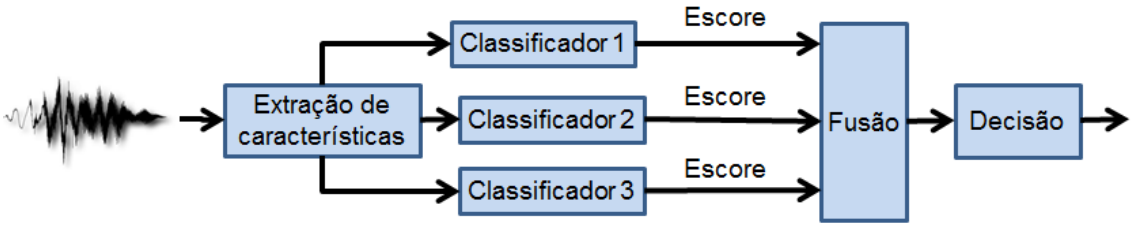

Figura 5. Representação do uso de fusão.

Existem várias técnicas de fusão que são utilizadas para a tarefa de combinar os resultados de diferentes classificadores. Para Shen et al. (2008, p. 819) estas técnicas podem ser agrupadas em dois tipos:

- **Fusão de Produto-Regra (em Inglês *Product-Rule Fusion*):** são baseadas em fórmulas fixas para calcular os resultados, onde os pesos informados para cada conjunto de dados são uniformes ou escolhidos empiricamente. Uma desvantagem destas técnicas é que resultados próximos a zero podem ter impacto indesejado no resultado final. Alguns desses métodos são: Adição Simples (em Inglês *Simple Addition*), Multiplicação Simples (em Inglês *Simple Multiplication*), Regra Máxima (em Inglês *Maximum Rule*) ou Peso de Pontuação Linear (em Inglês *Linear Score Weighting*).
- **Fusão de Classificadores (em Inglês *Classifiers Fusion*):** utiliza um classificador para treinar dados provenientes dos resultados individuais do sistema e gerar novas estatísticas. As estatísticas do conjunto de resultados são exploradas para produzir um resultado superior aos obtidos com a fusão produto-regra. Classificadores como GMM ou redes neurais podem utilizados para esta função.

3.4.1 Desempenho aplicando a fusão

A técnica de fusão de sistemas tem auxiliado muito para que o desempenho dos sistemas de reconhecimento de língua seja melhor. Alguns resultados obtidos em pesquisas no reconhecimento automático de língua aplicando a técnica de fusão de sistemas são apresentados na Tabela 4. Torres-Carrasquillo et al. (2002) utilizaram a base de LDC Callfriend para o treinamento e para o reconhecimento. A fusão dos escores dos tokenizadores GMM e do PPRLM foi realizada através da normalização da Análise Discriminante Linear (em Inglês *Linear Discriminant Analysis*, LDA) seguido do GMM. A LDA é utilizada para

maximizar a separação entre as classes e minimizar a variação dentro da classe (SHEN et al, 2008). Adami (2004) utilizou a base de do NIST 2003 para o treinamento e para o reconhecimento. A fusão dos escores da prosódia e do PPRLM foi realizada através da média dos escores dos respectivos segmentos. Campbell et al (2006) utilizaram a base de do NIST 2003 para o treinamento e para o reconhecimento. Para a fusão dos escores dos classificadores GMM e SVM foi realizada através da média dos escores. Nesta técnica de fusão todos os sistemas possuem o mesmo peso, logo é necessário que os escores sejam normalizados, removendo assim qualquer resultado distorcido. Gauvain, Messaoudi e Schwenk (2004) utilizaram uma rede neural artificial para realizar a fusão na base de dados do NIST 2003.

Tabela 4. Desempenho de Sistemas de Reconhecimento de Língua que aplicam a fusão.

Sistema	**Base**	**Desempenho**	**Fusão**
TOKENIZADOR GMM + PPRLM (TORRES-CARRASQUILLO et al, 2002).	CALLFRIEND	6.9 % EER	Fusão de classificadores: GMM.
PROSÓDIA (Frequência Fundamental e intensidade) + PPRLM (ADAMI, 2004).	NIST LRE 2003	6.6% EER	Fusão de Produto: Média
MFCC-SDC/GMM + MFCC-SDC/SVM (CAMPBELL et al, 2006).	NIST LRE 2003	3.2% EER	Fusão de Produto: Média
PPRLM/N-GRAMA + REDES NEURAIS (*multi-layer perceptron*) (GAUVAIN, MESSAOUDI e SCHWENK, 2004).	NIST LRE 2003	2.7% EER	Fusão de classificadores: Redes Neurais

4. SISTEMA DE RECONHECIMENTO DE LÍNGUA BASEADO EM TOKENS

O objetivo deste capítulo é descrever o sistema desenvolvido e as técnicas aplicadas para o sistema. A Seção 4.1 descreve a arquitetura utilizada para o desenvolvimento do sistema. A Seção 4.2 descreve as bases de áudio utilizadas para treinamento e teste. As Seções 4.3 e 4.4 explicam as técnicas utilizadas para a extração de características acústicas e prosódicas. A Seção 4.5 fala sobre o classificador utilizado no sistema. A Seção 4.6 apresenta os resultados obtidos pelo sistema de reconhecimento desenvolvido. A Seção 4.7 descreve como a fusão dos sistemas foi realizada e os resultados obtidos.

4.1 ARQUITETURA DO SISTEMA

O sistema desenvolvido para o reconhecimento da língua utiliza tokens para realizar esta tarefa. A arquitetura do sistema é composta por três tokenizadores, um utilizando as informações prosódicas e os outros dois utilizando as informações acústicas, conforme apresentado na Figura 6.

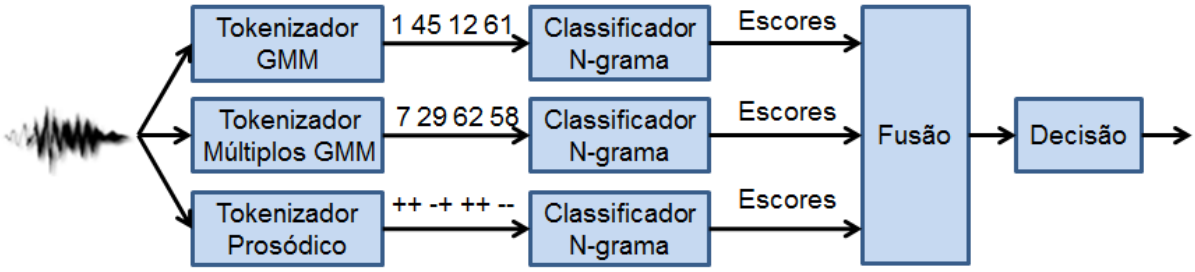

Figura 6. Arquitetura do sistema de reconhecimento.

Para os tokenizadores acústicos, as informações extraídas foram obtidas pelo uso do método de MFCC. O MFCC é extraído por um programa desenvolvido em pesquisa anteriores do Laboratório de Processamento de Voz da UCS. O processo de tokenização aplicado será o Tokenizador GMM, seguindo a ideia proposta por Torres-Carrasquillo, Reynolds e Deller (2002). O processo de reconhecimento da língua pode-se utilizar de duas maneiras dos tokenizadores acústicos: tokenizador acústico simples e tokenizador acústico múltiplo. Os sistemas com tokenizador acústico simples empregam um único tokenizador contendo informações acústicas de todas as línguas. Os sistemas com tokenizador acústico

múltiplo empregam um tokenizador específico para cada língua. A escolha do tokenizador de informações acústicas deve-se a possibilidade de utilizar informações espectrais, amplamente utilizadas para o reconhecimento de língua. O tokenizador acústico será descrito na Seção 4.3.

Para o tokenizador prosódico, as informações extraídas foram às variações na frequência fundamental e na intensidade, seguindo o modelo definido na pesquisa de Adami (2004). A escolha pelo tokenizador prosódico justifica-se pelo fato de serem extraídas outras características de fala, permitindo assim o uso da técnica de fusão para explorar as diferentes características de cada sistema. Outra motivação é a possibilidade de utilizar uma ferramenta já aplicada para a tokenização em outro projeto. O tokenizador prosódico será descrito na Seção 4.4.

Para estimar a verossimilhança dos tokens de informações acústicas e prosódicas, o classificador utilizado é o N-grama. O classificador N-grama foi escolhido levando em consideração que é um dos classificadores mais utilizados para a classificação de dados discretos. O classificador N-grama utilizado neste projeto é o SRILM desenvolvido pelo *SRI Speech Technology and Research Laboratory* (STOLCKE, 2002). O método de suavização de N-gramas selecionado foi a suavização de Katz por ser a técnica de suavização padrão do SRILM. O classificador será descrito na Seção 4.5.

Para a fusão dos resultados obtidos entre os sistemas de tokenização prosódica e de tokenização acústica foram utilizados os métodos de média e regressão logística. O método de média foi escolhida pela sua simplicidade. O método de regressão logística foi escolhido como um método alternativo para obter os resultados e por já existir uma ferramenta que possibilitasse realizar a fusão de escores de múltiplos sistemas. O método de regressão logística encontra-se na ferramenta chamada de Focal Multi-Class, desenvolvido por Brümmer e DataVoice (2007). A técnica de fusão de regressão logística será descrita na Seção 4.7.

4.2 BASES DE DADOS DE TREINAMENTO E TESTE

As bases de dados de áudio utilizadas neste trabalho são do NIST 2003, do OGI 22 *Language Corpus* e do OGI-TS. As bases de dados do OGI 22 *Language* e do OGI-TS foram utilizados para construir os modelos GMM utilizados para a tokenização dos dados acústicos. A base de dados do NIST 2003 é utilizada para avaliar o desempenho do sistema de

reconhecimento de língua desenvolvido. A forma como estão divididos os dados para treinamento e testes estão de acordo com os padrões definidos pelo paradigma do NIST para esta base de dados (NIST 2003, 2003). O NIST 2003 possui o Russo apenas nos dados de teste. O Russo representa a língua desconhecida do sistema. A quantidade de dados e o tempo de total de áudio utilizados para cada língua de cada base para a validação do sistema se encontram na Tabela 5.

Tabela 5. Distribuição de dados utilizados por língua (número de arquivos – tempo de gravação aproximado).

Línguas	Bases de Áudio			
	OGI 22	OGI-TS	NIST 2003 - Treinamento	NIST 2003 - Teste
Alemão	2475 – 4,53 horas	1059 – 2,86 horas	478 – 1,9 horas	240 – 0,99 hora
Árabe	2037 – 4,2 horas	Não há	466 – 1,84 horas	240 – 1 hora
Coreano	2526 – 5,04 horas	904 – 2,19 horas	470 – 1,85 horas	240 – 1 hora
Espanhol	2523 – 4,11 horas	1151 – 3,1 horas	921 – 3,64 horas	240 – 0,99 horas
Farsi	2472 – 5 horas	1003 – 2,55 horas	477 – 1,89 horas	240 – 0,99 horas
Francês	Não há	1082 – 2,91 horas	472 – 1,87 horas	240 – 1 hora
Hindi	2480 – 5,51 horas	198 – 2,57 horas	465 – 1,83 horas	240 – 1 hora
Inglês	2395 – 4,59 horas	2506 – 12,1 horas	1913 – 7,61 horas	720 – 2,98 horas
Japonês	2361 – 4,5 horas	930 – 2,35 horas	474 – 1,88 horas	480 – 1,99 horas
Mandarim	2463 – 5,4 horas	1103 – 2,6 horas	939 – 3,73 horas	240 – 1 hora
Russo	Não há	Não há	Não há	240 – 0,97 horas
Tâmil	2403 – 3,95 horas	1189 – 2,83 horas	448 – 1,75 horas	240 – 1 hora
Vietnamita	2350 – 3,75 horas	1026 – 2,46 horas	466 – 1,83 horas	240 – 1,01 horas

4.3 EXTRAÇÃO DOS TOKENS ACÚSTICOS

Nesta seção são abordadas as técnicas utilizadas para a extração dos tokens acústicos no sistema de reconhecimento. O tokenizador acústico utiliza informações espectrais para transformar em tokens. Essas informações espectrais são obtidas por técnicas de extração de características amplamente utilizadas em sistemas de reconhecimento de línguas. Com isso, a

tokenização acústica pode fornecer novas informações para serem exploradas nos sistemas de reconhecimento de línguas. A Seção 4.3.1 descreve o método de extração de características acústicas utilizado no sistema. A Seção 4.3.2 descreve o método de tokenização utilizado para rotular as informações acústicas.

4.3.1 Extração dos Coeficientes Cepstrais

Para Davis e Mermelstein (1980, p. 365) a concepção da representação do MFCC foi motivada por fatores perceptivos. A habilidade de capturar informações relevantes de percepção é uma vantagem importante. Esta concepção parte do princípio que, uma métrica de distância melhor pode resultar em uma modelagem mais precisa do comportamento perceptivo. A representação do MFCC é definida como um cepstrum real de um sinal de janela de curto prazo derivado do algoritmo da Transformada Rápida de Fourier (em Inglês *Fast Fourier Transform*, FFT) do sinal (HUANG, ACERO e HON, 2001). O cepstrum real utiliza a escala de frequência não linear, que se aproxima do comportamento auditivo do ser humano. A Transformada de Fourier Discreta (em Inglês *Discrete Fourier Transform*, DFT) do sinal de entrada é representado por:

$$X_N[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, 0 \leq k < N$$

onde *k* representa o comprimento do DFT, *N* representa o período da amostra. O banco de filtros para o MFCC é definido com *M* filtros (m = 1, ..., *M*) e representado por:

$$H_m[k] = \begin{cases} 0, & k < f[m-1] \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])}, & f[m-1] \leq k \leq f[m] \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m+1] - f[m])}, & f[m] \leq k \leq f[m+1] \\ 0, & k > f[m+1] \end{cases}$$

onde m representa o filtro triangular e f[m] representa os pontos de limite. Esses filtros calculam o espectro médio em torno de cada frequência com o incremento da largura de banda, representado pela Figura 7.

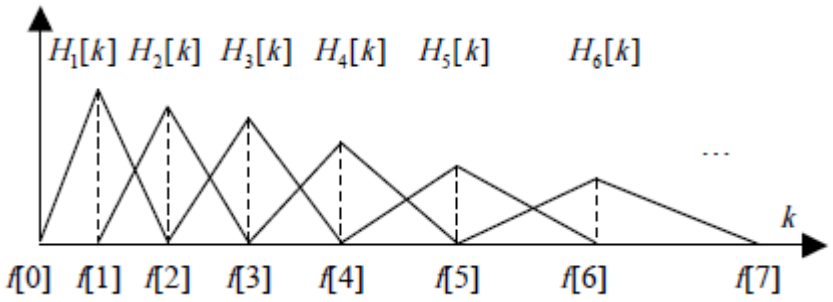

Figura 7. Filtros triangulares utilizados pelo MFCC.

Os pontos de limite (f[m]) estão espaçados uniformemente dentro da escala Mel e são representados pela equação:

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_l) + m\frac{B(f_h) - B(f_l)}{M+1}\right)$$

onde f_l e f_h representa as *m* frequências mais baixas e mais altas, respectivamente, do banco de filtros em Hz, F_s representa a frequência da amostragem em Hz, *N* representa o tamanho do FFT, *B*(*f*) representa a função da escala Mel representada pela equação

$$B(f) = 1125 \ln\left(1 + {}^{f}/_{700}\right)$$

e B^{-1} é a sua inversa, representada por:

$$B^{-1}(f) = 700\left(exp\left({}^{f}/_{1125}\right) - 1\right).$$

Em seguida, o logaritmo da energia é aplicado na saída de cada filtro, representado pela equação:

$$S[m] = \ln\left[\sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k]\right], \qquad 0 \le m < M$$

O cepstrum da frequência mel é a transformada do cosseno discreto (em Inglês *Discrete Cossine Transform*, DCT) das saídas de M filtros:

$$c[n] = \sum_{m=0}^{M-1} \mathrm{S[m]} \cos(\pi n(m + 0.5)/M), \qquad 0 \le n < M$$

4.3.2 Tokenizador GMM

O objetivo do Tokenizador GMM é produzir uma sequência de tokens a partir de um dado vetor de características extraídas (TORRES-CARRASQUILLO et al, 2002; NKADIMENG, 2010). Cada componente gaussiana é representado por um índice. Cada um

dos tokens são referentes ao índice do componente GMM de maior verossimilhança entre o dado de entrada e o modelo GMM utilizado como tokenizador. A representação para definir a maior verossimilhança é através da equação

$$Tok_t = \arg\max_i P(x_t|\mu_i, E_i), \qquad i = 1, 2, \ldots, M$$

onde x representa o vetor de características, $P(x_t|\mu_i, E_i)$ é a função de verossimilhança, *M* é o número de componentes do GMM, $\boldsymbol{\mu_i}$ representa o vetor de média e $\mathbf{E_i}$ representa a matriz de covariância do componente GMM.

A sequência resultante do processo de tokenização, como representado na Figura 8, é utilizada pelo classificador para estimar a similaridade com o modelo da língua. O modelo utilizado como tokenizador é construído pelo GMM para representar o dicionário acústico dos dados de treinamento.

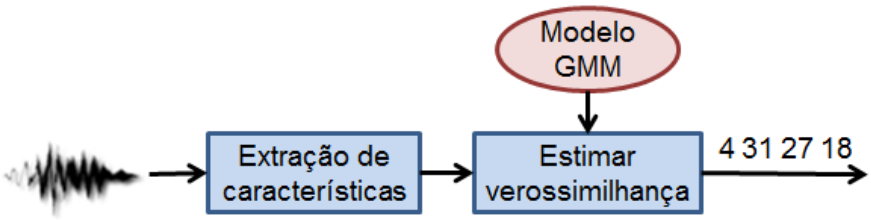

Figura 8. Representação do tokenizador GMM.

Modelo de Mistura Gaussiana

Para o tokenizador GMM, o modelo de mistura gaussiana modela as classes acústicas dos dados de fala por uma combinação linear de distribuições gaussianas (AMBIKAIRAJAH et al, 2011). No caso deste sistema este modelo representa as informações da língua. Uma distribuição gaussiana pode ser representada por sua média e variância, onde uma GMM possui *N* componentes.

O GMM possui características de um modelo paramétrico e não paramétrico (REYNOLDS, QUATIERI E DUNN, 2000). Entre as características de um modelo paramétrico, o GMM possui uma estrutura e parâmetros que controlam o comportamento da densidade de maneira conhecida, porém sem restrições que especifiquem o tipo de distribuição que os dados pertencem. Entre as características de um modelo não paramétrico, o GMM possui graus de liberdade para permitir arbitrariamente a modelagem das densidades, sem cálculos indevidos e sem a necessidade de armazenamento.

Para um vetor de características n-dimensional, a densidade do componente utilizado

para a função de verossimilhança é definida como

$$p(x|\lambda) = \sum_{i=1}^{M} w_i p_i(x).$$

onde x é o vetor de características, p(x|λ) é a função de verossimilhança, w_i são os pesos dos componentes, M é o número de componentes do GMM e $p_i(x)$ é a combinação linear dos pesos dos componentes gaussianas, representada pela equação

$$p_i(x) = \frac{1}{(2\pi)^{N/2}|E_i|^{1/2}} exp\left\{-\frac{1}{2}(x - \mu_i)' {E_i}^{-1}(x - \mu_i)\right\},$$

onde $\boldsymbol{\mu_i}$ é o vetor de média e $\mathbf{E_i}$ é a matriz de covariância do componente, λ é a representação do modelo $\{w_i , \mu_i, E_i\}$. Os pesos dos componentes no GMM possuem a restrição de $\sum_{i=1}^{M} w_i = 1$.

Para estimar os parâmetros de média, covariância e pesos do GMM é executado o algoritmo iterativo de máxima expectativa (em Inglês *Expectation-Maximization*, EM). Em um dado conjunto de vetores de treinamento, o algoritmo EM refina os parâmetros do GMM para incrementar a verossimilhança do modelo estimado para esses vetores, até se obter a convergência definida. Em cada iteração do algoritmo EM, os parâmetros de peso, média e variância são atualizados pelas equações:

$$w_i = \frac{1}{T}\sum_{t=1}^{T} p_i(x|\lambda)$$

$$\mu_i = \frac{\sum_{t=1}^{T} p_i(x|\lambda)x_t}{\sum_{t=1}^{T} p_i(x|\lambda)}$$

$$E_i = \frac{\sum_{t=1}^{T} p_i(x|\lambda)x_t}{\sum_{t=1}^{T} p_i(x|\lambda)} - \mu_i\mu_i'$$

Na pesquisa de Reynolds, Quatieri e Dunn (2000) foi utilizada a diagonal da matriz de covariância, ao invés da matriz de covariância completa para o cálculo do GMM, utilizaram apenas. Algumas das vantagens para o uso da matriz diagonal ao invés da matriz completa encontradas foram:

- A modelagem da densidade de uma covariância completa de ordem M pode ser obtida utilizando uma grande diagonal da matriz de covariância;
- O treinamento do modelo de GMM é mais eficiente computacionalmente quando é utilizada a diagonal da matriz de covariância se comparado ao uso da matriz completa;
- Os resultados obtidos pela aplicação com o uso da diagonal da matriz de

covariância foram melhores que os resultados obtidos com o uso da matriz completa.

Adaptação de Modelo

A ideia básica da adaptação é obter um modelo da língua através da atualização de parâmetros do GMM do Modelo Universal por adaptação (REYNOLDS, QUATIERI E DUNN, 2000). A adaptação entre o modelo de língua e os parâmetros do modelo independente de língua ocorre através do uso da técnica de adaptação Bayesiana. A técnica de adaptação Bayesiana, também chamada de estimação *maximum a posteriori* (MAP), tem por objetivo estimar o modelo adaptado de uma língua a partir de um modelo independente de língua, provendo assim um maior relacionamento entre o modelo da língua e o modelo independente da língua. Assim como o algoritmo EM, a adaptação é um processo estimativo de dois passos. O primeiro passo é idêntico ao passo de expectativa do algoritmo EM, onde as estimativas de peso, média e variância para o modelo da língua são calculadas para cada mistura no modelo independente. No segundo passo, os novos cálculos estatísticos são unidos com os antigos a partir dos parâmetros da mistura do modelo independente combinando os coeficientes. Nesta combinação de coeficientes, as misturas com alta contagem de dados da língua dependem mais dos novos dados estatísticos para a estimativa final. Enquanto as misturas com baixa contagem de dados da língua dependem mais dos antigos dados estatísticos.

Dado um modelo independente de língua e um vetor de treinamento, $X = \{x_1, ..., x_t\}$, primeiro é determinado o alinhamento probabilístico do vetor de treinamento com os componentes da mistura modelo independente, que é definida por

$$\Pr(i|x_t) = \frac{w_i p_i(x_t)}{\sum_{j=1}^{M} w_j p_j(x_t)},$$

onde *i* representa a mistura, $\Pr(i|x_t)$ e x_t são utilizados para calcular as estatísticas dos parâmetros de peso, média e variância.

$$n_i = \sum_{t=1}^{T} \Pr(i|x_t)\, x_t$$

$$E_i(x^2) = \frac{1}{n_i} \sum_{t=1}^{T} \Pr(i|x_t)\, x_t^2$$

As novas estatísticas obtidas dos dados de treinamento são utilizadas para atualizar as estatísticas antigas do modelo independente para a mistura *i* para criar os parâmetros

adaptados desta mistura com as equações

$$\hat{w}_i = \left[\propto_i{}^{w} \; n_i /_T + (1 - \propto_i{}^{w}) w_i \right] \gamma$$

$$\hat{\boldsymbol{\mu}}_i = \propto_i{}^{m} \; E_i(x) + (1 - \propto_i{}^{m}) \mu_i$$

$$\hat{\sigma}_i^2 = \alpha_i{}^{v} E_i(x^2) + (1 - \; \alpha_i{}^{v})(\sigma_i^2 + \mu_i^2) - \; \mu_i^2$$

onde $\alpha_i{}^{w}$, $\alpha_i{}^{m}$ e $\alpha_i{}^{v}$ representam os coeficientes de adaptação utilizados para controlar o balanceamento entre as novas e as antigas estimativas para os pesos, as médias e as variâncias respectivamente. γ representa o fator de escala que é calculado sobre todos os pesos contidos no modelo para assegurar que eles mantenham a unidade. Para cada mistura e cada parâmetro adaptado é utilizado um coeficiente de adaptação, que é definido pela equação

$$\propto_i{}^{p} = \frac{n_i}{n_i + r^p}$$

onde $\alpha_i{}^{p}$ representa os coeficientes de adaptação, $p \in \{w, m, v\}$, e r^p representa o fator de relevância fixo para o parâmetro p.

O uso do coeficiente de adaptação permite adaptar os parâmetros das misturas. Neste contexto, o fator de relevância é o modo para controlar quanto os novos dados para o novo modelo devem ser observados na mistura antes de substituir o novo parâmetro pelo antigo. Segundo Reynolds e Campbell (2008, p. 770) "a aproximação por adaptação produz um desempenho superior comparado a um sistema onde o modelo de língua é treinado independentemente do modelo independente". Para eles, o melhor desempenho de modelos adaptados se deve ao fato, de que a razão da verossimilhança não ser afetada por eventos acústicos despercebidos no reconhecimento da fala.

4.4 EXTRAÇÃO DOS TOKENS PROSÓDICOS

As informações prosódicas podem oferecer informações que utilizem adequadamente os aspectos temporais ou que relacionem corretamente as características prosódicas, (por exemplo, a frequência fundamental, intensidade e duração) (ADAMI, 2004). Um modo de utilizar as informações prosódicas é através da modelagem das variações dos aspectos temporais e a interatividade entre os contornos da frequência fundamental e da intensidade. Modelar padrões de contornos da frequência fundamental e da intensidade em conjunto está relacionado ao alto grau de interdependência entre eles. A combinação dos contornos da frequência fundamental, da intensidade e da duração tem por objetivo caracterizar gestos

prosódicos que sejam informações específicas de uma língua. A representação adotada para descrever a informação prosódica é caracterizada pela união entre os estados dinâmicos (queda e crescimento) da frequência fundamental e da intensidade e sua respectiva duração. Como nos segmentos não vozeado não possuem frequência fundamental, um quinto estado é utilizado para representar tais regiões.

Para criar essas representações/classes é necessário seguir cinco etapas: detectar mudanças na dinâmica de frequência fundamental, segmentar os contornos nos pontos das mudanças detectadas do contorno de frequência fundamental, calcular a razão das mudanças do contorno de energia para cada segmento, rotular os segmentos de acordo com as dinâmicas dos contornos de frequência fundamental e de energia e acrescentar o rótulo de duração no segmento.

1º Passo: Detecção das mudanças das dinâmicas de frequência fundamental

A detecção das mudanças das dinâmicas de frequência fundamental é o passo inicial para a criação das classes de estado. Os contornos da frequência fundamental são utilizados para segmentar e converter o sinal de fala na sequência de classes de estados. Os contornos de frequência fundamental são detectados a partir do algoritmo de estilização próximo da cópia (em Inglês *Close-Copy Stylization*) da frequência fundamental. Essa estilização é uma aproximação sintética do contorno da frequência fundamental utilizada para reduzir os ruídos e reduzir os efeitos de micro entonação.

O contorno da frequência fundamental é estimada utilizando a Função de Correlação Cruzada Normalizada (em Inglês *Normalized Cross-Correlation Function*, NCCF) para localizar as frequências fundamentais candidatas e a programação dinâmica é utilizada para selecionar a melhor candidata ou a hipótese sem voz. Para diminuir o custo computacional da NCCF, o sinal de fala coloca uma redução de amostragem para uma frequência F_{ds} definida por

$$F_{ds} = \frac{F_s}{round\left(\frac{F_s}{4F0_{max}}\right)}$$

onde F_s representa a amostra da frequência (em Hz) do sinal original e $F0_{max}$ representa a maior frequência fundamental (em Hz) que será encontrada. Então o NCCF é computado a partir da equação:

$$\emptyset_{i,k} = \frac{\sum_{j=m}^{m+n-1} s_j s_{j+k}}{\sqrt{e_m e_{m+k}}} \; m = iz; z = tF_{ds}; n = \; wF_{ds}; i = 0{,}1 \ldots, M-1$$

onde s_m é uma amostra do sinal de fala, k representa os atrasos do janelamento do sinal reduzido, w (em ms) representa a janela de análise, t (em ms) representa o intervalo da análise do janelamento, s representa uma amostra não zerada do sinal de fala com frequência de amostragem F_{ds}, i representa o índice do janelamento da fala, M representa a quantidade de janelas e

$e_j = \sum_{l=j}^{j+n-1} s_l^2$.

A programação dinâmica é utilizada para obter a melhor frequência fundamental expressando a estimativa para cada janela da fala. I_i representa o número de estados propostos para a janela i, onde o estado 0 é sem voz e os demais são as frequências fundamentais candidatas. Para cada janela $I_i - 1$ é proposto todos os estados possíveis. Para cada enquadramento i são estimados um valor de custo local (d_{ij}), para cada estado j e um custo de transição ($\delta_{i,j,k}$) da janela anterior k. Associado a cada estado há uma penalidade acumulativa ($D_{i,j}$), que representa a melhor combinação com o estado da janela anterior. A penalidade acumulativa é representada pela equação

$$D_{i,j} = d_{i,j} \min_{k \in I_{i-1}} \{D_{i-1,k} + \delta_{i,j,k}\}, 1 \leq j \leq I_i,$$

onde as condições iniciais são representadas pelo $D_{0,j} = 0$, $1 \leq j \leq I_0$ e $I_0 = 2$. O estado da janela anterior com a menor transição, se comparado com o atual, é salva para definir uma trajetória para a melhor combinação na janela anterior. A trajetória ótima é definida pela trajetória com a menor penalidade acumulada. Já que existem estados não vozeados por janela, a decisão de vozeamento também é estimada na trajetória.

Após obter os contornos da frequência fundamental, é aplicado um filtro nos contornos para reduzir as irregularidades no rastreamento do tom. Para cada região de voz, uma parte do modelo é carregada com valor estimado (logaritmo) dos medianos da frequência fundamental. O modelo linear por partes (método de splines) aproxima uma função f(x) utilizando um número de retas $f_i(x)$, para um subintervalo do contorno. A função de aproximação é representada por

$$\tilde{f}(x) = \begin{cases} f_1(x) = a_1 x + b_1, & x_0 \leq x \leq x_1 \\ f_2(x) = a_2 x + b_2, & x_2 \leq x \leq x_3 \\ \vdots & \\ f_m(x) = a_m x + b_m, & x_{m-1} \leq x \leq x_m \end{cases}$$

onde $f(x)$ representa o contorno da região vozeada, $\{(x_i, y_i)\}^{m}_{i=0}$ representa um grupo dos pontos com $x_{i-1} < x_i$ e $y_i = f(x_i)$, a_m representa a inclinação e b_m representa da intersecção da linhas definidas por (x_i, y_i) e (x_{i+1}, y_{i+1}). O número de pontos é proporcional a duração da região vozeada. O grupo de pontos $\{(x_i, y_i)\}^{m}_{i=0}$ é estimado pela minimização do erro

quadrático médio entre as aproximações das funções f(x) e $\tilde{f}$(x) nas regiões vozeada.

2º Passo: Segmentar os contornos nos pontos das mudanças detectadas do contorno de frequência fundamental

Os contornos de frequência fundamental, f_0(x), e de energia, e(x), são segmentados em todos os pontos de mudanças detectadas nas dinâmicas do contorno da frequência fundamental. O domínio definido para ambas as funções é um intervalo fechado e limitado em [0, N] e os segmentos são definidos em cada um dos intervalos (0, b_1), (b_2, b_3), ..., (b_{k-1}, b_k), (b_k, N).

3º Passo: Calcular a razão das mudanças do contorno de energia para cada segmento

Para cada um dos segmentos é calculado a razão de variação do contorno de energia. A razão de variação no interior de cada segmento é aproximada através da representação de uma reta das amostras de energia dentro do segmento.

4º Passo: Rotular os segmentos de acordo com as dinâmicas dos contornos de frequência fundamental e energia

A rotulação do segmento ocorre de acordo com a direção da razão de variação dos contornos de energia e de frequência fundamental contida nos segmentos (--, -+, +-, ++). O estado de um contorno dentro do segmento é definido como "crescimento" (+) quando a inclinação é positiva e de "queda" (-) quando a inclinação é negativa. Para as regiões não vozeadas os dados não são estimados, mas convertidos para uma classe específica (nv).

5º Passo: Acréscimo do rótulo de duração no segmento

A integração da informação de duração de cada segmento ocorre com o acréscimo de um rótulo extra a cada segmento. A duração é uma informação adicional, pois ela é o resultado de diversos fatores, como taxa de fala, ritmo e estresse. A duração é medida pelo número de janelas analisadas e quantizadas em dois rótulos: curto e longo. O rótulo de duração curto é atribuído ao segmento quando o segmento possui o número de janelas inferior a uma quantidade *X*. O rótulo de duração longo é atribuído ao segmento quando o segmento possui um número de janelas igual ou superior a uma quantidade *X*. O método de quantização da duração é diferente para regiões vozeadas e não vozeadas. Como as regiões vozeadas são

segmentadas de acordo com a variação dos contornos de energia e de frequência fundamental, elas são menores quando comparadas as regiões não vozeadas. Com isso, as classes que representam segmentos vozeados são superestimadas quando agrupados com segmentos não vozeados. Com o acréscimo do rótulo de duração, o número total de símbolos utilizados para representar as informações prosódicas do sinal de fala é de dez elementos (--S, --L, +-S, +-L, -+S, -+L, ++S, ++L, nvS, nvL).

4.5 CLASSIFICADOR: N-GRAMA

Um modelo de língua pode ser formulado como uma distribuição da probabilidade P(W) sobre o token W, que reflete a qual frequência de W ocorrer na sentença (HUANG, ACERO e HON, 2001). P(W) pode ser decomposto como

$$\begin{aligned} P(W) &= P(w_1, w_2, \dots, w_n) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \dots P(w_n|w_1, w_2, \dots, w_{n-1}) \\ &= \prod_{i=1}^{n} P(w_1, w_2, \dots, w_{i-1}) \end{aligned}$$

onde P(w_i| $w_{1,}$ $w_{2,...}$ w_{i-1}) representa a probabilidade que w_i seguirá, dado que a sequência do token $w_{1,}$ $w_{2,...}$ w_{i-1} estava presente anteriormente. Isto significa que a escolha de w_i depende do passado completo da entrada. Em um vocabulário v há v^{i-1} histórias diferentes e para representar todas as probabilidades possíveis, v^i valores devem ser estimados. Porém estimar todas as probabilidades não é possível quando existe uma quantidade considerável de valores para i, pois a maioria das histórias $w_{1,}$ $w_{2,...}$ w_{i-1} ocorrem poucas vez ou são únicas. A solução para isso é assumir que P(w_i| $w_{1,}$ $w_{2,...,}$ w_{i-1}) depende apenas de algumas classes equivalentes. Esta classe equivalência pode ser baseada nos tokens anteriores w_{i-N+1}, w_{i-N+2},..., w_{i-1}. Isto nos leva aos N-grama. Pode ser atribuído uma nomenclatura para se referenciar o N-grama de acordo com a quantidade de dependências de tokens anteriores. No unigrama o token não depende dos tokens anteriores, P(w_i), no bigrama o token depende de um token anterior P(w_i|w_{i-1}), no trigrama o token depende de dois tokens anteriores P(w_i|$w_{i-1,}$ w_{i-2}).

Para estimar um bigrama P(w_i|w_{i-1}), isto é, a frequência que o token w_i ocorre quando a palavra anterior é w_{i-1}, é realizada a contagem da quantidade de vezes que sequência ($w_{i-1,}$ w_i) ocorre e normalizado pela quantidade de vezes que o token w_{i-1} ocorre. A equação para estimar o bigrama é dada por

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})}$$

onde C(w_{i-1}, w_i) representa a quantidade de vezes que ocorre a sequência do trio de tokens e C(w_{i-1}) que representa quantidade de vezes que ocorre o par de tokens que precedem o token w_i. A estimativa desta equação se baseia no princípio da máxima verossimilhança, pois estas atribuições das probabilidades produzem o modelo de trigrama que atribui a maior probabilidade para os dados de treinamento de todos os modelos possíveis de trigramas.

Perplexidade

Perplexidade é uma das medidas utilizadas para estimar a probabilidade de um modelo de linguagem (HUANG, ACERO e HON, 2001). A perplexidade é derivada da medida de entropia cruzada. Dada a equação de entropia cruzada, representada por H(W), do modelo P(w_i| $w_{i-N+1,...,}$ w_{i-1})

$$H(W) = -\frac{1}{N_w}\log_2 P(W)$$

onde P(W) atribui a probabilidade para uma sequência de tokens W, $-\log_2 P(W)$ é o algoritmo de compressão que codifica o texto W e N_w representa o comprimento do texto W medido em tokens. A perplexidade representada por PP(W) de uma linguagem P(W) é dada pela equação

$$PP(W) = 2^{H(W)}$$

é definida como recíproca da probabilidade média atribuída pelo modelo para cada token no texto W. Ela possui dois parâmetros fundamentais: um modelo de linguagem e uma sequência de tokens. O conjunto de treinamento da perplexidade mede como o modelo de linguagem se ajusta aos dados de treinamento, assim como a verossimilhança. O conjunto de teste da perplexidade avalia a capacidade de generalização do modelo de linguagem.

De fato, a menor perplexidade se correlaciona com o melhor desempenho de reconhecimento. Quanto maior a perplexidade, mais ramificações o reconhecedor precisa considerar, isto é, o número de tokens da ramificação de um token anterior é maior em média. Neste sentido, a perplexidade indica a complexidade da linguagem, quando existe uma estimativa precisa de P(W). Para uma dada língua, a diferença entre a perplexidade de um modelo de linguagem e a verdadeira perplexidade da língua é uma indicação da qualidade do modelo. A perplexidade de um determinado modelo de linguagem pode mudar em termos de tamanho do vocabulário, o número de estados ou regras gramaticais, e as probabilidades estimadas.

Suavização do N-Grama

Um dos problemas existente na modelagem de N-grama é a escassez de dados de treinamento (HUANG, ACERO e HON, 2001). Em uma base de treinamento sem dados suficientes, muitas sequências de tokens podem não ser bem observadas, gerando muitas probabilidades pequenas. Se P(W) é zero, o token W não é levado em consideração como uma transcrição possível, pois um erro é gerado. O objetivo da suavização é prevenir que o erro aconteça atribuindo a todas as cadeias de tokens uma probabilidade diferente de zero, tornando a probabilidade mais robusta para dados não vistos. Outro objetivo das técnicas de suavização é melhorar a precisão do modelo como um todo. Isto ocorre quando uma probabilidade é estimada a partir de poucos dados, a suavização tem a capacidade de generalização, o que pode melhorar a estimação.

Uma das técnicas de suavização é conhecida por Suavização de *Katz Backoff*. Esta técnica é baseada na técnica de estimativas de Good-Turing (HUANG, ACERO e HON, 2001). A ideia da técnica de estimativas de Good-Turing é particionar o N-grama em grupos, onde o token é alocado de acordo com a sua frequência no N-grama.

Os estados das estimativas de Good-Turing para que qualquer N-grama que ocorre r vezes é definido como

$$r^* = (r+1)\frac{n_{r+1}}{n_r}$$

onde n_r representa o número de N-gramas que ocorre exatamente r vezes nos dados de treinamento. Para converter esta contagem em probabilidade é necessário normalizar pela equação

$$P(a) = \frac{r^*}{N}$$

onde N representa o número original de contagens na distribuição.

A estimativa de Good-Turing estima apenas os modelos de mais baixa ordem. Porém, a suavização de Katz estende a estimativa de Good-Turing adicionando a combinação de modelos de mais alta ordem com os modelos de mais baixa ordem. Utilizando um bigrama como exemplo, a suavização de Katz em conjunto com as estimativas de Good-Turing para contadores não zerados é definido por

$$C^*(w_{i-1}w_i) = \begin{cases} d_r\, r & r > 0 \\ \alpha(w_{i-1})P(w_i) & r = 0 \end{cases}$$

onde d_r é aproximadamente igual para r^*/r. Então todos os bigramas com o contador r não zerado é descontado de acordo com a razão de desconto d_r, que implica que os contadores subtraídos dos contadores não zerados são distribuídos entre os bigramas de contados zerados. O valor para $\alpha(w_{i-1})$ é calculado tanto que o bigrama suavizado é satisfeito pela restrição da probabilidade

$$\alpha(w_{i-1}) = \frac{1 - \sum_{w_i:C(w_{i-1}w_i)>0} P^*(w_i|w_{i-1})}{\sum_{w_i:C(w_{i-1}w_i)=0} P(w_i)}$$
$$= \frac{1 - \sum_{w_i:C(w_{i-1}w_i)>0} P^*(w_i|w_{i-1})}{1 - \sum_{w_i:C(w_{i-1}w_i)>0} P(w_i)}$$

Para calcular a probabilidade de $P^*(w_i|w_{i-1})$ do contador corrigido é preciso normalizar pela equação

$$P^*(w_i|w_{i-1}) = \frac{C^*(w_{i-1}w_i)}{\sum_{w_k} C^*(w_{i-1}w_i)}$$

Na suavização de Katz, a razão de desconto d_r para contadores de grande valor são considerados confiáveis e por isso não são descontados. Há duas restrições aplicadas para a razão de desconto na suavização de Katz. Na primeira, os descontos resultantes são equivalentes aos descontos previstos pela estimativa de Good-Turing. Na segunda, o total de contadores descontados na distribuição de bigramas é igual ao total de contadores que devem ser atribuídos aos bigramas com contador zerado de acordo com a estimativa de Good-Turing. Com isso a equação da razão de descontos d_r é representada por

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}}$$

O modelo de N-grama de Katz é definido em termos de modelo (n-1) - grama de Katz. Para o fim da recursão, o modelo unigrama de Katz considera a maior verossimilhança do modelo unigrama.

4.6 RESULTADOS OBTIDOS

Nesta seção é apresentada a configuração das bases de treinamento e testes dos sistemas, além dos resultados obtidos de cada tokenizador. Os tokenizadores utilizados foram baseados em informações acústicas e informações prosódicas. Para a extração de informações acústicas foram abordadas duas metodologias para a geração de tokens: tokenizador acústico

simples e tokenizador acústico múltiplo. Para a geração dos modelos e dos escores foi utilizado o classificador de N-gramas do SRILM. A configuração adotada para o N-grama é na ordem de três elementos (trigramas). Os trigramas foram adotados por estar entre uma das configurações mais adotadas no reconhecimento de língua (SHEN et al, 2008). Para obter os escores em cada um dos sistemas, é comparada a verossimilhança entre o modelo de língua que se alega para a amostra de áudio e o modelo de línguas impostoras. Cada modelo de língua possui dados referentes à sua própria língua. Porém, o modelo de línguas impostoras contêm os dados referentes de todas as línguas, exceto a língua alegada para a amostra de áudio.

Para os resultados obtidos também foi utilizada a técnica de normalização de escores (REYNOLDS, QUATIERI e DUNN, 2000). A normalização é utilizada para reduzir as distorções existentes nas gravações, causadas pelos dispositivos de captura de áudio. A normalização é representada pela equação

$$R^{norm}(X) = \frac{R(X) - \mu(X, L)}{\sigma(X, L)}$$

onde $R(X)$ representa o escore, $\mu(X, L)$ representa a média entre os escores obtidos para modelos para uma determinada língua e $\sigma(X, L)$ representa a variância entre os escores obtidos para modelos para uma determinada amostra da língua. Para cada amostra testada foram obtidos os escores referentes a cada um dos modelos de língua contidos no sistema.

4.6.1 Tokenizador Acústico Simples

As informações acústicas extraídas são constituídas por 19 coeficientes cepstrais de frequência-mel. Essas informações foram extraídas utilizando uma janela de 20 ms, um deslocamento de 10 ms, com frequências entre 300 Hz e 3400 Hz e com 24 filtros. Estes parâmetros são similares aos utilizados em pesquisas de CAMPBELL et al (2006) e Wong (2004). No tokenizador acústico simples, a tokenização acústica ocorre com o uso de apenas um modelo, que serve de referência como tokenizador. O modelo criado para a tokenização possui no seu conteúdo as informações das 12 línguas treinadas do sistema. A estrutura do tokenizador acústico simples e as bases de áudio são mostradas na Figura 9.

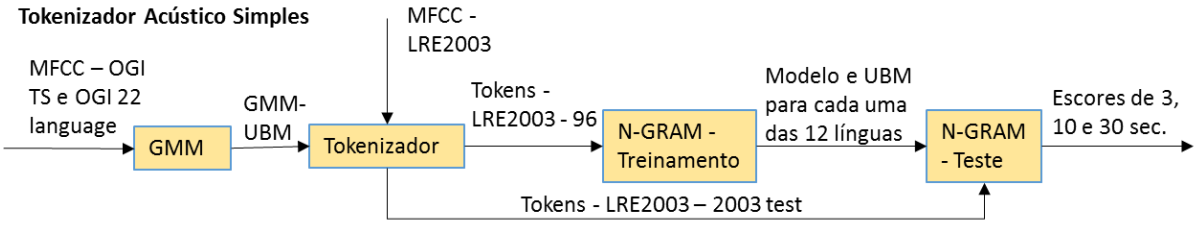

Figura 9. Estrutura do tokenizador acústico simples.

A Tabela 6 apresenta os resultados obtidos com o tokenizador acústico simples, para diferentes ordens de componentes gaussianas (64, 128, 256, 512, 1024). Os resultados apresentados mostram que quanto maior o tempo de duração dos arquivos de teste, melhor é o reconhecimento da língua. A normalização de escores também melhorou o desempenho do sistema. Porém, ao avaliar os resultados pela ordem de componentes gaussianas, observa-se que o melhor resultado obtido encontra-se na ordem de 64 componentes, com arquivos de 30 segundos, com o resultado de 38.12% e resultado normalizado de 35.21%.

Tabela 6. Resultados do tokenizador acústico simples (EER).

Ordem	Resultados			Resultados Normalizados		
	3s	10s	30s	3s	10s	30s
64	43.96%	40.42%	38.12%	41.98%	38.54%	35.21%
128	44.27%	41.98%	41.04%	42.40%	38.65%	36.46%
256	45.10%	44.69%	43.75%	43.85%	43.33%	42.71%
512	45.10%	43.02%	42.29%	41.88%	38.85%	37.29%
1024	44.38%	42.50%	42.19%	40.31%	38.33%	38.33%

A Tabela 7 representa a matriz de confusão do sistema do resultado de 30 segundos da ordem de 64 gaussianas. Em destaque estão definidas as três maiores porcentagens de estimativa para cada língua. A diagonal principal representa as línguas que foram corretamente identificadas, isto é, a língua da amostra é a mesma do que a língua estimada pelo sistema. Pode-se observar na matriz de confusão que a maioria dos áudios de teste foram atribuídos para o Inglês (ENG), o Mandarim (MAN) e o Espanhol (SPA). Estas três línguas são as línguas que mais possuem dados de treinamento da base de dados do NIST 2003, conforme apresentado na Tabela 5. Um melhor balanço de dados de treinamento deve ser

definido a fim de evitar este problema.

Tabela 7. Matriz de confusão de identificação de língua do tokenizador acústico simples.

		Língua estimada pelo sistema											
		ARA	ENG	FAR	FRE	GER	HIN	JAP	KOR	MAN	SPA	TAM	VIE
Língua original da amostra	ARA	9%	31%	3%	10%	3%	8%	8%	6%	5%	9%	9%	1%
	ENG	0%	71%	2%	3%	2%	1%	1%	1%	10%	6%	2%	0%
	FAR	4%	40%	3%	5%	5%	4%	0%	3%	18%	11%	3%	6%
	FRE	8%	19%	4%	20%	5%	1%	4%	0%	18%	16%	5%	1%
	GER	3%	25%	3%	8%	10%	6%	1%	1%	23%	18%	0%	4%
	HIN	10%	15%	3%	10%	0%	9%	6%	5%	15%	19%	6%	3%
	JAP	7%	12%	2%	2%	3%	6%	4%	3%	33%	16%	4%	10%
	KOR	4%	16%	4%	3%	4%	4%	10%	14%	28%	10%	1%	4%
	MAN	10%	14%	10%	8%	3%	5%	0%	8%	23%	9%	8%	5%
	SPA	6%	26%	3%	5%	3%	6%	9%	1%	11%	24%	3%	4%
	TAM	10%	6%	6%	4%	9%	5%	1%	5%	19%	20%	9%	6%
	VIE	3%	28%	3%	3%	0%	1%	3%	6%	26%	11%	4%	14%
	RUS	0%	79%	0%	0%	0%	1%	1%	0%	10%	5%	3%	1%

O tokenizador acústico simples mostrou que apenas acrescentar o número de componentes não significa que o sistema terá um resultado pior. A configuração de 256 componentes gaussianas apresentou os piores desempenhos em todos os testes realizados. Como o sistema estimou a maioria dos resultados para as línguas com mais dados de treinamento, é possível que os modelos dessas línguas tenham sido superestimadas capturando características que seriam relevantes para outras línguas.

4.6.2 Tokenizador Acústico Múltiplo

As informações acústicas foram extraídas com as mesmas configurações utilizadas para o tokenizador acústico simples. No tokenizador acústico múltiplo, a tokenização acústica utiliza um tokenizador para cada língua do sistema. Para obter um modelo para cada língua, o GMM utiliza a técnica de adaptação de modelo. A adaptação do modelo para uma determinada língua utiliza os dados referentes à língua em questão para adaptar ao modelo independente de língua. O modelo independente possui as informações relacionadas a todas as línguas do sistema, com exceção da língua que está sendo adaptada. O detalhe do tokenizador múltiplo é que para cada tokenizador, um escore diferente é produzido para se definir a identidade da língua da amostra. Logo, ao final do processo é necessário submeter todos os

escores obtidos da amostra em um processo de fusão para conseguir o real escore entre a amostra e a identidade que se diz pertencer a amostra. A estrutura do tokenizador acústico múltiplo e as bases de áudio são mostradas na Figura 10.

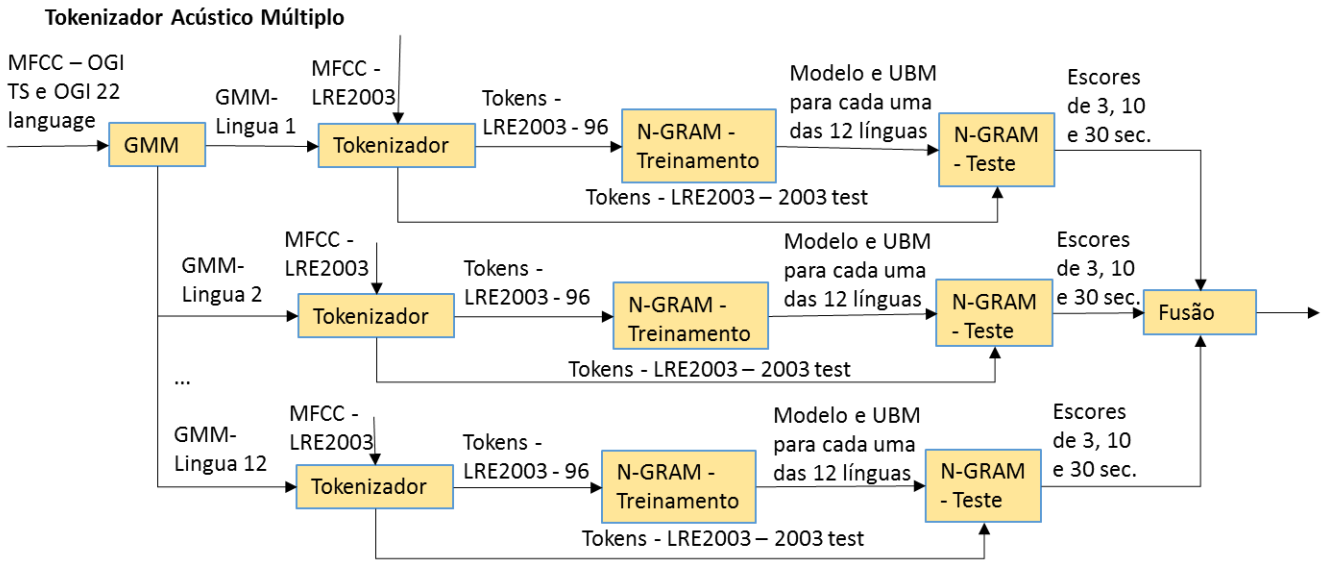

Figura 10. Estrutura do tokenizador acústico múltiplo.

A Tabela 8 apresenta os resultados obtidos com o tokenizador acústico múltiplo, para diferentes ordens de componentes gaussianas (64, 128, 256, 512, 1024). Os resultados apresentados mostram que quanto maior o tempo de duração dos arquivos de teste, melhor é o reconhecimento da língua. Porém, não houve uma mudança significativa entre os resultados obtidos entre os testes de 10 e 30 segundos. A normalização de escores melhorou o desempenho do sistema. Porém ao avaliar os resultados pela ordem de componentes gaussianas, observa-se que o melhor resultado obtido encontra-se na ordem de 64 componentes, com arquivos de 30 segundos, com o desempenho de 41.56% e o resultado normalizado encontra-se na ordem de 1024 componentes com o desempenho de 36.88%.

Tabela 8. Resultados do tokenizador acústico múltiplo (EER).

Ordem	Resultados			Resultados Normalizados		
	3s	10s	30s	3s	10s	30s
64	43.23%	42.29%	41.56%	41.04%	38.65%	37.81%
128	43.96%	43.96%	43.65%	41.88%	40.10%	39.90%
256	45.00%	43.85%	42.92%	40.83%	38.85%	38.54%
512	44.17%	43.85%	43.75%	39.79%	38.54%	37.81%
1024	44.90%	43.44%	43.96%	39.90%	38.33%	36.88%

A Tabela 9 representa a matriz de confusão do sistema do resultado de 30 segundos da ordem de 64 gaussianas. Em destaque estão definidas as três maiores porcentagens de estimativa para cada língua. Pode-se observar na matriz de confusão, que o tokenizador múltiplo sofre do mesmo problema que ocorreu para o tokenizador acústico simples: identificação dos áudios de teste como oriundos do Inglês (ENG), do Mandarim (MAN) e do Francês (FRE).

Tabela 9. Matriz de confusão de identificação de língua do tokenizador acústico múltiplo.

		Língua estimada pelo sistema											
		ARA	ENG	FAR	FRE	GER	HIN	JAP	KOR	MAN	SPA	TAM	VIE
Língua original da amostra	ARA	25%	35%	0%	18%	5%	0%	1%	1%	6%	9%	0%	0%
	ENG	1%	61%	6%	9%	0%	0%	2%	3%	14%	3%	2%	0%
	FAR	5%	34%	5%	4%	8%	0%	3%	3%	29%	1%	4%	6%
	FRE	6%	10%	0%	51%	8%	0%	3%	0%	15%	6%	1%	0%
	GER	11%	26%	1%	29%	8%	3%	5%	0%	13%	1%	1%	3%
	HIN	8%	19%	1%	13%	6%	3%	18%	0%	16%	15%	1%	1%
	JAP	6%	11%	1%	6%	1%	3%	20%	3%	29%	8%	6%	8%
	KOR	4%	11%	1%	20%	1%	0%	19%	6%	25%	3%	0%	10%
	MAN	13%	4%	1%	16%	5%	4%	5%	0%	33%	9%	4%	8%
	SPA	8%	15%	1%	15%	5%	1%	6%	0%	15%	30%	1%	3%
	TAM	18%	10%	1%	9%	1%	1%	8%	0%	21%	16%	6%	9%
	VIE	3%	31%	4%	4%	0%	0%	1%	0%	28%	3%	6%	21%
	RUS	0%	60%	4%	4%	0%	0%	0%	6%	20%	6%	0%	0%

O tokenizador acústico múltiplo mostrou que os desempenhos apresentados não foram superiores aos obtidos pela tokenização acústica simples. Porém, quando são comparadas as matrizes de confusão dos dois tokenizadores pode-se observar uma mudança no comportamento das línguas estimadas. No tokenizador acústico simples, as línguas estimadas na maioria das vezes eram as que mais possuíam dados de treinamento. Já no tokenizador acústico múltiplo, apesar de ainda existir uma concentração línguas estimadas entre o Inglês e o Mandarim, outras línguas como o Francês e o Japonês passaram a ser mais estimadas.

4.6.3 Tokenizador Prosódico

As características prosódicas de frequência fundamental e de energia foram extraídas apenas para a base de áudio do NIST 2003. Estas características foram estimadas a cada 10

ms do sinal de áudio. Diferentes configurações foram utilizadas para a parametrização relacionada a quantização dos rótulo de duração para segmentos vozeados e não vozeados. A parametrização foi utilizada levando em consideração o trabalho proposto por Adami (2004). O parâmetro de intervalo verifica qual o a quantidade máxima de segmentos não vozeados para considerar a mudança de uma sentença. Já os parâmetros de vozeado e não vozeado são referentes às quantidades de segmento com o mesmo rótulo para definir o rótulo de duração do segmento (curto ou longo). A estrutura do tokenizador prosódico e as bases de áudio utilizadas são mostradas na Figura 11.

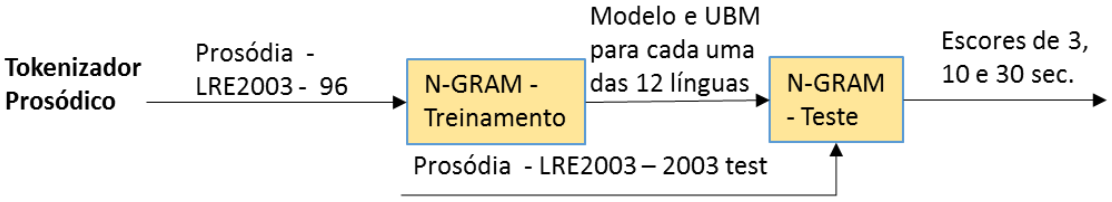

Figura 11. Estrutura do tokenizador prosódico.

A Tabela 10 apresenta resultados obtidos pelo tokenizador prosódico. Os desempenhos obtidos representam as diferentes parametrizações utilizadas para os testes do sistema com as amostras em 3, 10 e 30 segundos. Pode-se observar que os sistemas que utilizaram o intervalo de 25 segmentos obtiveram os melhores resultados em cada um dos grupos de testes. Além disso, os testes com as amostras de 30 segundos o EER possui o desempenho inferior a 28%. Outro detalhe importante é que a variação dos valores do parâmetro de segmentos não vozeados não produziu mudança significativa nos resultados.

Tabela 10. Resultados do tokenizador prosódico (EER).

Parâmetros			Resultados			Resultados Normalizados		
Intervalo	Vozeado	Não Vozeado	3s	10s	30s	3s	10s	30s
25	6	11	37.19%	31.98%	24.79%	36.98%	30.63%	23.85%
25	6	13	37.29%	32.40%	25.21%	36.77%	31.35%	23.44%
50	6	9	38.85%	32.71%	26.35%	38.44%	31.56%	25.52%
50	6	11	38.54%	33.02%	26.25%	39.27%	31.67%	25.00%
50	4	11	38.23%	33.65%	27.50%	38.33%	32.81%	26.25%
50	6	13	38.44%	33.33%	26.35%	38.54%	32.40%	25.73%
50	8	13	38.96%	32.60%	26.77%	38.33%	31.46%	25.52%
50	6	15	38.44%	32.81%	26.77%	38.23%	32.08%	25.62%

A matriz de confusão do sistema prosódico é apresentada na Tabela 11. Esta matriz de confusão é referente ao sistema parametrizado com o intervalo de 25 segmentos, o parâmetro de vozeado em 6 segmentos e de não vozeado parametrizada em 11 segmentos. Em destaque estão definidas as três maiores porcentagens de estimativa para cada língua. Pode-se observar que no sistema prosódico, os resultados com as maiores porcentagens de acerto são referentes à diagonal principal da matriz de confusão.

O tokenizador prosódico mostrou desempenho superior aos tokenizadores acústicos utilizando uma quantidade inferior de tokens, 10 tokens no total. Mesmo com o número reduzido de tokens foi possível observar que em cada conjunto de línguas das amostras, as línguas estimadas com maior índice de acerto eram as línguas da própria amostra.

Tabela 11. Matriz de Confusão de Identificação de Língua do Tokenizador Prosódico.

		Língua estimada pelo sistema											
		ARA	**ENG**	**FAR**	**FRE**	**GER**	**HIN**	**JAP**	**KOR**	**MAN**	**SPA**	**TAM**	**VIE**
Língua original da amostra	**ARA**	58%	11%	3%	5%	5%	3%	5%	4%	3%	4%	0%	1%
	ENG	4%	53%	2%	3%	3%	5%	7%	6%	11%	3%	3%	2%
	FAR	6%	19%	20%	1%	6%	10%	8%	6%	8%	13%	3%	1%
	FRE	11%	18%	6%	38%	5%	4%	9%	1%	1%	5%	1%	1%
	GER	8%	25%	3%	4%	49%	1%	6%	0%	1%	3%	0%	1%
	HIN	6%	11%	5%	4%	9%	20%	13%	6%	0%	11%	13%	3%
	JAP	4%	11%	1%	3%	1%	4%	39%	12%	12%	7%	5%	1%
	KOR	3%	19%	4%	0%	4%	6%	19%	26%	6%	11%	3%	0%
	MAN	1%	15%	0%	5%	4%	3%	13%	5%	39%	4%	9%	4%
	SPA	8%	6%	4%	5%	8%	10%	20%	0%	1%	31%	8%	0%
	TAM	8%	1%	1%	1%	0%	9%	4%	6%	5%	13%	50%	3%
	VIE	1%	0%	0%	1%	0%	0%	3%	0%	3%	9%	10%	74%
	RUS	10%	30%	5%	8%	6%	3%	15%	4%	11%	9%	0%	0%

4.7 FUSÃO DE SISTEMAS

Para unir os escores dos diferentes tokenizadores utilizados, foram adotadas três técnicas de fusão de resultados. O uso de três técnicas de fusão foi realizado para comparar os desempenhos e avaliar a existência de um método de fusão mais adequado para o sistema desenvolvido. As técnicas de média e média ponderada são classificadas como fusão de produto-regra. Na técnica de média, os escores obtidos para cada um dos sistemas foram submetidos ao cálculo de média, produzindo assim um novo escore. Na técnica de média ponderada, para cada um dos sistemas é atribuído um peso diferente. Os pesos atribuídos a

cada um dos sistemas foram obtidos a partir de experimentos produzidos em diferentes combinações.

A técnica de fusão por regressão logística é classificada como fusão de classificadores. Neste tipo de fusão, é definido um grupo de dados para treinar os dados de cada sistema individualmente para que possa ser gerado novos escores para o sistema. A regressão logística para este sistema foi treinado utilizando 90% dos escores obtidos. A partir dos coeficientes produzidos pelo treinamento da regressão logística, todos os escores obtidos pelos sistemas são submetidos à etapa de testes onde são produzidos novos escores.

4.7.1 Regressão Logística Multi-Classe

Regressão Logística Multi-Classe (em Inglês *Multi-Class Logistic Regression*, MLR) é uma técnica de aprendizagem discriminativa similar ao SVM (VAN LEEUWEN e BRÜMMER, 2006). Enquanto o SVM utiliza a estratégia de "um contra todos", um simples modelo MLR pode ser treinado para discriminar todas as línguas. Seus escores podem ser interpretados como a verossimilhança. Estas verossimilhanças são definidas pela equação

$$\lambda_{jt} = \log \frac{p(x_t|L_j, M)}{p(x_t|L_N, M)}, j = 1, \dots, N-1$$

onde *M* representa o modelo de regressão logística e *N* representa a quantidade de línguas do sistema.

A função objetiva de regressão linear básica é a probabilidade total do logaritmo posterior dos rótulos dos dados de treinamento, dados pelos vetores de características destes dados. Assumindo a independência em relação aos testes, isto pode ser representado pela equação

$$\log \prod_{t \epsilon T} P(L_t|x_t, M),$$

onde *T* representa o grupo de segmentos de fala treinados, L_t representa a classe da língua verdadeira para o segmento *t* e x_t representa o vetor de características para o segmento. Pode-se escrever a posterior, $P(L_i|x_t, M)$, em termos de escores de regressão logística, s_{it}, pela equação

$$P(L_t|x_t, M) = \sigma_{it} = \frac{e^{s_{it}+\gamma_i}}{\sum_{j=1}^{N} e^{s_{jt}+\gamma_j}}$$

onde $\gamma_i = \log P(L_i)$ representa a distribuição da probabilidade *prior* das classes da língua. É

utilizado P(L_i) = || T_i || / || T || onde T_i representa o subgrupo de segmentos de fala tendo a classe da língua L_i e onde ||...|| é o operador cardinal. Para evitar um excesso de treinamento, é necessário regularizar a função objetiva. Para isso, é utilizada uma penalidade quadrática nos pesos dos modelos, o que tende a limitar as magnitudes da verossimilhança. Com isto obtêm-se a função objetivo

$$O(M) = \sum_{i=1}^{N} \sum_{t \epsilon T} \log \sigma_{it} - \frac{\alpha}{2} |w_i|^2$$

onde $|w_i|^2 = w_i * w_i$ e α é a constante de regularização positiva que pode ser escolhida com um procedimento de validação cruzada em alguns dados que não foram utilizados. $O(M)$ representa o treinamento da regressão logística para um dado valor α.

4.7.2 Resultados da Fusão

Esta seção apresenta os resultados da fusão dos 3 tokenizadores utilizando a média, média ponderada e a regressão logística. Os resultados obtidos para as três técnicas de fusão foram separados de acordo com a duração dos arquivos de teste. Os resultados dos testes com duração de 3 segundos são apresentados na Tabela 12. Os resultados com testes de duração de 10 segundos estão na Tabela 13 e com duração de 30 segundos estão na Tabela 14. Para comparar os resultados, o tokenizador de referência é o tokenizador prosódico, por ser o tokenizador com melhores resultados. Os resultados do tokenizador prosódico são apresentados na Tabela 10. Na fusão para escores de 3 segundos todas as técnicas de fusão melhoraram o desempenho do sistema. Enquanto o tokenizador prosódico com parametrização de intervalo de 25 segmentos, duração 6 segmentos vozeados e 13 segmentos não vozeados obteve o resultado de 36.77%, todos os métodos de fusão obtiveram resultados inferiores a 34%. Para os escores de 10 segundos a média não obteve melhoras no desempenho do sistema (30.42%), enquanto o tokenizador prosódico obteve 31.35% para 10 segundos. Para os escores de 30 segundos, a média piorou o desempenho do sistema (27.6%), enquanto o tokenizador prosódico obteve 23.44% para 30 segundos. Para os escores de 10 e 30 segundos, os métodos de média ponderada e de regressão logística obtiveram os melhores desempenhos.

Tabela 12. Resultado da fusão de sistemas - EER (3s).

Configuração dos Tokenizadores			Método de Fusão (em 3s)		
Prosódico	Acústico Simples	Acústico Múltiplo	Regressão Logística	Média	Média Ponderada
25-6-13	1024	1024	32.81%	33.54%	33.12%
50-6-15	512	512	35.42%	35.52%	35.21%
50-6-9	256	256	35.42%	37.19%	35.73%
50-8-13	128	128	35.83%	35.83%	34.79%
25-6-11	64	64	34.69%	35.83%	33.96%

Tabela 13. Resultado da fusão de sistemas - EER (10s).

Configuração dos Tokenizadores			Método de Fusão (em 10s)		
Prosódico	Acústico Simples	Acústico Múltiplo	Regressão Logística	Média	Média Ponderada
25-6-13	1024	1024	28.23%	30.42%	29.06%
50-6-15	512	512	30.21%	32.19%	30.73%
50-6-9	256	256	30.21%	32.50%	30.42%
50-8-13	128	128	29.90%	31.15%	30.31%
25-6-11	64	64	29.58%	31.35%	29.27%

Tabela 14. Resultado da fusão de sistemas - EER (30s).

Configuração dos Tokenizadores			Método de Fusão (em 30s)		
Prosódico	Acústico Simples	Acústico Múltiplo	Regressão Logística	Média	Média Ponderada
25-6-13	1024	1024	22.08%	27.60%	22.40%
50-6-15	512	512	23.75%	28.23%	23.75%
50-6-9	256	256	23.44%	29.38%	24.17%
50-8-13	128	128	24.27%	28.33%	24.06%
25-6-11	64	64	22.29%	26.56%	22.71%

As técnicas de média ponderada e de regressão logística conseguiram melhorar os desempenhos obtidos pelo sistema prosódico. Considerando que os tokenizadores acústicos tiveram em poucos casos uma porcentagem maior de identificação de língua do que o tokenizador prosódico. O tokenizador acústico simples teve um índice de acerto maior para o Inglês e o tokenizador acústico múltiplo teve um índice de acerto maior para o Inglês e o Francês.

5. CONCLUSÕES

Neste trabalho de conclusão foi apresentado um sistema automático de reconhecimento de língua utilizando tokens. A tokenização foi empregada no sistema para representar as informações acústicas e prosódicas e a respectiva dinâmica. O uso das técnicas de fusão teve por objetivo explorar as diferentes informações do sistema para produzir um resultado superior para o sistema.

Os resultados do tokenizador acústico simples mostraram que apenas o acrescentar tokens não significa necessariamente que o desempenho do tokenizador será melhor. A configuração de 256 tokens apresentou desempenhos piores em todos os testes realizados. Os resultados foram inferiores inclusive aos resultados obtidos pela configuração de 1024 tokens. Como o sistema estimou a maioria dos resultados para as línguas com mais dados de treinamento, é possível que os modelos dessas línguas tenham sido superestimadas, capturando características que seriam relevantes para outras línguas.

Os desempenhos do tokenizador acústico múltiplo foram superiores aos obtidos pela tokenização acústica simples. Porém, ao comparar as matrizes de confusão dos dois tokenizadores pode-se observar uma mudança no comportamento das línguas estimadas. No tokenizador acústico simples, as línguas estimadas na maioria das vezes eram as que mais possuíam dados de treinamento. Já no tokenizador acústico múltiplo, apesar de ainda existir uma concentração línguas estimadas entre o Inglês e o Mandarim, o Francês passou a ser mais estimada.

O tokenizador prosódico não mostrou desempenho superior aos tokenizadores acústicos utilizando uma quantidade inferior de tokens (10 tokens no total). Mesmo com o número reduzido de tokens foi possível observar que em cada conjunto de línguas das amostras, as línguas estimadas com maior índice de acerto eram as línguas da própria amostra. Isto foi possível pelo fato dos modelos de línguas terem sido melhores definidos ao utilizarem as informações prosódicas.

As técnicas de fusão não conseguiram produzir melhoras significativas quando os resultados foram comparados aos resultados obtidos pelo sistema prosódico. Os tokenizadores acústicos em algumas poucas línguas obtiveram um índice de acerto superior ao tokenizador prosódico. O tokenizador acústico simples teve um índice de acerto maior para o Inglês e o tokenizador acústico múltiplo teve um índice de acerto maior para o Inglês e o Francês.

5.1 TRABALHOS FUTUROS

Neste trabalho, a tokenização de informações acústicas foi uma das técnicas utilizadas para representar as informações da língua. Entretanto, os resultados obtidos, principalmente com os tokenizadores acústicos, levaram a uma série de observações e alterações que poderiam ter sidos exploradas no projeto. Uma das alterações seria utilizar o mesmo tempo de áudio para cada língua em cada uma das etapas do sistema. Com isso seria garantido que nenhuma língua teria vantagem em relação a criação do modelo de língua. Para melhorar o desempenho do sistema a exploração de outros métodos de suavização de N-gramas poderiam ser uma alternativa. Outra forma é o uso de diferentes quantidades de dependências de tokens anteriores, como bigramas e unigramas, para ser mais uma fonte de informação para analisar os resultados.

APÊNDICE A – CÓDIGOS-FONTES E SCRIPTS

A seguir são apresentados os códigos-fontes dos programas e scripts utilizados no sistema de reconhecimento automático de língua. Entre os diversos programas e scripts desenvolvidos durante o trabalho, os mais relevantes são: gmm_token.c, _gmm_token.c, identificacao.pl, fusion_media.pl e gmm_token.c e normalize_scores.pl. O gmm_token.c e o _gmm_token.c são os códigos-fontes desenvolvidos para a tokenização acústica. O script identificacao.pl é utilizado para montar a matriz de confusão dos escores obtidos. O script fusion_media.pl é utilizado para realizar a fusão dos escores com as técnicas de média e média ponderada. O script normalize_scores.pl é utilizado para a normalização de escores.

gmm_token.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "_dataStruct.h"
#include "_feature.h"
#include "_gmm_io.h"
#include "_getopt.h"
#include "_nist.h"
#include "_gmm.h"

/* COMMAND LINE */
#define HEADERTYPE    0 //(0 RAW, 1 NIST)
#define VERBOSE       0
#define CALCTYPE      0 //(0 Only Distance, 1 Distance and Weight)

/*  Command Name  */
char *cmnd;

int option_index = 0;
static char options[] = "c:f:m:o:v:h";
static struct option long_options[] = {
      {"calctype",          required_argument, 0, 'c'},
      {"inputFile",         required_argument, 0, 'f'},
      {"model",             required_argument, 0, 'm'},
      {"outputFile",        required_argument, 0, 'o'},
      {"verbose",           required_argument, 0, 'v'},
      {"help",              no_argument,       0, 'h'},
      {0, 0, 0, 0}
};

void usage (int status){
      fprintf(stderr, "\n");
      fprintf(stderr, " %s - gmm_token\n",cmnd);
      fprintf(stderr, "\n");
```

```
      fprintf(stderr, "  usage:\n");
      fprintf(stderr, "       %s [options]\n", cmnd);
      fprintf(stderr, "  options:\n");
      fprintf(stderr, "       -f, -inputFile <str> [none]\n");
      fprintf(stderr, "            Input a feature file.\n");
      fprintf(stderr, "       -c, -calctype <int> [%d]\n", CALCTYPE);
      fprintf(stderr, "            Define what parameters is used to
define the tokens:\n");
      fprintf(stderr, "            0 Mean and Covariance\n");
      fprintf(stderr, "            1 Weight, Mean and Covariance\n");
      fprintf(stderr, "       -m, -model <str> [none]\n");
      fprintf(stderr, "            Input a speakers file.\n");
      fprintf(stderr, "       -o, -outputFile <str> [none]\n");
      fprintf(stderr, "            Output the test results.\n");
      fprintf(stderr, "       -v, -verbose <int> [%d]\n", 0);
      fprintf(stderr, "            Verbose level.\n");
      fprintf(stderr, "       -h, -help.\n");
      fprintf(stderr, "            Print this message\n");
      exit(status);
}

int main(int argc, char **argv) {
      Array     *mat;
      gmmObject *gmmobj = NULL;
      FILE      *fpout = NULL, *fpfeat = NULL, *fpmodel = NULL;
      double    distance, distanceUBM = 0.0;
      int       calctype = CALCTYPE, verb  = 0, i, *tokens, tok;
      char      outfile[FILENAME_MAX], infeat[FILENAME_MAX],
model[FILENAME_MAX];

// Leitura dos parâmetros de entrada.
      while ((i=getopt_long_only (argc, argv, options, long_options,
&option_index)) != -1) {
            switch(i) {
                  case 0:
                        // If this option set a flag, do nothing else
now.
                        if(long_options[option_index].flag != 0)
break;
                        fprintf(stderr,"option %s (%d)",
long_options[option_index].name,option_index);
                        if(optarg)  fprintf(stderr," with arg %s \n",
optarg);
                        break;
                  case 'f':{ strcpy(infeat, optarg);   break; }
                  case 'm':{ strcpy(model, optarg);   break; }
                  case 'o':{ strcpy(outfile, optarg);  break; }
                  case 'c':{ calctype = atoi(optarg);  break; }
                  case 'v':{ verb = atoi(optarg);      break; }
                  case 'h':
                  default:{ usage(1); }
            }
      }

      if(strlen(infeat) == 0){
```

```
          fprintf(stderr, "Error: Feature file (-inputFile) not
defined.\n");
          return -1;
     }

     if(strlen(outfile) == 0){
          fprintf(stderr, "Error: Output scores (-outputFile) not
defined.\n");
          return -1;
     }
// Leitura o modelo tokenizador.
     if(strlen(model) == 0){
          fprintf(stderr, "Error: Model (-model) not defined.\n");
          return -1;
     }
     if((fpmodel = fopen(model, "r")) == NULL){
          fprintf (stderr, "Error: Could not open <%s>\n", model);
          return -1;
     }

     gmmobj = readGMMFile(fpmodel);
     fclose(fpmodel);
// Leitura do arquivo que contêm as informações acústicas da fala.
     if((fpfeat = fopen(infeat, "r")) == NULL){
          fprintf (stderr, "Error: Could not open <%s>\n", infeat);
          return -1;
     }
     mat = readFeatureFile(fpfeat, gmmobj->iSampleDimension);
     fclose(fpfeat);

     if((fpout = fopen(outfile, "w")) == NULL){
         fprintf(stderr, "Error: Could not write <%s>\n", outfile);
          return -1;
     }

/* Define os parâmetros que o sistema utilizará para realizar a
tokenização. */
     if (calctype == 0) {
         tokens = getGmmTokenDistance(mat->mat, mat->nRows, gmmobj,
verb);
     } else {
          tokens = getGmmToken(mat->mat, mat->nRows, gmmobj, verb);
     }

     for(i = 0, tok = -1; i < mat->nRows; i++){
          if(tok != tokens[i]){
               fprintf(fpout, "%d ", tokens[i]);
               tok = tokens[i];
          }
     }

     fclose(fpout);
     if(mat != NULL) freeArray(mat);
     if(gmmobj != NULL) freeGmm(&gmmobj);
     if(tokens != NULL) free((char*) tokens);
```

```
     return 0;
}
```

_gmm_token.c

```
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "_dataStruct.h"
#include "_gmm_token.h"
#include "_gmm.h"

/* Função que calcula a distância entre as componentes gaussianas e
os dados de características da fala. Retorna a sequência de
gaussianas com menor distância. */
int* getGmmTokenDistance(real **data, int nrows, gmmObject *gmmobj,
int verbose){
     int i, j, l, *tokens;
     double bestprob, distance, diff;

     tokens = (int *) malloc(nrows * sizeof(int));

/* Calcula a distância entre as misturas gaussianas e as
características da fala */
     for(i = 0; i < nrows; i++){
          for(j = 0, bestprob = MAXDOUBLE; j < gmmobj->iMixtures;
j++){
               for(distance = 0, l = 0; l < gmmobj-
>iSampleDimension; l++){
                    diff = data[i][l] - gmmobj-
>pMixtures[j].dMeans[l];
                    distance += diff * diff / gmmobj-
>pMixtures[j].dCovariance[l];
               }
// Seleciona a componente gaussiana de menor distância.
               if(distance < bestprob){
                    bestprob = distance;
                    tokens[i] = j;
                    if(verbose > 9) fprintf(stdout, "new token %d
for line %d.\n", j, i);
               }
          }
     }
     return tokens;
}


/* Função que calcula a distância entre as componentes gaussianas e
os dados de características da fala. Retorna a sequência de
gaussianas a maior verossimilhança. Leva em consideração os pesos
das componentes gaussianas. */
int* getGmmToken(real **data, int nrows, gmmObject *gmmobj, int
verbose){
```

```
      int i, j, l, *tokens;
      double *determinant, bestprob = log (GMM_MINPROB), dens;

      determinant = (double *) malloc(gmmobj->iMixtures *
sizeof(double));
      tokens = (int *) malloc(nrows * sizeof(int));
      calculeDeterminant(gmmobj, determinant);

      for(i = 0; i < nrows; i++){
            for(j = 0, bestprob = log(GMM_MINPROB); j < gmmobj-
>iMixtures; j++){
                  dens = density(gmmobj, data[i], j, determinant);
                  if(dens < GMM_MINPROB) dens = GMM_MINPROB;
                  dens = log(dens);

                  if(dens > bestprob){
                        bestprob = dens;
                        tokens[i] = j;
                        if(verbose > 9) fprintf(stdout, "new token %d
for line %d.\n", j, i);
                  }
            }
      }
      free((char*) determinant);
      return tokens;
}
```

identificacao.pl

```
#!/usr/bin/perl

use Getopt::Long;
use File::Basename;
use File::Glob;

# Unbuffer STDOUT
$| = 1;

#-----------------------------------------------------------------------
# Parameters default
#-----------------------------------------------------------------------
$outputfile   = "result.txt";

#-----------------------------------------------------------------------
# Command line processing
#-----------------------------------------------------------------------
@COM_LINE_ARGS = ("help","inputfile=s","mapfile=s","outputfile=s");
@COM_LINE_ARGS_DESC = (
      "-inputfile   <string>  arquivo de entrada (scores).\n",
      "-mapfile     <string>  Arquivo de mapeamento de
resultados.\n",
      "-outputfile  <string>  Arquivo de saÃda.\n");

if ($#ARGV == -1 || GetOptions(@COM_LINE_ARGS) == 0 || $opt_help ==
1) {
```

```
    print STDERR "Usage: $0 [args]\n";
    print STDERR "Possible arguments are:\n";
    print STDERR @COM_LINE_ARGS_DESC;
    exit(-1);
}

$inputfile  = $opt_inputfile;
$mapfile    = $opt_mapfile;
$outputfile = $opt_outputfile  if (defined $opt_outputfile);
@trlang = ('ARABIC', 'ENGLISH', 'FARSI', 'FRENCH', 'GERMAN', 'HINDI',
'JAPANESE', 'KOREAN', 'MANDARIN', 'SPANISH', 'TAMIL', 'VIETNAMESE',
'RUSSIAN');
@falang = ('ARABIC', 'ENGLISH', 'FARSI', 'FRENCH', 'GERMAN', 'HINDI',
'JAPANESE', 'KOREAN', 'MANDARIN', 'SPANISH', 'TAMIL', 'VIETNAMESE');

if (!(-e $inputfile)) { print STDERR "-inputfile $inputfile does not
exist!\n"; exit(-1); }
if (!(-e $mapfile)) { print STDERR "-mapfile $mapfile does not
exist!\n"; exit(-1); }

# Faz o mapeamento das língua correta para cada amostra.
open(IN, $mapfile) || die "Could not open $mapfile to read\n";
$line = <IN>;
while($line){
      chomp($line);
      ($msg, $model) = split(" ",$line);
      $mapmodels{lc($msg)} = $model;
      $line = <IN>;
}
close IN;

for($i = 0; $i < @trlang; $i++){
      for($j = 0; $j < @falang; $j++){
            $nroresults{$trlang[$i]}{$falang[$j]} = 0.0;
      }
}

$msgant = "-";
$count = 1;
# Carrega as informações dos escores do arquivo de testes.
open(IN, $inputfile) || die "Could not open $inputfile to read\n";
$line = <IN>;
while($line){
      chomp($line);
      ($msg, $model, $score) = split(" ",$line);
# Ao mudar a amostra de teste, atribui 1 à língua estimada.
      if(!($msgant =~ $msg)){
            if(!($msgant =~ "-")){
                  $nroresults{$mapmodels{$msgant}}{$modelbest} += 1;
                  $count++;
            }
            $msgant = $msg;
            $bestscore = $score;
            $modelbest = $model;
      }
```

```
# Seleciona o melhor escore para a amostra de teste.
      if($bestscore < $score){
            $modelbest = $model;
            $bestscore = $score;
      }
      $line = <IN>;
}
close IN;

$nroresults{$mapmodels{$msgant}}{$modelbest} += 1;

# Escreve os resultados em arquivo.
open(OUT, ">$outputfile")  || die "Could not open $outputfile to
write\n";
print OUT "   ;ARA;ENG;FAR;FRE;GER;HIN;JAP;KOR;MAN;SPA;TAM;VIE;\n";

for($i = 0; $i < @trlang; $i++){
      $lang = substr($trlang[$i], 0, 3);
      print OUT "$lang;";
      for($j = 0; $j < @falang; $j++){
            $value =
sprintf("%03s",$nroresults{$trlang[$i]}{$falang[$j]});
            printf OUT "$value;";
      }
      print OUT "\n";
}
close OUT;
```

fusion_media.pl

```
#!/usr/bin/perl

use Getopt::Long;
use File::Basename;
use File::Glob;

# Unbuffer STDOUT
$| = 1;

#--------------------------------------------------------------------
# Parameters default
#--------------------------------------------------------------------
$outputfile   = "result.txt";

#--------------------------------------------------------------------
# Command line processing
#--------------------------------------------------------------------
@COM_LINE_ARGS = ("help","inputfile=s","outputfile=s","ratio=s");
@COM_LINE_ARGS_DESC = (
      "-inputfile   <string>  arquivo de entrada (lista de
scores).\n",
      "-outputfile  <string>  Arquivo de saÃda.\n",
      "-ratio       <string>  Razao entre arquivos.\n");
```

```
if ($#ARGV == -1 || GetOptions(@COM_LINE_ARGS) == 0 || $opt_help ==
1) {
    print STDERR "Usage: $0 [args]\n";
    print STDERR "Possible arguments are:\n";
    print STDERR @COM_LINE_ARGS_DESC;
    exit(-1);
}

$inputfile  = $opt_inputfile;
$outputfile = $opt_outputfile  if (defined $opt_outputfile);
$ratiofile  = $opt_ratio       if (defined $opt_ratio);

if (!(-e $inputfile)) { print STDERR "-inputfile $inputfile does not
exist!\n"; exit(-1); }
@rationindex = split(":",$ratiofile);
open(IN, $inputfile) || die "Could not open $inputfile to read\n";
$line = <IN>;
$countfiles = 0;
while($line){
     chomp($line);
     @data = split(" ",$line);
     if (!(-e $data[0])) { print STDERR "-file $data[0] does not
exist!\n"; exit(-1); }
# Verifica qual razão para a média a ser utilizado.
     if(@data > 1){
          $ratio = $data[1];
     } else {
          if(defined $opt_ratio){
               $ratio = $rationindex[$countfiles];
          } else {
               $ratio = 1;
          }
     }
     open(INDATA, $data[0]) || die "Could not open $data[0] to
read\n";
     $dataline = <INDATA>;
     $countlines = 0;
# Soma o escores.
     while($dataline){
          chomp($dataline);
          @scores = split(" ",$dataline);
          $sum[$countlines] += $scores[2] * $ratio;
          $countlines++;
          $dataline = <INDATA>;
     }
     close INDATA;
     $line = <IN>;
     $countfiles++;
}
close IN;

# Calcula a media.
for($i = 0; $i < @sum; $i++){
     $sum[$i] /= $countfiles;
}
```

```
# Escreve os resultados em arquivo.
open(OUT, ">$outputfile")  || die "Could not open $outputfile to
write\n";
open(INDATA, $data[0]) || die "Could not open $data[0] to read\n";
for($i = 0; $i < @sum; $i++){
     $dataline = <INDATA>;
     chomp($dataline);
     @scores = split(" ",$dataline);
     print OUT "$scores[0] $scores[1] $sum[$i]\n";
}
close INDATA;
close OUT;
```

normalize_scores.pl

```
#!/usr/bin/perl

use Getopt::Long;
use File::Basename;
use File::Glob;

# Unbuffer STDOUT
$| = 1;

#---------------------------------------------------------------------
# Parameters default
#---------------------------------------------------------------------
$outputfile   = "result.txt";

#---------------------------------------------------------------------
# Command line processing
#---------------------------------------------------------------------
@COM_LINE_ARGS = ("help","inputfile=s","outputfile=s");
@COM_LINE_ARGS_DESC = (
      "-inputfile   <string>  arquivo de entrada (scores).\n",
      "-outputfile  <string>  Arquivo de saÃda.\n");

if ($#ARGV == -1 || GetOptions(@COM_LINE_ARGS) == 0 || $opt_help ==
1) {
    print STDERR "Usage: $0 [args]\n";
    print STDERR "Possible arguments are:\n";
    print STDERR @COM_LINE_ARGS_DESC;
    exit(-1);
}

$inputfile  = $opt_inputfile;
$outputfile = $opt_outputfile  if (defined $opt_outputfile);

if (!(-e $inputfile)) { print STDERR "-inputfile $inputfile does not
exist!\n"; exit(-1); }
open(IN, $inputfile) || die "Could not open $inputfile to read\n";
open(OUT, ">$outputfile")  || die "Could not open $outputfile to
write\n";
$line = <IN>;
$sum  = 0.0;
```

```
$sum2 = 0.0;
$count  = 0;
while($line){
      chomp($line);
      @data = split(" ",$line);
# Acumula estatísticas para a normalização.
      $models[$count] = $data[1];
      $scores[$count] = $data[2];
      $sum += $data[2];
      $sum2 += $data[2] * $data[2];
# Ao carrega as estatísticas de todos os modelos para a amostra,
calcula a normalização.
      if($count == 11){
            $mean = $sum / 12;
            $std = sqrt($sum2 / 12 - $mean * $mean);
# Calcula e imprime os resultados normalizados.
            for($i = 0; $i <= $count; $i++){
                  $scores[$i] = ($scores[$i] - $mean) / $std;
                  print OUT "$data[0] $models[$i] $scores[$i]\n";
            }
# Reseta as variáveis acumuladoras.
            $sum  = 0.0;
            $sum2 = 0.0;
            $count  = 0;
      } else {
            $count++;
      }
      $line = <IN>;
}
close OUT;
close IN;
```

BIBLIOGRAFIA

ABDI, Hervé. Signal Detection Theory (SDT). In Neil Salkind. **Encyclopedia of measurement and statistics.** v. 3, 2007, Califórnia, Estados Unidos. p. 886 – 889.

ADAMI, André Gustavo et al. **Modeling Prosodic Differences for Speaker Recognition.** In: ICASSP, v. 4, 2003, Hong Kong, China, p. 788 – 791.

ADAMI, André Gustavo. **Modeling Prosodic Differences for Speaker and Language Recognition**. 2004. 152 f. Dissertação (Doutorado) – Oregon Health & Science University, OGI School of Science & Engineering, 2004.

AMBIKAIRAJAH, Elithamby. et al. Language identification: a tutorial. **Circuits and Systems Magazine, IEEE**, Nova York, Estados Unidos, v. 11, n. 2, p. 82 – 108, 2011.

BISHOP, Christopher M. **Neural networks for pattern recognition**. Clarendon Press, Oxford, Inglaterra, 1995.

BOURNE, Mike et al. Implementing performance measurement systems: a literature review. **International Journal of Business Performance Management**, 2003, New York, Estados Unidos. v. 5 n. 2 p. 1 – 24.

BRÜMMER, Niko; DATAVOICE, Spescom. FoCal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scores – Tutorial and user manual – "Software disponível em https://sites.google.com/site/nikobrummer/focalmulticlass ". Acesso em 9 de novembro de 2013.

CANAVAN, Alexandra; ZIPPERLEN, George. The CALLFRIEND Corpus. Linguist Data Consortium, 1996.

CANAVAN, Alexandra; GRAFF, David; ZIPPERLEN, George. The CALLHOME corpus. Linguist Data Consortium, 1997.

CAMPBELL, William M. et al. High-level speaker verification with support vector machines, In: **Acoustics, Speech and Signal Processing**, 2004, IEEE International Conference , p. 73 – 76, Montreal, Canadá, 2004.

CAMPBELL, William M. et al. Support vector machines for speaker and language recognition, **Computer Speech & Language**, 20.2, 2006, p. 210 – 229, Nova York, EUA.

CAMPBELL, William M. et al. Advanced Language Recognition using Cepstra and Phonotactics: MITLL System Performance on the NIST 2005 Language Recognition Evaluation, In: **Speaker and Language Recognition Workshop**, 2006, IEEE Odyssey 2006, p. 1 – 8, San Juan, Porto Rico, 2006.

CIERI, Christopher; MILLER, David; WALKER, Kevin. The Fisher Corpus: a resource for the next generation of speech-to-text. In: Proceedings of LREC. 2004. Lisboa, Portugal, p. 69 – 71.

CIERI, Christopher; et al. The Mixer Corpus, multichannel speaker recognition data. Pennsylvania Univ Philadelphia, 2004 Estados Unidos. p. 1 – 4.

DAVIS, Steven B.; MERMELSTEIN, Paul. Comparison of Parametric representations for monosyllabic Word Recognition in Continuously Spoken Sentences. **Acoustic, Speech and Signal Processing**, IEEE Transactions, 28.4, 1980, p. 357 - 367, Piscataway, Estados Unidos, 1980.

DUDA, Richard O.; HART, Peter E. e STORK, David G.. Pattern Classification. 2 Nova York, Estados Unidos, 2012.

FERRI, Csar; HERNÁNDEZ-ORALLO, José; MODROIU, R. An experimental comparison of performance measures for classification. **Pattern Recognition Letters**, Nova York, Estados Unidos, v. 30, p. 27-38, 2009.

FURUI, Sadaoki. Comparison of Speaker Recognition Methods using statistical features and dynamic features. **Acoustic, Speech and Signal Processing**, IEEE Transactions, 29.3, 1981, p. 342 - 350, Piscataway, Estados Unidos, 1980.

GAUVAIN, Jean-Luc; MESSAOUDI, Abdel: SCHWENK, Holger. Language recognition using phone lattices. In: Interspeech 2004, Jeju, Coréia do Sul, v. 4, p. 1283 – 1286.

HARPER, M. P; MAXWELL, M. Spoken Language Characterization In: BENESTY, Jacob. **Springer handbook of speech processing**. Berlin, Alemanha, Ed. Springer-Verlag Berlin Heidelberg, 2008, p. 798 - 807.

HARWART, David; HASEGAWA-JOHNSON, Mark. PHONETIC Landmark Detection for Automatic Language Identification. University of Illinois at Urbana-Champaign, Estados Unidos. v. 51, 2010.

HAYKIN, Simon. **Redes Neurais. Princípios e Práticas**. Porto Alegre, Brasil, Ed. Bookman Companhia Editora, 2001, 2ª Edição.

HERMANSKY, Hynek. Perceptual linear predictive (PLP) analysis of speech. The Journal of the Acoustical Society of America, Estados Unidos, 1990, v. 87, n. 4, p. 1738 – 1752.

HUANG, Xuedong; ACERO, Alejandro; HON, Hsiao-Wuen. **Spoken Language Processing**. New Jersey, Estados Unidos, Ed. Prentice Hall PTR, 2001.

LANDER, T. et al. The OGI 22 Language Telephone Speech Corpus, Oregon Graduate Institute, Portland, Estados Unidos. 1995.

LI, Haizhou; MA, bin; LEE, Chin-Hui. A vector space modeling approach to spoken language identification. **Audio, Speech and Language Processing**, IEEE Transactions, 15.1, 2007, p. 271-284, Piscataway, Estados Unidos, 2007.

MAKHOUL, John. Linear Prediction: A tutorial review. In: Proceedings of the IEEE. 1975, v. 63, n. 4, Piscataway, Estados Unidos, 1975.

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich**. An Introduction to Information Retrieval**. Cambridge, Inglaterra, Ed. Cambridge University Press, 2009.

MARY, Lena; YEGNANARAYANA, B. Extraction and representation of prosodic features for language and speaker recognition. **Speech communication**, Nova York, Estados Unidos, 2008, p. 782-796.

MUTHUSAMY, Yeshwant Kumar. **A segmental approach to automatic language identification**. 1993. 288 f. Dissertação (Doutorado) – Oregon Graduate Institute of Science & Technology, 1993.

NAVRATIL, Jiri. Automatic Language Identification. In SHULTZ, Tanja, KIRCHHOFF, Katrin. **Multilingual Speech Processing**. Estados Unidos, Academic Press, p. 233-272, 2006.

NG, Raymond W. M. et al. Prosodic Attribute Model for Spoken Language Identification. In: Acoustic Speech and Signal Processing (ICASSP), 2010, IEEE International Conference, p. 5022 – 5025.

NIST 1996 Language Recognition Evaluation Plan (LRE96), 1996, Disponível em: <http://www.itl.nist.gov/iad/mig//tests/lang/1996/LRE96EvalPlan.pdf>. Acesso em: 25 maio. 2013.

NIST 2003 Language Recognition Evaluation Plan (LRE03), 2003, Disponível em: <http://www.nist.gov/speech/tests/lre/2003/LRE03EvalPlan-v1.pdf>. Acesso em: 27 abr. 2013.

NIST 2005 Language Recognition Evaluation Plan (LRE05), 2005, Disponível em: <http://www.nist.gov/speech/tests/lre/2005/LRE05EvalPlan-v5-2.pdf>. Acesso em: 27 abr. 2013.

NIST 2007 Language Recognition Evaluation Plan (LRE07), 2007, Disponível em: <http://www.nist.gov/speech/tests/lre/2007/LRE07EvalPlan-v8b.pdf>. Acesso em: 27 abr. 2013.

NIST 2009 Language Recognition Evaluation Plan (LRE09), 2009, Disponível em: <http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf>. Acesso em: 27 abr. 2013.

NKADIMENG, Calvin. **Language identification using Gaussian mixture models**. 2010. 70 f. Dissertação (Mestrado) – Stellenbosch University, Department of Electrical & Eletronic Engineering, 2010.

NOOR, Elad; ARONOWITZ, Hagai. Efficient language identification using anchor models and support vector machines. In: **Speaker and Language Recognition Workshop**, 2006, IEEE Odyssey 2006, p. 1 – 6, San Juan, Porto Rico, 2006.

PELECANOS, Jason; SRIDHARAN, Sridha. Feature Warping for Robust Speaer Verification. In: The Speaker Recognition Workshop, 2001, Creta, Grécia.

REYNOLDS, Douglas A.; QUATIERI, Thomas F.; DUNN, Robert B. Speaker Verification

using adapted Gaussian Mixture Models. Digital Signal Processing, Estados Unidos v. 10, p. 19 – 41, 2000.

REYNOLDS, D. A; CAMPBELL W. M. Text-Independent Speaker Recognition. In: BENESTY, Jacob. **Springer handbook of speech processing**. Berlin, Alemanha, Ed. Springer-Verlag Berlin Heidelberg, 2008, Alemanha, p. 763 - 779.

RONG, Tong. **Towards high performance phonotatic feature for spoken language recognition**. 2012. 140 f. Dissertação (Doutorado) – Nanyang Technological University, School of Computer Engineering, 2012.

ROSENBERG, A. E.; BIMBOT, F.; PARTHASARATHY, S. Overview of Speaker Recognition. In: BENESTY, Jacob. **Springer handbook of speech processing**. Berlim, Alemanha, Ed. Springer-Verlag Berlin Heidelberg, 2008, p. 725 – 729.

ROUAS, Jean-Luc. Modeling Long and Short-term prosody for language identification. In: European Conf. on Speech Communication and Technology (INTERSPEECH2005-EUROSPEECH), 9, 2005, Lisboa, Portugal, 2005, p. 2257 – 2260.

SHEN, Wade et al. Automatic language recognition via spectral and token based approaches. In: BENESTY, Jacob. **Springer handbook of speech processing**. Berlin, Alemanha, Ed. Springer-Verlag Berlin Heidelberg, 2008, p. 811 – 823.

SINISCALCHI, Sabato Marco et al. Universal attribute characterization of spoken languages for automatic spoken language recognition. **Computer Speech and Language**, Nova York, Estados Unidos, v. 27, p.209 – 227, 2012.

SOKOLOVA, Marina; LAPALME, Guy. A systematic analysis of performance measures for classification tasks. **Information Processing and Management**, Nova York, Estados Unidos, v. 45, p. 427 – 437, 2009.

STOLCKE, Andreas, SRILM – An extensible language modeling toolkit, In: Interspeech 2002, Denver, Estados Unidos, 2002.

THIRUVARAN, Tharmarajah; AMBIKAIRAJAH, Eliathaby; EPPS, Julien. FM Features for Automatic Forensic Speaker Recognition. In Forensic Speaker Recognition, Interspeech 2008, Brisbane, Austrália, 2008.

TORRES-CARRASQUILLO, Pedro A.; REYNOLDS, Douglas A.; DELLER, John R., Language identification using Gaussian mixture model tokenization, In: Acoustics, Speech and Signal Processing (ICASSP), 2002 IEEE International Conference, V. 1, p. 757 – 760, Orlando, Estados Unidos, 2002.

TORRES-CARRASQUILLO, Pedro A. et al, Approaches to language identification using Gaussian mixture models and shifted delta cepstral features, In: Interspeech 2002, V. 2, p. 33 – 36, Denver, Estados Unidos, 2002.

VAN LEEUWEN, David A., BRÜMMER, Niko. Channel-dependent GMM and Multi-class Logistic Regresssion models for language recognition. In: **Speaker and Language Recognition Workshop**, 2006, IEEE Odyssey 2006, p. 1 – 8, San Juan, Porto Rico, 2006.

WANG, Liang. **Automatic Spoken Language Identification**. 2008. 135 f. Dissertação (Doutorado) – The University of New South Wales, School of Electrical Engineering and Telecommunications Faculty of Engineering, 2008.

WONG, Ka-keung; SIU, Man-hung. Automatic language identification using discrete hidden Markov model. In: Interspeech 2004, Jeju, Coréia do Sul, v. 4, p. 1283 - 1286

WONG, Kim-Yung Eddie. **Automatic Spoken Language Identification Utilizing Acoustic and Phonetic Speech Information**. 2004. 153 f. Dissertação (Doutorado) – Quensland University of Technology, School of Electrical & Electronic Systems Engineering, 2004.

ZISSMAN, Marc A.; BERKLING, Kay M. Automatic language identification. **Speech Communication**, Nova York, Estados Unidos, v. 35, p. 115-124, 2001.