Pós-Graduação em Ciência da Computação

Leonardo Valeriano Neri

Extração de Características para Segmentação de Locutores

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2019

Leonardo Valeriano Neri

Extração de Características para Segmentação de Locutores

Tese de Doutorado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de Concentração: processamento de sinais e reconhecimento de padrões
Orientador: Tsang Ing Ren
Coorientador: George Darmiton da Cunha Cavalcanti

Recife
2019

Catalogação na fonte
Bibliotecária Mariana de Souza Alves CRB4-2106

N445e Neri, Leonardo Valeriano
Extração de Características para Segmentação de Locutores – 2019.
113f.: il., fig., tab.

Orientador: Tsang Ing Ren
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da computação. Recife, 2019.
Inclui referências e apêndices.

1. Processamento de sinais e reconhecimento de padrões. 2. Diarização de locutores. 3. Segmentação de locutores. 4. Sobreposição de fala. I. Ren, Tsang Ing (orientador). II. Título.

006.4 CDD (22. ed.) UFPE-MEI 2019-149

Leonardo Valeriano Neri

Extração de Características para Segmentação de Locutores

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 21/02/2019.

_____

Orientador: Prof. Dr. Tsang Ing Ren

BANCA EXAMINADORA

_____

Prof. Dr. Aluízio Fausto Ribeiro Araújo
Centro de Informática /UFPE

_____

Prof. Dr. Paulo Salgado Gomes de Mattos Neto
Centro de Informática /UFPE

_____

Prof. Dr. Leandro Maciel Almeida
Centro de Informática / UFPE

_____

Prof. Dr. Jugurta Rosa Montalvao Filho
Departamento de Engenharia Elétrica / UFS

_____

Prof. Dr. André Gustavo Adami
Centro de Computação e Tecnologia da Informação / UCS

Dedico este trabalho à minha esposa, família, orientadores e amigos que tanto me ajudaram a completar essa jornada. Muito grato e contem sempre comigo para novos desafios!

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me dado força, perseverança e pessoas que me ensinaram a nunca desistir e que me guiaram em momentos muito difíceis. Dentre essas pessoas estão minha esposa Paula, meus familiares, meus orientadores e amigos desde o tempo da faculdade. Estes sempre me apoiaram nos meus objetivos, me ensinaram a ter foco e resiliência para trabalhar e finalizar cada etapa dessa caminhada. Agradeço também aos amigos que criei durante a jornada, por serem verdadeiros exemplos de esforço, dedicação e sucesso.

Agradecimentos a Tsang, George, Hector e André por me auxiliarem na realização dos meus objetivos acadêmicos.

Agradecimentos aos participantes da banca de avaliação por aceitarem o convite da minha defesa de Tese. Espero que apreciem o trabalho desenvolvido.

RESUMO

A transcrição de locutores em conversações determina "quem falou e quando?", identificando o número de locutores presentes e os intervalos onde cada locutor fala. Um sistema de transcrição de locutores implementa quatro etapas fundamentais: Detecção de atividade de voz, extração de características acústicas, segmentação e clusterização dos locutores. A tarefa de segmentação torna-se um grande desafio em conversas de estilo livre, nas quais as transições entre locutores são frequentes e em muitas delas ocorrem a sobreposição da fala de dois ou mais locutores. Nesse cenário, a detecção de transições/mudanças, precisa ser feita utilizando segmentos curtos da fala de dois ou mais locutores, para não incluir duas ou mais mudanças na mesma amostra, e assim evitando perdas durante o processo. O estado da arte i-vector representa as características da fala correspondentes à identidade do locutor, projetada para discriminar pessoas. No entanto, seu desempenho é afetado pelo tamanho da amostra da fala, de tal modo que no cenário de conversações de estilo livre, seu desempenho é comparável com métodos tradicionais de modelagem das características acústicas utilizando misturas gaussianas. Propomos o Mel Cepstral Affinity Features (MCAF) um extrator de características da fala projetado para amostras curtas e próprio para a tarefa de segmentação de locutores. A característica proposta discrimina os diferentes tipos de fala: homogênea (amostra contendo um único locutor), heterogênea (dois locutores presentes sem sobreposição) e a sobreposta (ao menos dois locutores falando simultaneamente). Um método de janelas deslizantes utiliza essa discriminação para detectar as mudanças de locutor. Experimentos utilizando o corpora da AMI mostram que nossa proposta exibe um desempenho na métrica F_1 *score* 38% superior ao método de segmentação tradicional utilizando as características *Mel Frequency Cepstral Coefficients* (MFCC) e a distância *Generalized Likelihood Ratio* (GLR), e 15% superior ao método utilizando i-vector, considerado estado da arte para a tarefa, mas com menor custo computacional.

Palavras-chaves: Diarização de locutores. Segmentação de locutores. Sobreposição de fala. MFCC. i-vector. Características da matriz de afinidade.

ABSTRACT

Speaker diarization determines "who spoke and when?" in a conversation, detects the number of speakers and the intervals where each speaker is active. A speaker diarization system has at least four fundamental steps: voice activity detection, acoustic feature extraction, speaker segmentation, and speaker clustering. The segmentation step becomes a big challenge in spontaneous conversations scenario, because transitions between speakers occur frequently, andaround the transitions the speech from the speakers overlap. In this scenario, the detection of a speaker change is performed using short segments of speech, in order to avoid to have more than one speaker change per segment, so no change is missed. The state of the art i-vector represents the speech characteristics corresponding to the identity of the speaker, designed to discriminate people. However, its performance is affected by speech sample size, so that in the spontaneous talk scenario, its performance is comparable to traditional acoustic modeling methods using Gaussian mixture models. We propose the use of Mel Cepstral Affinity Features (MCAF), designed for short samples and the task of speaker segmentation. The proposed feature discriminates the different types of speech segments: homogeneous (segment containing a single speaker), heterogeneous (two speakers present without overlap) and overlapped (at least two speakers speaking simultaneously). A two sliding window method uses this discrimination to detect speaker changes. Experiments using the AMI corpora show that our proposed feature exhibits superior performance of F_1 score in 38% to traditional segmentation method using MFCC and GLR distance, and it is 15% superior to the i-vector-based method, which is considered state of the art for the task, but with lower computational cost.

Key-words: Speaker diarization. Speaker segmentation. Overlap speech. MFCC. i-vector. Features from affinity matrix.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Nas últimas duas décadas, a transcrição de locutores (tradução livre de *Speaker Diarization*) emergiu como uma área de pesquisa em processamento de voz de grande importância para as demais áreas de estudo da voz, como rastreamento de locutores e reconhecimento automático de locutores (EVANS et al., 2012). De maneira geral, a transcrição de áudio é o processo de dividir o sinal de áudio em segmentos e agrupá-los de acordo com cada tipo de áudio existente. Mais especificamente, na transcrição de locutores, a segmentação e agrupamento do áudio ocorre de acordo com a identidade dos locutores presentes. O problema de transcrição é popularmente denominado "quem falou e quando?"(ANGUERA; WOOTERS; PARDO, 2006b; EVANS et al., 2012; MIRO; BOZONNET, 2012), no qual não é conhecido *a priori* o número de locutores e nem a identidade dos mesmos. A solução do problema requer a detecção dos instantes em que cada locutor está presente (MOATTAR; HOMAYOUNPOUR, 2012). Os resultados gerados são as marcações dos locutores no áudio, assim como as marcações de trechos que não são interessantes para a aplicação (ruído, silêncio, etc.).

Um sistema de transcrição de locutores serve como etapa de pré processamento para maioria das aplicações de processamento de fala, entre elas:

- Busca e indexação de conteúdo falado: a busca recebe como entrada o texto e o áudio, ou então uma base de áudio na qual deve-se recuperar a informação. O texto pode conter palavras chave, ou temas de interesse. Enquanto que o áudio/base deve conter pessoas falando sobre temas mais diversos possíveis. O sistema de transcrição fornece as locuções de cada pessoa presente no áudio/base. Então, um sistema de reconhecimento automático de fala retorna os trechos das locuções que citam os assuntos de interesse.

- Reconhecimento e rastreamento automático de locutores: em palestras, entrevistas, reuniões e filmes, deseja-se escutar/visualizar o conteúdo apresentado por determinadas pessoas. O sistema de transcrição fornece as locuções de cada pessoa na gravação. Então, um sistema de reconhecimento de locutores identifica e filtra na gravação apenas as locuções geradas pelo indivíduo de interesse.

- Tradução de fala: A mídia gerada em diversos lugares do mundo possui também diversos idiomas. Para que esse conteúdo possa ser acessado por diversas pessoas, é preciso traduzir de fala para o idioma do ouvinte (voz para texto, ou voz para voz). O sistema de transcrição deve detectar os instantes em que cada locutor fala, independente do idioma. Após a detecção, um sistema de reconhecimento de idioma vai agrupar os locutores pelo idioma. E por fim, um sistema de tradução de multi-idiomas pode traduzir o que foi falado por cada grupo para a língua mãe do ouvinte apenas dos locutores presentes, para reconhecer o idioma e traduzir a fala de cada participante. O sistema de

transcrição fornece as informações dos locutores presentes, ignorando as marcações que não representam a fala.

Grande parte das aplicações de voz trabalha com gravações de áudio ou vídeo, as quais contêm mais de um locutor. Como exemplo, temos gravações de conversações telefônicas (*call centers* e teleconferências), noticiários de rádio e tv, *podcasts*, debates, shows, filmes, reuniões, conferências entre outras (REYNOLDS; TORRES-CARRASQUILLO, 2004). O desempenho dessas aplicações depende da precisão da segmentação e agrupamento do conteúdo falado por cada locutor. Dessa forma, a transcrição de locutores torna-se crucial como etapa de pré-processamento.

Na literatura (MIRO; BOZONNET, 2012; EVANS et al., 2012; MOATTAR; HOMAYOUNPOUR, 2012; KOTTI; MOSCHOU; KOTROPOULOS, 2008; REYNOLDS; TORRES-CARRASQUILLO, 2004), pode-se destacar duas principais abordagens para implementar sistemas de transcrição de locutores. As duas possuem inicialmente as etapas de detecção de atividade de voz e extração de características acústicas. Segundo Reynolds e Torres-Carrasquillo; Kotti, Moschou e Kotropoulos; Moattar e Homayounpour; Miro e Bozonnet; Evans et al., a mais popular delas é a abordagem *Bottom-up*, a qual possui uma etapa que divide os trechos com fala em segmentos homogêneos (um locutor ativo por segmento), e outra que agrupa os segmentos de cada locutor. A outra abordagem é a *Top-down*, a qual combina as atividades de agrupamento e segmentação em uma única etapa. Neste trabalho, pelo fato das características propostas utilizarem amostras curtas da fala, consideramos que a abordagem *Bottom-up* é mais adequada por permitir o processamento do sinal de voz em pequenas partes. Não consideramos viável o trabalho com a abordagem *Top-down*, pois esta trata o sinal de voz como um todo e procura segmentos de locutor de forma global, com amostras longas. As abordagens possuem quatro etapas fundamentais ao todo: detecção de atividade de voz/ruído/silêncio, extração de características acústicas, segmentação de locutores e agrupamento de locutores.

A detecção de atividade de voz é a etapa de pré-processamento responsável por identificar os trechos que contêm voz. Logo, os trechos do sinal que possuam ruído excessivo ou silêncio predominante são descartados da transcrição. Essa etapa é fundamental para evitar o processamento desnecessário de partes do sinal que não são importantes, diminuindo os erros de transcrição.

A etapa de extração de características é uma etapa básica em qualquer sistema de processamento de sinais acústicos. Ela é elaborada para cada tipo de problema em processamento de voz, com o objetivo de obter do áudio as informações mais relevantes para cada problema. Para o problema de transcrição de locutores, o tipo de característica escolhido é aquele que permite discriminar não somente os locutores, mas também voz de ruído, música e silêncio. O tipo de característica e o modo de extração é fundamental para as próximas etapas do sistema, pois todo o estudo e processamento das informações são feitos em cima dessas características.

Na abordagem *Bottom-up*, a etapa de segmentação de locutores é crucial (REYNOLDS; TORRES-CARRASQUILLO, 2004; KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUN-

POUR, 2012; EVANS et al., 2012), pois a perda de transições entre locutores faz com que existam mais de um locutor ativo por segmento, o que consequentemente polui os grupos gerados na etapa de agrupamento (mais de um locutor presente por grupo). A segmentação ideal é um importante passo para criação de grupos de locutores homogêneos (apenas um locutor por grupo). As abordagens dos métodos de segmentação são geralmente classificadas em duas categorias principais (KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012):

- Segmentação baseada em modelos: a detecção de mudanças é realizada de forma supervisionada, ou seja, é necessário conhecimento *a priori* de alguns locutores presentes. Os trechos de fala são segmentados uniformemente e cada segmento é classificado em uma/nenhuma das identidades modeladas *a priori*.
- Segmentação baseada em medidas de distância: a detecção de mudanças utiliza duas janelas deslizantes e uma métrica de distância/dissimilaridade. O algoritmo de detecção avalia se os dois segmentos amostrados pelas janelas pertencem ao mesmo locutor, caso o valor de distância calculado estiver acima de um limiar. A segmentação de locutores baseada em distância é a abordagem mais popular no desenvolvimento de sistemas de transcrição de locutores.

A etapa de agrupamento de locutores também é crucial para a transcrição, responsável por organizar os segmentos em grupos homogêneos onde cada grupo contém apenas um locutor. Esta etapa decide a qual grupo um determinado segmento deve se unir, qual segmento deve sair (reagrupamento) e quais grupos devem se unir ou se separar, dependendo da estratégia de agrupamento (KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012; MIRO; BOZONNET, 2012). Essa etapa atribui a cada segmento um identificador do grupo ao qual pertence, de modo que a quantidade de grupos no final do processo se aproxima ao número real de locutores presentes no sinal.

1.1 DEFINIÇÃO DO PROBLEMA E MOTIVAÇÃO

A atividade de segmentação torna-se difícil para conversações com uma alta frequência de transições entre locutores, cenário comum das conversações de estilo espontâneo/livre (debates, *brainstorms*, discussões e entrevistas sem programação/ensaio). É preciso trabalhar com segmentos curtos da fala, de modo que só exista uma transição entre locutores, e assim evitar perdas em cascata no processo de segmentação. A discriminação de locutores utilizando amostras curtas de fala ainda é considerada uma tarefa complexa (PODDAR; SAHIDULLAH; SAHA, 2018b; PODDAR; SAHIDULLAH; SAHA, 2018a).

Além dessa dificuldade, as conversações de estilo livre contêm uma considerável quantidade de segmentos com sobreposição de fala (dois ou mais locutores falando simultaneamente), por ser um comportamento natural humano de interação (EVANS et al., 2012; MOATTAR;

HOMAYOUNPOUR, 2012; MIRO; BOZONNET, 2012). Muitos trabalhos não consideram trechos com sobreposição de fala no processo de transcrição de locutores (EVANS et al., 2012; MOATTAR; HOMAYOUNPOUR, 2012; MIRO; BOZONNET, 2012; KOTTI; MOSCHOU; KOTROPOULOS, 2008; ANGUERA; WOOTERS; PARDO, 2006b). Características acústicas e técnicas de modelagem da fala são utilizados para descartar esses trechos do sinal antes da transcrição, ou simplesmente, usando o *ground-truth* da base de áudio para descartá-los. O descarte automático de trechos com sobreposição de fala demanda uma etapa adicional antes da segmentação, o que leva a um maior custo computacional do sistema.

A segmentação de locutores poderia considerar os trechos de sobreposição como transições de fala. Posteriormente, os segmentos com tamanho menor que um dado limiar arbitrário poderiam ser processados para identificar se contém mais de um locutor. A identificação pode ser feita através de modelos de aprendizagem de máquina treinados em uma base de dados com segmentos de tamanho até um dado limiar, com os dois tipos de fala. Esse processo pode reduzir o custo computacional por não ter que processar todos os trechos com fala.

Neste trabalho, consideramos que a solução para o problema de segmentação em conversas de estilo livre precisa de uma quantidade limitada de dados da fala, como uma forma de detectar o elevado número de transições de locutores e evitar a perda de transições que estejam presentes na amostra. Também consideramos a sobreposição de fala como um novo tipo de transição, já que esse é um evento esperado no cenário de conversações de estilo livre, sendo assim uma abordagem mais realista do que simplesmente ignorar que a sobreposição de fala existe. Isso nos motiva a definir uma nova abordagem para segmentação de locutores. Ao invés de capturar características únicas dos locutores utilizando segmentos curtos de fala, propomos novas características para discriminar 3 possíveis tipos de segmentos: homogêneo (apenas um locutor ativo), heterogêneo (dois locutores ativos em intervalos diferentes, com possível sobreposição de fala de forma não predominante) e *overlap* (dois ou mais locutores falando simultaneamente na maior parte do tempo). Neste trabalho, consideramos que os tipos de segmento possuem estilos diferentes de fala, e portanto, as características propostas precisam capturar as informações dos diferentes estilos para que a discriminação dos segmentos seja possível. O estilo para a fala é interpretado aqui como um conceito abstrato da forma individual de expressar a fala. Como exemplo, um segmento heterogêneo contém ao menos dois estilos de fala, um segmento com *overlap* tem a sobreposição de ao menos dois estilos e um segmento homogêneo possui apenas um estilo de fala. Utilizamos as características propostas para detectar as transições entre locutores e os instantes em que ocorrem sobreposição de fala. As caraterísiticas são extraídas diretamente dos coeficientes *mel* (Mel Frequency Cepstral Coefficients - MFCCs) estimados para cada segmento. A extração ocorre através da matriz de afinidade calculada em cada segmento, daí a origem do nome: *Mel Cepstral Affinity Features* (MCAFs).

1.2 OBJETIVOS

O trabalho desenvolvido nesta tese tem como objetivos principais:

- Formular uma nova abordagem de segmentação de locutores, sem o alto custo computacional de adaptação de modelos oriundos dos *frameworks* de verificação de locutores;
- Definir características que possam discriminar os tipos de segmento de fala (homogêneo, heterogêneo e *overlap*);
- Trabalhar com uma quantidade de dados limitada (segmentos curtos), para segmentar conversações de estilo livre;
- Segmentação com desempenho superior ou ao menos compatível com o estado da arte.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho é organizado como segue:

- Capítulo 2: traz uma visão geral dos sistemas de transcrição de locutores e dos desafios que ainda existem na área. Descreve o entendimento das técnicas clássicas e das técnicas estado da arte das características, modelagem da fala, métodos de segmentação de locutores e da etapa de agrupamento.
- Capítulo 3: discute as limitações dos modelos da fala para amostras curtas. Define a nova abordagem de segmentação, bem como as caraterísticas MCAF;
- Capítulo 4: descreve o protocolo de experimentos realizados, apresenta os resultados, bem como a análise e discussão dos mesmos;
- Capítulo 5: revisita o trabalho desenvolvido, ressaltando os objetivos atingidos, apresentando formas para continuidade do trabalho e novas aplicações.

2 SISTEMAS DE TRANSCRIÇÃO DE LOCUTORES

Dado o grande volume de mídia em voz produzido na atualidade, surge a necessidade de organização, busca e categorização desses dados através do conteúdo falado, pessoas envolvidas e pelo tipo de aplicação na qual a mídia foi produzida. Alguns exemplos de aplicações que produzem mídias de áudio são: conversas telefônicas (*Call Centers*, escutas telefônicas e gravações em geral), teleconferências, programas de TV e rádio, *podcasts*, filmes, shows e gravações de reuniões. Cada aplicação possui diferentes fontes de sons, como a voz de cada locutor presente, música, ruído ambiente, entre outros. Os sons são afetados de acordo com a forma e tipo de canal usado na captura do sinal digital do áudio. Podemos ter um microfone para cada locutor, ambientes de gravação distintos e meios distintos de captura (canal telefônico, distâncias diferentes entre o locutor e o microfone). O cenário ilustrado na Figura 1 retrata a diversidade de sons (nomes e números) e meios de captura de cada fonte de som (cores). As tecnologias de transcrição, busca, indexação e separação de áudio têm como objetivo viabilizar o acesso ao conteúdo do áudio, identificando os instantes de tempo nos quais estão presentes as fontes de som de interesse, locutores, assunto da conversa, palavras chave, entre outros.

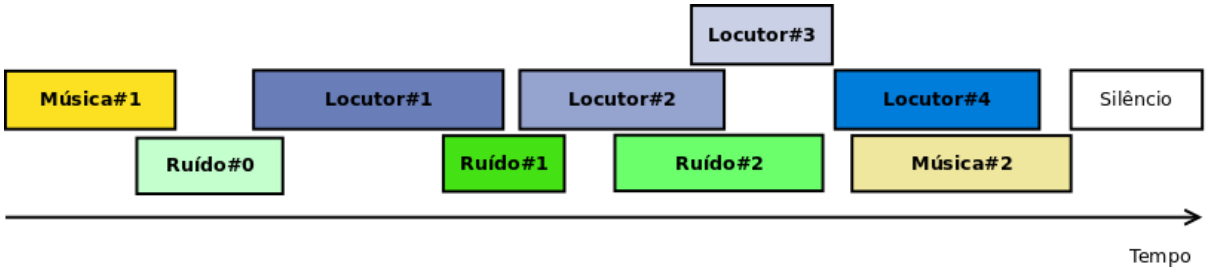

Figura 1 – Exemplo da variedade de tipos de sons em um sinal de áudio. Os retângulos na imagem são trechos de um tipo de som. A sobreposição de fontes sonoras é representada como retângulos posicionados em cima dos demais quando as posições no tempo coincidem. O nome e o número representam a fonte sonora que predomina em termos de intensidade. As cores distintas retratam a diferença que pode existir entre os canais de captura, ambientes, ou distâncias entre as fontes e o microfone.

A transcrição de locutores é uma aplicação com o objetivo de encontrar os trechos do áudio com voz, e neles registrar os instantes de tempo onde cada locutor está presente, sendo ainda possível identificar a quantidade de locutores presentes no áudio. A transcrição de locutores é geralmente referenciada como a tarefa "quem falou e quando?". A *priori* não é conhecido o número de locutores e nem a identidade dos mesmos. Esta aplicação é útil como uma etapa para pré-processamento de outras aplicações em voz, como indexação, rastreamento, verificação e identificação automática de locutores. Sendo crucial a precisão da transcrição para evitar erros nas outras aplicações.

A tarefa de transcrição de locutores recebeu atenção de vários grupos de pesquisa do mundo

inteiro. Muitos dos trabalhos que compõem o estado da arte em segmentação (DESPLANQUES; DEMUYNCK; MARTENS, 2015; GUPTA, 2015; YELLA; STOLCKE, 2015) e agrupamento de locutores (ROUVIER; BOUSQUET; FAVRE, 2015; SENOUSSAOUI et al., 2014; SHUM et al., 2013; KENNY, 2010) foram desenvolvidos através de competições organizadas pelo *National Institute of Standards and Technology* (NIST), nos Estados Unidos, conhecidas como *Rich Transcriptions Evaluations* (RT) (EVANS et al., 2012). Determinar o estado da arte para diarização é algo complexo, pois cada domínio de aplicação apresenta métodos, características acústicas e técnicas de modelagem adaptadas ao domínio. Segundo Tranter e Reynolds; Kotti, Moschou e Kotropoulos; Moattar e Homayounpour, as aplicações em transcrição de locutores se encaixam em três domínios principais:

- Conversas telefônicas: gravações de conversas telefônica em um único canal, entre duas ou mais pessoas.
- Redes de TV e rádio: programas de diversos tipos e conteúdos, das redes de TV e rádio, contendo usualmente intervalos comerciais e música, em um único canal.
- Reuniões ou conferências: diversas pessoas interagindo em salas ou auditórios, com gravações feitas com um ou mais microfones. No caso de múltiplos microfones, os dados de voz obtidos diferem em localização e se cada participante possui um microfone próprio. Aplicações com um único microfone são categorizadas como *Single Distant Microphone* (SDM), com múltiplos microfones são categorizadas como *Multiple Distant Microphones* (MDM). Nos casos em que cada participante tenha o seu microfone ou *headset*, as aplicações são categorizadas como *Individual Personal Microphone* (IPM) ou *Individual Headset Microphone* (IHM), respectivamente.

Alguns dos grupos de pesquisa que investem na solução de problemas de transcrição são apresentados a seguir, assim como é apresentado um resumo do histórico de pesquisas realizadas para cada domínio de aplicação exemplificado anteriormente. Através dos trabalhos com dados de gravações telefônicas, as aplicações das redes de TV e rádio se tornaram o foco principal de pesquisa que consta da metade da década de 1990 até os primeiros anos da década de 2000, onde o uso de sistemas de transcrição de locutores tinha como objetivo facilitar as anotações de conteúdo falado, nas transmissões de progamas de TV e rádio diários (MIRO; BOZONNET, 2012). O *Laboratoire d'Informatique d'Avignon* (LIA) e o *Communication Langagiere et Interaction Personne Systeme-Institut* (CLIPS) participaram das competições RT da NIST desde 2000, contribuindo com publicações dos seus sistemas de transcrição (MORARU et al., 2004; FREDOUILLE et al., 2004; MEIGNIER et al., 2004). Dois outros projetos bastante referenciados em transcrição de locutores são os sistemas desenvolvidos pela *Cambridge University Engineering Department* (CUED) (REYNOLDS; TORRES-CARRASQUILLO, 2004) e pelo *Massachusetts Institute of Tecnology-Lincon Labs* (MIT-LL) (SINHA et al., 2005). O interesse em dados de reuniões e conferências cresceu significativamente a partir de 2002,

com competições RT da NIST até 2009, com o desenvolvimento de sistemas de transcrição pelo *International Computer Science Institute* (ICSI, Berkley, Califórnia), que posteriormente culminaram no sistema de transcrição ICSI-SRI (ANGUERA; WOOTERS; PARDO, 2006b). Além das competições, foram lançados projetos de pesquisa como o *European Union Mulimodal Meeting Manager* (EUM4), o *Swiss Interface Multimodal Information Manager* (IM2), o *EU Augmented Multi-party Interaction* (AMI), *EU Augmented Multi-party Interaction with Distant Access* (AMIDA) e o *EU Computers in the Human Interaction Loop* (CHIL). Esses projetos têm como objetivo extrair conteúdo automaticamente das reuniões, disponibilizando-os para os participantes da reunião, ou para propósito de estudo futuro.

Para a construção de um sistema de transcrição de locutores que atenda às aplicações de forma generalizada, é ideal que ele funcione sem possuir nenhum conhecimento *a priori* da aplicação, ou das pessoas envolvidas. Esta é uma definição geral utilizada nas avaliações de transcrições obtidas de reuniões e redes de noticiário padronizadas pelo NIST (EVANS et al., 2012; TRANTER; REYNOLDS, 2006).

Segundo Reynolds e Torres-Carrasquillo; Tranter e Reynolds; Kotti, Moschou e Kotropoulos; Moattar e Homayounpour; Miro e Bozonnet; Evans et al., a tarefa de transcrição de locutores possui 4 etapas fundamentais. São as etapas de extração de características acústicas, detecção de atividade de voz/ruído/silêncio (etapa de pré processamento), a etapa de segmentação de locutores e a etapa e agrupamento de locutores. A segmentação e agrupamento também podem ser realizados em uma única etapa, como é explicado em Moattar e Homayounpour; Miro e Bozonnet; Evans et al.. Neste trabalho, essas etapas são tratadas de forma distinta. A segmentação de locutores é responsável por particionar os trechos com voz, criando segmentos contendo apenas um locutor em atividade. O agrupamento de locutores é responsável por juntar todos os segmentos de cada locutor no áudio. Essas etapas são consideradas as duas principais etapas de um sistema de transcrição de locutores, pois elas implementam a identificação de mudanças e coleta dos trechos falados para cada locutor presente.

Existem duas abordagens principais para o desenvolvimento de sistemas de transcrição de locutores (EVANS et al., 2012; MIRO; BOZONNET, 2012):

- *Bottom-up*: é a abordagem mais popular de acordo com Reynolds e Torres-Carrasquillo; Kotti, Moschou e Kotropoulos; Moattar e Homayounpour; Miro e Bozonnet; Evans et al.. A popularidade se dá pela facilidade de implementação, integração entre as etapas, possibilidade de acrescentar sub etapas de pré e pós processamento para segmentação e agrupamento, e pela possibilidade de realizar a ressegmentação e reagrupamento de forma independente e iterativa. O sinal de áudio é pré-processado por um *Voice Activity Detector* (VAD) para segmentar a fala no sinal. Os segmentos com fala são dados como entrada para o extrator de características. A etapa de segmentação utiliza as características para detectar todas as possíveis mudanças entre locutores em cada segmento de fala. Em seguida, o agrupamento hierárquico dos segmentos é realizado até formar um número mínimo de k locutores, ou a distância entre os grupos atinja um

estado estacionário (Figura 2);

- *Top-down*: O sinal de áudio é pré-processado por um VAD (*Voice Activity Detector*) para segmentar a fala no sinal. Os segmentos com fala são dados como entrada para o extrator de características. Estimam-se os grupos de locutores com um número inicial mínimo de locutores e iterativamente realiza-se a descoberta de novos grupos, seja pela divisão ou aglomeração dos grupos existentes. O critério de parada pode ser um número máximo k de locutores, ou se as distâncias entre grupos e *intra* grupo atingirem um estado estacionário (Figura 3).

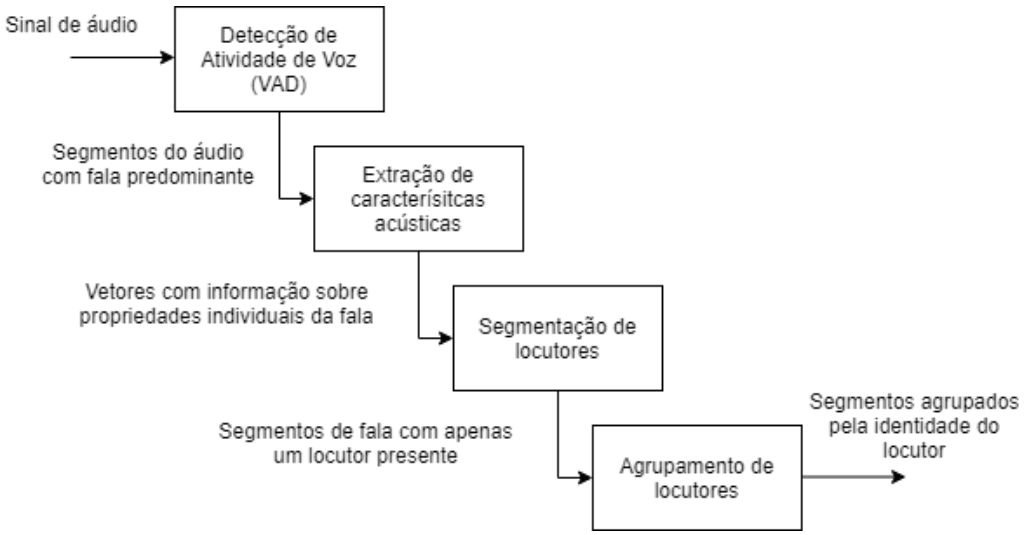

Figura 2 – Arquitetura *Bottom-up* genérica de um sistema de transcrição de locutores. O sinal de áudio é pré processado por um VAD (*Voice Activity Detector*) para segmentar a fala no sinal. Os segmentos com fala são dados como entrada para o extrator de características. A segmentação de locutores utiliza as características extraídas para detectar mudanças de locutor nos segmentos de fala. Por fim, os segmentos criados pelas mudanças de locutor são agrupados por identidade.

A Seção 2.1 revisa as principais técnicas e abordagens para a etapa de pré processamento. A Seção 2.2 traz uma revisão das características da fala mais utilizadas e recentemente propostas em transcrição de locutores. A Seção 2.3 explora os trabalhos realizados na tarefa de segmentação de locutores. A Seção 2.4 apresenta uma revisão das principais técnicas e abordagens para a tarefa de agrupamento de locutores. Na Seção 2.5, discutimos sobre as limitações das técnicas da literatura para a segmentação de locutores.

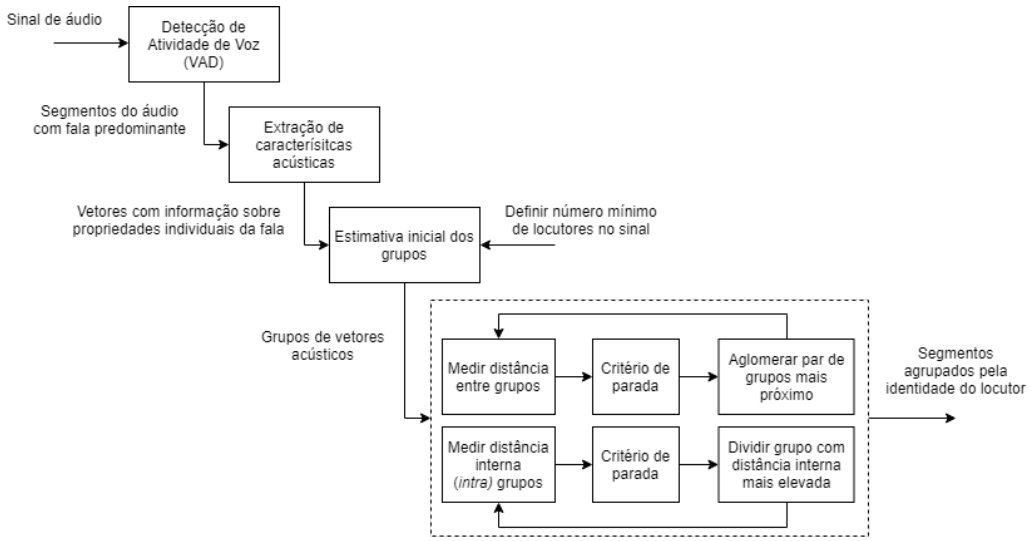

Figura 3 – Arquitetura *Top-down* genérica de um sistema de transcrição de locutores. O sinal de áudio é pré processado por um VAD (*Voice Activity Detector*) para segmentar a fala no sinal. Os segmentos com fala são dados como entrada para o extrator de características. Através de um número inicial mínimo de locutores, estimam-se os grupos de locutores e iterativamente realiza-se a descoberta de novos grupos, seja pela divisão ou aglomeração dos grupos existentes. O algoritmo retorna os segmentos agrupados por identidade quando um número máximo k de locutores é alcançado, ou se não for mais recomendável aglomerar ou dividir os grupos.

2.1 PRÉ-PROCESSAMENTO DO SINAL DE ÁUDIO

No contexto de transcrição de locutores, a detecção de atividade de fala (SAD), ou detecção de atividade de voz (VAD) é realizada como uma etapa de pré-processamento. Nessa etapa, ocorre a divisão do sinal de áudio entre os segmentos que contêm e os que não contêm voz. A precisão da segmentação da voz é crucial para o desempenho das outras etapas, pois o erro cometido nesta etapa é propagado para as demais. Segmentos que não contêm voz representam diversos tipos de sons, onde a quantidade de tipos podem variar de acordo com a aplicação processada: silêncio, música, ruído do ambiente, pessoas falando em segundo plano, risadas, aplausos, propagandas de TV e rádio, etc. Quando existe ruído ou música com alta intensidade (mais intenso que a intensidade da voz) no plano de fundo de um segmento contendo voz, o segmento será descartado. Caso a intensidade da música ou ruído seja inferior que o da voz, o segmento não será descartado. Portanto, um VAD baseado em detecção de energia abaixo da média, combinado com limiar de aceitação sinal ruído pode não ser uma solução eficiente para acaplicação estudada (MOATTAR; HOMAYOUNPOUR, 2012; MIRO; BOZONNET, 2012).

Sinha et al. utiliza um reconhecedor de fonemas para remover os trechos com silêncio do sinal. Em Tranter e Reynolds é utilizado um limiar ajustável de energia para a mesma tarefa. Zhou e Hansen utilizam um limiar de energia como estágio inicial e adiciona um estágio final com um modelo de reconhecimento de uma única palavra, para refinar os resultados.

As regiões que contêm ruído, música e sons ambiente provenientes da aplicação podem ser automaticamente detectadas e removidas (TRANTER; REYNOLDS, 2004; JOHNSON, 1999; KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012; MIRO; BOZONNET, 2012). A abordagem geral para a detecção de voz nesses sinais é a classificação por máxima verossimilhança com modelos de misturas Gaussianas (GMMs) treinadas com dados categorizados de acordo com a aplicação, o que inclui: voz, silêncio, voz sobre ruído, voz sobre música, voz sobre voz (*overlap speech*), música, entre outros (TRANTER; REYNOLDS, 2006). O sistema mais simples utiliza modelos de voz e não-voz (agrupa todas as outras categorias no grupo não-voz) (WOOTERS et al., 2004). Em Nguyen quatro modelos para voz são utilizados para possíveis combinações de gêneros ou largura de banda. É proposto algo similar em Sun et al., utilizando dois modelos iniciais, classificando os *frames* como voz ou não-voz. Esses frames são então utilizados para retreinar os modelos iterativamente através da máxima probabilidade *a posteriori* (*Maximum a posteriori Probability* - MAP). Ruído e música são explicitamente modelados em Zhu et al.; Reynolds e Torres-Carrasquillo; Gauvain, Lamel e Adda, enquanto Hain et al.; Sinha et al. utilizam modelos para voz capturada em canais de banda larga, banda curta música e voz sobre música.

Em Anguera, Wooters e Pardo, é utilizada inicialmente uma máquina de estados finita (*Finite State Machine* - FSM) para detectar regiões com voz e não-voz. As marcações iniciais são então utilizadas para treinar dois modelos GMM-HMM (*Hidden Markov Models* - HMM) de voz e não-voz. O sistema proposto segmenta e treina os modelos iterativamente até a verossimilhança geral parar de aumentar. Em Moattar e Homayounpour; Miro e Bozonnet são citados alguns trabalhos que utilizam uma abordagem híbrida para VAD, aplicando inicialmente um limiar de energia, e filtrando os resultados com modelos GMM ou GMM-HMM de voz e não-voz.

Para o domínio de conversações telefônicas, limiares de energia/espectro são tipicamente utilizados para desenvolver algoritmos VAD (MOATTAR; HOMAYOUNPOUR, 2012), mas a abordagem utilizando modelos GMM também tem alcançado resultados satisfatórios em aúdio gravado com um único microfone (TRANTER; REYNOLDS, 2004), ou múltiplos microfones (LILT; KUBALA, 2004). Para o domínio de gravações de reuniões e conferências, pode existir uma variedade de fontes de ruído, como o som de folhear papéis ou cadernos, risadas, bocejo, etc. Para esse tipo de aplicação, modelos GMM pré treinados para as classes voz e não-voz são preferencialmente utilizados para a tarefa VAD (ANGUERA; HERNANDO, 2005; MOATTAR; HOMAYOUNPOUR, 2012; MIRO; BOZONNET, 2012).

2.2 CARACTERÍSTICAS DA FALA

As características extraídas do sinal precisam trazer informações relevantes para a discriminação das pessoas presentes no áudio, permitindo aos sistemas de transcrição de locutores formas de identificá-las e realizar o agrupamento das mesmas de maneira otimizada. Podemos dividir as características da fala mais utilizadas em sistemas de transcrição de locutores em duas

categorias:

- Características acústicas de análise de tempo curto: engloba características como os coeficientes Mel Cepstrais (MFCC - *Mel Frequency Cepstral Coefficients*), PLP (*Perceptual Linear Predictive)* e LPCC (*Cepstral Linear Predictive Coding*) (ANGUERA; WOOTERS; PARDO, 2006b; KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012; EVANS et al., 2012; MIRO; BOZONNET, 2012);
- Caraterísticas de alto nível: são aquelas extraídas de distribuições de probabilidade modeladas a partir de características acústicas. Como exemplo temos os *speaker factors* e i-vector (*identity vector*) (CASTALDO et al., 2008; KENNY; REYNOLDS; CASTALDO, 2010; DEHAK; KENNY; DUMOUCHEL P., 2011; SHUM et al., 2011; SENOUSSAOUI et al., 2014; DESPLANQUES; DEMUYNCK; MARTENS, 2015; NERI et al., 2017).

2.2.1 Características de Tempo Curto

Nesse tipo de características, o sinal voz é dividido em *frames* de curta duração (20-30 milisegundos), assumindo que dentro desse pequeno intervalo de tempo as propriedades estacionárias do sinal de voz se conservam. Observando essas propriedades no domínio da frequência, é possível determinar padrões para caracterizar o sinal analisado. A transformação do sinal de áudio do domínio do tempo para o domínio da frequencia é feita pela transformada de Fourier. O domínio da frequência mostra a intensidade do sinal em diferentes valores de frequência, o que traz mais *insights* para análise do sinal. Por se tratar de um sinal digital, utilizamos a Transformada Discreta de Fourier (DFT). A *Fast Fourier Transform* (FFT) é uma implementação otimizada da DFT, com menor custo computacional e amplamente utilizada para estimar o espectro da fala (RABINER; JUANG, 1993; CAMPBELL, 1997). A FFT assume que o espectro contínuo obtido para um conjuto finito de amostras corresponde a um número inteiro de períodos de um sinal periódico. A amostragem do sinal de análise utilizando um janelamento abruto considera que a amostra no início e fim do conjunto se conectam para compor o próximo período. Porém, quando o sinal não é periódico ou se o conjunto não capturar um número inteiro de períodos, a amostra no início e fim do conjunto apresentarão descontinuidade. Essa descontinuidade aparece no espectro como componentes de alta frequência que não existem no espectro original do sinal. Esse fenômeno é conhecido como *spectral leakage*, aparentando um "vazamento"da energia de uma frequência para as demais frequências. Para minimizar esse fenômeno, utiliza-se a técnica de janelamento do sinal, a qual é conhecida como uma filtragem com filro do tipo passa baixa. O janelamento reduz a amplitude das descontinuidades nas bordas (amostras de início e fim) de cada conjunto de amostras do sinal. A técnica consiste em multiplicar o conjunto de amostras por uma janela de tamanho finito com amplitude que varia suave e gradualmente do valor máximo 1 no centro da janela, até o valor muito próximo de zero nas bordas. Dessa forma, a descontinuidade é menos aparente e como resultado, o espectro estimado possui menos ruído nas componentes de alta frequência.

Coeficientes Mel Cepstrais

A técnica de extração dos coeficientes MFCC foi proposta por Davis e Mermelstein. O MFCC faz uma análise de características espectrais de tempo curto, analisando o sinal de voz em intervalos de 10 a 30ms. Baseia-se no uso do espectro da voz alterado segundo a escala Mel, uma representação que procura se aproximar de características únicas presentes no ouvido humano. Os coeficientes mel-cepstrais surgiram devido aos estudos na área de psicoacústica (ciência que estuda a percepção auditiva humana), pois verificou-se que a percepção humana de frequências de tons puros ou de sinais de voz não seguem uma escala linear. Isso estimulou a criação de uma escala da seguinte forma: para cada tom com uma determinada frequência, medida em Hz, associa-se um valor medido em uma escala chamada Mel. Como referência, definiu-se através de experimentos que o mapeamento de uma frequência em Hz para a escala Mel é realizado da seguinte maneira no domínio da frequência:

$$mel(f) = 1127 \ln(1 + \frac{f}{700}). \tag{2.1}$$

Após a segmentação do sinal de voz em *frames* e tranformação para o domínio da frequência, são necessárias as seguintes etapas: aplicação do banco de filtros mel e transformada discreta do cosseno (DAVIS; MERMELSTEIN, 1980; KINNUNEN et al., 2007).

O espaçamento dos filtros é feito regularmente na escala Mel e não regularmente na escala linear, o que faz com que a aplicação do banco de filtros no espectro se pareça com uma conversão do mesmo da escala linear para a escala Mel. O espaçamento da escala Mel é apresentada na Equação 2.1. O espectro, quando submetido a um filtro triangular, tem as componentes que estão próximas ao centro deste filtro enfatizadas, enquanto as demais são atenuadas. Dessa forma, as frequências Mel são enfatizadas. A utilização desses filtros reduz a dimensão final das características e suaviza a magnitude do espectro do sinal (RABINER; JUANG, 1993).

Na etapa final, converte-se o logaritmo do espectro Mel de volta ao domínio do tempo. Como o logaritmo do espectro Mel fornece números reais, eles podem ser convertidos usando a transformada discreta do cosseno (*Discrete Cosine Transform - DCT*). Os valores dessas energias no domínio do tempo estão numa frequência chamada *quefrequency*, ou o domínio do tempo na escala Mel, sendo esses valores descorrelacionados, o que facilita a aplicação de modelos bayesianos. A DCT pode ser calculada da seguinte forma:

$$x[k] = \sum_{n=1}^{N} x[n] \cos[\frac{\pi}{N}(n + \frac{1}{2})l], \quad l = 0, ..., L \tag{2.2}$$

onde L é o número de coeficientes MFCC que são extraídos e N é o número de filtros triangulares utilizados.

A aplicação do banco de filtros no espectro e a aplicação da DCT irão reduzir a dimensionalidade das informações que existem no espectro em sua forma original obtida pela FFT. A redução é projetada para que as informações da voz humana permaneçam nas características

MFCC. O tamanho do banco de filtros e a quantidade de coeficientes da DCT são parâmetros que precisam ser determinados para cada domínio de aplicação de processamento de fala. Essas características são comumente utilizadas em transcrição de locutores (KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012). A quantidade de coeficientes não é um consenso na literatura, Tritschler e Gopinath; Chen et al.; Ajmera, Lathoud e McCowan; Cheng e Wang utilizam 24 coeficientes MFCC. Em Kotti, Moschou e Kotropoulos; Moattar e Homayounpour são citados diversos outros trabalhos que utilizam 12 coeficientes MFCC, e outros que utilizam a energia do espectro como um 13º coeficiente.

Coeficientes de Predição Linear

A predição linear (*Linear Prediction - LP*) (MAKHOUL, 1975) é uma alternativa ao MFCC para extração de características espectrais. A principal característica da LP é ser uma técnica simples e intuitiva de ser interpretada, tanto no domínio do tempo quanto no domínio da frequência. A ideia básica é modelar o processo de produção da voz. A modelagem estima um filtro linear, também denominado filtro de síntese, cujo objetivo é predizer valores futuros de um sinal de voz a partir de amostras passadas deste mesmo sinal. A predição em um instante t do sinal $s[t]$ é uma soma linear dos p instantes anteriores à t. A No domínio do tempo, este filtro é representado por:

$$\tilde{s}[n] = \sum_{k=1}^{p} a_k s[n-k], \tag{2.3}$$

onde $s[n]$ é o sinal de voz observado, a_k são os coeficientes de predição e $\tilde{s}[n]$ é o sinal predito. O erro da predição é obtido através de $e[n] = s[n] - \tilde{s}[n]$. O objetivo é encontrar os coeficientes a_k que gerem o menor erro de predição. A minimização do erro é realizada aplicando o algoritmo de *Levinson-Durbin* (MAMMONE; ZHANG; RAMACHANDRAN, 1996), baseado no algoritmo de aprendizagem por mínimos quadráticos (*Least-squares estimate*). Esse algoritmo é utilizado por ser computacionalmente tratável para encontrar a matriz inversa que corresponde à solução do sistema.

A partir dos coeficientes LPC, são extraídos os coeficientes cepstrais (*Linear Predictive Cepstral Coefficients - LPCCs*). A extração é realizada através da transformada inversa de Fourier do $\log$ do espectro de potência do sinal (HARRINGTON; CASSIDY, 2000).

Coeficientes de Predição Linear Perceptual

Os coeficientes PLP foram introduzidos por Hermansky, a partir da análise dos coeficientes de predição linear. Tem como objetivo aproximar a representação desses coeficientes da sensibilidade da audição humana para certas faixas de frequencia, estimadas em um ambiente de auditório. As propriedades da audição humana são simuladas com aproximações práticas, utilizando filtros não lineares e deformações no sinal de voz.

O *frame* do sinal de voz é levado para o domínio da frequência utilizando o janelamento e a transformada Discreta de Fourier (DFT). O espectro obtido pela transformada é então convertido para a escala Bark de frequência (Ω):

$$\Omega(\theta) = 6\ln(\frac{\theta}{1200\pi} + [(\frac{\theta}{1200\pi})^2 + 1^{0.5}]), \tag{2.4}$$

onde θ é a frequência angular do espectro em rad/s. O resultado dessa conversão é convolvido com o espectro do filtro de banda crítica $\Psi(\Omega)$:

$$\Psi(\Omega) = \begin{cases} 0 & \text{para } \Omega < -1.3 \\ 10^{2.5(\Omega+0.5)} & \text{para } -1.3 \leq \Omega \leq -0.5 \\ 1 & \text{para } -0.5 < \Omega < 0.5 \\ 10^{-(\Omega-0.5)} & \text{para } 0.5 \leq \Omega \leq 2.5 \\ 0 & \text{para } \Omega > 2.5. \end{cases} \tag{2.5}$$

Esse filtro de banda crítica simulado é a aproximação do que é conhecido a respeito dos filtros para sinais de voz em um ambiente de auditório. As caudas do filtro são truncadas em -40 dB. A convolução resulta na redução da resolução do espectro, permitindo o *down-sampling* do resultado da convolução.

Em seguida é realizada a etapa de pré-ênfase do espectro w com a resolução reduzida, utilizando a simulação da curva de igualdade de intensidade $E(w)$:

$$E(w) = \frac{(\Psi(\Omega)^2 + 56.8 \times 10^6)w^4}{(\Psi(\Omega)^2 + 6.3 \times 10^6)^2(\Psi(\Omega)^2 + 0.38 \times 10^9)}, \tag{2.6}$$

para sons em nível moderado, esta conversão é razoavelmente boa até a frequecia 5 kHz.

Após a pré ênfase, é realizada a raiz cúbica da compressão da amplitude do sinal:

$$\Phi[w] = (E(w)\Theta[\Omega])^{0.33}, \tag{2.7}$$

onde $\Theta[\Omega]$ é o espectro do sinal após a pré ênfase. Essa operação é uma aproximação para a lei de potência da capacidade auditiva e simula a relação não linear entre a intensidade do som e a intensidade percebida pela audição humana. Após essa etapa, o sinal é recuperado no domínio do tempo e os coeficientes PLP são extraídos da técnica de predição linear (LP).

Os coeficientes cepstrais PLP são extraídos através da transformada inversa de Fourier do $\log$ do espectro de potência do sinal. Essas características são utilizadas em Tranter e Reynolds na implementação da etapa de agrupamento, considerando o tempo de processamento da extração sem deteriorar a capacidade de discriminação entre grupos de locutores distintos.

Frequência Fundamental

Em Sun, para cada *frame*, considere $A[f]$ como a amplitude do espectro de uma amostra de sinal $a[t]$. Considere o F_0 e F_{max} como a frequência fundamental e a frequência máxima de

$a[t]$, respectivamente. Então, a soma das amplitudes das harmônicas é calculada como:

$$SH = \sum_{n=1}^{N} A[nF_0], \tag{2.8}$$

onde N é o número máximo de harmônicas contidas no espectro. Se restringirmos a busca pela frequência fundamental pelo intervalo [F_{min}, F_{max}], então $N = floor(\frac{F_{max}}{F_{min}})$. Assumindo que a frequência mais baixa das subharmônicas é a metade da F_0, a soma das amplitudes das subharmônicas é definida como:

$$SS = \sum_{n=1}^{N} A[(n - \frac{1}{2})F_0]. \tag{2.9}$$

As somas SH e SS são usadas para decompor os efeitos das harmônicas e sub-harmônicas e analisar se as sub-harmônicas são fortes o suficiente para serem candidatas à frequência fundamental. A frequência que apresentar o pico mais proeminente é considerada a F_0.

Em Yamaguchi, Yamashita e Matsunaga é proposto na etapa de segmentação de locutores o uso da energia, frequência fundamental (F_0), centróide da frequência de pico e a faixa da frequência de pico, e a adição de três novas características: a estabilidade temporal da potência do espectro, a forma (*shape*) do espectro e as similaridades do ruído branco. A F_0 é uma característica prosódica, ou seja, atua na modelagem do ritmo, entonação e estilo da fala.

Características Temporais

Em Soong e Rosenberg, afirma-se que as propriedades dinâmicas do espectro da voz contêm informações úteis para a discriminação de locutores. Essas propriedades são a velocidade e a aceleração da fala. Uma prática comum para incorporar essas propriedades é utilizar as derivadas de primeira e segunda ordem das características acústicas, conhecidas como coeficientes delta (Δ) e delta ao quadrado (Δ^2), respectivamente. Os coeficientes Δ em um instante t são computados como

$$d[t] = \frac{\sum_{k=1}^{p} k(c[t-k] - c[t+k])}{2\sum_{k=1}^{p} k^2} \tag{2.10}$$

onde $d[t]$ são os coeficientes, p é o intervalo de análise da dinâmica do espectro e $c[t]$ é o vetor das características acústicas no instante t. Os coeficientes Δ^2 são computados aplicando a (2.10) nos coeficientes $d[t]$. Os novos coeficientes são anexados aos vetores das características acústicas, aumentando a dimensionalidade: se os vetores de MFCCs possuem 13 coeficientes de base, a adição do Δ e Δ^2 geram um vetor final com 39 coeficientes.

Segundo Moattar e Homayounpour, com relação às características MFCC, não há um consenso na literatura sobre o uso das derivadas de primeira e segunda ordem. Em Delacourt e Wellekens, o uso da derivada de primeira ordem demostra deteriorar os resultados, enquanto que em Wu e Hsieh é demonstrado que melhora o desempenho geral da segmentação. Em Sinha et al.; Lu e Zhang; Kotti, Moschou e Kotropoulos as derivadas de primeira e/ou segunda ordem são utilizadas com sucesso no desempenho dos sistemas de transcrição.

A configuração sobre o uso e a quantidade de coefientes Δ e Δ^2 são parâmetros que precisam ser determinados durante os experimentos em cada domínio de aplicação e até mesmo em cada base de conversação (KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012). A configuração ideal pode ser determinada para cada etapa do sistema de transcrição de forma independente, de acordo com o desempenho em cada uma delas (KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012).

Normalização e Mapeamento das Características

Diferentes técnicas de normalização de características têm sido aplicadas em indexação de áudio (MOATTAR; HOMAYOUNPOUR, 2012). São técnicas como a normalização de média e variância cepstral, com filtro RASTA (REYNOLDS; TORRES-CARRASQUILLO, 2004) e a normalização de média cepstral (CMN) (ZAMALLOA et al., 2010).

O mapeamento de características (*feature warping*) considera que os efeitos de canal e ruído ambiente são aditivos no espaço de características Cepstrais e possuem uma distribuição Gaussiana de média μ_C e variância σ^2. Ao mapear as características da fala para uma Gaussiana de média 0, desvio padrão 1, os efeitos do canal são amortecidos pela subtração da média da distribuição original μ. Como resultado, os ruídos ambiente e outros eventos acústicos que não estejam relacionados ao locutor são atenuados (PELECANOS; SRIDHARAN, 2001; OUELLET; BOULIANNE; KENNY, 2005). Em Sinha et al.; Zhu et al., essas técnicas conseguiram melhorar o desempenho da transcrição de locutores para o domínio de redes de noticiários e reuniões gravadas, respectivamente. O mapeamento de características tem se mostrado mais eficiente do que técnicas de normalização de características para a tarefa de verificação de locutores (MOATTAR; HOMAYOUNPOUR, 2012).

2.2.2 Caracterísiticas de Alto Nível

Extração via Modelos de *Factor Analysis*

Dehak, Kenny e Dumouchel P. introduz um framework utilizando modelos de *factor analysis* para encontrar os fatores que explicam tanto a variabilidade de distribuições de milhares de locutores quanto a variabilidade dos diversos canais de captura de telefone. Eles definem o espaço de Variabilidade Total (TV) contendo simultaneamente informações de variabilidade dos locutores e do canal. Esse espaço é derivado do espaço de super vetores (*super vectors*) obtidos de um modelo *Universal Background Model* (UBM). O UBM é um modelo GMM que foi treinado a partir dos vetores de características MFCC extraídos de uma base vasta de locutores e de canais de captura. Para cada locução nessa base, extraem-se os vetores MFCC e estima-se um *super vector*, através da concatenação das médias da GMM adaptada aos dados via o algoritmo de *Maximum a Posteriori* (MAP). Utilizando *factor analysis*, podemos definir um *super vector* $\mathbf{m}_l$ da locução l através do espaço de Variabilidade Total (TV), como

$$\mathbf{m}_l = \mathbf{m} + T\mathbf{w} \tag{2.11}$$

onde **m** é o *super vector* obtido do modelo UBM, T é o espaço de Variabilidade Total, representado como uma matriz retangular de rank baixo, and **w** é um vetor aleatório de distribuição normal, chamado de i-vector. Os componentes desse vetor são conhecidos como *total factors*. Eles representam a locução no espaço de variabilidade de locutores e canal. A matriz T pode ser estimada utilizando o algoritmo de *Expectation and Maximization* (EM) descrito em Kenny e Dumouchel.

Extração via *Deep Learning*

Em Chen e Salman é proposa uma arquitetura de aprendizagem profunda (*Deep Learning*) para extrair características específicas do locutor a partir de características acústicas, como o MFCC. Essas novas caracaterísticas foram avaliadas na tarefa de segmentação de locutores, demonstrando desempenho superior que o método de segmentação tradicional BIC. Em Gupta, *Deep Neural Networks* (DNNs) são utilizadas para a extração de características de segmentos contendo mudanças de locutor. A rede é utilizada no reconhecimento dos trechos e identifica o ponto de mudança de locutor. As novas características são extraídas a partir das características acústicas TRAP e MFCC. Já em Ma e Bao, DNNs esparsas são utilizadas para extrair características específicas a partir dos *super vectors* obtidos do modelo GMM, adaptados a partir do modelo de voz *Universal Background Model* (UBM) independente de locutor. Ma e Bao demonstra resultados melhores que o de Chen e Salman.

2.3 SEGMENTAÇÃO DE LOCUTORES

A tarefa de segmentação de locutores também é chamada de detecção de mudança de locutor e tem uma definição muito próxima da detecção de mudança acústica. Dado um sinal de voz/áudio a segmentação de locutor localiza os instantes de tempo onde ocorre uma mudança de locutores. De forma mais generalizada, a detecção de mudança acústica localiza os instantes onde existe uma mudança acústica no áudio, que pode ser uma mudança de voz para silêncio, música para voz, ruído para voz entre outras. Essa tarefa é crucial para a transcrição de locutores, pois a segmentação visa separar os trechos de cada locutor de forma homogênea para que os grupos formados no final não contenham fragmentos de mais de um locutor por grupo.

As abordagens de segmentação de áudio geralmente são classificadas em duas categorias principais:

- Segmentação baseada em distância: uma métrica de distância é definida para medir dois segmentos adjacentes de fala, depois um algoritmo de detecção de mudança é definido a partir dessa métrica para avaliar se ambos os segmentos pertencem ao mesmo locutor (REYNOLDS; TORRES-CARRASQUILLO, 2004; TRANTER; REYNOLDS, 2006; KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012; EVANS et al., 2012; MIRO; BOZONNET, 2012);

- Segmentação baseada em modelo: detecta mudanças de locutor de forma supervisionada, ou seja, é necessário algum conhecimento *a priori* sobre os locutores, ou sobre algum outro padrão que seja útil para detectar mudanças (KARTIK; SATISH; SEKHAR, 2005; LIN et al., 2007; NERI et al., 2013; GUPTA, 2015).

As métricas mais comuns para avaliação de desempenho da segmentação são:

- *Recall*: percentual de mudanças reais detectadas

$$RCL = \frac{\text{Número de mudanças reais detectadas}}{\text{Número de mudanças reais}} \times 100\%.$$

- Precisão: percentual de mudanças detectadas que são de fato mudanças reais

$$PRC = \frac{\text{Número de mudanças reais detectadas}}{\text{Número de mudanças detectadas}} \times 100\%.$$

- F-*score*: relação entre as métricas PRC e RCL, definida como a média harmônica

$$F = \frac{2PRCRCL}{PRC + RCL}.$$

- Taxa de perda (*Miss Detection Rate* - MDR): percentual de mudanças reais perdidas

$$MDR = \frac{\text{Número de mudanças reais perdidas}}{\text{Número de mudanças reais}} \times 100\%.$$

- Taxa de falsos alarmes (*False Alarm Rate*) - FAR): percentual de mudanças detectadas que não são mudanças reais

$$FAR = \frac{\text{Número de mudanças detectadas que não são reais}}{\text{Número de mudanças reais}} \times 100\%.$$

Na Seção 2.3.1, as abordagens baseadas em distância são revisadas. Na Seção 2.3.2 as abordagens baseadas em modelo serão descritas. Na Seção 2.3.3 falamos sobre trabalhos recentes na área de *Deep Learning*, utilizando redes neurais artificiais.

2.3.1 Abordagem baseada em distâncias

Os algoritmos de segmentação de locutor baseados em distância são os mais populares na construção de sistemas de transcrição de locutores (ANGUERA; WOOTERS; PARDO, 2006b; KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012; MIRO; BOZONNET, 2012). Em geral, eles utilizam janelas móveis e uma métrica para avaliar se as características da fala extraídas de cada janela pertencem ao mesmo locutor. Um valor da métrica é calculado para cada passo das janelas, iterativamente até a janela alcançar o final do segmento de fala. Os valores computados formam uma curva de distância. Define-se algum critério de análise dessa curva para definir a relevância de valores de pico. As possíveis mudanças de locutor são consideradas como os instantes de tempo desses valores de pico mais relevantes.

As abordagens de segmentação de locutor utilizando medidas de distância iniciaram-se com Chen e Gopalakrishnan propondo um algoritmo que detecta múltiplos pontos de mudança em duas etapas, e depois Tritschler e Gopinath; Sivakumaran, Fortuna e Ariyaeeinia; Cheng e Wang; Lu e Zhang; Cettolo e Vescovi; Vescovi, Cettolo e Rizzi seguiram a mesma ideia com algoritmos de um ou dois passos. Todos eles propuseram um sistema usando uma janela de tempo que cresce de acordo com uma variável de tamanho de análise de segmentos para iterativamente achar os pontos de mudança. Em Tritschler e Gopinath são propostas algumas formas para tornar o algoritmo de detecção de múltiplas mudanças mais rápido e com foco voltado para detecção de mudanças de locutor em segmentos mais curtos ($t < 5$ segundos).

Revisaremos na Subseção 2.3.1.1 as distâncias mais adotadas na literatura. Detalharemos por primeiro as medidas utilizadas para características acústicas da fala e em seguinte apresentamos as distâncias utilizadas para características de alto nível. Na Subseção 2.3.1.2 apresentamos os algoritmos de janelas móveis para a detecção de mudança de locutor.

2.3.1.1 Medidas de Distância

Bayesian Information Criterion

O *Bayesian Information Criterion* (BIC) é utilizado em modelos derivados de características acústicas. Provavelmente é a medida mais usada tanto em segmentação quanto em agrupamento de locutores, pela sua simplicidade e eficiência. O BIC é um critério de verossimilhança com penalidade pela complexidade do modelo e foi introduzido por Schwarz e em Schwarz como um critério de seleção de modelos. Dado um conjunto de vetores de características acústicas $\mathbf{X}$ e seu modelo probabilístico $\mathcal{M}$, o valor BIC é calculado pelo log da verossimilhança de $\mathbf{X}$ em relação a $\mathcal{M}$, determinado como

$$\text{BIC}(\mathcal{M}) = \log \mathcal{L}(\mathbf{X}_i, \mathcal{M}) - \lambda \frac{1}{2} \#(\mathcal{M}) \log(N). \tag{2.12}$$

Sendo $\log \mathcal{L}(\mathbf{X}, \mathcal{M})$ o log da verossimilhança do conjunto de vetores em relação ao modelo $\mathcal{M}$, λ é um parâmetro de penalidade que pode ser determinado de forma livre, N é o tamanho do conjunto e $\#(\mathcal{M})$, o número de parâmetros livres para estimar $\mathcal{M}$.

Para a segmentação de locutores, BIC foi aplicado em (CHEN; GOPALAKRISHNAN, 1998; CHEN; GOPALAKRISHNAN, 1998; CHEN et al., 2002) em um algoritmo que detectava mudanças de locutor utilizando apenas uma janela móvel de tamanho variável. A janela amostrava um segmento das características da fala, $\mathbf{X}$, e dividia a amostra em duas partições, $\mathbf{X}_i$ e $\mathbf{X}_j$. A modelagem dos dados é feita por uma distribuição normal com a matriz de covariância completa. Duas hipóteses são estabelecidas para avaliar se existe uma mudança de locutor no segmento:

- Hipótese nula, H_0: afirma que as partições são oriundas do mesmo locutor. Ela é definida como

$$H_0 : \mathbf{X} = \mathbf{X}_i \cup \mathbf{X}_j \sim \mathcal{N}(\mu, \Sigma) \tag{2.13}$$

- Hipótese alternativa, H_1: afirma que as partições são oriundas de locutores diferentes.

$$H_1 : \mathbf{X}_i \sim \mathcal{N}(\mu_i, \Sigma_i), \mathbf{X}_j \sim \mathcal{N}(\mu_j, \Sigma_j) \tag{2.14}$$

Uma forma de avaliação das hipóteses é o teste da razão da verossimilhança entre H_0 e H_1

$$\text{LR} = \frac{\mathcal{L}(\mathbf{X}, \mathcal{M})}{\mathcal{L}_i(\mathbf{X}_i, \mathcal{M}_i)\mathcal{L}_j(\mathbf{X}_j, \mathcal{M}_j)}, \tag{2.15}$$

porém, essa forma não considera que o tamanho do conjunto de dados utilizados para estimar $\mathcal{M}_i = \mathcal{N}(\mu_i, \Sigma_i)$ e $\mathcal{M}_j = \mathcal{N}(\mu_j, \Sigma_j)$ é menor que o tamanho utilizado para estimar $\mathcal{M} = \mathcal{N}(\mu, \Sigma)$. O ΔBIC utiliza a informação do teste de verossimelhança e adiciona como penalidade a complexidade do modelo $\mathbf{X}$:

$$\Delta\text{BIC} = -log(\text{LR}) - \gamma = BIC(H_1, \mathbf{X}) - BIC(H_0, \mathbf{X}), \tag{2.16}$$

sendo γ a complexidade de $\mathcal{M}$

$$\gamma = \frac{1}{2}\lambda(d + \frac{1}{2}(d(d+1))\log(N) \tag{2.17}$$

onde d é a dimensão do vetor de características acústicas e N a quantidade de vetores no segmento $\mathbf{X}$. A hipótese H_1 só será verdadeira se $\Delta BIC > 0$.

Considerando que os modelos são distribuições Gaussianas de apenas um componente, a (2.16) pode ser reescrita através de uma aproximação:

$$\Delta BIC = \frac{N}{2}\log|\Sigma_{\mathbf{X}}| - \frac{N_i}{2}\log|\Sigma_{\mathbf{X}_i}| - \frac{N_j}{2}\log|\Sigma_{\mathbf{X}_j}| - \gamma, \tag{2.18}$$

onde N, N_i e N_j são o número de vetores de $\mathbf{X}$, $\mathbf{X}_i$ e $\mathbf{X}_j$, respectivamente, e $\Sigma_{\mathbf{X}}$, $\Sigma_{\mathbf{X}_i}$ e $\Sigma_{\mathbf{X}_j}$ são as matrizes completas de covariância de $\mathbf{X}$, $\mathbf{X}_i$ e $\mathbf{X}_j$.

Sivakumaran, Fortuna e Ariyaeeinia; Cheng e Wang; Lu e Zhang; Cettolo e Vescovi; Vescovi, Cettolo e Rizzi propuseram maneiras mais rápidas de calcular as médias e variâncias dos modelos gaussianos utilizados, reescrevendo o BIC para utilizá-lo com misturas gaussianas. Mais recentemente, Cheng, Wang e Fu propuseram um método que utiliza o BIC com um algoritmo com estratégia de dividir para conquistar, o cálculo do BIC é realizado recursivamente em um segmento com voz, até um tamanho mínimo do segmento ser alcançado. Em Chen e Salman a distância BIC é utilizada como distância utilizando características da fala extraídas de um modelo de *deep learning*. Essas novas caracaterísticas demonstram desempenho superior ao método de segmentação tradicional BIC utilizando MFCC.

Generalized Likelihood Ratio

A *Log Likelihood Ratio* (LLR) também é utilizada para avaliar modelos derivados de características acústicas. Ela foi inicialmente proposta para a detecção de mudança acústica por Willsky e Jones; Appel e Brandt. Trata-se da razão entre duas hipóteses H_0, H_1, as mesmas

hipóteses das Equações 2.13 e 2.14. São dados os segmentos $\mathbf{X}$, $\mathbf{X}_i$, $\mathbf{X}_j$, representados pelas distribuições Gaussianas $\mathcal{M}$, $\mathcal{M}_i$ e $\mathcal{M}_j$, respectivamente. Da Equação (2.15), é determinada a distância como

$$D(\mathbf{X}_i, \mathbf{X}_j) = -\log(LR(\mathbf{X}_i, \mathbf{X}_j)),$$

que, uma vez definido um limiar apropriado, pode decidir se dois segmentos pertencem ao mesmo locutor ou não. O LLR não conhece *a priori* as funções de distribuição de probabilidade dos modelos, elas devem ser estimadas a partir dos dados em cada segmento. Na segmentação de locutor, a distância LLR é geralmente usada com dois segmentos adjacentes de mesmo tamanho que são deslocados através do sinal, e o limiar é pré fixado ou então dinamicamente adaptado.

Em Bonastre et al., o LLR é usado para segmentar o sinal em mudanças de locutor com uma única etapa de processamento para o rastreamento de locutores. O limiar é determinado com o objetivo de aumentar o número de detecções de mudança de locutor para diminuir a perda das mudanças, mas com o custo de aumentar os falsos alarmes.

O algoritmo que mais representa o uso do LLR para segmentação de locutor é o DISTBIC (DELACOURT; KRYZE; WELLEKENS, 1999; DELACOURT; WELLEKENS, 2000), onde o LLR é proposto como uma primeira etapa de um algoritmo de segmentação de duas etapas de processamento (usando o BIC como a segunda medida).

Kullback-Leibler Divergence

A divergência de Kullback-Leibler (KL) e a distância KL2 (SIEGLER et al., 1997; HUNG; WANG; LEE, 2000) também avaliam modelos derivados de características acústicas. Elas são conhecidas pelo bom desempenho computacional e resultados de segmentação semelhantes ao LLR. Dados dois conjuntos de vetores de características acústicas $\mathbf{X}$ e $\mathbf{Y}$, a divergência KL é definida como

$$KL(\mathbf{X}; \mathbf{Y}) = E_{\mathbf{X}}(\log(\frac{P_{\mathbf{X}}}{P_{\mathbf{Y}}})), \tag{2.19}$$

onde $E_{\mathbf{X}}$ é o valor esperado com respeito a função de distribuição de probabilidade de $\mathbf{X}$. Quando as duas distribuições podem ser aproximadas para a Gaussiana $\mathcal{N}(\mu, \Sigma)$, a seguinte expressão é obtida (CAMPBELL, 1997):

$$KL(\mathbf{X}; \mathbf{Y}) = \frac{1}{2} tr[(\Sigma_{\mathbf{X}} - \Sigma_{\mathbf{X}\mathbf{Y}})(\Sigma_{\mathbf{X}}^{-1} - \Sigma_{\mathbf{Y}}^{-1})] + \frac{1}{2} tr[(\Sigma_{\mathbf{X}}^{-1} - \Sigma_{\mathbf{Y}}^{-1})(\mu_{\mathbf{X}} - \mu_{\mathbf{Y}})(\mu_{\mathbf{X}} - \mu_{\mathbf{Y}})^T], \tag{2.20}$$

sendo tr a operação matricial de traço de uma matriz, a qual é a soma dos elementos da diagonal principal. A distância KL2 pode ser obtida simetrizando a KL, como segue:

$$KL2(\mathbf{X}; \mathbf{Y}) = KL(\mathbf{X}; \mathbf{Y}) + KL(\mathbf{Y}; \mathbf{X}). \tag{2.21}$$

Em Delacourt e Wellekens; Zochova e Radova, a distância KL2 é usada como a primeira de duas etapas para o algoritmo de detecção de mudança de locutor. Em Hung, Wang e Lee,

os vetores de MFCC são inicialmente processados via o método *Principal Component Analisys* (PCA) para redução de dimensionalidade. Então, a divergência KL é utilizada na detecção de mudanças de locutor.

Distâncias Geométricas

Para as características da fala de alto nível como os speaker factors e i-vectors, utilizam-se distâncias de interpretação geométrica como a Euclidiana e do cosseno. Os modelos extraídos para o locutor utilizando *Factor Analysis*, são vetores em um espaço linear. Nos trabalhos anteriores, vimos que os modelos utilizados para representar os locutores são funções de distribuição de probabilidades com máxima distribuição a posteriori (MAP) (CASTALDO et al., 2008; SHUM et al., 2011; SENOUSSAOUI et al., 2014; DESPLANQUES; DEMUYNCK; MARTENS, 2015; NERI et al., 2017). Dado dois segmentos de características acústicas $\mathbf{X}_i$ e $\mathbf{X}_j$, estima-se um vetor de representação do locutor para cada segmento. Sejam w_1 e w_2 os vetores de alto nível computados para $\mathbf{X}_i$ e $\mathbf{X}_j$, respectivamente. A distância Euclidiana entre esses vetores é definida como:

$$\text{dist}_{\text{euc}}(w_1, w_2) = \sqrt{\sum_{k=1}^{d}(w_1^k - w_2^k)^2}, \tag{2.22}$$

onde d é a dimensão do espaço vetorial de w_1 e w_2. A distância do cosseno entre esses vetores é definida como:

$$\text{dist}_{\cos}(w_1, w_2) = 1 - \frac{w_1^\top w_2}{||w_1|| \cdot ||w_2||}, \tag{2.23}$$

onde $||\cdot||$ é a norma de um vetor.

2.3.1.2 Algoritmos de Janelas Móveis

As estratégias mais populares para segmentação baseada em distância são (ANGUERA; WOOTERS; PARDO, 2006b; KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012):

- Janela de tamanho variável: nomeada de *Window Growing* (WinGrow) em Cheng, Wang e Fu, o algoritmo utiliza uma janela crescente e móvel para detecção de mudanças de locutor. Esse algoritmo é o mais popular dentre os que utilizam o critério BIC como medida de distância (CHEN; GOPALAKRISHNAN, 1998; TRITSCHLER; GOPINATH, 1999; SIVAKUMARAN; FORTUNA; ARIYAEEINIA, 2001; CHENG; WANG, 2003; LU; ZHANG, 2002; CETTOLO; VESCOVI, 2003; VESCOVI; CETTOLO; RIZZI, 2003; ANGUERA; HERNANDO, 2005). O objetivo é detectar se existe mudança de locutor dentro da janela. Para isso, o segmento amostrado pela janela é particionado dois conjuntos de vetores de caracaterísticas e o critério BIC avalia se existe mudança. (CHEN; GOPALAKRISHNAN, 1998; CETTOLO; VESCOVI, 2003) propõe que uma curva de valores BIC deve ser calculada para essa janela, a fim de determinar qual o instante t que, ao particionar os dados, gera o maior valor BIC. Caso não exista mudança, a janela aumenta de tamanho e o processo se repete.

De início, a janela assume um tamanho de N_{ini} vetores de características. Caso haja mudança, a janela com tamanho N_{ini} é movimentada para a localização da mudança e o algoritmo de detecção se repete para localizar a próxima mudança. Caso não haja mudança, a janela cresce de tamanho por N_g vetores e o algoritmo de detecção se repete para esse novo segmento. Caso a janela alcance um tamanho máximo de N_{max} vetores, ela será movimentada por N_s vetores até que o algoritmo de detecção encontre uma mudança. O algoritmo termina quando a janela alcança o fim segmento de fala. Os demais segmentos de fala serão então processados para que o processamento do sinal de áudio termine.

- *FixSlid*: utiliza duas janelas adjacentes de tamanho fixo de N_{fix} vetores acústicos que deslizam no tempo. Esse é o algoritmo mais utilizado para detecção de mudanças de locutor Anguera, Wooters e Pardo; Kotti, Moschou e Kotropoulos; Castaldo et al.; Miro e Bozonnet; Moattar e Homayounpour; Desplanques, Demuynck e Martens; Neri et al.. As janelas se movimentam com um passo de tamanho N_s vetores até alcançarem o fim do segmento de fala. Para cada passo, um valor de distância é calculado entre os dois segmentos amostrados pelas janelas. A curva de distância é computada com os valores calculados durante a movimentação. A curva é analisada segundo algum critério que define quais são os valores de pico de distância mais relevantes na curva. As localizações no tempo desses valores serão consideradas como possíveis mudanças de locutor. Os demais segmentos de fala serão então processados para que o processamento do sinal de áudio termine. Essa estratégia, utilizando o i-vector como representação do locutor e distâncias geométricas para detecção de mudanças é considerada estado da arte para segmentação de locutores (DESPLANQUES; DEMUYNCK; MARTENS, 2015).

- DISTBIC: esse algoritmo propõe uma segunda etapa após a execução do algoritmo *FixSlid*, para validar os segmentos gerados. Ele avalia todos os pares de segmentos adjacentes utilizando uma métrica de distância diferente da etapa inicial e mais critério para decidir se os segmentos pertencem de fato a locutores distintos (DELACOURT; KRYZE; WELLEKENS, 1999; DELACOURT; WELLEKENS, 2000; ZHOU; HANSEN, 2000; KIM; ERTELT; SIKORA, 2005; TRANTER; REYNOLDS, 2004; LU; ZHANG, 2002). Em Delacourt, Kryze e Wellekens; Delacourt e Wellekens, o LLR é utilizado como distância no *FixSlid*, e na segunda etapa utiliza-se o BIC e o critério $\Delta\text{BIC} > 0$ para validar se os segmentos pertencem a locutores distintos. Outras distâncias são utlizadas no *FixSlid*, como a *Hotelling's* T^2 em Kim, Ertelt e Sikora, e Lu e Zhang propõem a utilização da divergência KL2 para o *FixSlid* e o BIC para a segunda etapa.

2.3.2 Abordagem baseada em modelos

Modelos iniciais, como GMMs (REYNOLDS, 1995), são treinados para um conjunto de tipos de sons (voz em telefone, voz masculina e feminina, música, silêncio, sobreposição de tipos de

sons, entre outros) a partir de um conjunto de dados de treinamento. O segmento de fala é processado utilizando uma janela para extração das características acústicas. As características são dadas como entrada para os modelos treinados e a classificação da amostra é o tipo do modelo que obteve a máxima verossimilhança com os dados (GAUVAIN; LAMEL; ADDA, 1998; KEMP et al., 2000; BAKIS; CHEN; GOPALAKRISHNAN, 1997; SANKAR et al., 1998; KUBALA et al., 1997). Utilizando os modelos gerados de cada locutor, e considerando cada locutor como um estado, um modelo oculto de Markov (*Hidden Markov Model* - HMM) é estimado para modelar as transições entre locutores. A localização dessas transições podem ser estimadas utilizando o algoritmo de decodificação Viterbi (TRANTER; REYNOLDS, 2006; KOTTI; MOSCHOU; KOTROPOULOS, 2008).

Em Ajmera, Bourlard e Lapidot; Ajmera e Wooters, uma decodificação iterativa seguindo a abordagem *bottom-up* (começando com um número alto de locutores e então eliminando-os até obter um número ótimo) é proposta. Em Meignier, Bonastre e Igounet; Miró e Pericas, é construída uma abordagem *top-down* (começando com um segmento e adicionando segmentos extras até um número desejado de segmentos serem alcançados). Em Meignier et al., é analisado o uso de sistemas evolutivos onde modelos pré treinados também são usados na modelagem de sons ambientes, demonstrando que, no geral, quanto mais informações *a priori* for possível obter do áudio analisado, melhores serão os resultados alcançados. Em Neri et al., é proposta uma arquitetura de segmentação de locutores que combina as respostas de segmentação criadas de diversas características acústicas (MFCC, LPC, LPCC e F_0) por uma rede neural (RNA), que demostra ter melhores resultados que a segmentação por um único tipo de característica acústica.

2.3.3 Abordagems utilizando *Deep Learning*

Em Chen e Salman é proposta uma arquitetura de aprendizagem profunda (*Deep Learning*) para extrair características específicas do locutor a partir de características acústicas, como o MFCC. Essas novas caracaterísticas foram avaliadas na tarefa de segmentação de locutores, demonstrando desempenho superior ao método de segmentação tradicional BIC. Em Gupta, a utilização de DNNs (*Deep Neural Networks*) é proposta para extração de características de segmentos contendo mudanças de locutor. A rede é utilizada no reconhecimento dos trechos e identifica o ponto de mudança de locutor. As novas características são extraídas a partir das características acústicas TRAP e MFCC. Já em Ma e Bao, DNNs esparsas são utilizadas para extrair características específicas a partir dos GMM *super vectors*, adaptados a partir do modelo de voz UBM treinado em uma base de locutores externa. Essa abordagem demonstra resultados melhores que a abordagem proposta por Chen e Salman.

2.4 AGRUPAMENTO DE LOCUTORES

A etapa de agrupamento de locutores é responsável por aglomerar os segmentos obtidos na etapa de segmentação que pertencerem ao mesmo locutor. Essa etapa finaliza o processo de transcrição de locutores no sinal de áudio. Ele tem o objetivo de criar grupos homogêneos a partir dos segmentos de locutores.

Esta seção traz uma revisão das técnicas e algoritmos de agrupamento que não utilizam a informação real sobre o número de locutores presentes no áudio (heurísticas são adotadas para inferir esse número), ou sobre suas identidades. Por um lado é crucial termos dados de transcrição que mostram com uma boa precisão o número de locutores presentes, o que por consequência prejudica a homogeneidade dos grupos. Por outro lado, para utilizar as respostas do agrupamento em sistemas de reconhecimento de locutor, é importante ter grupos com a maior homogeneidade possível, a partir dos quais os modelos de locutor são estimados sem o ruído de mais de uma identidade. A ressalva com relação a homogeneidade porém, é que também é importante que não gere muitos grupos com poucos vetores, afim de se obter uma boa estimação dos modelos. Como resultado, a quantidade de grupos acaba aumentando (MOATTAR; HOMAYOUNPOUR, 2012; MIRO; BOZONNET, 2012).

A etapa de agrupamento de locutores pode ser classificada em duas abordagens principais: *Bottom-up* e *Top-down*. A *Bottom-up* utiliza técnicas de agrupamento hierárquico, criando os grupos de forma iterativa com junções de segmentos ou criando novas partições a partir de grupos formados. A *Top-down* utiliza técnicas/heurísticas para inferir o número mínimo de grupos, e em seguida utiliza técnicas de agrupamento com essa informação *a priori* (KOTTI; MOSCHOU; KOTROPOULOS, 2008; MOATTAR; HOMAYOUNPOUR, 2012; MIRO; BOZONNET, 2012). Nas abordagens, dois itens precisam ser definidos:

1. A distância entre grupos e/ou segmentos que será usada para determinar a similaridade. Uma matriz de distâncias de pares de grupos/segmentos é criada para determinar quais ou qual par deverá ser agrupado ou separado. Muitas vezes as distâncias usadas nesse procedimento são as mesmas utilizadas na segmentação, como descrito na Seção 2.3.1.

2. Um critério de parada (aceitação) para o algoritmo iterativo. O critério pode ser um número máximo k de locutores, estabelecido com alguma heurística (lembrando que não há informação *a priori* sobre o número de locutores), ou se as distâncias entre grupos e *intra* grupo atingirem um estado estacionário.

Revisamos as abordagens *Bottom-up* e *Top-down* na Seção 2.4.1 e 2.4.2, respectivamente. Outras abordagens de agrupamento são revisadas na Seção 2.4.3

2.4.1 Abordagem *Bottom-up*

Em Kotti, Moschou e Kotropoulos; Moattar e Homayounpour; Miro e Bozonnet, essa abordagem é considerada a mais utilizada no desenvolvimento de sistemas de transcrição de locutores. Os

segmentos de locutor obtidos na etapa de segmentação de locutores são considerados como os grupos iniciais para o algoritmo de agrupamento hierárquico. Uma matriz de distâncias é computada com valores de similaridade de cada possível par de grupos. Os dois grupos com maior valor de similaridade são agrupados. O processo se repete iterativamente até o critério de parada ser alcançado.

Uma das primeiras pesquisas feitas seguindo esta abordagem para agrupamento de locutores está em Hubert, Francis e Schwartz utilizando a distância Gish (GISH; SIU; ROHLICEK, 1991). O critério de parada é a obtenção do valor mínimo da seguinte função de penalidade:

$$W_J = |\sum_{k=1}^{K} N_k \Sigma_k| \sqrt{k}, \tag{2.24}$$

onde K é o número de grupos considerado, Σ_k a matriz de covariância do grupo k, com N_k vetores de características acústicas.

Em Siegler et al., a divergência KL2 é utilizada e o critério de aceitação é definido com um limiar de agrupamento. Em Zhou e Hansen a divergência KL2 também é utilizada, mas primeiro os segmentos são separados em dois grupos (homem e mulher) e depois o agrupamento é executado em cada um deles. Com isso, o tempo de processamento torna-se menor e melhores resultados são alcançados.

Além das medidas de distância entre vetores de características acústicas, também foram propostas distâncias entre modelos em Rougui et al., utilizando GMMs e baseando-se na divergência KL, definida como

$$d(\mathcal{M}_1, \mathcal{M}_2) = \sum_{i=1}^{K_1} W_1(i) \min_{j=1}^{K_2} KL(\mathcal{N}_1(i), \mathcal{N}_2(j)), \tag{2.25}$$

onde $\mathcal{M}_1$ e $\mathcal{M}_2$ são dois modelos com K_1 e K_2 componentes Gaussianas, pesos $W_1(i), i = 1...K_1$ e $W_2(j), j = 1...K_2$, respectivamente, e $\mathcal{N}_1(i)$ e $\mathcal{N}_2(2)$ são componentes das Gaussianas de $\mathcal{M}_1$ e $\mathcal{M}_2$, respectivamente.

A distância e critério de aceitação mais usado é novamente o BIC, que foi inicialmente proposto para a agrupamento de locutores em Chen e Gopalakrishnan. O par de grupos na matriz de distâncias que possuir o menor valor BIC será agrupado. O algoritmo termina quando todos os pares restantes tiverem um BIC > 0. Em algumas pesquisas posteriores (CHEN et al., 2002; TRITSCHLER; GOPINATH, 1999; TRANTER; REYNOLDS, 2004; CETTOLO; VESCOVI, 2003; MEINEDO; NETO, 2003) são propostas modificações para o parâmetro de penalidade λ e algumas alterações na etapa de segmentação.

Em Barras et al.; Zhu et al.; Zhu et al. e em Sinha et al. são propostas etapas de agrupamento de locutores que utilizam técnicas de reconhecimento de locutores. Uma etapa de agrupamento inicialmente proposta por Gauvain, Lamel e Adda é desenvolvida para determinar uma segmentação inicial em Barras et al.; Zhu et al.; Zhu et al., enquanto o algoritmo de Chen e Gopalakrishnan para detecção de mudança de locutor é utilizado em Sinha et al.. Os sistemas propostos utilizam o algoritmo de Chen e Gopalakrishnan para agrupamento aglomerativo

hierárquico via BIC, com o valor da penalidade λ ajustado para obter mais grupos do que o número ótimo de grupos.

2.4.2 Abordagem *Top-down*

Em Woodland, uma abordagem de agrupamento *top-down* é proposta para sistemas de reconhecimento de locutores, e em Johnson; Tranter e Reynolds ela é aplicada em sistemas de transcrição de locutores. O algoritmo particiona os dados iterativamente em quatro sub-grupos e permite o agrupamento dos que são mais similares entre si. Em Woodland são propostas duas implementações desse algoritmo.

Em Meignier, Bonastre e Igounet; Miró e Pericas, um grupo inicial é treinado com todas as classes acústicas presentes nos dados. Então, iterativamente é feita a decodificação de novos modelos onde novos grupos são particionados usando uma métrica pela verossimilhança medida a partir de uma janela de tempo aplicada ao sinal de áudio. A variação global da verossimilhança dos dados com todos os modelos criados é usado como um critério de parada. Em Miró e Pericas, uma abordagem similar é seguida e um repositório de modelos acústicos é introduzido para melhorar a pureza dos grupos criados.

2.4.3 Outras técnicas de agrupamento

Enquanto as abordagens que seguem a estratégia *bottom-up* são mais populares que as que seguem a estratégia *top-down*, ainda não há um consenso de qual das duas alcançam melhores resultados e quais as condições para isso. Na aplicação de transcrição de programas de noticiários, em Hain et al. ambas as estratégias são comparadas. A *Bottom-up* usa a distância KL como medida e uma quantidade mínima de grupos como critério de parada. A *Top-down* usa a distância esférica harmônica e o mesmo critério de parada.

Em Tranter, uma estratégia híbrida é proposta com o intuito de complementar o que individualmente as abordagens não conseguiam alcançar. É proposto um algoritmo de agrupamento por esquema de voto, permitindo o aprimoramento do resultado final, agrupando os resultados de dois sistemas que utilizam cada uma estratégias.

Em Moraru et al.; Fredouille et al., duas diferentes combinações de abordagens são apresentadas para combinar as respostas de estratégias *Top-down* e *bottom-up* e depois são utilizadas em aplicações de programas de noticiários e gravações de reuniões. A primeira abordagem é chamada de *hybridization* e propõe um sistema como fase de inicialização do outro sistema. A segunda abordagem é chamada de *fusion* e propõe um casamento dos resultados em comum seguida por uma resegmentação dos resultados que diferiram.

Algumas referências propõem técnicas que não se encaixam na definição do agrupamento hierárquico. Em Tsai e Wang, um algoritmo genético é proposto para obter o melhor agrupamento de forma a maximizar a verossimilhança geral dos modelos através da atribuição aleatória do número do grupo e avaliação iterativa da verossimilhança e mutação. Para alcançar a quantidade ótima de locutores eles computam o BIC nos modelos resultantes. Em Lapidot,

os modelos de mapas auto-organizáveis (*Self-Organized Maps - SOM*) descrito em Lapidot, Guterman e Cohen; Kohonen são propostos para o agrupamento dado um número conhecido de locutores. Trata-se de um algoritmo de *Vector Quantization (VQ)* para treinar os dicionários, representando cada um dos locutores. Um conjunto inicial de dicionários é criado e então o SOM é executado iterativamente, retreinando-os até o número de vetores acústicos trocados entre os grupos se aproximar a 0. O número de locutores é definido como sendo aquele que maximiza o critério BIC.

2.5 DISCUSSÃO SOBRE AS LIMITAÇÕES DAS TÉCNICAS DA LITERATURA PARA SEGMENTAÇÃO DE LOCUTORES

O problema de segmentar conversações de estilo espontâneas/livre exige trabalhar com amostras curtas da fala para evitar a perda das frequentes transições de locutor. A dificuldade em estimar bons modelos do locutor a partir de dados limitados da fala é mostrada em trabalhos recentes (PODDAR; SAHIDULLAH; SAHA, 2018b; PODDAR; SAHIDULLAH; SAHA, 2018a). Os sistemas de verificação de locutor que utilizam o *framework* GMM-UBM e i-vector requerem uma quantidade suficiente de dados de fala para funcionarem com razoável precisão (PODDAR; SAHIDULLAH; SAHA, 2018b; PODDAR; SAHIDULLAH; SAHA, 2018a). A métrica *Equal Error Rate* (EER) é utilizada para comparar o desempenho de sistemas de verificação de locutor. Ela é calculada como o ponto no qual a taxa de falsos positivos seja igual a taxa de falsos negativos. Em Poddar, Sahidullah e Saha, experimentos realizados nesses sistemas mostram uma degradação significativa no desempenho, ao reduzir a duração da fala das locuções de treinamento ou de validação: de 3,48%, para 22.09% da ERR, quando a duração das locuções foram encurtadas de 2,5 minutos para 2 segundos. Essa degradação ocorre pelo fato da estimativa do i-vector ter se tornado dependente da baixa variabilidade entre-locutores (fator importante para discriminação de locutores), e tornou-se dependente da alta variabilidade do conteúdo léxico falado (fator independente da tarefa de discriminação de locutores) (PODDAR; SAHIDULLAH; SAHA, 2018b).

Para entender as limitações para estimar o i-vector, voltaremos para a definição dos *super-vectors* na Seção 2.2. Os *super-vectors* são descritos pelas estatísticas de ordem zero e de primeira ordem de Baum-Welch (DEHAK; KENNY; DUMOUCHEL P., 2011). As estatísticas de ordem zero são representadas em um vetor contendo a soma das probabilidades *a posteriori* dos vetores de caracterísiticas acústicas de uma amostra para cada componente do modelo GMM. Quando o segmento possui curta duração, o vetor com essas estatísticas torna-se esparso. Essa esparsidade faz com que a estimativa do i-vector seja pouco precisa (HAUTAMÄKI et al., 2013; PODDAR; SAHIDULLAH; SAHA, 2018b). Por outro lado, quando o segmento possui uma duração razoável, o vetor de estatísticas torna-se denso, permitindo uma estimativa mais precisa do i-vector (HAUTAMÄKI et al., 2013; PODDAR; SAHIDULLAH; SAHA, 2018b). Mais detalhes sobre a degradação do desempenho pelo tempo de duração de fala são fornecidos em (HAUTAMÄKI et al., 2013; PODDAR; SAHIDULLAH; SAHA, 2018a; PODDAR; SAHIDULLAH; SAHA, 2018b).

A dificuldade em estimar modelos precisos do locutor a partir segmentos curtos da fala nos

leva a propor uma nova abordagem e características para o problema de segmentação. Em vez de estimar representações dos locutores, as características propostas têm como objetivo caracterizar de forma distinta os possíveis casos de segmentos da fala. As propriedades da voz que investigamos para representar cada caso são referentes ao estilo da fala (caracterizados no Capítulo 3). Consideramos que cada caso tem um estilo geral da fala distinto dos demais. Interpretamos o estilo geral da fala como todas as variações de estilo presentes em um segmento de fala. Esta informação pode ser útil para a tarefa de segmentação, porque são esperados os casos de segmentos heterogêneos, homogêneos e com sobreposição de fala, e a detecção de seus intervalos estão relacionados com as transições de locutor.

3 CARACTERÍSTICAS PARA SEGMENTAÇÃO DE LOCUTORES

3.1 REPRESENTAÇÃO PARA O ESTILO DA FALA

Neste trabalho, consideramos que os tipos de segmento possuem estilos diferentes da fala, e portanto, as características propostas precisam capturar as informações dos diferentes estilos para que a discriminação entre os tipos seja possível. Definimos estilo da fala como uma forma de expressá-la de forma individual. Tratamos o estilo como uma propriedade explicada por fatores suprasegmentais (CRYSTAL, 2011), os quais são elementos contrastivos que são difíceis de se analisar separadamente, e que estão presentes ao mesmo tempo na fala: velocidade, aceleração, emoção, intenção e sotaque. Esses fatores são importantes para a compreensão da fala pois podem mudar o entendimento do significado de uma sentença, ou mesmo uma palavra. Alguns desses fatores são dinâmicos (velocidade, aceleração), outros alteram a forma da pronúncia ou entonação de palavras (intenção e sotaque), assim como podem alterar a forma de emitir a voz (emoção) (FLANAGAN; ALLEN; HASAGAWA-JOHNSON, 2008). Assumimos as seguintes premissas de caracterização do estilo geral da fala para um segmento:

- Segmento homogêneo: Representação da informação do estilo da fala de um indivíduo;
- Segmento heterogêneo: Representação da informação de ao menos dois estilos de fala existentes em intervalos de tempo diferentes;
- Segmento com sobreposição de fala: Representação da sobreposição de dois estilos de fala. Interpretamos como uma indefinição de estilo predominante no segmento como um todo.

Realizamos uma investigação sobre como codificar ou extrair informações do estilo da fala. Levantamos trabalhos relacionados na área de processamento de imagens com aplicação em transferência de estilo de uma imagem de referência para outra imagem de destino. A Seção 3.1.1 traz esse levantamento, descrevendo as principais técnicas que conseguem extrair informações do estilo das imagens de forma satisfatória para o problema de transferência de estilo.

3.1.1 Modelo para Informações do Estilo na Área de Processamento de Imagens

Recentemente, Gatys, Ecker e Bethge mostram que o estilo visual de uma imagem pode ser codificado através de uma matriz contendo as correlações entre as características extraídas de camadas densas de uma rede convolucional. Eles propõem um método de transferência de estilo baseado na minimização da diferença das matrizes de correlação de duas imagens de estilos diferentes. Quando a diferença é mínima, assume-se que transferência de estilo foi concluída de forma ótima.

Segundo Gatys, Ecker e Bethge; Li et al.; Li et al., um algoritmo de transferência de estilo deve ser capaz de extrair o conteúdo da imagem semântica da imagem de destino (por exemplo, objetos e cenário) e, em seguida, informar um procedimento de transferência de textura para processar o conteúdo semântico da imagem de destino no estilo de a imagem de origem. Portanto, um pré-requisito fundamental é encontrar representações de imagens que modelem independentemente variações no conteúdo da imagem semântica e no estilo em que ela é apresentada. Geralmente, separar o conteúdo do estilo em imagens naturais ainda é um problema extremamente difícil. No entanto, o recente avanço das *Deep Convolutional Neural Networks* produziu poderosos sistemas de visão computacional que aprendem a extrair informações semânticas de alto nível de imagens naturais.

(GATYS; ECKER; BETHGE, 2016) utilizam um conjunto de características definidos nas camadas *Fully-Connected* de uma rede neural convolucional com 19 camadas ao todo (arquitetura VGG-19), proposta por (SIMONYAN; ZISSERMAN, 2015). Duas imagens de estilos diferentes I_1 e I_2 são dadas como entrada para a VGG e dois conjuntos de características $\mathcal{F}_1$ e $\mathcal{F}_2$ são extraídos da rede. A matriz que codifica as correlações entre as características é chamada de Gram (GATYS; ECKER; BETHGE, 2016), definida como:

$$G(\mathcal{F}) = \begin{bmatrix} < \mathbf{f}_1, \mathbf{f}_1 > & < \mathbf{f}_1, \mathbf{f}_2 > & \dots & < \mathbf{f}_1, \mathbf{f}_l > \\ < \mathbf{f}_2, \mathbf{f}_1 > & < \mathbf{f}_2, \mathbf{f}_2 > & \dots & < \mathbf{f}_2, \mathbf{f}_l > \\ \vdots & \vdots & \ddots & \vdots \\ < \mathbf{f}_l, \mathbf{f}_1 > & < \mathbf{f}_l, \mathbf{f}_2 > & \dots & < \mathbf{f}_l, \mathbf{f}_l > \end{bmatrix} \tag{3.1}$$

onde $\mathbf{f}_i$ e $\mathbf{f}_j$ são vetores de características em $\mathcal{F}$ extraídos da imagem nas camadas densas i e j, respectivamente, e l é a quantidade de camadas densas da VGG. A matriz é calculada para $\mathcal{F}_1$ e $\mathcal{F}_2$ e uma função de perda é definida utilizando essas matrizes para retreinar a rede com o objetivo de transferir o estilo sem a perda da informação de conteúdo da imagem. A matriz Gram mede a similaridade entre vetores de um conjunto através do produto interno desses vetores.

Em (LI et al., 2018), observa-se que a função de perda utilizando as matrizes Gram é equivalente a computar a diferença entre alguns aspectos que são discrepantes entre duas imagens. Outras métricas de similaridade como *maximum mean discrepancy* (MMD) são avaliadas e obtiveram resultados similares aos obtidos com as matrizes Gram. Observa-se que as informações sobre o estilo podem ser capturadas através da similaridade de características da imagem sem a necessidade de correlação espacial entre os objetos presentes.

3.2 *MEL CEPSTRAL AFFINITY FEATURES*

Neste trabalho, investigamos se a similaridade das características MFCC também podem codificar o estilo da fala para segmentos de diferentes tipos. Ao invés da matriz Gram utilizamos a matriz de afinidade, por possuir uma métrica não-linear de distância, tendo portanto menos

restrições geométricas para o cálculo da similaridade. Assumimos então, que a similaridade das características acústicas não precisa estar definida em um espaço linear, pois o objetivo não é se preocupar com a proximidade real entre dois vetores, e sim ponderar a semelhança entre eles. A distribuição dessas semelhanças é que pode revelar os padrões de diferentes tipos de segmentos da fala. Além disso, a função de afinidade cria uma matriz positiva semi-definida, o que permitirá a decomposição do espaço de afinidade para obter variáveis latentes sem a necessidade de aproximações estimativa do modelo.

As características *Mel Cepstral Affinity Features* (MCAF) são computadas dadas as características MFCCs extraídas de um segmento de fala. A extração possui 5 etapas (Figura 4): amostragem dos MFCCs, cálculo da matriz de afinidade, computar o vetor de afinidade, aplicar transformação invariante no tempo e extração de características latentes.

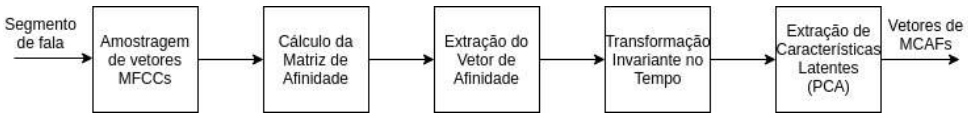

Figura 4 – Diagrama de blocos do processo de extração das características MCAF. Começa com a amostragem dos vetores de MFCCs de um segmento de fala. Daí, segue para o cálculo da matriz de afinidade. Em seguida, removemos informação redundante desta matriz e extraímos o vetor de afinidade. Aplicamos uma transformação para que as mesmas informações não variem no tempo conforme ocorre a amostragem de mais vetores MFCCs. Por fim, com todos os vetores de afinidade extraídos do segmento de fala, aplicamos a transformação PCA para estimar o espaço latente de informações de afinidade.

3.2.1 Amostragem dos MFCCs

A amostragem dos MFCCs extraídos do segmento de fala é feita utilizando uma janela móvel com um tamanho de n vetores e com um passo (*step*) de tamanho s (Figura 5). A amostra extraída em um instante t de um segmento de fala é representada pelo conjunto $X(t)$ (Figura 6).

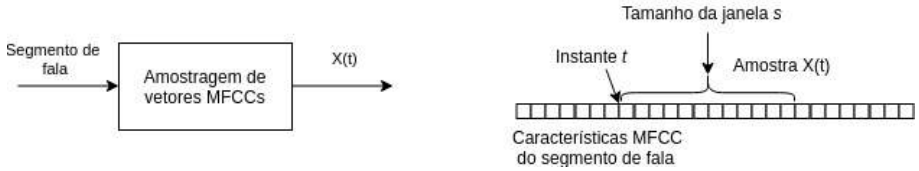

Figura 5 – Etapa de amostragem dos vetores MFCCs extraídos do segmento de fala. A amostragem é realizada por uma janela deslizante de tamanho s. O passo da janela vai determinar a quantidade de vetores MCAF que serão extraídos do segmento de fala, para cada passo um vetor MCAF.

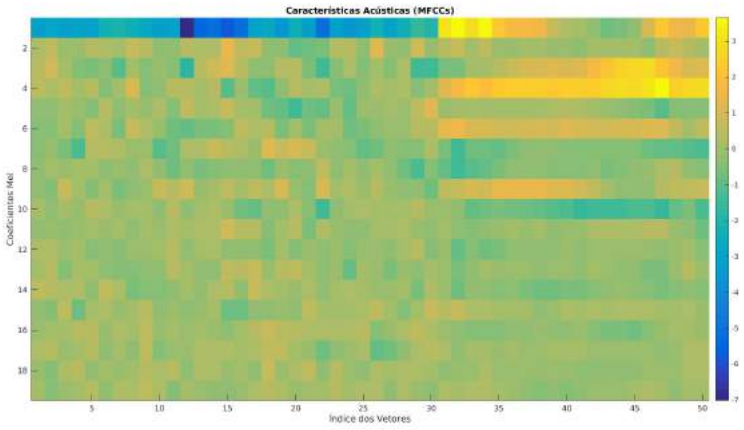

Figura 6 – Características MFCCs de uma amostra de 0,5 segundos. Ao todo são 19 coeficientes Mel sem a energia e coeficientes Δ e Δ^2.

3.2.2 Cálculo da Matriz de Afinidade

Dado o conjunto de características MFCCs $X(t)$, a matriz de afinidade $A(t)$ $n \times n$ é definida como:

$$A(t) = \begin{cases} exp(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}), & \forall i \neq j, \quad i,j = 1,2,\ldots,n \\ 0, & \text{caso contrário,} \end{cases} \tag{3.2}$$

onde $||.||$ é a norma do vetor, $\mathbf{x}_i$ e $\mathbf{x}_j$ são vetores de características i e j de $X(t)$ e σ é um fator que controla a velocidade de decaimento dos níveis de afinidade dada distância entre $\mathbf{x}_i$ e $\mathbf{x}_j$. A Figura 7 ilustra a matriz calculada a partir da amostra $X(t)$ (mais detalhes da matriz $A(t)$ na Figura 8).

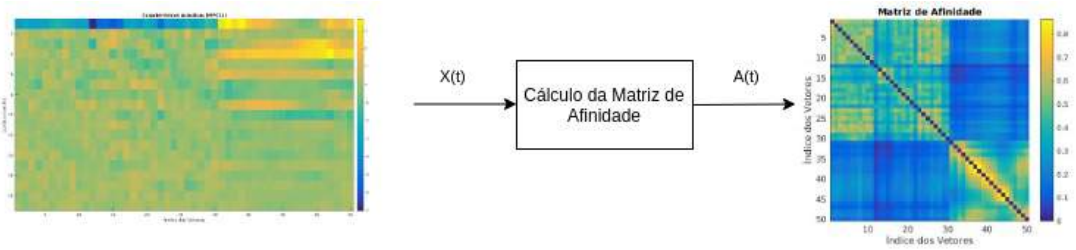

Figura 7 – Entrada e saída da etapa de cálculo da matriz de afinidade. Calcula-se a afinidade dos pares de vetores em X(t), de acordo com a Equação 3.2.

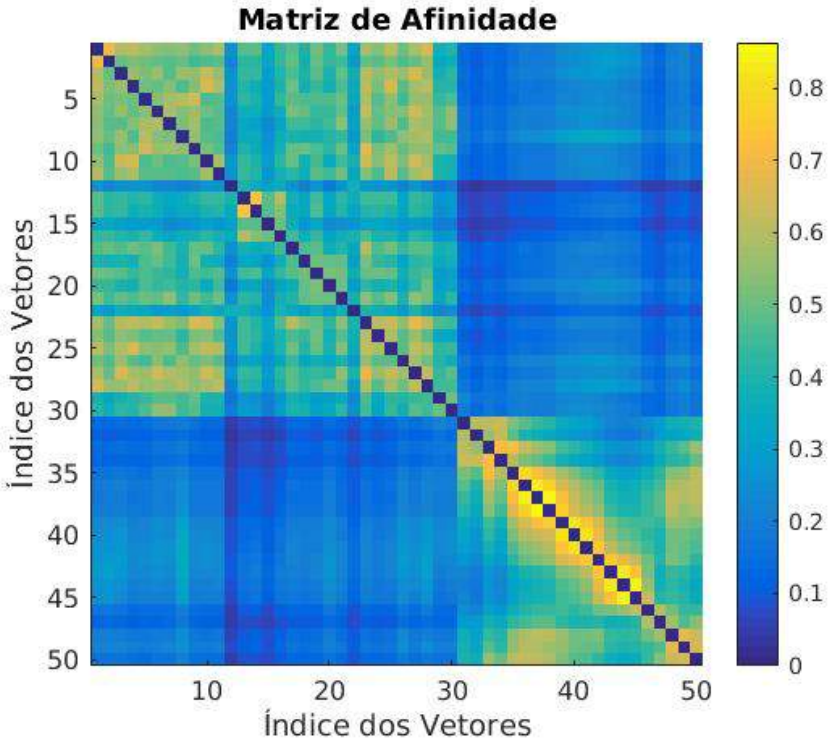

Figura 8 – Matriz de afinidade calculada para a amostra de 0,5 segundos de MFCCs. Neste exemplo, utilizamos $\sigma = 0,75$

3.2.3 Vetor de Afinidade

Esta terceira etapa computa o vetor de afinidade $\mathbf{v}(t)$ de $A(t)$ (Figura 9) através de dois procedimentos:

- Remoção de informação redundante: a matriz de afinidade é simétrica e sua diagonal tem um valor constante independente da amostra $X(t)$, pois trata-se do valor 0 (caso em que $i = j$). Com isso, podemos desconsiderar o conteúdo da diagonal e considerar apenas os elementos que ficam abaixo dela (Figura 10);
- Definir o vetor: os elementos da diagonal inferior de $A(t)$ formarão o vetor de afinidade. Aplica-se na matriz a operação *flatting*, responsável por reformatar os elementos da matriz em uma única dimensão (Figura 11).

A dimensionalidade do vetor de afinidade independe da dimensionalidade do vetor de características acústicas. Ela é definida a partir da quantidade n de vetores amostrados:

$$p = \frac{n^2 - n}{2}. \tag{3.3}$$

Figura 9 – Entrada e saída da etapa de extração do vetor de afinidade $\mathbf{v}(t)$. A etapa engloba as operações de *flatting* da matriz de afinidade inferior, descartando os elementos da diagonal superior e da diagonal da matriz A(t).

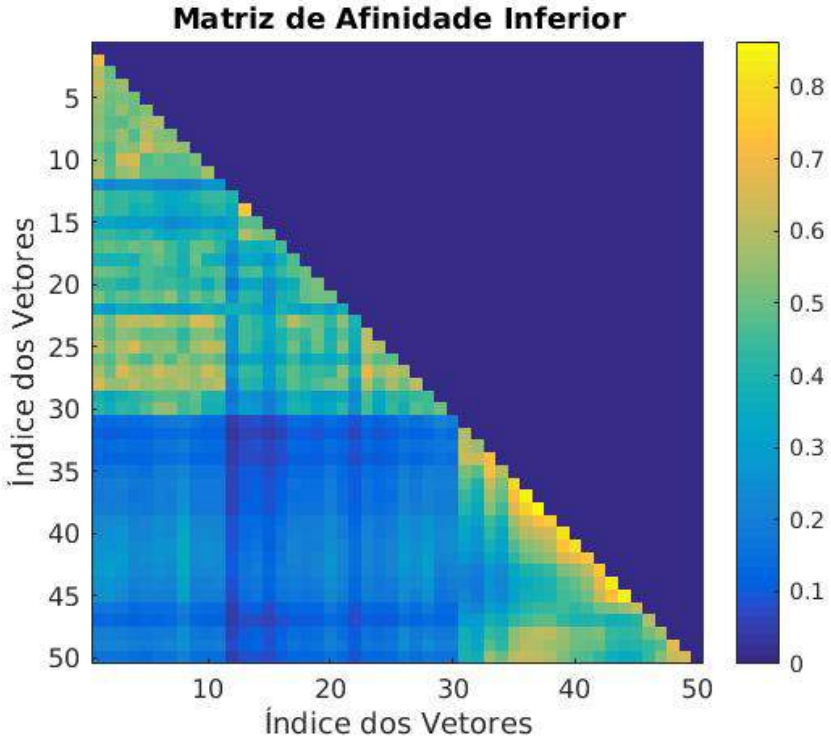

Figura 10 – Matriz de afinidade sem redundância de informação. Apenas os elementos abaixo da diagonal principal são utilizados.

3.2.4 Transformada Invariante no Tempo

Durante a amostragem dos MFCCs, a janela desliza no sinal para a direita (Figura 5), avançando no tempo. Inevitavelmente, essa movimentação faz com que o conteúdo falado translade da direita para a esquerda. Essa translação traz como consequência o deslocamento de grande parte dos elementos de $A(t)$ para $A(t + \Delta t)$, no eixo da diagonal principal, em direção ao elemento na posição (1,1). Esse deslocamento dos elementos causa variação no vetor de afinidade pelo simples fato da variação temporal do conteúdo falado. Para evitar essa variabilidade, propopomos uma transformada que remova a referência temporal (i, j) dos valores de afinidade (Figura 12). A transformada proposta é o *sorting* dos elementos de afinidade, em ordem

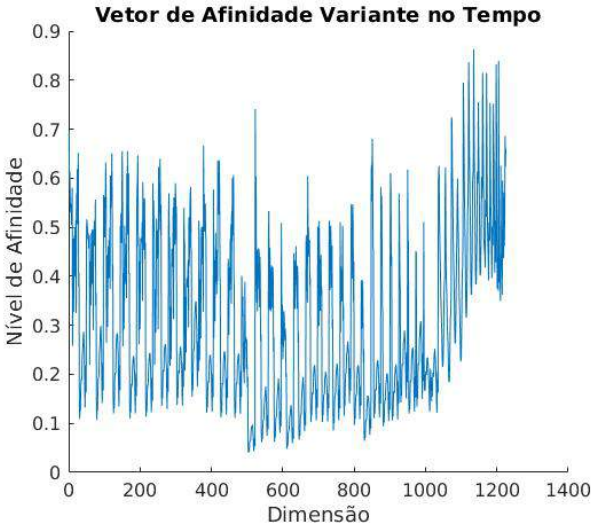

Figura 11 – Vetor de afinidade $\mathbf{v}(t)$ obtido a partir da operação de *flatting* da matriz de afinidade inferior, , descartando os elementos da diagonal superior e da diagonal da matriz A(t).

ascendente (o ordenamento não precisa ser necessariamente ascendente, porém precisa ser fixado e não deve ser modificado após a definição). Com isso, um novo vetor de afinidade invariante no tempo $\mathbf{v}_{IT}(t)$ ((Figura 13) e de mesma dimensão p é definido:

$$\mathbf{v}_{IT}(t) = \text{sort}(\mathbf{v}(t)). \tag{3.4}$$

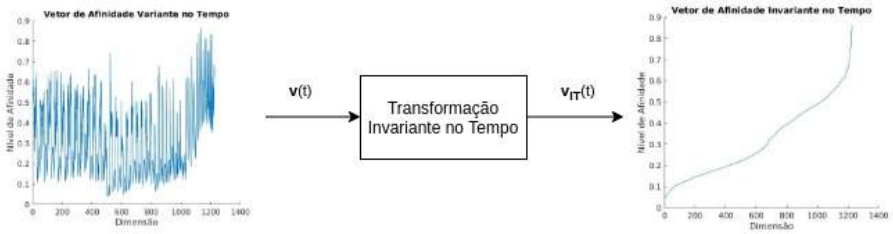

Figura 12 – Entrada e saída da etapa de cálculo do vetor de afinidade invariante no tempo $\mathbf{v}_{IT}(t)$.

3.2.5 Características Latentes

Nesta quinta e última etapa, trataremos o conjunto V_{IT} composto pelos vetores $\mathbf{v}_{IT}(t)$ computados para cada instante t de amostragem (cada linha de V_{IT} é um vetor $\mathbf{v}_{IT}$). Então, utilizamos a Análise de Compontentes Principais (PCA) para estimar um novo espaço de menor dimensionalidade e que preserve as informações em dimensões não correlacionadas (Figura 14). Os vetores de características MCAF Y são definidos a partir de V_{IT} como:

$$Y = V_{IT}W \tag{3.5}$$

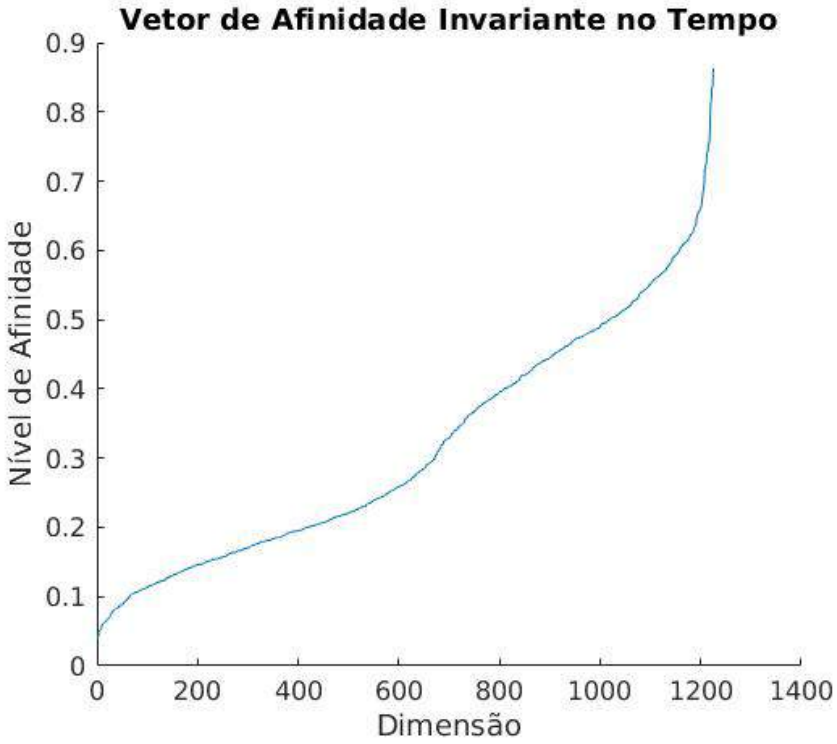

Figura 13 – Vetor de afinidade invariante no tempo $\mathbf{v}_{IT}(t)$ obtido a partir do *sorting* dos componentes de $\mathbf{v}(t)$.

onde as colunas de W são os autovetores da matriz de covariância de V_{IT} na ordem decrescente dos autovalores. Os vetores MCAF são as linhas de Y. A dimensionalidade é igual ao número de componentes principais definido para execução do PCA.

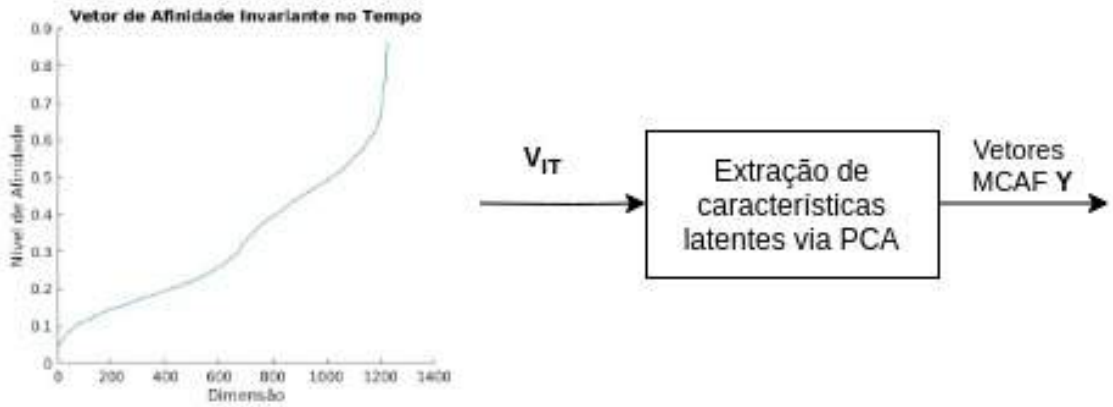

Figura 14 – Entrada e saída da etapa da extração de características latentes dos vetores de afinidade. A matriz V_{IT} representa os vetores $\mathbf{v}_{IT}(t)$ computados em cada passo da janela de amostragem dos MFCCs. A matriz Y são as projeções de V_{IT} no espaço dos principais componentes, correspondendo às variáveis latentes que explicam a variabilidade de V_{IT}.

3.3 DISCRIMINAÇÃO DOS TIPOS DE SEGMENTO DE FALA

Investigamos se as características MCAF apresentam padrões que evidenciem a discriminação dos segmentos de fala homogêneos, heterogêneos e com sobreposição de fala. A investigação contempla todos os tamanhos de amostra avaliados: 0,5, 0,75, 1 e 1,25 segundos, e utiliza todas as conversações do conjunto de treinamento. Para categorizar as amostras, utilizamos a identidade dos locutores presentes no *ground truth* de cada conversação. Os vetores MCAF são categorizados como sendo do tipo sobreposição de fala, caso essa sobreposição exista na amostra de origem por mais de 20% da duração da mesma. Vetores categorizados como sendo do tipo fala heterogênea são provenientes de amostras que possuem mais de um locutor em momentos distintos (permite-se sobreposição de fala por no máximo 10% da amostra). Vetores categorizados como fala homogênea são provenientes de amostras com apenas um locutor presente. Utilizamos o valor de $\sigma = 0,75$.

A Figura 15 mostra a distribuição populacional dos tipos de segmentos da fala para amostras de 0,5 segundos. Com esse tamanho de amostra, observa-se um número maior de segmentos com sobreposição de fala do que do tipo heterogêneo. Na Figura 16, observamos a diferença no comportamento das correlações para cada tipo de segmento da fala. A discrepância é aparentemente maior quando compara-se as correlações do tipo homogêneo com os demais tipos, do que ao comparar as correlações do tipo heterogêneo com o tipo sobreposição de fala. Na Figura 17, também observamos a diferença de comportamento, agora relacionado às médias dos vetores de cada tipo de segmento.

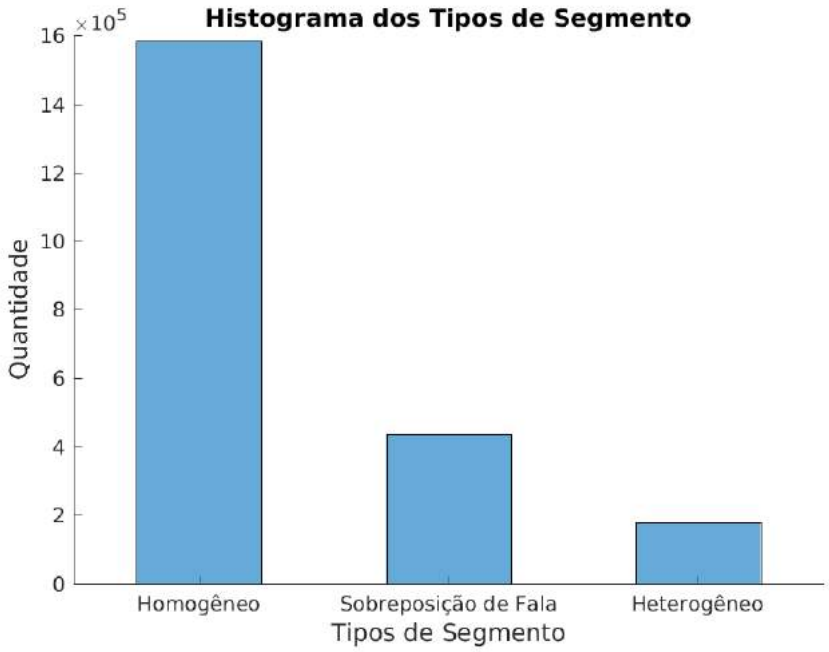

Figura 15 – Tamanho da população cada tipo de segmentos da fala em amostras de 0,5 segundos. A população foi calculada utilizando todas as conversações do conjunto de treinamento da AMI *Corpus*.

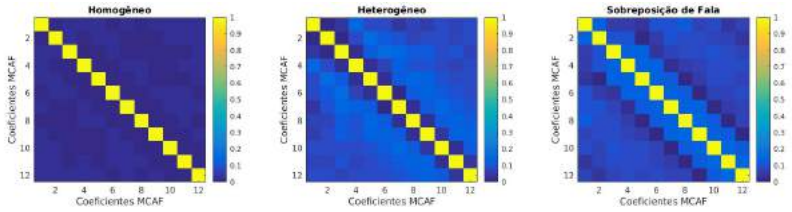

Figura 16 – Matriz de correlação de cada tipo de segmento da fala. O tamanho da amostra utilizada para extração dos MCAF é de 0,5 segundos. As matrizes de correlação foram computadas utilizando os vetores MCAF extraídos de todas as conversações do conjunto de treinamento da AMI *Corpus*.

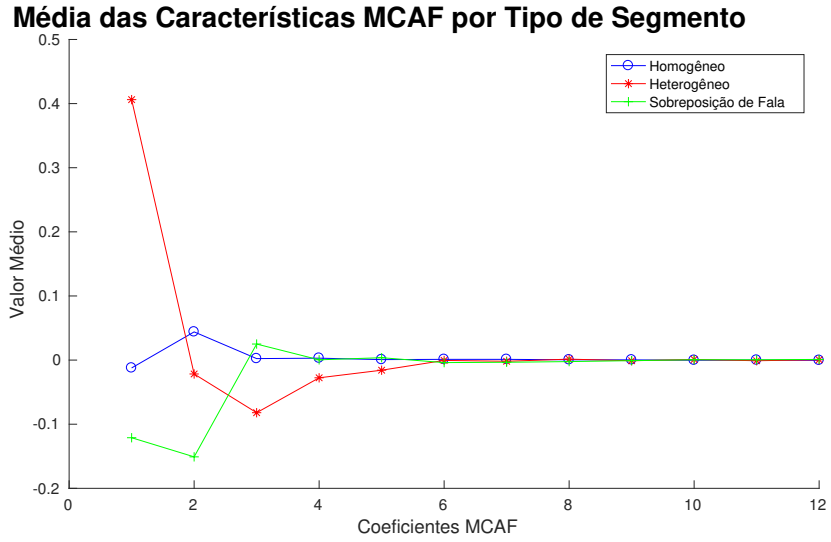

Figura 17 – Médias dos vetores MCAF extraídos de cada tipo de segmento da fala. O tamanho da amostra utilizada para extração dos MCAF é de 0,5 segundos. As médias foram computadas utilizando os vetores MCAF extraídos de todas as conversações do conjunto de treinamento da AMI *Corpus*.

A Figura 18 mostra a distribuição populacional dos tipos de segmentos da fala para amostras de 0,75 segundos. Com esse tamanho de amostra, ainda observa-se um número maior de segmentos com sobreposição de fala do que do tipo heterogêneo. Na Figura 19, ainda podemos observamos a diferença no comportamento das correlações para cada tipo de segmento da fala. A discrepância é aparentemente maior quando compara-se as correlações do tipo homogêneo com os demais tipos, do que ao comparar as correlações do tipo heterogêneo com o tipo sobreposição de fala. Na Figura 20, também observamos a diferença de comportamento, agora relacionado às médias dos vetores de cada tipo de segmento.

A Figura 21 mostra a distribuição populacional dos tipos de segmentos da fala para amostras de 1 segundo. Com esse tamanho de amostra, observa-se um número maior de segmentos com sobreposição de fala do que do tipo heterogêneo. Na Figura 22, observamos a diferença no comportamento das correlações para cada tipo de segmento da fala. A discrepância não tão evidente quando compara-se as correlações do tipo homogêneo com o tipo de sobreposição. Ao

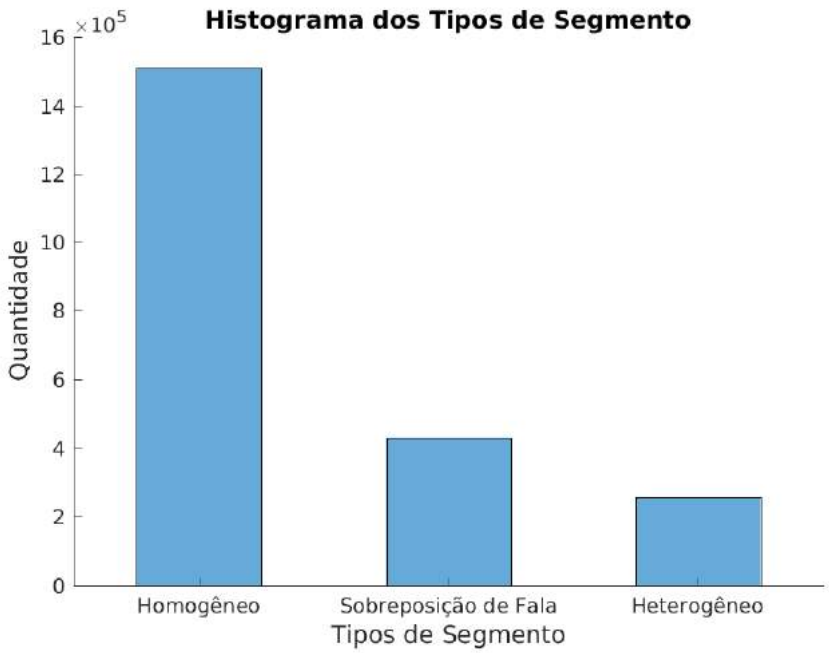

Figura 18 – Tamanho da população cada tipo de segmento da fala em amostras de 0,75 segundos. A população foi calculada utilizando todas as conversações do conjunto de treinamento da AMI *Corpus*.

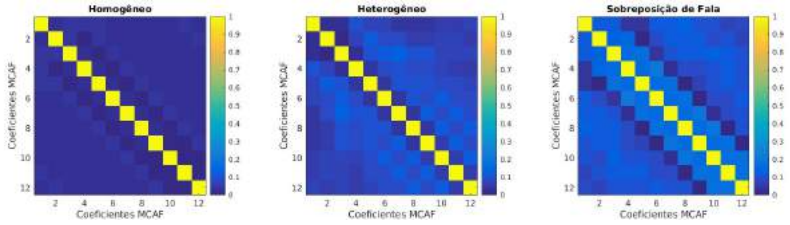

Figura 19 – Matriz de correlação de cada tipo de segmento da fala. O tamanho da amostra utilizada para extração dos MCAF é de 0,75 segundos. As matrizes de correlação foram computadas utilizando os vetores MCAF extraídos de todas as conversações do conjunto de treinamento da AMI *Corpus*.

comparar as correlações do tipo heterogêneo com o tipo sobreposição e homogêneo é possível enxergar discrepância. Na Figura 23, também observamos a diferença de comportamento, agora relacionado às médias dos vetores de cada tipo de segmento.

A Figura 24 mostra a distribuição populacional dos tipos de segmentos da fala para amostras de 1,25 segundos. Com esse tamanho de amostra, observa-se um número equiparado de segmentos do tipo heterogêneo e com sobreposição de fala. Na Figura 25, observamos a diferença no comportamento das correlações para cada tipo de segmento da fala. A discrepância não é mais aparentemente compara-se as correlações dos tipo homogêneo e sobreposição. Ao comparar as correlações do tipo heterogêneo com os demais, ainda percebe-se a discrepância entre as correlações. Na Figura 26, também observamos a diferença de comportamento, agora relacionado às médias dos vetores de cada tipo de segmento.

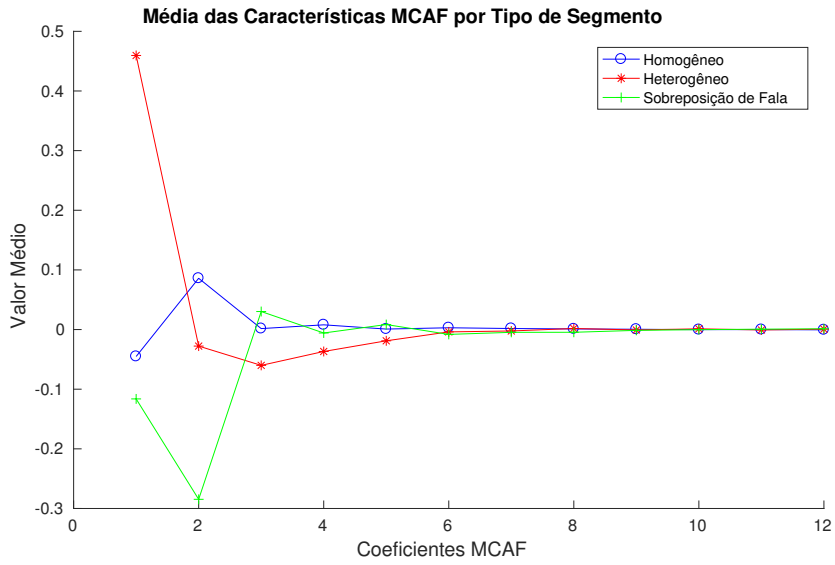

Figura 20 – Médias dos vetores MCAF extraídos de cada tipo de segmento da fala. O tamanho da amostra utilizada para extração dos MCAF é de 0,75 segundos. As médias foram computadas utilizando os vetores MCAF extraídos de todas as conversações do conjunto de treinamento da AMI *Corpus*.

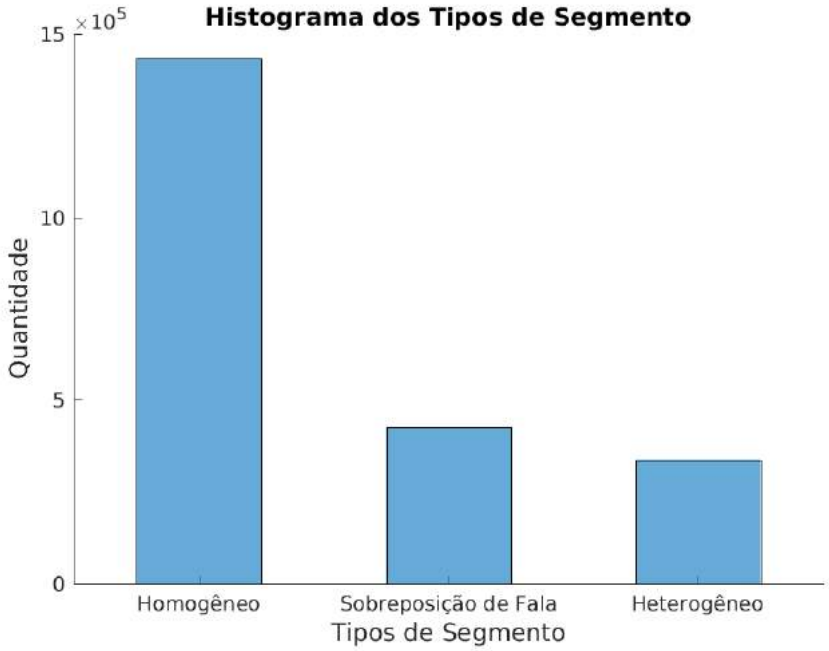

Figura 21 – Tamanho da população cada tipo de segmento da fala em amostras de 1 segundo. A população foi calculada utilizando todas as conversações do conjunto de treinamento da AMI *Corpus*.

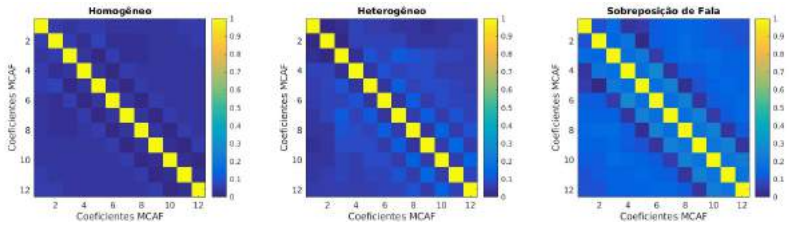

Figura 22 – Matriz de correlação de cada tipo de segmento da fala. O tamanho da amostra utilizada para extração dos MCAF é de 1 segundo. As matrizes de correlação foram computadas utilizando os vetores MCAF extraídos de todas as conversações do conjunto de treinamento da AMI *Corpus*.

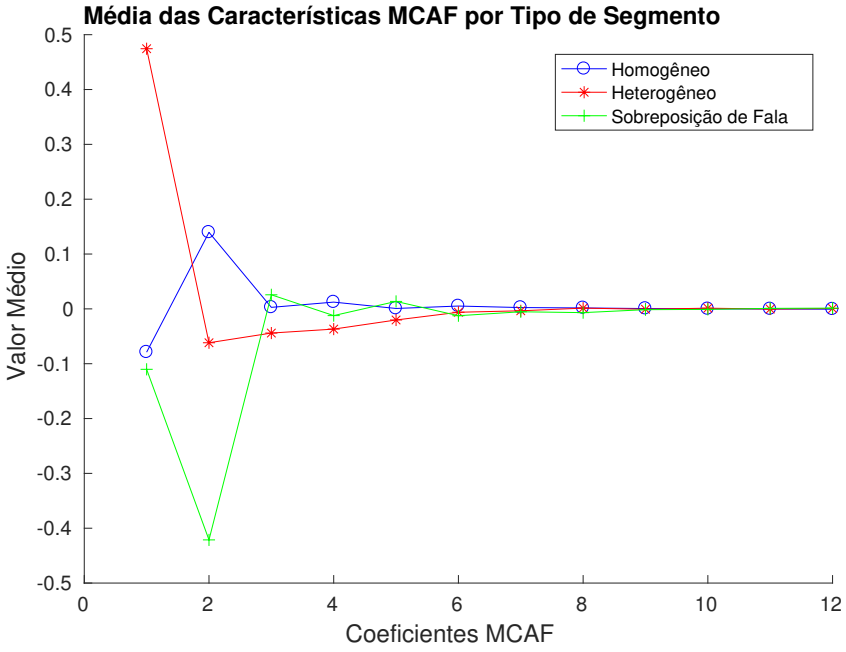

Figura 23 – Médias dos vetores MCAF extraídos de cada tipo de segmento da fala. O tamanho da amostra utilizada para extração dos MCAF é de 1 segundo. As médias foram computadas utilizando os vetores MCAF extraídos de todas as conversações do conjunto de treinamento da AMI *Corpus*.

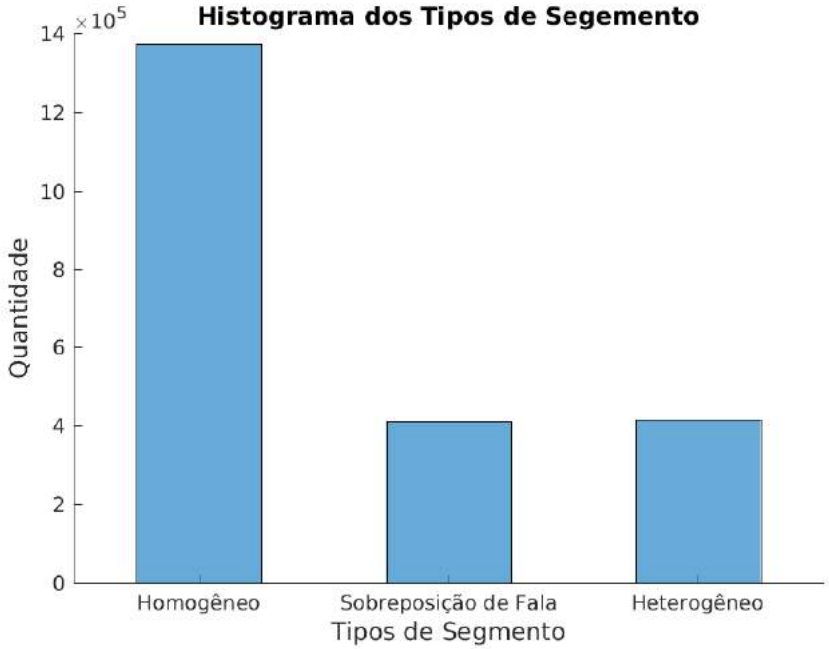

Figura 24 – Tamanho da população cada tipo de segmento da fala em amostras de 1,25 segundos. A população foi calculada utilizando todas as conversações do conjunto de treinamento da AMI *Corpus*.

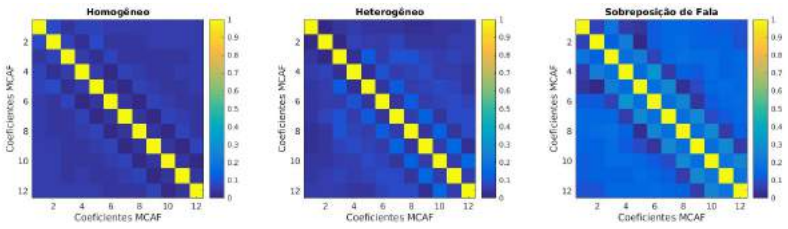

Figura 25 – Matriz de correlação de cada tipo de segmento da fala. O tamanho da amostra utilizada para extração dos MCAF é de 1,25 segundos. As matrizes de correlação foram computadas utilizando os vetores MCAF extraídos de todas as conversações do conjunto de treinamento da AMI *Corpus*.

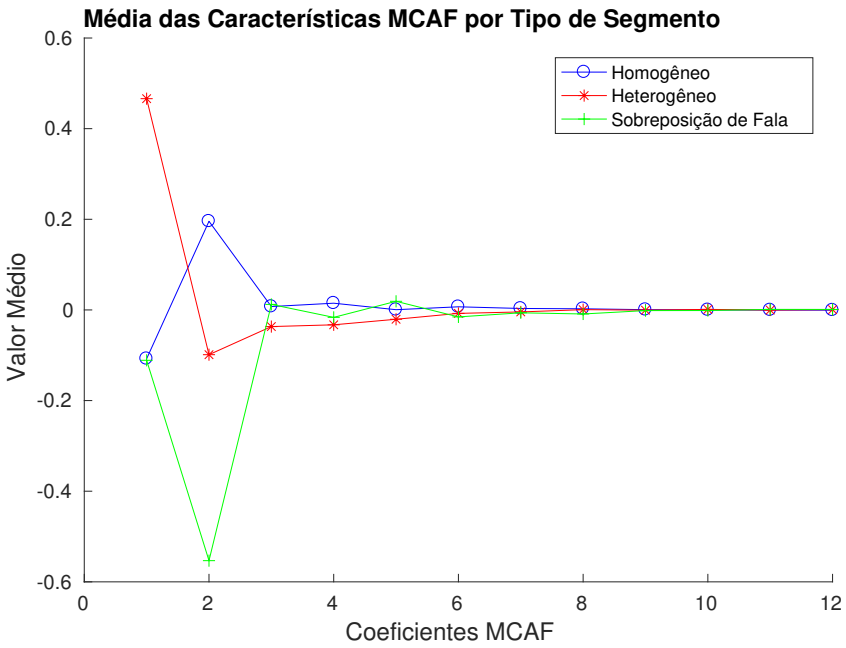

Figura 26 – Médias dos vetores MCAF extraídos de cada tipo de segmento da fala. O tamanho da amostra utilizada para extração dos MCAF é de 1,25 segundos. As médias foram computadas utilizando os vetores MCAF extraídos de todas as conversações do conjunto de treinamento da AMI *Corpus*.

3.3.1 Método de Segmentação de Locutores

Os métodos tradicionais de segmentação utilizam duas janelas deslizantes e uma métrica de distância para avaliar se dois segmentos pertencem ao mesmo locutor. Eles estimam modelos para representar a informação discriminante dos locutores e para calcular a distância entre os segmentos. No final do processo de movimentação, obtém-se uma curva de distância. Os valores de pico presentes na curva indicam quando os segmentos pertencem a locutores distintos. Uma possível mudança de locutor é um valor de pico considerado relevante na curva: é um valor de um máximo local, cuja proeminência está acima de um limite pré-configurado. A proeminência é calculada como a diferença entre o valor de pico e o maior dos dois valores mínimos da vizinhança, cada um localizado em um lado (esquerdo e direito) do valor máximo (DELACOURT; WELLEKENS, 2000; KOTTI; MOSCHOU; KOTROPOULOS, 2008; DESPLANQUES; DEMUYNCK; MARTENS, 2015).

O método de segmentação proposto extrai as características MCAF também utilizando duas janelas deslizantes e também uma métrica de distância. Na nossa abordagem, os valores de pico presentes na curva revelam intervalos de tempo em que segmentos de diferentes tipos são vizinhos. Dentro de cada intervalo existe um ponto de mudança de locutor. Durante a movimentação, sempre que a janela da direita se torna em um tipo diferente da janela da esquerda, teremos um valor alto de distância e talvez um valor de pico. Detectamos picos relevantes quando sua proeminência está acima de um limite pré-fixado. Neste trabalho, assumimos que o ponto de mudança de locutor é o ponto médio de cada intervalo com duração máxima de 0,5 segundos. Acima disso, consideramos que se tratam de dois intervalos distintos. As distâncias utilizadas são a Euclidiana (Equação 2.22) e a do cosseno (Equação 2.23).

4 EXPERIMENTOS E RESULTADOS

Este capítulo apresenta o protocolo de experimentos adotado para a avaliação das características MCAF na segmentação de locutores em conversações de estilo livre. Selecionamos os métodos de duas janelas deslizantes, aqui nomeados de *FixSlid*, utilizando diferentes características, modelos e distâncias da literatura, para comparação e discussão dos resultados. Ao todo são três métodos. com até duas variações de distância:

- *MFCC-LLR*: utilizando as características MFCCs e a distância LLR. A modelagem é feita através de distribuições Gaussianas;
- *I-VECTOR-COS* e *I-VECTOR-EUC*: utilizando o i-vector como característica e modelo. Avaliamos o desempenho utilizando as distâncias do cosseno e Euclidiana;
- *MCAF-COS* e *MCAF-EUC*: utilizando as características MCAF para discriminar os tipos de segmentos. Avaliamos o desempenho utilizando as distâncias do cosseno e Euclidiana;

Realizamos os experimentos de segmentação de locutores utilizando como base de conversações a AMI *Corpus*. Apresentamos o comportamento das transições de locutor nas conversas de estilo livre nessa base. Detalhamos como configuramos a extração das características acústicas para cada método selecionado. Também descrevemos a configuração para extração das características do locutor, ou da fala, bem como as configurações adotadas para modelagem dessas características. Apresentamos a distribuição dos tipos de segmentos da fala, bem como mais evidências observadas em mais conversações sobre o comportamento das MCAF para os diferentes tipos de segmento. Seguimos descrevendo como os métodos de segmentação foram configurados para cada característica. Apresentamos os resultados gerais da segmentação e definimos o melhor resultado para cada método avaliado de acordo com as métricas de desempenho apresentadas na Seção 2.3. Por fim, discutimos os resultados alcançados em conjuto com os dados de custo computacional coletados durante os experimentos.

4.1 ANÁLISE DOS DADOS DA AMI *CORPUS*

O AMI *Meeting Corpus* é um conjunto de dados de acesso aberto que consiste em 100 horas de gravações de reuniões. A configuração de corpus é particionada em treinamento, desenvolvimento e conjunto de avaliação. O conjunto de treinamento existe para construir modelos de aprendizagem de máquina ou algoritmos de reconhecimento de padrões. O conjunto de desenvolvimento existe para validar os modelos e algoritmos nos problemas estabelecidos (segmentação, agrupamento de locutores, etc.). O conjunto de avaliação existe para medir o desempenho final em cada problema. O conjunto de treinamento contém 4 partes gravadas de cada uma das 42 reuniões existentes, com a duração total de 75 horas e 166 locutores. O

conjunto de desenvolvimento contém 4 partes gravadas de cada uma das 11 reuniões existentes, com duração total entre 12 e 13 horas e 44 locutores (alguns locutores estão presentes em reuniões do conjunto de treinamento). O conjunto de avaliação contém 4 partes gravadas de cada uma das 6 reuniões existentes, com duração total próxima a 13 horas com 24 locutores distintos dos demais conjuntos do *Corpus*.

A Figura 27 indica que os conjuntos de treinamento e desenvolvimento são compostos de conversas rápidas, dado o número elevado de segmentos homogêneos de curta duração. A duração média observada nos segmentos de fala é menor que 1 segundo. Na Figura 28, observamos que no estilo livre, os locutores podem se sobrepor sem prévia permissão, organização ou roteiro. Esse comportamento aparece na elevada quantidade de segmentos com sobreposição de fala e na existência de uma quantidade considerável de segmentos com duração acima de 3 segundos.

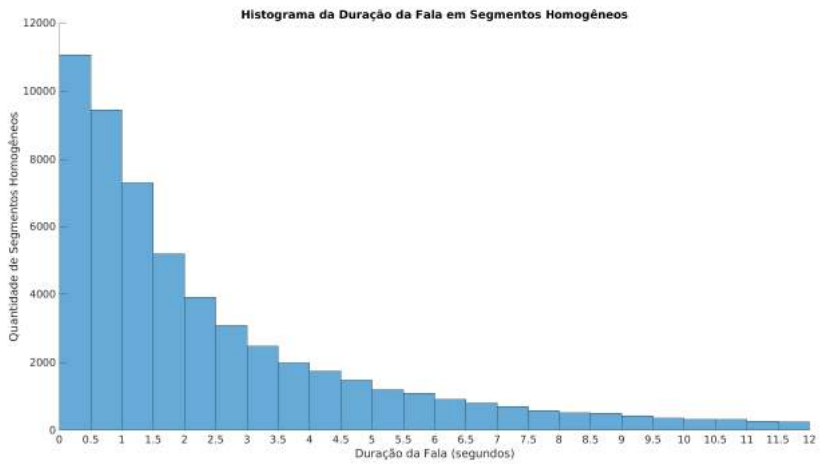

Figura 27 – Comportamento da duração da fala dos locutores nos conjuntos de treinamento e desenvolvimento da AMI *Corpus*. Observa-se que as conversações são rápidas pela quantidade elevada de segmentos homogêneos de curta duração (Δt próximo de 1 segundo).

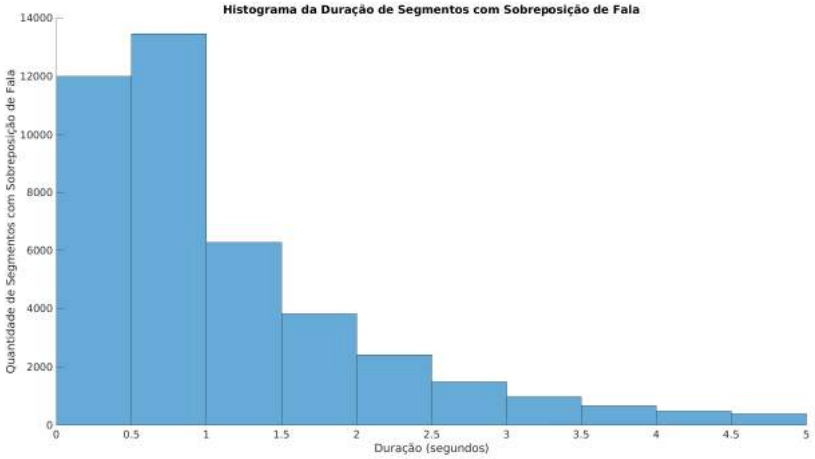

Figura 28 – Comportamento da duração dos segmentos com sobreposição de fala nos conjuntos de treinamento e desenvolvimento da AMI *Corpus*. Observa-se que os locutores conversam de forma livre, sem seguir nenhum roteiro, dada a quantidade elevada de segmentos com sobreposição de fala.

4.2 *SETUP* DE CARACTERÍSTICAS ACÚSTICAS

Diversas configurações de características acústicas precisam ser experimentadas de modo a encontrar a configuração com melhor desempenho para cada método avaliado. Detalhamos as configurações de cada método e mostramos neste trabalho apenas os resultados das configurações que com melhor desempenho.

4.2.1 Método *MFCC-LLR*

Extraímos os MFCCs com 12 e 19 coeficientes, com e sem a energia, de *frames* de voz de 20ms para cada 10 ms, usando 26 bancos de filtros Mel com largura de banda limitada à faixa de 120-3800Hz. Também avaliamos a utilização dos coeficientes Δ e Δ^2, resultando em configurações de características de 12 a 60 dimensões. Nenhuma técnica de normalização ou mapeamento é aplicada. A configuração que apresenta melhor desempenho na segmentação foi a de 12 coeficientes sem a energia e sem coeficientes delta, um total de 12 dimensões.

4.2.2 Métodos *I-VECTOR-COS* e *I-VECTOR-EUC*

Extraímos os MFCCs com 19 coeficientes e a energia, de *frames* da voz de 20ms para cada 10 ms, usando 26 bancos de filtros Mel com largura de banda limitada à faixa de 120-3800Hz. Também avaliamos a utilização dos coeficientes Δ e Δ^2, resultando em configurações de características de 20 a 60 dimensões. Nenhuma técnica de normalização ou mapeamento é aplicada. A configuração que apresenta melhor desempenho na segmentação foi a de 19 coeficientes com a energia e com os coeficientes delta, um total de 60 dimensões.

4.2.3 Métodos *MCAF-COS* e *MCAF-EUC*

Extraímos os MFCCs com 12 e 19 coeficientes, com e sem a energia, de *frames* de voz de 20ms para cada 10 ms, usando 26 bancos de filtros Mel com largura de banda limitada à faixa de 120-3800Hz. Também avaliamos a utilização dos coeficientes Δ e Δ^2, resultando em configurações de características de 12 a 60 dimensões. Nenhuma técnica de normalização ou mapeamento é aplicada. A configuração que apresenta melhor desempenho na segmentação foi a de 19 coeficientes sem a energia e sem coeficientes delta, um total de 19 dimensões.

4.3 *SETUP* PARA MODELAGEM E EXTRAÇÃO DAS CARACTERÍSTICAS DA FALA

Experimentamos diferentes configurações de modelagem e extração de características da fala para encontrar a configuração com melhor desempenho e menor custo computacional para cada método avaliado. Detalhamos as configurações de cada método e mostramos neste trabalho apenas os resultados das configurações que com melhor desempenho.

4.3.1 Método *MFCC-LLR*

Para a modelagem das características acústicas, experimentamos a distribuição Gaussiana configurando a matriz de covariância como diagonal e completa. Também experimentamos uma mistura de Gaussianas com 2 componentes configurando as matrizes de covariância como diagonal, completa e compartilhada completa (as duas componentes com mesma covariância). A distribuição e configuração com melhores resultados na segmentação foi a distribuição Gaussiana com matriz de covariância completa.

4.3.2 Métodos *I-VECTOR-COS* e *I-VECTOR-EUC*

Utilizamos os dados do conjunto de treinamento da base Fisher David04 para treinar inicialmente o UBM e a matriz de variabilidade total (TV) T. Os dados consistem em locuções de 4000 pessoas (2000 por gênero), 20 locuções por pessoa com a duração acima de 3 segundos em média, um total de 80000 locuções.

Treinamos inicialmente o modelo UBM$_{ini}$ independente de gênero de 512 distribuições com uma matriz de covariância diagonal, combinando dois UBMs dependentes de gênero de 256 componentes. Um novo UBM é treinado no conjunto de treinamento AMI usando os parâmetros do UBM$_{ini}$ como parâmetros iniciais. Utilizamos os três tipos de segmentos de fala. Todas as etapas de treinamento executam 50 iterações do algoritmo EM para estimativa dos parâmetros do modelo.

Treinamos incialmente a matriz T_{ini} para ser independente de gênero, com a configuração de rank 200 e 400 (dimensão do i-vector) utilizando 10 iterações do algoritmo EM, conforme descrito em (KENNY; DUMOUCHEL, 2004). Coletamos as estatísticas de Baum-Welch dos segmentos de fala no conjunto de treinamento da AMI utilizando quatro configurações de

tamanho de janela deslizante: 0,5, 0,75, 1 e 1,25 segundos para cada 0,1 segundo. Uma nova matriz T é treinada para cada tamanho utilizando a T_{ini} como ponto inicial. A configuração de rank que apresentou melhor desempenho na segmentação foi de rank 400.

4.3.3 Métodos *MCAF-COS* e *MCAF-EUC*

Extraímos as características propostas utilizando quatro configurações de tamanho de amostra n: 0,5, 0,75, 1 e 1,25 segundos para cada 0,1 segundo. Definimos o número de componentes principais para 12 na aplicação do PCA, que foi a dimensão mais baixa encontrada e que preserva acima de 99% da variabilidade dos dados. Configuramos o parâmetro σ com os valores: 0,25, 0,5, 0,75, 1 e 1,25. A configuração de σ que apresentou melhor resultado na segmentação foi o valor 0,75.

4.4 *SETUP* DOS MÉTODOS DE SEGMENTAÇÃO

Utilizamos o *ground-truth* dos segmentos de voz e não voz para avaliar o desempenho da segmentação de locutores em condições ideais. O conjunto de desenvolvimento da AMI *Corpus* é utilizado para escolher o tamanho da janela e o limiar de proeminência dos valores de pico da curva de distância. Os tamanhos de janela correspondem aos tamanhos da amostra avaliados para cada método: 0,5, 0,75, 1 e 1,25 segundos. O limiar de proeminência é calculado usando um parâmetro α que multiplica o desvio padrão da curva de distância. Avaliamos 2000 valores de α uniformemente distribuídos a partir de 0,1 até 0,9. O desempenho da segmentação é relatado usando a taxa de detecção de erros (MDR), taxa de alarmes falsos (FAR), F_1 *score*, taxa de precisão (PCR) e taxa de recall (RCL). Consideramos uma tolerância de 0,5 segundos para detectar as verdadeiras mudanças de locutor. A curva de erro MDR x FAR é definida para escolher o melhor ponto de operação. O melhor ponto de operação é definido como o par (MDR, FAR) da curva de erro mais próximo da origem (0,0). Selecionamos o melhor parâmetro α e tamanho da janela para cada método avaliado utilizando o melhor ponto de operação como referência. A linha tracejada que existe nas curvas de erro e de precisão x recall, ligando os pontos (0,0) e (100,100) serve como suporte para comparar uma determinada curva com as demais na direção do ponto ideal ((0,0) no caso de taxas de erro e (100,100) no caso de taxas de acerto).

As Figuras 29, 30, 31 e 32 exibem as curvas de erro MDR x FAR para as configurações de tamanho da janela de 0,5, 0,75, 1, 1,25 segundos, respectivamente. As Figuras 33, 34, 35 e 36 exibem as curvas com taxas de acerto de precisão e recall para as configurações de tamanho da janela de 0,5, 0,75, 1, 1,25 segundos, respectivamente. As Figuras 37, 38, 39 e 40 exibem as curvas de F_1 *score* para cada valor do parâmetro α, para as configurações de tamanho da janela de 0,5, 0,75, 1, 1,25 segundos, respectivamente. Compilamos os resultados dos experimentos seguindo o critério do melhor ponto de operação na curva de erro MDR x FAR. A Tabela 1 reúne as configurações de α correspondentes aos melhores pontos de operação

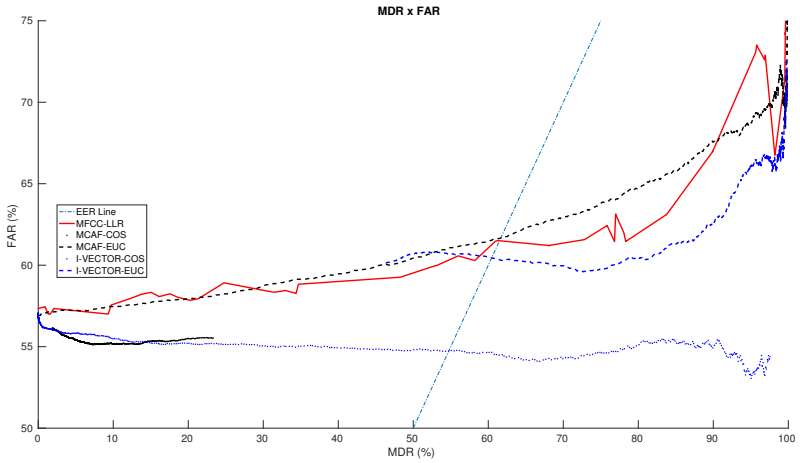

Figura 29 – Curva de erro MDR x FAR para cada método avaliado. O tamanho da janela de amostragem é de 0,5 segundos. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus* utilizando uma configuração do parâmetro α.

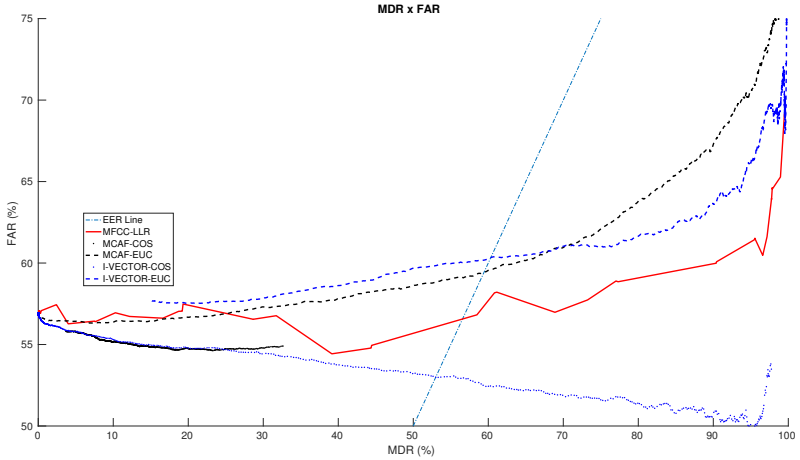

Figura 30 – Curva de erro MDR x FAR para cada método avaliado. O tamanho da janela de amostragem é de 0,75 segundos. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus* utilizando uma configuração do parâmetro α.

do método MFCC-LLR para cada tamanho de janela avaliado. Utilizamos o critério do maior F_1 *score* para escolher o tamanho da janela. Da mesma forma, as Tabelas 2 e 3 reúnem as configurações de α correspondentes aos melhores pontos de operação para as características i-vector e MCAF, respectivamente, considerando as variações de distância e tamanho de janela. Utilizamos o critério de maior F_1 *score* para selecionar o tamanho da janela.

A Tabela 4 resume os melhores resultados no conjunto de desenvolvimento da AMI *Corpus* para as características, modelos e distâncias avaliadas: MFCCs, distribuição Gaussiana e distância LLR; i-vector e distância do cosseno; e as características MCAF e distância euclidiana.

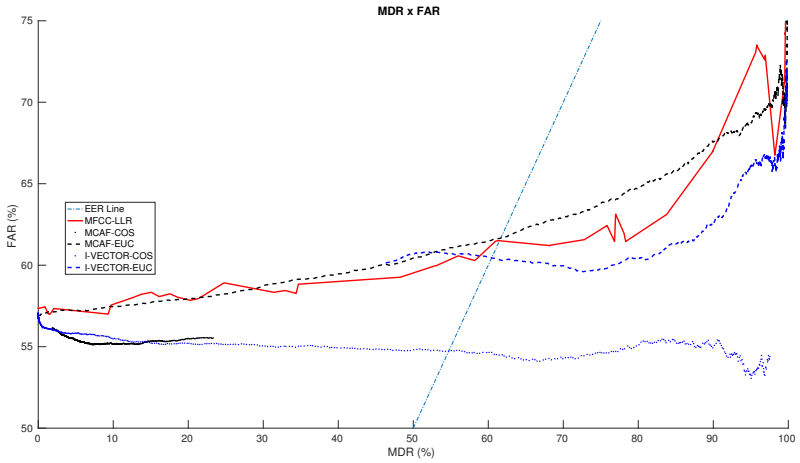

Figura 31 – Curva de erro MDR x FAR para cada método avaliado. O tamanho da janela de amostragem é de 1 segundo. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus* utilizando uma configuração do parâmetro α.

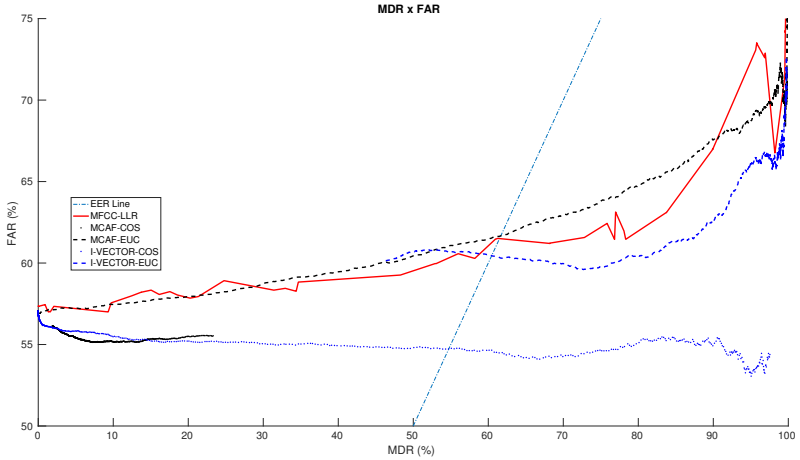

Figura 32 – Curva de erro MDR x FAR para cada método avaliado. O tamanho da janela de amostragem é de 1,25 segundos. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus* utilizando uma configuração do parâmetro α.

Já a Tabela 5 exibe os resultados obtidos pelos métodos da Tabela 4 no conjunto de avaliação da AMI *Corpus*. As configurações de tamanho de janela e α de cada método estão descritas nas Tabelas 1, 2 e 3.

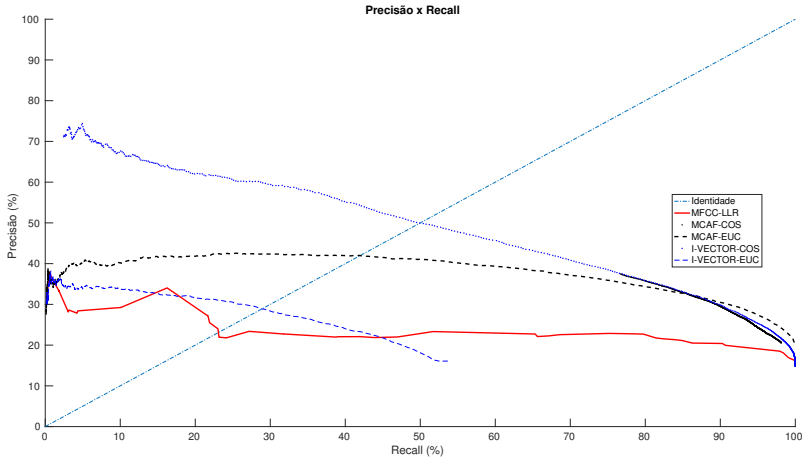

Figura 33 – Curva de PCR x RCL para cada método avaliado. O tamanho da janela de amostragem é de 0,5 segundos. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus* utilizando uma configuração do parâmetro α.

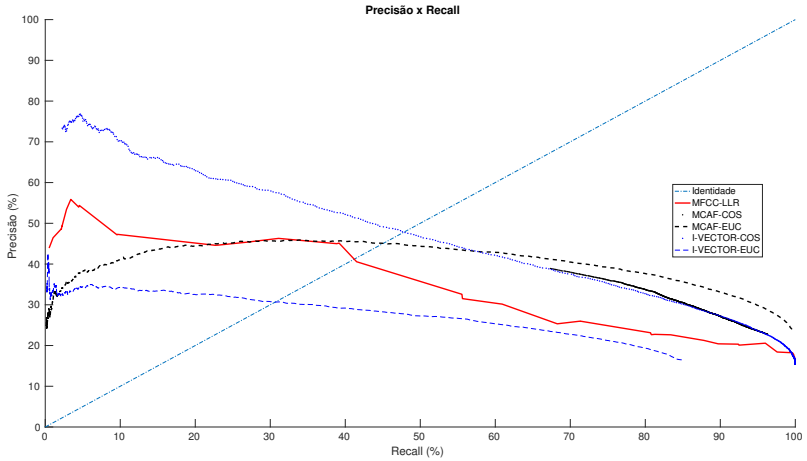

Figura 34 – Curva de PCR x RCL para cada método avaliado. O tamanho da janela de amostragem é de 0,75 segundos. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus* utilizando uma configuração do parâmetro α.

Tabela 1 – Compilação dos resultados do método MFCC-LLR correspondentes aos melhores pontos de operação das curvas MDR x FAR. O destaque em negrito indica qual configuração de tamanho de janela e α serão selecionados para o conjunto de avaliação. O critério utilizado é o maior valor de F_1 *score*.

Método	MDR (%)	FAR (%)	F_1 score	α
MFCC-LLR ($n = 0,5$ segundos)	1,36	57,00	30,01	0,45
MFCC-LLR (n = 0,75 segundos)	4,00	56,27	**33,88**	**0,50**
MFCC-LLR ($n = 1$ segundo)	0,08	56,98	27,26	0,10
MFCC-LLR ($n = 1,25$ segundos)	0,16	57,10	28,80	0,50

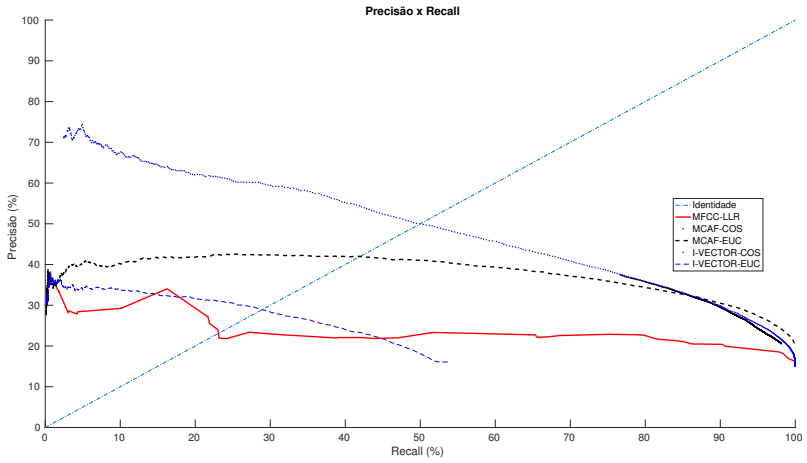

Figura 35 – Curva de PCR x RCL para cada método avaliado. O tamanho da janela de amostragem é de 1 segundo. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus* utilizando uma configuração do parâmetro α.

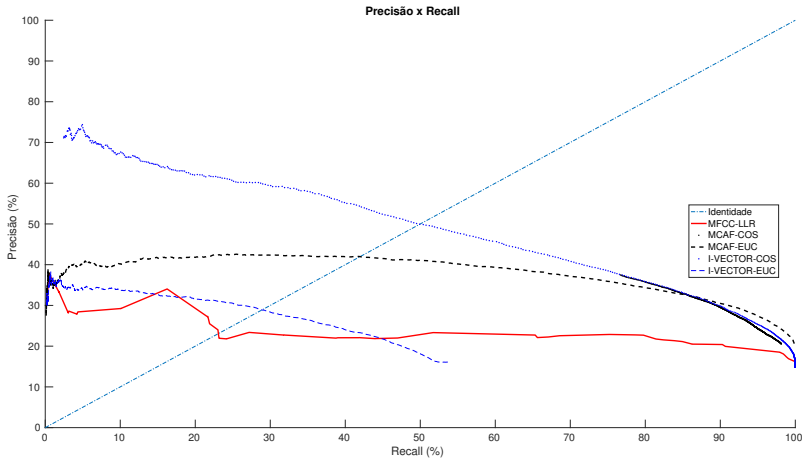

Figura 36 – Curva de PCR x RCL para cada método avaliado. O tamanho da janela de amostragem é de 1,25 segundos. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus* utilizando uma configuração do parâmetro α.

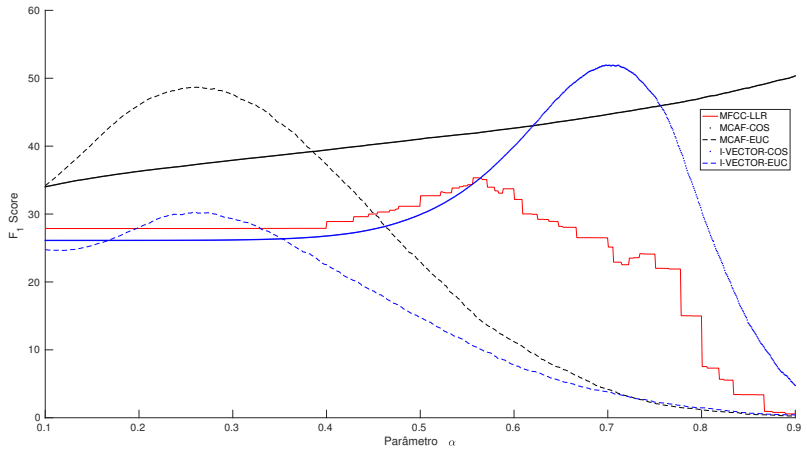

Figura 37 – Curva de F_1 *score* para cada parâmetro α de cada método avaliado. O tamanho da janela de amostragem é de 0,5 segundos. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus*.

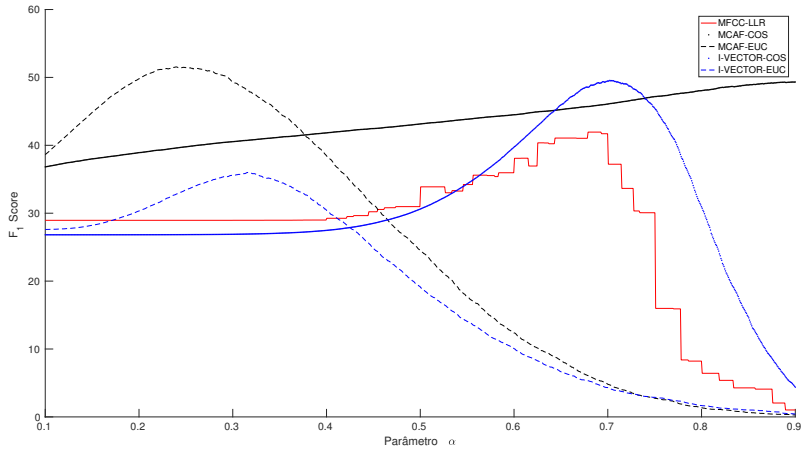

Figura 38 – Curva de F_1 *score* para cada parâmetro α de cada método avaliado. O tamanho da janela de amostragem é de 0,75 segundos. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus*.

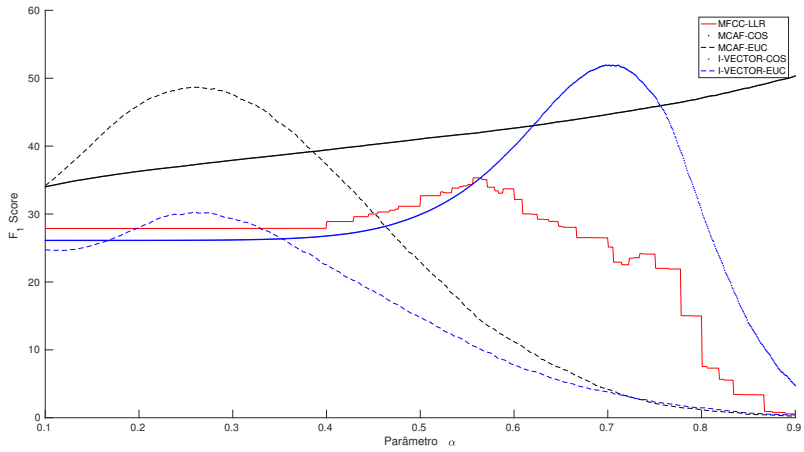

Figura 39 – Curva de F_1 *score* para cada parâmetro α de cada método avaliado. O tamanho da janela de amostragem é de 1 segundo. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus*.

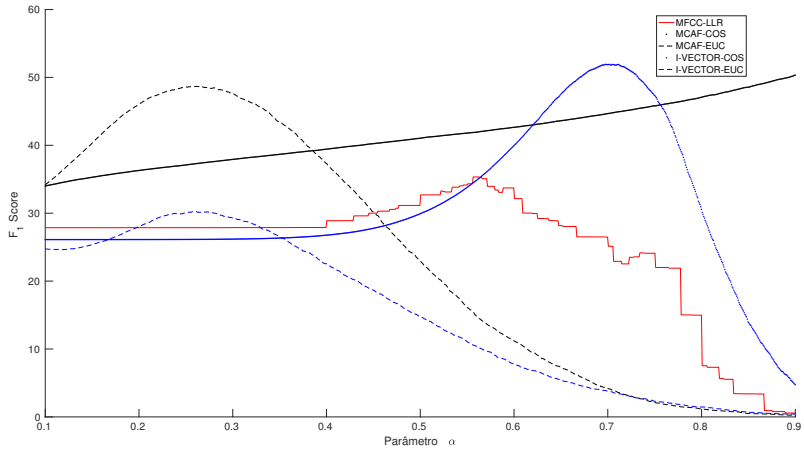

Figura 40 – Curva de F_1 *score* para cada parâmetro α de cada método avaliado. O tamanho da janela de amostragem é de 1,25 segundos. Cada ponto da curva é o resultado da segmentação das conversas do conjunto de desenvolvimento da AMI *Corpus*.

Tabela 2 – Compilação dos resultados do i-vector correspondentes aos melhores pontos de operação das curvas MDR x FAR, considerando também as variações de distância. O destaque em negrito indica qual medida de distância, configuração de tamanho de janela e α serão selecionados para o conjunto de avaliação. O critério utilizado é o maior valor de F_1 *score*.

Método	MDR (%)	FAR (%)	F_1 score	α
I-VECTOR-COS ($n = 0,5$ segundos)	3,30	55,84	38,00	0,59
I-VECTOR-COS ($n = 0,75$ segundos)	6,27	55,61	39,17	0,60
I-VECTOR-COS ($n = 1$ segundo)	5,61	55,13	38,58	0,57
I-VECTOR-COS (n = 1, 25 segundos)	6,87	53,65	**39,44**	**0,57**
I-VECTOR-EUC ($n = 0,5$ segundos)	46,33	60,15	24,76	0,10
I-VECTOR-EUC ($n = 0,75$ segundos)	15,16	57,67	27,61	0,10
I-VECTOR-EUC ($n = 1$ segundo)	1,13	56,56	30,92	0,18
I-VECTOR-EUC ($n = 1,25$ segundos)	0,26	56,83	29,52	0,14

Tabela 3 – Compilação dos resultados das características MCAF correspondentes aos melhores pontos de operação das curvas MDR x FAR, considerando também as variações de distância. O destaque em negrito indica qual medida de distância, configuração de tamanho de janela e α serão selecionados para o conjunto de avaliação. O critério utilizado é o maior valor de F_1 *score*.

Método	MDR (%)	FAR (%)	F_1 score	α
MCAF-COS ($n = 0,5$ segundos)	5,63	55,30	39,79	0,42
MCAF-COS ($n = 0,75$ segundos)	8,16	55,28	40,43	0,29
MCAF-COS ($n = 1$ segundo)	6,08	55,24	38,46	0,13
MCAF-COS ($n = 1,25$ segundos)	7,13	55,27	39,01	0,11
MCAF-EUC ($n = 0,5$ segundos)	0,18	56,90	34,59	0,10
MCAF-EUC ($n = 0,75$ segundos)	1,38	56,46	41,19	0,12
MCAF-EUC ($n = 1$ segundo)	3,63	55,83	44,32	0,14
MCAF-EUC (n = 1,25 segundos)	3,33	55,89	**45,19**	**0,13**

Tabela 4 – Resumo dos melhores resultados no conjunto de desenvolvimento da AMI *Corpus*. Os destaques em negrito para cada taxa indicam os melhores resultados naquela taxa. O destaque em negrito para o método indica aquele que possui as melhores taxas.

Método	MDR (%)	FAR (%)	F_1 score
MFCC-LLR	4,00	56,27	33,88
I-VECTOR-COS	6,87	**53,65**	39,44
MCAF-EUC	**3,33**	55,89	**45,19**

Tabela 5 – Resultados obtidos no conjunto de avaliação da AMI *Corpus*. Os destaques em negrito para cada taxa indicam os melhores resultados naquela taxa. O destaque em negrito para o método indica aquele que possui as melhores taxas.

Método	MDR (%)	FAR (%)	F_1 score
MFCC-LLR	**4,79**	61,63	30,63
I-VECTOR-COS	9,03	**59,19**	36,86
MCAF-EUC	5,21	61,05	**42,43**

4.5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Através dos resultados de segmentação obtidos no conjunto de desenvolvimento, ao analisar as curvas de Precisão x Recall (Figuras 33, 34, 35 e 36), podemos observar que a maioria dos métodos que utilizam as características MCAF possuem desempenho superior ao método tradicional que utiliza as características MFCCs e a distância LLR. Com o auxílio da reta chamada de "Identidade", podemos identificar com mais facilidade o ponto de operação no qual a precisão tem o mesmo valor do recall (ponto de intersecção com da curva com a reta identidade). Nas configurações de tamanho de janela avaliadas, a maioria dos pontos de intersecção dos métodos que utilizam as MCAF são mais próximos do valor ideal (100, 100) do que os pontos do método tradicional. Esse mesmo tipo de observação também pode ser feita para o i-vector em comparação com o método tradicional. Analisando os valores máximos da F_1 *score* (Figuras 37, 38, 39 e 40), considerando todas as configurações de tamanho de janela, percebemos que o método tradicional não consegue desempenho superior aos métodos que utilizam i-vector e MCAF.

Ainda através dos resultados obtidos no conjunto de desenvolvimento, não é possível observar nas tanto nas curvas MDR x FAR (Figuras 29, 30, 31 e 32) e Precisão x Recall que os métodos que utilizam as MCAFs possuem alguma vantagem. Apenas analisando os valores máximos de F_1 *score* pode-se observar desempenho competitivo. Ao selecionar os melhores pontos de operação nas curvas MDR x FAR nas Tabelas 2 e 3 , voltamos a observar competitividade entre i-vector e MCAF. Na Tabela 3, vemos que alguns dos valores de F_1 *score* para as MCAFs estão perto dos maiores valores observados nas Figuras 37, 38, 39 e 40 (acima de 40 e abaixo de 50). E estes valores correspondem aos melhores pontos de operação nas curvas MDR x FAR. Este fato não ocorre para o i-vector, basta observar na Tabela 2 que os melhores pontos de operação possuem valores de F_1 *score* que não estão entre os maiores nas Figuras 37, 38, 39 e 40 (acima de 40 e abaixo de 50). Isso indica que, apesar das taxas de perda e falsos alarmes serem parecidas, a natureza que origina essas taxas é diferente, pois as taxas de precisão e recall levam a diferentes valores de F_1 *score*. Com o F_1 *score* mais elevado, e com taxas de perda e falso alarme parecidas, temos que a quantidade de detecções de mudança é menor. Um menor número detecções para uma quantidade parecida de verdadeiros positivos indica uma maior taxa de precisão. Ao selecionarmos a configuração de janela e distância, observamos que as características MCAF obtém o maior F_1 *score* e menor taxa de perda. No quesito custo computacional (Tabelas 6, 7, 8 e 9), observamos vantagens para a utilização das MCAFs, pois o tempo de processamento médio em segundos é cerca de 10 vezes menor que o tempo necessário para o i-vector (a implementação dos processos foi feita para utilização de GPU), e tem a mesma ordem de grandeza do tempo médio de processamento do método tradicional.

Nas Tabelas 4 e 5 temos os melhores desempenhos encontrados no conjunto de desenvolvimento e o desempenho correspondente às mesmas configurações de distância, tamanho de janela e α no conjunto de avaliação. Os desempenhos são consistentes para a taxa de

falso alarme e F_1 *score*. Porém a taxa de perda mostrou-se melhor para o método tradicional. As taxas de perda e falso alarme foram maiores para o conjunto de avaliação do que o de desenvolvimento, comportamento esperado, considerando que são dados não vistos e que podem ter mais locutores e estilo de conversa mais rápido, ou não, que o estilo visto no conjunto de desenvolvimento. Dadas essas condições, também é esperado o deterioramento dos valores F_1 *score*.

Tabela 6 – Tempo de processamento para execução dos principais processos da segmentação com tamanho de janela de 0,5 segundos. Coletamos o tempo para cada conversação processada nos experimentos.

Processo	Tempo Médio (segs)	Desvio (segs)
Extração do i-vector	407	108
Extração das MCAF	33	8
Curvas de Distância LLR	17	0
Curvas de Distância Cosseno para MCAF	0.0017	0.0254
Curvas de Distância Euclidiana para MCAF	0.0011	0.0078
Curvas de Distância Cosseno para ivector	0.0013	0.0078
Curvas de Distância Euclidiana para ivector	0.0021	0.0862

Tabela 7 – Tempo de processamento para execução dos principais processos da segmentação com tamanho de janela de 0,75 segundos. Coletamos o tempo para cada conversação processada nos experimentos.

Processo	Tempo Médio (segs)	Desvio (segs)
Extração do i-vector	409	111
Extração das MCAF	39	7
Curvas de Distância LLR	18	0
Curvas de Distância Cosseno para MCAF	0.0007	0.0005
Curvas de Distância Euclidiana para MCAF	0.0013	0.0006
Curvas de Distância Cosseno para i-vector	0.0012	0.0006
Curvas de Distância Euclidiana para i-vector	0.0021	0.0005

As vantagens em utilizar o método proposto são o fato de não precisarmos de base treinamento para aprendizagem dos padrões dos tipos de segmentos, ou locutores, todo o processamento e modelagem funciona para cada conversação, de forma independente das demais conversações, e o fato do método de segmentação por tipos de segmento ter demonstrado consegue ser mais preciso nas detecções de mudança, do que o método tradicional que busca discriminar locutores, o que pode ser evidenciado nos resultados reportados pelo F_1 *score* mais elevado que o estado da arte i-vector, mantendo as taxas de perda em um mesmo patamar, indicando por sua vez, que o método consegue detectar as mudanças sem sofre com o problema de segmentação demasiada (algoritmo de detecção que na maioria dos casos vai "chutar"que

Tabela 8 – Tempo de processamento para execução dos principais processos da segmentação com tamanho de janela de 1 segundo. Coletamos o tempo para cada conversação processada nos experimentos.

Processo	Tempo Médio (segs)	Desvio (segs)
Extração do i-vector	410	100
Extração das MCAF	80	7
Curvas de Distância LLR	18	0
Curvas de Distância Cosseno para MCAF	0.0007	0.0003
Curvas de Distância Euclidiana para MCAF	0.0011	0.0005
Curvas de Distância Cosseno para i-vector	0.0012	0.0005
Curvas de Distância Euclidiana para i-vector	0.0021	0.0005

Tabela 9 – Tempo de processamento para execução dos principais processos da segmentação com tamanho de janela de 1,25 segundos. Coletamos o tempo para cada conversação processada nos experimentos.

Processo	Tempo Médio (segs)	Desvio (segs)
Extração do i-vector	411	108
Extração das MCAF	139	7
Curvas de Distância LLR	18	0
Curvas de Distância Cosseno para MCAF	0.0007	0.0003
Curvas de Distância Euclidiana para MCAF	0.0011	0.0005
Curvas de Distância Cosseno para i-vector	0.0012	0.0005
Curvas de Distância Euclidiana para i-vector	0.0021	0.0005

existe mudança de locutor). Ainda como vantagens, a proposta obtém o menor tempo de processamento e pode paralelizar em GPU a computação das etapas com as matrizes e a estimativa do PCA.

5 CONCLUSÃO

Este trabalho propõe uma nova abordagem para segmentação de locutores através da detecção de transições entre 3 diferentes tipos de segmento de fala: Homogêneo, heterogêneo e com sobreposição de fala *overlap*. Consideramos que os tipos de segmento possuem estilos diferentes da fala, então desenvolvemos características para representar de forma distinta as informações dos diferentes estilos. Neste trabalho, definimos estilo da fala como uma forma de expressá-la de forma individual. Assumimos as seguintes premissas de caracterização do estilo de fala de um segmento:

- Segmento homogêneo: Representação da informação do estilo da fala de um indivíduo;
- Segmento heterogêneo: Representação da informação de ao menos dois estilos de fala existentes em intervalos de tempo diferentes;
- Segmento com (*overlap*): Representação da sobreposição de dois estilos de fala.

As caraterísiticas nomeadas Mel Cepstral Affinity Features (MCAFs) são extraídas diretamente das características *mel* (MFCCs) computadas para cada segmento através do cálculo da matriz de afinidade. O trabalho mostra que tanto as matrizes de correlação das MCAFs quanto os vetores de média têm aspecto diferente para cada tipo de segmento definido.

As MCAFs são utilizadas no método de segmentação de duas janelas deslizantes utilizando as distâncias euclidiana e cosseno para detectar as transições os 3 tipos de segmentos. Realizamos experimentos utilizando o *corpus* da AMI e avaliamos métodos tradicionais e estado da arte de segmentação baseados em distâncias utilizando tanto os coeficientes MFCCs quanto o i-vector para caracterizar os locutores, e considerando a métrica de log da verossimelhança (LLR) como distância para comparar os modelos estimados a partir dos MFCCs, assim como as distâncias euclidiana e cosseno para comparar os i-vectors. Os resultados mostram que as MCAFs obtêm desempenho comparável ao i-vector e possui menor tempo para processamento. O desempenho obtido pelas MCAFs é superior aos métodos tradicionais, porém tem um tempo de processamento superior.

As conclusões desse trabalho são que a representação do estilo da fala contida em segmentos pode ser caracterizada de tal forma que os 3 tipos definidos para os segmentos tornem-se distintos. Essa distinção pode ser utilizada para segmentar os locutores em conversação de estilo livre, pois o cenário apresenta os três tipos de segmento ao redor das transições entre locutores. A abordagem não necessita de grandes amostras da fala para discriminar os tipos e não utiliza modelos ou uma fase de aprendizado para discriminação de locutores, o que torna a mesma menos custosa computacionalmente que o estado da arte i-vector. Os resultados obtidos são compatíveis aos do i-vector e demonstram ser superiores aos métodos tradicionais,

viabilizando a aplicação da abordagem em sistemas de transcrição de locutores. Sendo assim, o trabalho alcança os objetidos definidos para a pesquisa.

Para continuação dos trabalhos, propomos a aplicação das MCAFs para detectar e remover segmentos com sobreposição da fala como pós processamento da segmentação de locutores, visando melhorar a pureza dos grupos formados na etapa final de transcrição de locutores. Propomos também abordagens supervisionadas de segmentação, considerando os 3 tipos de segmento como classes, de modo que será utilizada apenas uma janela deslizante e o modelo para classificação. O complemento da probabilidade de um segmento ser do tipo homogêneo

$$\bar{p_{hom}} = 1 - p_{hom}$$

pode ser interpretado como a distância na detecção das transições entre locutores. Mais domínios da aplicação de transcrição podem ser considerados como conversações telefônicas e noticiários, para avaliação de como/se a informação do canal de gravação do áudio afeta o comportamento das características MCAF.

Além dos cenários e aplicações em diarização de locutores, outros trabalhos com foco nas características do espaço de afinidade podem ser desenvolvidos, avaliando o impacto de outras características Cepstrais, ou até mesmo computando a afinidade no próprio espectro sem filtros e sem a transformada DCT. Assim como avaliar com outras técnicas de extração de variáveis latentes do espaço de afinidade original, como a aplicação de *Deep Neural Networks*.

REFERÊNCIAS

AJMERA, J.; BOURLARD, H.; LAPIDOT, I. Unknown-multiple speaker clustering using HMM. In: *International Conference Spoken Language Processing*. Denver: [s.n.], 2002. p. 573–576.

AJMERA, J.; LATHOUD, G.; MCCOWAN, I. Clustering and segmenting speakers and their locations in meetings. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*. Montreal: [s.n.], 2004. v. 1, p. 605–608.

AJMERA, J.; WOOTERS, C. A robust speaker clustering algorithm. In: *IEEE Workshop on Automatic Speech Recognition and Understanding*. St Thomas: [s.n.], 2003. p. 411 – 416.

ANGUERA, X.; HERNANDO, J. Xbic: Real-time cross probabilities measure for speaker segmentation. *ICSI - Berkeley Technical Report*, v. 1, n. 1, p. 05–08, 2005.

ANGUERA, X.; WOOTERS, C.; PARDO, J. M. Robust Speaker Diarization for meetings. In: *International Workshop on Machine Learning for Multimodal Interaction*. Berlim: [s.n.], 2006. p. 346–358.

ANGUERA, X.; WOOTERS, C.; PARDO, J. M. Robust Speaker Diarization for Meetings: ICSI RT06S Meetings Evaluation System. In: *Proceedings of the Third International Conference on Machine Learning for Multimodal Interaction*. Berlim: [s.n.], 2006. p. 346–358.

APPEL, U.; BRANDT, A. V. Adaptive sequential segmentation of piecewise stationary time series. *Information Sciences*, v. 29, n. 1, p. 27–56, 1983.

BAKIS, R.; CHEN, S.; GOPALAKRISHNAN, P. Transcription of broadcast news shows with the IBM large vocabulary speech recognition system. In: *Proceedings of the Speech Recognition Workshop*. Chantilly: [s.n.], 1997. v. 87, p. 4–9.

BARRAS, C.; ZHU, X.; MEIGNIER, S.; GAUVAIN, J. L. Improving Speaker Diarization. In: *Proceedings of the Fall 2004 Rich Transcription Workshop*. Palisades: [s.n.], 2004. p. 1–7.

BONASTRE, J.-F.; DELACOURT, P.; FREDOUILLE, C.; MERLIN, T.; WELLEKENS, C. A speaker tracking system based on speaker turn detection for NIST evaluation. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Istambul: [s.n.], 2000. v. 2, p. 1177–1180.

CAMPBELL, J. P. Speaker recognition: a tutorial. *Proceedings of the IEEE*, v. 85, n. 9, p. 1437–1462, 1997.

CASTALDO, F.; COLIBRO, D.; DALMASSO, E.; LAFACE, P.; VAIR, C. Stream-based speaker segmentation using speaker factors and eigenvoices. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*. [S.l.: s.n.], 2008. p. 4133—-4136.

CETTOLO, M.; VESCOVI, M. Efficient audio segmentation algorithms based on the BIC. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Hong-Kong: [s.n.], 2003. v. 6, p. 537–540.

CHEN, K.; SALMAN, A. Learning speaker-specific characteristics with a deep neural architecture. *IEEE Transactions on Neural Networks*, v. 22, n. 11, p. 1744–1756, 2011.

CHEN, S.; GOPALAKRISHNAN, P. Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion. In: *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. Lansdowne: [s.n.], 1998. p. 67–72.

CHEN, S. S.; EIDE, E.; GALES, M. J. F.; GOPINATH, R. A.; KANVESKY, D.; OLSEN, P. Automatic transcription of Broadcast News. *Speech Communication*, v. 37, n. 1-2, p. 69–87, 2002.

CHEN, S. S.; GOPALAKRISHNAN, P. S. Clustering via the Bayesian information criterion with applications in speech recognition. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Seattle: [s.n.], 1998. v. 2, p. 645–648.

CHENG, S. S.; WANG, H. M. A sequential metric-based audio segmentation method via the Bayesian information criterion. In: *International Conference Spoken Language Processing*. Geneva: ISCA, 2003. p. 945–948.

CHENG, S. S.; WANG, H. M.; FU, H. C. BIC-Based Speaker Segmentation Using Divide-and-Conquer Strategies With Application to Speaker Diarization. *IEEE Transactions On Audio, Speech, and Language Processing*, v. 18, n. 1, p. 141–157, 2010.

CRYSTAL, D. *A dictionary of linguistics and phonetics*. [S.l.]: John Wiley & Sons, 2011.

DAVIS, S.; MERMELSTEIN, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 28, n. 4, p. 357–366, 1980.

DEHAK, N.; KENNY, P. J.; DUMOUCHEL P., O. P. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, v. 19, n. 4, p. 788–798, 2011.

DELACOURT, P.; KRYZE, D.; WELLEKENS, C. Detection of speaker changes in an audio document. In: *European Conference Spoken Language Processing*. Budapeste: ISCA, 1999. p. 1195–1198.

DELACOURT, P.; WELLEKENS, C. DISTBIC: A speaker-based segmentation for audio data indexing. *Speech Communication*, v. 32, n. 1-2, p. 111–126, 2000.

DESPLANQUES, B.; DEMUYNCK, K.; MARTENS, J. P. Factor analysis for speaker segmentation and improved speaker diarization. In: *International Speech Communication Association, INTERSPEECH*. [S.l.: s.n.], 2015. p. 3081–3085.

EVANS, N.; BOZONNET, S.; WANG, D.; FREDOUILLE, C.; TRONCY, R. A Comparative Study of Bottom-Up and Top-Down Approaches to Speaker Diarization. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 20, n. 2, p. 382–392, 2012.

FLANAGAN, J.; ALLEN, J.; HASAGAWA-JOHNSON, M. *Speech Analysis Synthesis and Perception*. [S.l.]: Springer, 2008.

FREDOUILLE, C.; MORARU, D.; MEIGNIER, S.; BESACIER, L.; BONASTRE, J.-F. The NIST 2004 spring rich transcription evaluation: Two-axis merging strategy in the context of multiple distant microphone based meeting speaker segmentation. In: *NIST 2004 Rich Transcription Evaluation Workshop*. Montreal: [s.n.], 2004. p. 15–20.

GATYS, L. A.; ECKER, A. S.; BETHGE, M. Image style transfer using convolutional neural networks. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2016. p. 2414–2423.

GAUVAIN, J.-L.; LAMEL, L.; ADDA, G. Partitioning and transcription of broadcast news data. In: *International Conference Spoken Language Processing*. Sidney: [s.n.], 1998. v. 98, n. 5, p. 1335–1338.

GISH, H.; SIU, M.-H.; ROHLICEK, R. Segregation of speakers for speech recognition and speaker identification. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Toronto: [s.n.], 1991. p. 873–876.

GUPTA, V. Speaker change point detection using deep neural nets. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Brisbane, Austrália: [s.n.], 2015. p. 4420–4424.

HAIN, T.; JOHNSON, S. E.; TUERK, A.; WOODLAND, P. C.; YOUNG, S. J. Segment Generation and Clustering in the HTK Broadcast News Transcription System. In: *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. Lansdowne: [s.n.], 1998. p. 133–137.

HARRINGTON, J.; CASSIDY, S. *Techniques in Speech Acoustics*. 26. ed. [S.l.]: Springer, 2000. v. 26. 294–295 p.

HAUTAMÄKI, V.; CHENG, Y. C.; RAJAN, P.; LEE, C. H. Minimax i-vector extractor for short duration speaker verification. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. [S.l.: s.n.], 2013. p. 3708–3712.

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

HUBERT, J.; FRANCIS, K.; SCHWARTZ, R. Automatic Speaker Clustering. In: *Proceedings of the DARPA Speech Recognition Workshop*. Chantilly: [s.n.], 1997. v. 2, p. 1–9.

HUNG, J.-W.; WANG, H.-M.; LEE, L.-S. Automatic metric-based speech segmentation for broadcast news via principal component analysis. In: *International Conference Spoken Language Processing*. Pequim: ISCA, 2000. p. 121–124.

JOHNSON, S. E. Who spoke when?-Automatic segmentation and clustering for determining speaker turns. In: *European Conference Spoken Language Processing*. Budapeste: [s.n.], 1999. p. 2211–2214.

KARTIK, V.; SATISH, D. S.; SEKHAR, C. C. Speaker Change Detection using Support Vector Machines. In: *ISCA Tutorial and Research Workshop on Non-Linear Speech Processing (NOLISP'05)*. Madison: [s.n.], 2005. p. 130–136.

KEMP, T.; SCHMIDT, M.; WESTPHAL, M.; WAIBEL, A. Strategies for automatic segmentation of audio data. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Istambul: [s.n.], 2000. v. 3, p. 1423–1426.

KENNY, P. Bayesian speaker verification with heavy-tailed priors. In: *The speaker and language recognition workshop (Odyssey)*. [S.l.: s.n.], 2010. p. 14.

KENNY, P.; DUMOUCHEL, P. Disentangling speaker and channel effects in speaker verification. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*. [S.l.: s.n.], 2004. p. 37—-40.

KENNY, P.; REYNOLDS, D.; CASTALDO, F. Diarization of Telephone Conversations using Factor Analysis. *IEEE Journal of Selected Topics in Signal Processing*, v. 4, n. 6, p. 1–7, 2010.

KIM, H.-G.; ERTELT, D.; SIKORA, T. Hybrid Speaker-Based Segmentation System Using Model-Level Clustering. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Filadélfia: [s.n.], 2005. v. 1, p. 745–748.

KINNUNEN, T.; ZHANG, B.; ZHU, J.; WANG, Y. Speaker verification with adaptive spectral subband centroids. In: *Advances in Biometrics*. Xx. [S.l.]: Springer, 2007. p. 58–66.

KOHONEN, T. The self-organizing map. *Proceedings of the IEEE*, v. 78, n. 9, p. 1464–1480, 1990.

KOTTI, M.; MOSCHOU, V.; KOTROPOULOS, C. Speaker segmentation and clustering. *Signal Processing*, v. 88, n. 5, p. 1091–1124, 2008.

KUBALA, F.; JIN, H.; MATSOUKAS, S.; NGUYEN, L.; SCHWARTZ, R.; MAKHOUL, J. The 1996 BBN byblos HUB-4 transcription system. In: CITESEER. *Proceedings of the DARPA Speech Recognition Workshop*. Chantilly, 1997. p. 90–93.

LAPIDOT, I. SOM as likelihood estimator for speaker clustering. In: *International Conference Spoken Language Processing*. Geneva: ISCA, 2003. p. 3001–3004.

LAPIDOT, I.; GUTERMAN, H.; COHEN, A. Unsupervised speaker recognition based on competition between self-organizing maps. *IEEE Transactions on Neural Networks*, v. 13, n. 4, p. 877–887, jul 2002.

LI, Y.; LIU, M.-Y.; LI, X.; YANG, M.-H.; KAUTZ, J. A closed-form solution to photorealistic image stylization. In: FERRARI, V.; HEBERT, M.; SMINCHISESCU, C.; WEISS, Y. (Ed.). *European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 468–483.

LI, Y.; WANG, N.; LIU, J.; HOU, X. Demystifying neural style transfer. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2017. (IJCAI'17), p. 2230–2236.

LILT, D.; KUBALA, F. Online speaker clustering. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Montreal: [s.n.], 2004. v. 1, p. 333–339.

LIN, P. C.; WANG, J. C.; WANG, J. F.; SUNG, H. C. Unsupervised speaker change detection using SVM training misclassification rate. *IEEE Transactions on Computers*, v. 56, n. 9, p. 1234–1244, 2007.

LU, L.; ZHANG, H. Real-Time Unsupervised Speaker Change Detection. In: *International Conference on Pattern Recognition*. Quebec: [s.n.], 2002. p. 358–361.

MA, Y.; BAO, C.-C. Sparse DNN-based speaker segmentation using side information. *Electronics Letters*, v. 51, n. 8, 2015.

MAKHOUL, J. Linear prediction: A tutorial review. *Proceedings of the IEEE*, v. 63, n. 4, p. 561–580, 1975.

MAMMONE, R. J.; ZHANG, X. Z. X.; RAMACHANDRAN, R. P. Robust speaker recognition: a feature-based approach. *IEEE Signal Processing Magazine*, v. 13, n. 5, p. 58–62, 1996.

MEIGNIER, S.; BONASTRE, J.-F.; IGOUNET, S. E-HMM approach for learning and adapting sound models for speaker indexing. In: *ODYSSEY-2001 - The Speaker Recognition Workshop*. Creta: [s.n.], 2001. p. 175–180.

MEIGNIER, S.; MORARU, D.; FREDOUILLE, C.; BESACIER, L.; BONASTRE, J. Benefits of prior acoustic segmentation for automatic speaker segmentation. *IEEE International Conference on Acoustics, Speech and Signal Processing*, v. 1, n. 1, p. 397–400, 2004.

MEINEDO, H.; NETO, J. Audio segmentation, classification and clustering in a broadcast news task. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Hong-Kong: [s.n.], 2003. v. 2, p. 2–5.

MIRO, X. A.; BOZONNET, S. Speaker diarization: A review of recent research. *IEEE Transactions On Audio, Speech, and Language Processing*, v. 20, n. 2, p. 356–370, 2012.

MIRó, X. A.; PERICAS, J. H. Evolutive speaker segmentation using a repository system. In: *International Conference Spoken Language Processing*. Jeju Island: ISCA, 2004. p. 605–608.

MOATTAR, M. H.; HOMAYOUNPOUR, M. M. A review on speaker diarization systems and approaches. *Speech Communication*, v. 15, 2012.

MORARU, D.; MEIGNIER, S.; FREDOUILLE, C.; BESACIER, L.; BONASTRE, J. The ELISA consortium approaches in broadcast news speaker segmentation during the NIST 2003 rich transcription evaluation. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Montreal: [s.n.], 2004. v. 1, p. 373–376.

NERI, L. V.; PINHEIRO, H. N. B.; TSANG, I. R.; CAVALCANTI, G. D. C.; ADAMI, A. G. Speaker segmentation using i-vectors in meetings domain. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*. [S.l.: s.n.], 2017. p. 5455–5459.

NERI, L. V.; REN, T. I.; CAVALCANTI, G. D. C.; JYH, T. I.; SIJBERS, J. A Combined Features Approach for Speaker Segmentation Using BIC and Artificial Neural Networks. In: *IEEE International Conference on Systems, Man, and Cybernetics*. Manchester: [s.n.], 2013. p. 4332–4335.

NGUYEN, P. SWAMP: An isometric frontend for speaker clustering. In: *NIST 2003 Rich Transcription Evaluation Workshop*. Boston: [s.n.], 2003. p. 24–28.

OUELLET, P.; BOULIANNE, G.; KENNY, P. Flavors of Gaussian warping. In: *International Conference Spoken Language Processing*. Lisboa: ISCA, 2005. p. 2957–2960.

PELECANOS, J.; SRIDHARAN, S. Feature Warping for Robust Speaker Verification. In: *ODYSSEY-2001 - The Speaker Recognition Workshop*. Creta: [s.n.], 2001. p. 213–218.

PODDAR, A.; SAHIDULLAH, M.; SAHA, G. Improved i-vector extraction technique for speaker verification with short utterances. *International Journal of Speech Technology*, v. 21, n. 3, p. 473–488, 2018.

PODDAR, A.; SAHIDULLAH, M.; SAHA, G. Speaker verification with short utterances: a review of challenges, trends and opportunities. *International Journal of Speech Technology*, v. 7, n. 2, p. 91–101, 2018.

RABINER, L. L.; JUANG, B.-H. B. *Fundamentals of Speech Recognition*. 1. ed. [S.l.]: PTR Prentice Hall, 1993. v. 103.

REYNOLDS, D. A. Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, v. 17, n. 2, p. 91–108, 1995.

REYNOLDS, D. A.; TORRES-CARRASQUILLO, P. *The MIT Lincoln Laboratory RT-04F diarization systems: Applications to broadcast audio and telephone conversations*. Palisades, 2004. 20–30 p.

ROUGUI, J.; RZIZA, M.; ABOUTAJDINE, D.; GELGON, M.; MARTINEZ, J. Fast Incremental Clustering of Gaussian Mixture Speaker Models for Scaling up Retrieval In On-Line Broadcast. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Toulouse: [s.n.], 2006. v. 5, p. 5–10.

ROUVIER, M.; BOUSQUET, P.-M.; FAVRE, B. Speaker diarization through speaker embeddings. In: *European Signal Processing Conference*. Nice, França: [s.n.], 2015. p. 2082–2086.

SANKAR, A.; WENG, F.; RIVLIN, Z.; STOLCKE, A.; GADDE, R. R. The development of SRI's 1997 Broadcast News transcription system. In: *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. Landsdowne: [s.n.], 1998. p. 91–96.

SCHWARZ, G. A sequential Student test. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 42, n. 3, p. 855–1156, 1971.

SCHWARZ, G. Estimating the dimension of a model. *The annals of statistics*, v. 6, n. 2, p. 239–472, 1978.

SENOUSSAOUI, M.; KENNY, P.; STAFYLAKIS, T.; DUMOUCHEL, P. A study of the cosine distance-based mean shift for telephone speech diarization. *IEEE Transactions on Audio, Speech and Language Processing*, v. 22, n. 1, p. 217–227, 2014.

SHUM, S.; REYNOLDS, D. A.; GLASS, J. R.; DEHAK, N.; CHUANGSUWANICH, E.; GLASS, J. Exploiting intra-conversation variability for speaker diarization. In: *IEEE International Conference on Speech and Language Processing*. [S.l.: s.n.], 2011. p. 945–948.

SHUM, S. H.; DEHAK, N.; DEHAK, R.; GLASS, J. R. Unsupervised methods for speaker diarization: An integrated and iterative approach. *IEEE Transactions on Audio, Speech and Language Processing*, v. 21, n. 10, p. 2015–2028, 2013.

SIEGLER, M. A.; JAIN, U.; RAJ, B.; STERN, R. M. Automatic segmentation, classification and clustering of broadcast news audio. In: *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. Chantilly: [s.n.], 1997. p. 97–99.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2015.

SINHA, R.; TRANTER, S. E.; GALES, M. J. F.; WOODLAND, P. C. The Cambridge University March 2005 speaker diarisation system. In: *International Conference Spoken Language Processing*. Lisboa: [s.n.], 2005. p. 2437–2440.

SIVAKUMARAN, P.; FORTUNA, J.; ARIYAEEINIA, A. M. On the use of the Bayesian information criterion in multiple speaker detection. In: DALSGAARD, P.; LINDBERG, B.; BENNER, H.; TAN, Z.-H. (Ed.). *International Conference Spoken Language Processing*. Aalborg: ISCA, 2001. p. 795–798.

SOONG, F.; ROSENBERG, A. On the use of instantaneous and transitional spectral information in speaker recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 36, n. 6, p. 871–879, 1988.

SUN, H.; BIN, M.; KHINE, S. Z. K.; LI, H. Speaker diarization system for RT07 and RT09 meeting room audio. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Dallas, EUA: [s.n.], 2010. p. 4982–4985.

SUN, X. Pitch determination and voice quality analysis using Subharmonic-to-Harmonic Ratio. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Orlando: [s.n.], 2002. v. 1, p. 333–336.

TRANTER, S. Two-Way Cluster Voting to Improve Speaker Diarisation Performance. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Filadélfia: [s.n.], 2005. v. 1, p. 753–756.

TRANTER, S.; REYNOLDS, D. An Overview of Automatic Speaker Diarisation Systems. *IEEE Transactions On Audio, Speech, and Language Processing*, v. 14, n. 5, p. 1557–1565, 2006.

TRANTER, S. E.; REYNOLDS, D. A. Speaker diarisation for broadcast news. In: *ODYSSEY-2004 - The Speaker Recognition Workshop*. Toledo: [s.n.], 2004. v. 14, p. 337–344.

TRITSCHLER, A.; GOPINATH, R. A. Improved speaker segmentation and segments clustering using the bayesian information criterion. In: *European Conference Spoken Language Processing*. Budapeste: ISCA, 1999. p. 679–682.

TSAI, W.-H.; WANG, H.-M. On maximizing the within-cluster homogeneity of speaker voice characteristics for speech utterance clustering. In: IEEE. *IEEE International Conference on Acoustics, Speech and Signal Processing*. Toulouse, 2006. v. 1, p. 10–14.

VESCOVI, M.; CETTOLO, M.; RIZZI, R. A DP algorithm for speaker change detection. In: *International Conference Spoken Language Processing*. Geneva: ISCA, 2003. p. 2997–3000.

WILLSKY, A. S.; JONES, H. L. A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems. *IEEE Transactions on Automatic Control*, v. 21, n. 1, p. 108–112, 1976.

WOODLAND, S. E. J. P. C. Speaker clustering using direct maximisation of the MLLR-adapted likelihood. In: *International Conference Spoken Language Processing*. Sydney: [s.n.], 1998. p. 1775–1779.

WOOTERS, C.; FUNG, J.; PESKIN, B.; ANGUERA, X. Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system. In: *Fall 2004 Rich Transcription Workshop (RT-04)*. Palisades, NY: [s.n.], 2004. p. 23.

WU, C. H.; HSIEH, C. H. Multiple change-point audio segmentation and classification using an MDL-based Gaussian model. *IEEE Transactions on Audio, Speech and Language Processing*, v. 14, n. 2, p. 647–657, 2006.

YAMAGUCHI, M.; YAMASHITA, M.; MATSUNAGA, S. Spectral cross-correlation features for audio indexing of broadcast news and meetings. In: *International Conference Spoken Language Processing*. Lisboa: ISCA, 2005. p. 613–616.

YELLA, S. H.; STOLCKE, A. A comparison of neural network feature transforms for speaker diarization. In: *IEEE International Conference on Speech and Language Processing*. Dresden, Alemanha: [s.n.], 2015. p. 3026–3030.

ZAMALLOA, M.; RODRÍGUEZ-FUENTES, L. J.; BORDEL, G.; PENAGARIKANO, M.; URIBE, J. P. Low-latency online speaker tracking on the AMI corpus of meeting conversations. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. Dallas, EUA: [s.n.], 2010. p. 4962–4965.

ZHOU, B.; HANSEN, J. H. L. Unsupervised audio stream segmentation and clustering via the Bayesian information criterion. In: *International Conference Spoken Language Processing*. Pequim: ISCA, 2000. p. 714–717.

ZHOU, B.; HANSEN, J. H. L. Efficient audio stream segmentation via the combined T2 statistic and bayesian information criterion. *IEEE Transactions on Speech and Audio Processing*, v. 13, n. 4, p. 467–474, 2005.

ZHU, X.; BARRAS, C.; LAMEL, L.; GAUVAIN, J.-L. Speaker diarization: From broadcast news to lectures. In: *Machine Learning for Multimodal Interaction*. Xii. [S.l.]: Springer, 2006. p. 396–406.

ZHU, X.; BARRAS, C.; MEIGNIER, S.; GAUVAIN, J. Combining speaker identification and BIC for speaker diarization. In: *International Conference Spoken Language Processing*. Lisboa: [s.n.], 2005. p. 2441–2444.

ZOCHOVA, P.; RADOVA, V. Modified DISTBIC algorithm for speaker change detection. In: *International Conference Spoken Language Processing*. Lisboa: ISCA, 2005. p. 3073–3076.

APÊNDICE A – ARTIGOS DE CONFERÊNCIAS

A.1 PUBLICAÇÕES

Submissão em Agosto/2016 para o ICASSP - International Conference on Audio, Speech, and Signal Processing de 2017. Artigo anexado.

SPEAKER SEGMENTATION USING I-VECTOR IN MEETINGS DOMAIN

Leonardo V. Neri[1] *Hector N. B. Pinheiro*[1]
Tsang Ing Ren[1] *George D. da C. Cavalcanti*[1] *André G. Adami*[2]

[1]Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
Recife, Brazil

[2] Centro de Ciencias Exatas e da Tecnologia (CCET)
Universidade de Caxias do Sul (UCS)
Caxias do Sul, Brazil

ABSTRACT

In this paper, we propose a speaker segmentation method for meeting audio based on i-vector. The motivation is to utilize the Total Variability (TV) framework as a feature extractor and to exploit the potential of modeling the speaker and channel variabilities for speaker segmentation in meetings. A distance-based segmentation method is designed with the cosine distance. A sliding window with variable length searches for speaker turns, through the distance between the i-vectors extracted from two segments with the same size. The experiments are conducted on the AMI Meeting Corpus, covering several conversation scenarios. For the training data of the UBM and TV matrix, 5 conversations from AMI Meeting Corpus are sampled. Other 10 conversations from AMI Meeting Corpus to compose the test data. The experiments show an improvement in the MDR and FAR curves compared with the FixSlid approach with different distance metrics, and for most of the operating points when compared with the classical BIC based WinGrow. The proposed method has on average a better computational performance, improving in 61.5% compared with the XBIC based FixSlid, and improving in 86.7% compared with the BIC based WinGrow.

Index Terms— speaker diarization, speaker segmentation, i-vector, total variability, meeting conversations

1. INTRODUCTION

Speaker Diarization is the process of splitting an audio stream into homogeneous segments that belongs to each participating speaker. Speaker segmentation is the task of determining the time instants when a transition from one speaker to another occurs, called speaker turns or speaker change points. The main challenge of this task is to form homogeneous speaker segments (only one speaker per segment) [1, 2].

Most of speaker segmentation algorithms relies on a sliding window and a distance metric to evaluate whether two segments belongs to the same speaker. This type of algorithms are categorized as distance-based segmentation [3]. This category assumes no *a priori* information about the speakers in the audio stream. The change points are localized by a sliding window, that can have a fix length (FixSlid approach [4,5]) or a variable length (WinGrow approach [6–8]). Commonly, MFCCs are extracted from the window [2, 9], and a distance between two segments of the features vectors is calculated. A speaker change point is detected when a distance is above a threshold. The Bayesian Information Criterion (BIC) is the most popular distance metric used for this task [2]. Other examples of popular distance metrics are the symmetric Kullback-Leibler (KL2) [4] and the Generalized Likelihood Ratio (GLR) [10].

In speaker recognition, factor analysis methods have proved to be very effective, particularly to telephone speech [11]. These methods are applied to speaker diarization, for telephone conversations [12–15] and broadcast news scenarios [16, 17]. Castaldo et al. [12] introduce the eigenvoices as speaker factor to pre-segment the speech, in a stream segmentation system for multi-speaker telephone conversations. In Desplanques et. al [16], eigenvoices are also proposed for speaker segmentation, in the context of broadcast news, utilizing single-pass and two-pass algorithms. In Shum et. al [14] and Senoussaoui et. al [15], the i-vector as speaker factors is proposed for speaker clustering in telephone conversations.

This work introduces the i-vector for the WinGrow segmentation approach. The i-vectors are extracted through a UBM and a Total Variability (TV) matrix trained for meetings domain. The segmentation relies on detecting change points with a sliding window with variable length. The i-vectors are extracted from two segments with the same length and the cosine distance is calculated between i-vectors. A threshold value is set to decide if there is a speaker change point between the segments boundaries. The UBM and the TV matrix are modeled using the AMI Meeting Corpus [18], sampling meeting audio streams. The experiments are conducted on the AMI Meeting Corpus, with different audio streams than the streams utilized for training of the UBM and TV matrix.

The remainder of this paper is organized as follows. Section 2 presents a brief introduction to the total variability modeling. Section 3 describes the baseline segmentation methods

Thanks to CNPq (446831/2014-0) and Facepe (APQ-0192-1.03/14) agencies for funding.

evaluated in this work. In Section 4, we detail the proposed speaker segmentation method utilizing i-vector. The experimental results are shown in Section 5. Finally, conclusions and discussions in relation to prior work are given in Section 6.

2. TOTAL VARIABILITY MODELING

In Dehak et al. [11], the total variability modeling is introduced, defining a single space containing both speaker and channel variabilities simultaneously. The total variability matrix defines this single space, containing the eigenvectors with the largest eigenvalues of the total variability covariance matrix. It ignores the distinction between the effects of speaker and channel in the Gaussian Mixtures Model (GMM) supervector space. A GMM supervector M is formed by concatenating the means of each mixture component. Given a speaker utterance, the speaker- and channel-dependent supervector M is written as,

$$M = m + Tw \tag{1}$$

where m is the speaker- and channel-independent supervector, taken from the Universal Background Model (UBM) supervector, T is a rectangular matrix of low rank, representing the total variability space, and w is a random vector with standard normal distribution. The vector w is called i-vector and its components are the total factors.

3. BASELINE SEGMENTATION METHODS

The proposed method is compared with the classical WinGrow approach [7], utilizing the BIC distance, and the FixSlid approaches, utilizing the distances: GLR [19], KL2 [4] and XBIC [8].

3.1. Window Growing Segmentation Approach

In the Window Growing approach (WinGrow) [7], the multiple change points detection in an audio stream is done with a defined distance metric, a criterion λ, or a threshold value and a sliding window with variable length, to sequentially detect one possible change point inside the window. The length of the window grows N_g vectors if no change is present. To detected one change point, a distance curve is calculated iteratively, splitting the window into two partitions. The length of these partitions is changed for each iteration, the length of the left partition starts with N_{min} vectors and increases over the right partition until the right partition length becomes N_{min}. The distance between partitions is calculated for each iteration and the defined criterion is applied to the curve, to detect the location of a possible speaker change. This approach has a high computation cost if the audio has long homogeneous speech segments and no upper limit is set to the window length. In [7], a upper limit length N_{max} together a slid length N_S parameter shows an efficiently computation performance and no deterioration in the speaker segmentation results. The algorithm presented in [7] is adopted in our method.

3.2. Fixed and Sliding Window Approach

The FixSlid approach detects multiple speaker change points with a defined distance metric, a criterion, or a threshold value and a sliding window with fixed length. The window length N_D is divided in two adjacent segments to calculate a distance value between them. The slide length N_{Slid} is defined to overlap the window and to calculate several distances across the audio stream. An analysis window is defined to find the peaks in the curve that are above the threshold, or are selected as candidates by a defined criterion. The length of the analysis window N_A is defined to evaluate a local set of points under the criterion or threshold and to find a speaker change point.

The criterion proposed in Delacourt et. al [19] is utilized in the baseline method. The absolute differences between a local peak value and the local lowest values on the left and right positions are computed. These differences are compared with the standard deviation of the curve multiplied by an adjustable parameter α. A relevant peak is detected if its value meets this criterion.

4. I-VECTOR BASED SPEAKER SEGMENTATION

In the distance curve computation, the low-level features vectors from the two partitions are modeled by using a single Gaussian density. As the partition length decreases, the voice patterns can become diverse due to the differences in their acoustic contents. The modeling inherently rely on averaging out the phonetic differences between partitions. Therefore, when the spoken material is from the same speaker, the partitions being compared can appear dissimilar, which can lead to an increased number of false alarms.

The i-vector as speaker factors relies on modeling the specific speaker features, considering the variability from its voice, and the channel effect present in the speech acquisition. Given an window analysis, the i-vectors are extracted from two partitions with same size. Then, a distance value is calculated between i-vectors. It is assumed that each partition has sufficient information in the i-vector representation. If the analysed spoken content comes from the same speaker, the extracted i-vectors tends to be similar, even with the variation of the phonetic acoustic information.

The cosine distance is utilized in the segmentation task, to measure the dissimilarity between the i-vectors:

$$cos\_dist(w_1, w_2) = 1 - \frac{\langle w_1, w_2 \rangle}{\|w_1\| \cdot \|w_2\|} \tag{2}$$

this distance has a normalized scale in the range 0 to 2, being easier to set a threshold θ. With no amplitude information,

this range appears independent from the audio application or audio data set.

This approach allows the detection of multiple speaker change points in a continuous speech segment. The analysis window has an initial length of N_{ini} low-level feature vectors. The speaker change candidate is in the middle of the window. The cosine distance is calculated between adjacent segments, corresponding to the left and right middle of the window. If the distance is above the threshold θ, the speaker change candidate is a change point. Then, the window is repositioned to this point and its length is reset to N_{ini}. If no change point is detected, the window length grows N_g features vector and the detection process starts again. If the length reaches the upper limit N_{max}, the window slides N_S features vector until a change point is detected. The algorithm stops when the window reaches the end of the signal. Fig. 1 shows how the proposed i-vector based WinGrow detects multiple speaker change points.

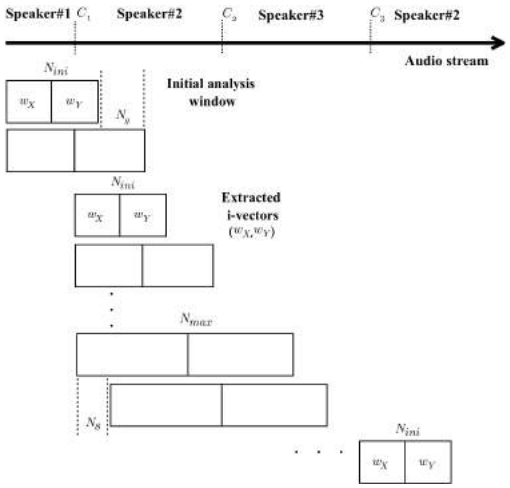

Fig. 1. Multiple speaker change points detection utilizing an i-vector based WinGrow approach. The cosine distance is calculated between i-vectors extracted from two window partitions with same length.

5. EXPERIMENTS

5.1. Data

The AMI Meeting Corpus [18] is an open access corpora with 100 hours of meeting recordings. For the experiments, we obtained 15 conversation having different speaker for all the samples. We divide the conversations into training and test data sets. In the training data, 5 conversations are sampled for both UBM and TV matrix modeling. The total of speakers for all the conversations is 10, and total duration for all conversations is approximately 4 hours. The evaluation data set is composed by 10 conversations. The conversations have between 4 and 5 speakers. The total duration of all conversations is approximately 6 hours and 24 minutes. There is a total of 6916 speaker change points in these conversations.

5.2. Speech Activity Detection

Before parametrization, a Speech Activity Detection (SAD) method based on energy is applied to extract all non speech segments longer than 0.25 second. In addition, speech segments shorter than 0.5 second are also discarded.

5.3. Evaluation measures

Given the inconsistencies in the labelling process and the uncertainty of when a speaker change precisely occur, a non-scoring collar around every reference speaker change is defined in evaluating the performance of the systems. In this work, a 500 ms collar is used to account for such issues.

Two types of errors are computed: False Alarm Rate (FAR) and Miss Detection Rate (MDR) [2, 9]. The MDR refers to the percentage of reference speaker changes that were not detected by the system. The FAR refers to the percentage of detected speaker changes that are not present in the audio.

The MDR and FAR from different operating points are used to compute the Receive Operation Characteristics (ROC) curve. The Area Under the Curve (AUC) is also used as a evaluation measure. In our experiments, we choose two error rates to compute the ROC curve, therefore, the best AUC is the most closely to 0.

5.4. Baseline Methods Configuration

For all baseline methods we extract 12 MFCCs from each cluster, with 32 ms frame length for every 10 ms.

For the BIC based WinGrow the following configuration is set: N_{ini}, N_g, N_{max} and N_S are set to 2, 1, 10 and 2 seconds, respectively. N_{min} is set to 20 frames of MFCC. We evaluate 10 values of λ, in a range from 0.5 to 5.5 in intervals of 0.5.

For the FixSlid approaches, the following configuration is set: N_D, N_{Slid} and N_A are set to 2, 0.2 and 2 seconds, respectively. We evaluate 10 values of α, in a range from 0.1 to 1.9 in intervals of 0.1.

5.5. Proposed Method Configuration

From each speaker cluster in the training data we extract 12 MFCCs, with a frame length of 32 ms for every 10 ms. The UBM is trained with all features vectors. The number of Gaussian components is set to 8. The matrix T is trained considering the variability intra-cluster. The rank of both T and the i-vector is set to 100.

Table 1. The Area Under the Curve (AUC) for the ROC curves of each segmentation method in the test data set.

Methods	AUC
WinGrow i-vector (COSINE)	**46.57**
WinGrow MFCC (BIC)	47.41
FixSlid MFCC (KL2)	52.07
FixSlid MFCC (GLR)	52.01
FixSlid MFCC (XBIC)	52.29

Table 2. Computational performances of the evaluated methods, measured in seconds. The mean and standard deviation (dev) are computed for the processing time for 10 runs of experiments.

Methods	mean (secs)	dev
WinGrow i-vector (COSINE)	**2.64**	**2.44**
WinGrow MFCC (BIC)	18.46	16.31
FixSlid MFCC (KL2)	16.18	20.29
FixSlid MFCC (GLR)	10.21	12.91
FixSlid MFCC (XBIC)	6.85	8.85

For the proposed approach, the WinGrow configuration follows the baseline using the BIC based WinGrow. We evaluate 10 values of θ, in a range from 0.1 to 1.9 in intervals of 0.1.

5.6. Speaker Segmentation Results

Fig. 2 shows the ROC curves for all evaluated methods. The points of the curve are obtained varying the adjustable parameters or threshold value. Table 1 shows the AUC for the ROC curves presented in Fig. 2. The ROC curves show that the proposed method outperforms the FixSlid approaches evaluated, in both MDR and FAR. The proposed method outperforms the BIC based WinGrow in some operating points. Table 1, the proposed method yields the lowest AUC.

Table 2 presents the average computational performance for 10 runs of experiments in the test data set for each evaluated method. The proposed method have the best average computation performance among the baseline methods, with an improvement of 61.5% compared to the FixSlid approach with the XBIC distance, and in 86.7% compared to the BIC based WinGrow.

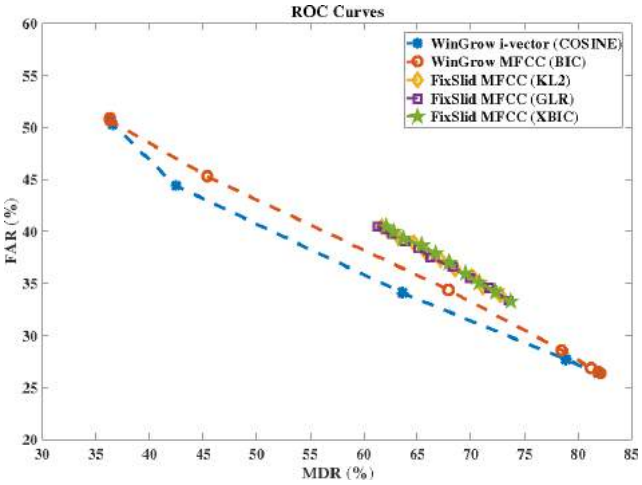

Fig. 2. The MDR x FAR curves of the evaluated methods. Each point is obtained varying the threshold value or the adjustable parameters.

6. CONCLUSION

This work presented an i-vector based WinGrow approach for speaker segmentation in meeting audio, using a cosine distance. In the experiments using the AMI Meeting Corpus, the i-vector and the cosine distance demonstrate a higher discrimination capacity among different speakers compared with the Gaussian distributions modeled from MFCC using the distances BIC, GLR, KL2 and XBIC. The results showed that the proposed method obtains better MDR and FAR for most of operating points compared with the classical BIC based WinGrow, and outperforms the classical FixSlid approaches with the distances GLR, KL2 and XBIC. The AUC of the proposed segmentation method is less than all baseline methods. The proposed method has a superior average computational performance.

As future work, we will investigate the results with more data for the UBM and T matrix modeling, and varying the rank of both T and i-vector. More experiments are needed in different application domains, such as telephone conversations and broadcast news.

7. ACKNOWLEDGEMENTS

This research are sponsored by the Conselho Nacional de Pesquisa (CNPq).

8. REFERENCES

[1] N Evans, S Bozonnet, Dong Wang, C Fredouille, and R Troncy, "A Comparative Study of Bottom-Up and Top-Down Approaches to Speaker Diarization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 20, no. 2, pp. 382–392, 2012.

[2] M. H. Moattar and M. M. Homayounpour, "A review on speaker diarization systems and approaches," *Speech Communication*, vol. 15, 2012.

[3] Shih Sian Cheng, Hsin Min Wang, and Hsin Chia Fu, "BIC-Based Speaker Segmentation Using Divide-and-Conquer Strategies With Application to Speaker Diarization," *IEEE Transactions On Audio, Speech, and Language Processing*, vol. 18, no. 1, pp. 141–157, 2010.

[4] Matthew A Siegler, Uday Jain, Bhiksha Raj, and Richard M Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Chantilly, 1997, pp. 97–99.

[5] Lie Lu and HongJiang Zhang, "Real-Time Unsupervised Speaker Change Detection," in *International Conference on Pattern Recognition*, Quebec, 2002, pp. 358–361.

[6] S Chen and P Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion," in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, 1998, number 6, pp. 67–72.

[7] M Cettolo and M Vescovi, "Efficient audio segmentation algorithms based on the BIC," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Hong-Kong, 2003, vol. 6, pp. 537–540.

[8] X Anguera and J Hernando, "Xbic: Real-time cross probabilities measure for speaker segmentation," *ICSI - Berkeley Technical Report*, vol. 1, no. 1, pp. 05–08, 2005.

[9] Margarita Kotti, Vassiliki Moschou, and Constantine Kotropoulos, "Speaker segmentation and clustering," *Signal Processing*, vol. 88, no. 5, pp. 1091–1124, 2008.

[10] Perrine Delacourt and Christian Wellekens, "DISTBIC: A speaker-based segmentation for audio data indexing.," *Speech Communication*, vol. 32, no. 1-2, pp. 111–126, 2000.

[11] Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[12] F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, and C. Vair, "Stream-based speaker segmentation using speaker factors and eigenvoices," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 4133–4136.

[13] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1059–1070, 2010.

[14] Stephen Shum, Najim Dehak, Ekapol Chuangsuwanich, Douglas A. Reynolds, and James R. Glass, "Exploiting intra-conversation variability for speaker diarization," in *INTERSPEECH*, 2011.

[15] Mohammed Senoussaoui, Patrick Kenny, Themos Stafylakis, and Pierre Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 1, pp. 217–227, 2014.

[16] Brecht Desplanques, Kris Demuynck, and Jean-Pierre Martens, "Factor analysis for speaker segmentation and improved speaker diarization," in *INTERSPEECH*, 2015, pp. 3081–3085.

[17] Jan Silovsky and Jan Prazak, "Speaker diarization of broadcast streams using two-stage clustering based on i-vectors and cosine distance scoring," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4193–4196.

[18] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner, "The ami meeting corpus: A pre-announcement," in *Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction*, Berlin, Heidelberg, 2006, pp. 28–39, Springer-Verlag.

[19] Perrine Delacourt, David Kryze, and Christian Wellekens, "Detection of speaker changes in an audio document," in *European Conference Spoken Language Processing*, Budapeste, 1999, pp. 1195–1198, ISCA.

APÊNDICE B – ARTIGOS DE PERIÓDICOS

B.1 REJEIÇÕES

Submissão para a revista IET Eletronic Letters em Agosto/2017. Artigo em anexo.

Singular Features for Speaker Segmentation

Leonardo V. Neri, Hector N.B. Pinheiro, Tsang Ing Ren and George D. C. Cavalcanti

This work proposes a speaker segmentation method using Singular Value Decomposition (SVD) applied to the scaled Mel-Frequency Cepstral Coefficients (MFCCs) as feature extractor. The SVD obtains a speaker representation given a short speaker utterance which preserves the discriminant information. We define a feature vector for that representation. A sequential speaker segmentation method uses this feature to detect multiple speaker turns. Experiments using the AMI dataset shows an improvement on the precision rate and F_1 measure compared to approaches that use Gaussian Mixture Model (GMM) and i-vectors.

Introduction: Speaker segmentation is the process for detecting the separation between two distinct speakers over the audio stream [1, 2]. It also includes the identification of the boundaries of the overlap speech segments, when two or more speakers are talking at same time. The detection of these points is also known as speaker turns detection. The main challenge is to form homogeneous speaker segments (only one speaker per segment), serving as a previous step for speaker tracking and diarization [1, 2]. Most of the speaker segmentation algorithms rely on sliding windows and a distance metric to evaluate if two segments belong to the same speaker.

Commonly, the segmentation approaches use low-level features like Mel-Frequency Cepstral Coefficients (MFCCs) to estimate Gaussian Mixture Models (GMMs) or adapted GMMs from Universal Background Model (UBM) to represent the speakers [1, 2]. High-level representations like identity vectors (i-vectors) use the MFCCs and UBM to compute the Baum-Welch statistics [3]. The i-vector is a speaker representation given an utterance from a single speaker and recent works utilize the i-vectors for speaker segmentation task [4, 5].

An intrinsic problem of these probabilistic representations is to derive a good model from short utterances (between 0.5 and 1.0 seconds) using high dimensional features. This is a common situation for fast speaker turns detection, when there are few data for each speaker. To skip this situation, features with lower dimension are preferred [1, 2], discarding information of energy or delta coefficients, or using a lower number of coefficients than the well established speaker verification systems [3]. Also, the number of Gaussian components and the dimension of i-vectors are smaller than state of art speaker verification systems, because the insufficient amount of data to adapt a large number of GMM components and for fast computation of the audio stream [1, 2, 4, 5].

This work proposes a speaker representation for short utterances derived directly from the high dimensional MFCCs space. The representation is a vector definition of the left singular vectors given a MFCCs sample. We use the Singular Value Decomposition (SVD) to obtains a basis for the MFCCs similarity space (left singular vectors) associated the most relevant correlation components (right singular vectors). The MFCCs similarity space is the inner product map of all possible pairs of MFCCs vectors. We aim to represent the speaker information through the similarity space instead of probabilistic parameters estimation with insufficient data. It is expected neighbors vectors be similar while they represent the same phoneme, or occur in different time instants with same phonemes (or phonemes with similar pronunciation). The SVD defines the left singular vectors to explain the MFCC space. The concatenation of the left singular vectors forms a single feature vector, named as singular feature, to represent the speaker. Then, we propose a speaker segmentation method with this singular feature, using the cosine distance to compare two adjacent speaker samples.

Background: Let X be a d-dimension MFCCs space after mean and variance normalization, represented as a $N \times d$ matrix. The k-components singular value decomposition is an approximation $\tilde{X}$ of X, defined as the truncated SVD:

$$\tilde{X} = U_k S_k V_k^\top. \tag{1}$$

The columns of U_k are a set of k eigenvectors of $XX^\top$ matrix. They are also called the left singular vectors of X, and form an orthonormal basis for $XX^\top$. The columns of V are the eigenvectors of $X^\top X$, meaning the correlation matrix (remembering that X is the result of mean and variance normalization). They are also called the right singular vectors of X, and form an orthonormal basis for the correlation matrix. Both sets of eigenvectors are associated with the same largest singular values of X, which are the squared roots of non-negative eigenvalues of both $XX^\top$ and $X^\top X$. The $XX^\top$ matrix is the weighted cosine similarity map of X. Its elements are the inner products of all vectors in X:

$$< \mathbf{x}_i, \mathbf{x}_j >= ||\mathbf{x}_i|| \cdot ||\mathbf{x}_j|| \cos \alpha_{ij}, \qquad i, j = 1, 2, \cdots, N \tag{2}$$

the cosine similarity $\cos \alpha_{ij}$ weighted by the norms of $\mathbf{x}_i$ and $\mathbf{x}_j$. The diagonal of $XX^\top$ is the squared norm of $\mathbf{x}_i$. That map contains the speaker information in the utterance and suppress the spoken content, because some map regions with close levels of similarity can correspond to distinct phonemes. On the other hand, map regions of distinct speakers for the same phonemes can have very different levels of similarity, given the differences of vocal tract and speaker style. The left singular vectors of X are a orthonormal basis for that map. This basis describes a speaker representation given a utterance. We form a feature vector concatenating that basis

$$\mathbf{s} = [\mathbf{u}_1^\top \mathbf{u}_2^\top ... \mathbf{u}_k^\top] \tag{3}$$

where $\mathbf{u}_k$ is the kth $\mathbf{U_k}$ column.

Sahidullah and Kinnunen [6] also decompose the space of MFCCs with SVD using a short window. They propose a feature to represent the local spectral variability of a speaker in a short time interval. They use a window length at most 130 ms to norm $X^\top$ by the mean and to calculate the covariance matrix as $XX^\top$. In their work, the feature vector $\mathbf{f}$ is also the concatenation the eigenvectors in U_k, but they are a decomposition of the covariance matrix, a different meaning compared to our proposal.

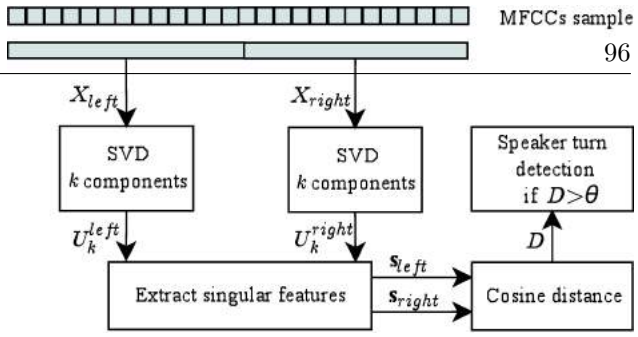

Fig. 1 *Speaker turn detection algorithm using the proposed singular features. Two adjacent windows split the MFCCs sample into X_{left} and X_{right}. The truncated SVD with k components of both X_{left} and X_{right} give the left singular vectors in U_k^{left} and U_k^{right}, respectively. The proposed features $\mathbf{s}_{left}$ and $\mathbf{s}_{right}$ are the concatenation of the transposed columns of U_k^{left} and U_k^{right}, respectively. The cosine distance between $\mathbf{s}_{left}$ and $\mathbf{s}_{right}$ gives a value D, which is tested against a threshold θ. The existence of a speaker turn between the windows is confirmed if D is above θ.*

Proposed Technique: Given the speech segments from an audio stream and the corresponding d-dimensional MFCCs feature vectors of each segment. We apply the Short Time Mean and Variance Normalization (STMVN) on each segment of vectors to reduce the linear channel effects [7]. A sequential method similar to [5] detects multiple speaker turns to generate the speaker segments. The sequential method uses a speaker turn detection algorithm shown in Fig. 1 to search the speaker turns on each segment. From Fig. 1, two adjacent windows samples feature vectors from a segment to form the **MFCCs sample**. Let t be the number of vectors sampled by each window, and $\mathbf{X_{left}}$ and $\mathbf{X_{right}}$ be the sampled vectors by the left and right windows, respectively. The proposed speaker representation are the left singular vectors $\mathbf{U_k^{left}}$ and $\mathbf{U_k^{rigth}}$ extracted for each $\mathbf{X_{left}}$ and $\mathbf{X_{right}}$, respectively. The extraction demands a definition of a number $\mathbf{k}$ of components for the **SVD**. We define the feature vectors $\mathbf{s_{left}}$ from $\mathbf{U_k^{left}}$ and $\mathbf{s_{right}}$ from $\mathbf{U_k^{right}}$ using the Equation 3. We compare both feature vectors with the **Cosine distance**

$$D = 1 - \frac{< \mathbf{s}_{left}, \mathbf{s}_{right} >}{||\mathbf{s}_{left}|| \cdot ||\mathbf{s}_{right}||}. \quad (4)$$

A speaker turn between the windows exists if D is above a threshold θ. If a speaker turn is detected, both windows slid t vectors. If not, they grow g vectors until find a speaker turn, or reach a maximum length of t_{max}. If they reach the maximum length and do not detect a speaker turn, they slid t_{slid} vectors until find a speaker turn or reach the end of segment.

Experiments: The AMI Meeting Corpus is an open access corpora consisting of 100 hours of meeting recordings. We use the full corpus configuration already divided into training, development, and evaluation sets, using the seen data for training and development, and the unseen data for evaluation. The training set contains 42 meeting recordings with total duration of 75 hours and 166 distinct speakers. We use this set to train the UBM with 128 components and the TV matrix with 100 factors. The development set contains 11 meeting recordings with total duration between 12 and 13 hours and 44 distinct speakers. We utilize the development set to choose the best window length and threshold/penalty for each segmentation method shown in Table 1. We choose carefully these parameters to make a fair comparison of the performance on the evaluation set. The evaluation set contains 6 meeting recordings with total duration close to 13 hours and 24 distinct speakers. We report the segmentation performance using the Precision Rate (PRC), Recall Rate (RCL), and the F_1 measure [1, 2]. We use a collar of 0.5 second to count the true detected speaker turns.

Table 1: The window length and threshold/penalty per method that achieve the best F_1 measure in the AMI development set. The λ parameter is the penalty factor of BIC distance [1, 2]. The α parameter is a factor of standard deviation of distance curve to detect the relevant peaks [1, 2]. The θ is a threshold to the cosine distance [5].

Methods	Window (secs)	Threshold
WinGrow	1.50	$\lambda = 0.45$
FixSlid	0.60	$\alpha = 0.05$
i-vector FixSlid	0.75	$\alpha = 0.35$
i-vector WinGrow	1.0	$\theta = 0.5$
Proposed Method	0.5	$\theta = 0.05$

We include the overlap speech segments (two or more speakers talking at same time) in our evaluation to increase the number of fast speaker turns. In this scenario, the development set has approximately 15k of speaker turns and the speaker duration is up to 1 second in 49% of speaker segments. About 70% of the speaker segments has a duration up to 2 seconds, indicating a fast conversation style in the development set.

In this work, we utilize the ground-truth labels of speech/non-speech segments to evaluate the speaker segmentation performance on ideal conditions. We extract 19 MFCCs plus the Δ, and Δ^2 coefficients, with 20 ms frame length for every 10 ms, resulting on 57 features dimension. We normalize each vector applying the STMVN method with a sliding window of 1 second length.

The proposed singular features computes the SVD with 8 components. The other evaluated methods are: the window growing BIC method (WinGrow) [1, 2, 8], the two fixed sliding windows method (FixSlid) [1, 2, 8], the FixSlid method using i-vectors with cosine distance (i-vector FixSlid), proposed as first step segmentation in [4], and the WinGrow method using i-vectors with the cosine distance (i-vector WinGrow), proposed in [5]. For the WinGrow methods, the growing step g, maximum length t_{max} and slid step t_{slid} (see [5, 8] for more details) are set to 0.1, 3, and 0.75 times the window length parameter.

Table 2 presents the performance results in the evaluation set. The higher PRC rate indicates the proposed speaker representation suppress the spoken content in favor of speaker information. The i-vector speaker representation with WinGrow achieves a PRC rate close (< 1%) to our proposal, but with the expenses of i-vector extraction and the necessity of a UBM and a Total Variability space modeling using a large amount of data. The RCL rate is bellow of the traditional WinGrow in 1.6% but it is still a high rate (> 95%), reasonable for speaker diarization systems. The F_1 measure of our proposal shows the better trade-off considering both PRC and RCL.

Table 2: The PRC, RCL, and F_1 measure of segmentation methods in the AMI evaluation set. The bold points to the higher rates.

Methods	PRC (%)	RCL (%)	F_1 measure (%)
WinGrow [8]	32.24	**98.66**	48.60
FixSlid [8]	28.77	94.57	44.11
i-vector FixSlid [4]	32.19	89.49	47.35
i-vector WinGrow [5]	39.51	92.53	55.37
Proposed Method	**40.11**	96.06	**56.59**

Conclusion: This work proposes a feature vector from a short speech utterance to perform speaker segmentation. The feature models the speaker information using only the normalized MFCCs and their delta and double delta coefficients. The proposed feature does not require a GMM/UBM, or a Total Variability factor model to compare two speech utterances. The truncated SVD finds a speaker space basis with k vectors for a short time window (maximum length of 1.5 secs). The concatenation of these vectors forms a feature vector representing the speaker present on the window. The proposed speaker segmentation method uses two adjacent windows to extract the features, the cosine distance to compare them, and a threshold to decide if the distance corresponds to a speaker turn. Each window grows equally if the distance is under the threshold. When the window lengths reach a up limit, they slid until find a speaker turn. Experiments in fast speaker turns scenarios shows the proposed feature achieves a higher F_1 score compared with traditional approaches using single Gaussian, and using i-vectors.

Acknowledgment: This work was partially supported by CAPES, CNPq and FACEPE

Leonardo V. Neri, Hector N.B. Pinheiro, Tsang Ing Ren, George D. C. Cavalcanti *(Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil).*
E-mail: {lvn,hnbp,tir,gdcc}@cin.ufpe.br

References

1 X.A. Miro, S. Bozonnet, Speaker diarization: A review of recent research, IEEE Transactions on Audio, Speech, and Language Processing 20 (2) (2012) 356–370.
2 N. Evans, S. Bozonnet, D. Wang, C. Fredouille, R. Troncy, A Comparative Study of Bottom-Up and Top-Down Approaches to Speaker Diarization, IEEE Transactions on Acoustics, Speech, and Signal Processing 20 (2) (2012) 382–392.
3 N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification, IEEE Transactions on Audio, Speech, and Language Processing 19 (4) (2011) 788–798.
4 B. Desplanques, K. Demuynck, J.-P. Martens, Factor analysis for speaker segmentation and improved speaker diarization, in: INTERSPEECH, 2015, pp. 3081–3085.
5 L.V. Neri, H.N.B. Pinheiro, I.R. Tsang, G.D.C. Cavalcanti, A.G. Adami, Speaker segmentation using i-vectors in meetings domain, in: International Conference on Acoustics, Speech and, Signal Processing, 2017, pp. 5455–5459.
6 Md. Sahidullah, T. Kinnunen, Local spectral variability features for speaker verification, Digital Signal Processing 50 (2016) 1–11.
7 M.J. Alam, P. Ouellet, P. J. Kenny, D. O'Shaughnessy, Comparative Evaluation of Feature Normalization Techniques for Speaker Verification, in: International Conference on Nonlinear Speech Processing, 2011, pp. 246–253.
8 S.S. Cheng, H.M. Wang, H.C. Fu, BIC-Based Speaker Segmentation Using Divide-and-Conquer Strategies With Application to Speaker Diarization, IEEE Transactions on Audio, Speech, and Language Processing 20 (2) (2012) 356–370.

B.2 EM REVISÃO

Submissão para a revista EURASIP Journal on Audio, Speech, and Music Processing em Agosto/2018. Os revisores levantaram alguns pontos de correção/clarificação e pediram para resubmeter o artigo revisado até o dia 02/04/2019. Artigo em anexo.

Mel Cepstral Affinity Features for Speaker Segmentation

Leonardo V. Neri · Hector N. B. Pinheiro · Ing R. Tsang · George D. C. Cavalcanti · André G. Adami

Received: date / Accepted: date

Abstract A speaker diarization system aims to categorize the segments of speech signals according to the speakers' identity. An important step of this process is the segmentation of the speech content into homogeneous parts (one speaker per segment). The segmentation task is difficulty in audio conversations with highly frequent speaker transitions. In this scenario, a limited speech sample should be used to segment the speakers, aiming not to miss the transitions. We propose a feature designed for limited speech data to the speaker segmentation task. The proposed segmentation approach utilizes the discriminant information between homogeneous and heterogeneous speech (two or more speakers per segment) to detect the speaker transitions. We show that the Mel Cepstral Affinity Features (MCAF) contains discriminant information between homogeneous and heterogeneous speech. Experiments using the AMI corpus show that our proposal exhibits comparable performance to i-vector as the front-end feature, but with lower computational cost.

Keywords speaker diarization · speaker segmentation · heterogeneous/homogeneous speech · MFCC · affinity space · i-vector

1 Introduction

Audio diarization is the process of splitting an audio stream into segments and clustering them according to the different audio sources. More specifically, in speaker diarization, the segmentation and clustering are performed according to the act "Who spoke when" with no prior information about the number or the identities of the speakers.

L.V. Neri, H.N.B. Pinheiro, I.R. Tsang, G.D.C. Cavalcanti
Centro de Informática (CIn), Universidade Federal de Pernambuco
E-mail: {lvn,hnbp,tir,gdcc}@cin.ufpe.br

A.G. Adami
Área das Ciências Exatas e Engenharias, Universidade de Caxias do Sul
E-mail: agadami@ucs.br

There are two main approaches in the literature for speaker diarization systems [1–3]. The most popular is to segment the speech stream into homogeneous speaker segments (one speaker per segment), followed by a speaker clustering step to merge the segments from the same speaker. The other approach is to combine speaker segmentation and clustering in a single step process.

Speaker segmentation aims to detect the time instants when a transition from one speaker to another occurs, called speaker turns or speaker change points. The most common approaches are distance-based and model-based segmentation [1–3]. The distance-based segmentation relies on two sliding windows and a distance metric to evaluate whether two segments belong to the same speaker. In the model-based segmentation, prior knowledge of the audio sources is necessary to train a model to classify the audio segments [7–10].

Commonly, the Mel Frequency Cepstral Coefficients (MFCCs) are extracted from the speech segments to estimate Gaussian distributions or Gaussian Mixture Models (GMM) as the speaker representations to segment the audio [1–3]. The i-vector framework [11] is considered the state-of-the-art of speaker representation, obtained from factor analysis modeling. For distance-based segmentation approaches, the metrics often used for Gaussian models are the Bayesian Information Criterion (BIC) [4], the symmetric Kullback-Leibler (KL2) [5], and the Generalized Likelihood Ratio (GLR) [6]. The distance metrics used for i-vector are the Euclidean, cosine, and Mahalanobis [13,19,20].

Despite all the efforts, the task of verifying if two short segments belong to the same speaker remains a challenge [23,22]. This challenge appears in the speaker segmentation of conversations with highly frequent speaker transitions. The limitation of speech data demands a tuning phase on the number of Gaussian mixtures and the i-vector dimension for robustness, which has a high computational cost [16, 18–20, 23]. In this work, we propose an approach to address the problem of limited speech data. We modify the traditional algorithm of two sliding windows to detect speaker transitions in the presence of a heterogeneous speech segment. The Mel Cepstral Affinity Features (MCAF) are the core of our segmentation approach, representing a feature space containing discriminant information between heterogeneous and homogeneous speech. Modeling MCAF can improve the discrimination, so we choose the Probabilistic Linear Discriminant Analysis (PLDA) to model the types of speech and to determine whether two segments belong to the same distribution through the PLDA score.

The rest of the paper is organized as follows. Section 2 gives a brief overview of the i-vector and related work. Section 3 presents the proposed feature MCAF. Section 4 describes the segmentation method. We describe the experiments and results in Section 5. Then, we discuss the results in Section 6 and presents the conclusions in Section 7.

2 Background and Related Work

In this section, we present a brief overview of the i-vector and the modeling framework to represent the speaker. Related works to speaker segmentation using factor analysis are also described. We also discuss issues of the i-vector framework related to limited speech data.

2.1 The i-vector framework

Dehak *et al.* [11] introduced a factor analysis framework to model both speaker and channel variabilities. They defined the Total Variability (TV) space to explain both speaker and channel variabilities, simultaneously. The TV space is derived from the super-vector space of the Gaussian Mixtures Model (GMM), where a speaker super-vector is formed by concatenating the means of each adapted-GMM component. The TV space ignores the distinction between speaker and channel effects. A super-vector $\mathbf{m_s}$ of a speaker s can be expressed using factor analysis as

$$\mathbf{m}_s = \mathbf{m} + T\mathbf{w} \tag{1}$$

where $\mathbf{m}$ is the super-vector obtained from the Universal Background Model (UBM), T is a rectangular matrix of low rank representing the TV space, and $\mathbf{w}$ is a random vector with standard normal distribution. The vector $\mathbf{w}$ is called i-vector, and its components are the total factors. The Expectation and Maximization (EM) algorithm as described in [15] estimates the matrix T.

There is redundant information about the session and channel effects projected on the i-vector dimensions. Different compensation methods are applied to reduce this redundancy, as described in [11]. Among the techniques, the PLDA-based method is the most widely used [11,22]. In this paper, we use a variant of the PLDA method, known as Gaussian PLDA (GPLDA) [12] in which a full covariance matrix models the inter-speaker variability. The i-vector $\mathbf{w}$ extracted from a speech sample is expressed as

$$\mathbf{w} = \boldsymbol{\mu} + \Psi\mathbf{z} + \boldsymbol{\epsilon} \tag{2}$$

where, $\boldsymbol{\mu}$ is the mean of the i-vectors extracted from a background data set, the columns of Ψ are the eigenvoices (the basis for the speaker-specific subspace) [12], and the latent variables $\mathbf{z}$ are the projection of $\mathbf{w}$ onto the speaker-specific subspace. The $\boldsymbol{\epsilon}$ term represents the residual variability not captured by the latent variables. The log-likelihood ratio tests whether two speech samples belong to the same speaker or not. The GPLDA score is the log-likelihood ratio between the alternative hypothesis H_1 and null hypothesis H_0, defined as

$$\Phi_{\mathrm{GPLDA}}(\mathbf{w}_1, \mathbf{w}_2) = \log \frac{p(\mathbf{w}_1, \mathbf{w}_2|H_1)}{p(\mathbf{w}_1|H_0)p(\mathbf{w}_2|H_0)} \tag{3}$$

where the alternative hypothesis claims that the given two i-vectors belong to the same speaker, whereas the null hypothesis claims that they belong to different speakers. In speaker verification, we test the i-vector model of a speaker identity (average of the i-vectors extracted in the enrollment phase) against the i-vector extracted from the evaluation data. The i-vectors are length normalized and transformed by the PLDA model. Then, the PLDA score is calibrated for the verification system to accept/reject the null hypothesis.

2.2 Framework for Speaker Segmentation

The factor analysis and the i-vector framework establish the state-of-the-art for many areas other than speaker recognition, especially on the speaker diarization

problem [13,14,16–20]. UBM is a pre-requisite for most recent works on speaker diarization [13,14,16,18–20], as well as the TV matrix T [16,18–20] and the PLDA model [16,18,19].

Castaldo *et al.* [13] introduce the eigenvoices (subspace for the speaker-dependent information [12]) as speaker representation to pre-segment the speech of telephone conversations. They sample MFCC features from speech using short windows (around 1 second) to mitigate the presence of multiple speakers in a window. The proposed speaker representation is low dimensional, and the stream processing is designed for fast computation, using only 256 mixtures for UBM, and 20 speaker factors. They reported that the speaker factors are affected by the phonetic content variation because of the short window size. Their approach yielded a reduction in the segmentation error rates of 18% compared to previous works.

In Desplanques *et al.* [19], the speaker segmentation method utilizes the i-vector as speaker representation. They also sample MFCC features from speech using short windows (around 1 second) and use low dimension configurations for UBM and the matrix T. The TV space was defined considering heterogeneous speech, aiming to model the total factors variability to discriminate heterogeneous samples from speaker samples. They also comment that the phonetic content variation caused by the limited speech data affects the total factors, and propose the Mahalanobis distance to compensate for this variation. The method reduces speaker turns detection errors by 50% compared to a BIC-based segmentation baseline.

2.3 Data limitation issues

Recent works [23,22] show that i-vector based verification systems with reasonable accuracy require a sufficient amount of speech data in the enrollment and evaluation samples. In [23], experiments on GMM-UBM and i-vector-GPLDA systems show a significant degradation on the performance when reducing the speech duration on training or evaluation samples: 3.48% to 22.09% of the Equal Error Rate (EER) when the duration of the evaluation utterances was shortened from 2.5 min to 2 seconds. This degradation occurs because the total factors estimation becomes dependent on both low intra-speaker variability and high variability of lexical content [23].

The zeroth-order and first-order Baum-Welch statistics are the descriptions of super-vectors. The zeroth-order statistics are the probabilistic counts and defines the covariance of the posterior distribution given a speech sample. When the window sample is short, the covariance matrix becomes sparse, and as a consequence, the i-vector is poorly estimated [23,24]. On the other hand, when the window sample is long enough, the covariance becomes sharper, and the i-vector can be better estimated [24,23]. More details about the performance degradation for short speech duration are provided in [22–24].

3 Mel Cepstral Affinity Features

The problem to segment fast style conversations demands to work with limited speech data to avoid missing the speaker transitions. The difficulty to estimate

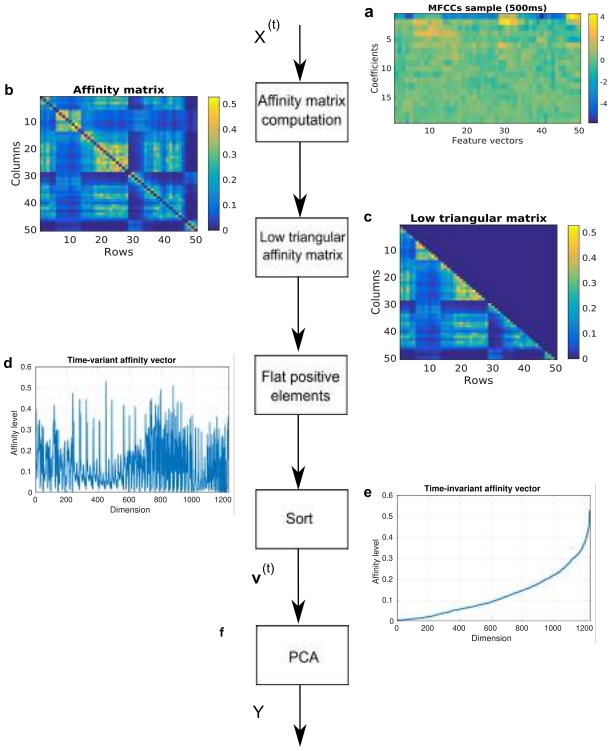

Fig. 1 Affinity features block diagram. Starts with the MFCCs sampling $X^{(t)}$ and follows with the affinity matrix computation, redundant information removal through low triangular matrix, time-invariant transformation through flatting and sorting the remain elements, and PCA transformation to estimate an uncorrelated low dimensional space.

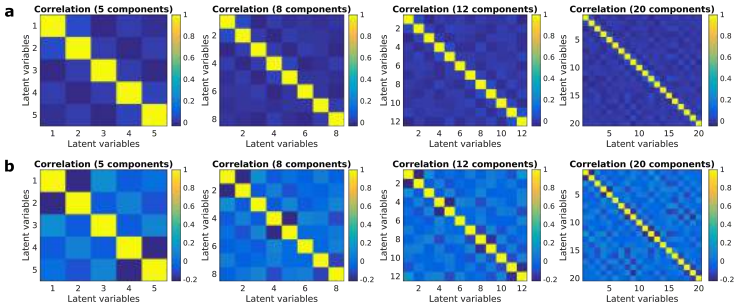

Fig. 2 MCAF correlation for different groups of speech. The correlation matrices in **a** (homogeneous speech) exhibit distinct behavior compared with **b** (heterogeneous speech).

good speaker models from limited speech data leads us to propose a novel approach and features for the problem. Instead of deriving discriminant speaker representations from Gaussian or factor analysis models using limited data, the proposed features are a discriminant representation between homogeneous and heterogeneous speech derived directly from the Mel cepstral domain. This infor-

mation can be useful for the segmentation task, because heterogeneous segments are expected during the process and their correct identification delimit intervals close to the real speaker transitions.

The proposed features are computed given the MFCCs extracted from a speech stream. The extraction of the Mel Cepstral Affinity Features (MCAF) has five steps (Figure 1): MFCCs sampling, affinity matrix computation, affinity vector computation, time-invariant transformation and latent features estimation. The MFCCs sampling utilizes a short sliding window to sample n MFCCs vectors $X^{(t)}$ on instant t of the speech stream (Figure 1a). The affinity matrix A (Figure 1b) is a square symmetric matrix ($n \times n$) defined as [25]

$$a_{ij} = \begin{cases} exp(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}), & \text{for } i \neq j, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where a_{ij} is an affinity element of A, $||.||$ is the Euclidean norm, $\mathbf{x}_i$ is the i-th vector of $X^{(t)}$, and σ is a scale parameter controlling how fast the affinity levels decay with the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. The third step extracts the affinity vector $\mathbf{v}$ from A through the following procedures: the redundant information is discarded using only the strictly lower triangular matrix of A (Figure 1c). Following by flatting (Figure 1d) and sorting the remaining elements (Figure 1e). In short, $\mathbf{v}$ is defined as

$$\mathbf{v} = \text{sort}([a_{ij}]), \qquad \text{for } i < j. \tag{5}$$

During the MFCCs sampling, the time translation of the spoken content occurs to the left when the window slides to the right. Consequently, the elements of A are shifted up in the direction of the main diagonal. Sorting the elements removes the translation effect because it loses the reference (i, j) of each element. In the fourth step, the set of row vectors $\mathbf{v}^{(t)}$ computed for every instant t forms the matrix V. Then, the PCA transformation estimates a lower dimension space, defining the MCAF vectors Y as

$$Y = VW \tag{6}$$

where the columns of W are the eigenvectors of the covariance matrix of V in order of decreasing eigenvalues. The MCAF vectors are the rows of Y (the output of the process shown in Figure 1).

The MCAF vectors extracted from groups of homogeneous and heterogeneous speech exhibit a distinct correlation behavior, this can be observed when comparing the plots in Figure 2a with 2b. We sample a random conversation from the AMI training corpus [26] and extract 19 MFCCs and energy. Then, the MCAF vectors are extracted using window of 0.5 secs for every 0.1 secs, setting σ to 1 and the MCAF dimension to 5. We use the ground truth of the speaker segments to label the MFCCs vectors according to the identity of the speakers. The MCAF vectors extracted from samples containing overlap speech are discarded. We defined that the samples from the heterogeneous group have two speakers in more than 10% of a sample.

4 Speaker Segmentation Method

Traditional segmentation methods use two sliding windows and a distance metric to evaluate whether two segments belong to the same speaker. They estimate

models to represent the discriminant information of the speakers and compute the distance between segments. The sliding process creates a distance curve, yielding peak values when the segments come from distinct speakers. A speaker change candidate (or relevant peak) is a peak value representing a local maximum, which prominence is above a threshold. The prominence is calculated as the difference between the peak value and the higher of two neighbor minimum values, each one located on a peak side (left and right) [3,6,19].

The proposed segmentation method extract the MCAF using two sliding windows and also use a distance metric. Our approach yields peak values spaced in small time intervals when these intervals contain speaker change points. During the sliding process, one peak is yielded when the right window becomes heterogeneous and a second peak is also yielded when the left window becomes heterogeneous. We detect relevant peaks when their prominence is above a threshold. The speaker changes are detected as the midpoint of intervals of at most 0.5 seconds.

We compare the segmentation performance of two front-end features: the i-vector as a baseline and the affinity features. The evaluated distances considering two feature vectors $\mathbf{f}_1$ and $\mathbf{f}_2$ are the Euclidean distance

$$dist_{\text{Euclidean}}(\mathbf{f}_1, \mathbf{f}_2) = \sqrt{\sum_{i=1}^{p}(f_1^i - f_2^i)^2}, \tag{7}$$

and the cosine distance

$$dist_{\text{cosine}}(\mathbf{f}_1, \mathbf{f}_2) = 1 - \frac{\mathbf{f}_1^\top \mathbf{f}_2}{||\mathbf{f}_1|| \cdot ||\mathbf{f}_2||}. \tag{8}$$

Modeling the features with PLDA can improve the segmentation performance by augmenting the inter-speaker variability. The PLDA score (Equation 3) between two segments is high if the speech content comes from the same speaker. We adapt the score for segmentation, defining a dissimilarity score as

$$dis_{\text{GPLDA}}(\mathbf{f}_1, \mathbf{f}_2) = -\Phi'_{\text{GPLDA}}(\mathbf{f}_1, \mathbf{f}_2) \tag{9}$$

where Φ'_{GPLDA} is the curve yielded using the PLDA score after normalization by the mean and standard deviation.

5 Experiments and Results

In this section, we describe the speaker segmentation experiments and the results using the AMI Corpus. First, the characteristics of fast style conversations are shown, following the acoustic features setup is described. We explain the setup of the i-vector framework and the affinity features. The tuning phase of the segmentation methods is also described. Then, the results are shown.

5.1 AMI dataset

The AMI Meeting Corpus is an open-access dataset consisting of 100 hours of meeting recordings. The corpus configuration is partitioned into training, development, and evaluation set. The training set contains 42 meeting recordings with the

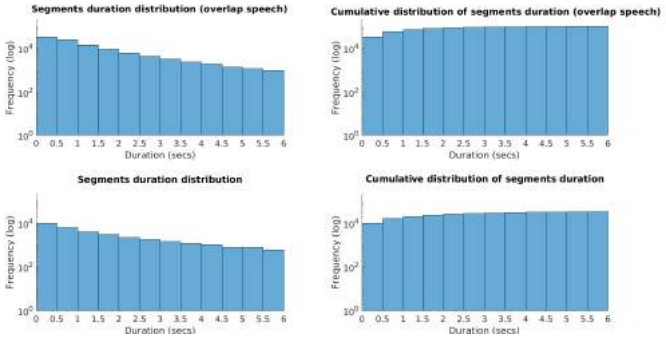

Fig. 3 Speaker duration of the segments in the training and development sets of AMI corpus. The fast style can be observed by the high frequency of short segments (< 1 sec).

total duration of 75 hours and 166 distinct speakers. The development set contains 11 meeting recordings with the total duration between 12 and 13 hours and 44 distinct speakers. The evaluation set contains 6 meeting recordings with the total duration close to 13 hours with 24 distinct speakers.

Figure 3 indicates that the training and development sets are composed of fast style conversations, given the number of speech segments of short duration. For the AMI corpus, the average duration of the speech segments is less than 1 second. We observe that the presence of overlap speech (two or more speakers talking at same time) increases the number of short segments. In this work, we do not consider the speaker changes with overlap speech [19,13].

5.2 Front-end features and background models

We extract 19 MFCCs and the energy from 20ms speech frames for every 10 ms, using 26 Mel-filter banks having the bandwidth limited to the 120–3800Hz range. Then, the Δ, and Δ^2 coefficients are computed, resulting in a 60-dimensional feature vector and no normalization or compensation techniques are applied.

5.2.1 The i-vector framework

We use the training data from the Fisher dataset [27] to train the UBM and the TV matrix T. The data consists of utterances from 4000 speakers (2000 per gender), 20 utterances per speaker with the duration above 3 seconds on average.

We train a gender-independent UBM of 512 distributions with a diagonal covariance matrix by combining two gender-dependent UBMs of 256 components. A new UBM is trained in the AMI training set using the pre-trained UBM parameters as initial parameters. Only pure speech segments are used (overlap speech parts are removed). All training steps run 50 iterations of the EM algorithm.

The matrix T is gender independent of rank 400 and trained using 10 iterations of the Expectation Maximization (EM) algorithm as described in [15]. A new matrix T is also trained in the AMI training set using the pre-trained T as an initial estimation. We compute the Baum-Welch statistics using a sliding window

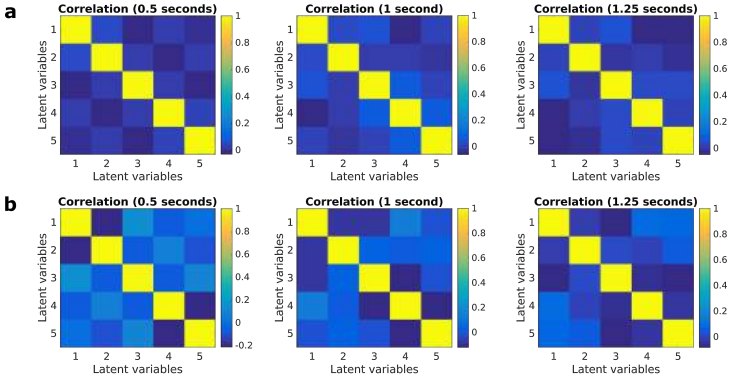

Fig. 4 Affinity features correlation for different sample sizes. The matrices in **a** belong to the homogeneous samples, and **b** shows the correlation from heterogeneous samples.

for three lengths: 0.5, 1, and 1.25 seconds for every 0.1 second. For each length, a different TV matrix is estimated.

For the PLDA training, we label the i-vectors using the speaker annotations of conversations in the AMI training set. During the extraction, if a speaker is active in more than 90% of the sampled speech, the sample receives the speaker's label. Otherwise, if a second speaker is also active in more than 10% of the sample, it receives the label "heterogeneous". We evaluate the PLDA using three different approaches: only homogeneous samples (each speaker is a class) named as i-vector PLDA (I), homogeneous and heterogeneous samples (each speaker is a class plus the "heterogeneous" class) named as i-vector PLDA (II), and for two classes ("homogeneous" and "heterogeneous") named as i-vector PLDA (III). We train the PLDA model for each matrix T. The rank of the matrix Ψ, the eigenvoices as described in Equation 2, is set to 200 (for the first two approaches) and is set to 1 for the third approach, because the last is a two-class problem and the LDA seeks for $c-1$ dimensions, where $c=2$. The matrix $\boldsymbol{\epsilon}$ is a full covariance matrix.

5.2.2 Affinity features

The affinity features are also extracted using three different window lengths: 0.5, 1, and 1.25 seconds for every 0.1 second. We set the number of principal components to 5, because it is the lowest dimension in Figure 2 that already exhibits discriminant information between the speech types. We set the parameter σ to 1 to simplify the study and to replicate the same results of Figure 2. Figure 4 shows a clear difference of correlations for different window lengths, if we group the data into: **a** homogeneous samples and **b** heterogeneous samples.

We train the PLDA following the same procedure as the i-vector PLDA. Here, the three approaches are named as MCAF PLDA (I), MCAF PLDA (II), and MCAF PLDA (III). We conserve the dimensionality setting the rank of the matrix Ψ to 5 for the first two approaches and to 1 for the third approach.

5.3 Setup of the segmentation method

We utilize the ground-truth labels of speech/non-speech segments to evaluate the speaker segmentation performance on ideal conditions. The development set is used to choose the best window length and prominence threshold. Three window lengths are evaluated: 0.5, 1, and 1.25 seconds. The prominence threshold is computed using a factor α of the distance curve standard deviation. We evaluate 1000 uniformly distributed α values starting from 0.002 to 2. The segmentation performance is reported using the Miss Detection Rate (MDR), False Alarms Rate (FAR), and the F_1 measure [1–3]. We use a collar of 0.5 seconds to count the true detected speaker turns. The Error Rate (ERR) is defined as the distance of the pair (MDR, FAR) to the ideal error (0,0). We select the best result as the one with the minimum ERR.

Table 1 Results for the development set using a window length of 0.5 seconds and choosing the α using the minimum ERR. The best results are shown in bold.

Distance	Features	MDR (%)	FAR (%)	F_1 measure (%)
Cosine	MCAF	5.810	78.605	27.964
Cosine	i-vector	4.449	78.621	30.165
Euclidean	MCAF	**0.168**	79.381	20.921
Euclidean	i-vector	0.183	79.287	18.969
PLDA	MCAF PLDA (I)	4.281	78.941	26.051
PLDA	i-vector PLDA (I)	14.570	76.620	39.302
PLDA	MCAF PLDA (II)	4.495	78.862	26.400
PLDA	i-vector PLDA (II)	15.120	**76.369**	**39.990**
PLDA	MCAF PLDA (III)	0.841	79.497	28.105
PLDA	i-vector PLDA (III)	3.914	78.917	30.798

Table 2 Results for the development set using a window length of 1 second and choosing the α using the minimum ERR. The best results are shown in bold.

Distance	Features	MDR (%)	FAR (%)	F_1 measure (%)
Cosine	MCAF	4.984	78.654	26.814
Cosine	i-vector	1.926	78.883	24.530
Euclidean	MCAF	1.345	79.600	23.742
Euclidean	i-vector	**0.688**	79.075	20.640
PLDA	MCAF PLDA (I)	1.529	79.128	23.151
PLDA	i-vector PLDA (I)	35.759	**66.620**	**50.918**
PLDA	MCAF PLDA (II)	1.269	79.206	21.978
PLDA	i-vector PLDA (II)	15.716	74.853	43.562
PLDA	MCAF PLDA (III)	2.263	79.356	24.677
PLDA	i-vector PLDA (III)	5.626	78.833	27.802

5.4 Segmentation Results

Tables 1, 2, and 3 show the best results for the respective window lengths: 0.5, 1, and 1.25 seconds. Table 4 and 5 summarize the best parameters and results found in the development set, respectively. These parameters are used in the AMI evaluation set.

Analyzing the i-vector results without the PLDA model, the lowest MDR (0.183% in Table 1) is obtained for the Euclidean distance with the window length

Table 3 Results for the development set using a window length of 1.25 seconds and choosing the α using the minimum ERR. The best results are shown in bold.

Distance	Features	MDR (%)	FAR (%)	F_1 measure (%)
Cosine	MCAF	5.320	78.950	25.682
Cosine	i-vector	1.911	78.669	23.299
Euclidean	MCAF	2.094	79.550	23.732
Euclidean	i-vector	**0.963**	79.041	22.113
PLDA	MCAF PLDA (I)	1.559	79.259	22.677
PLDA	i-vector PLDA (I)	17.123	**75.514**	**41.180**
PLDA	MCAF PLDA (II)	1.376	79.205	22.610
PLDA	i-vector PLDA (II)	15.288	76.258	39.613
PLDA	MCAF PLDA (III)	4.158	79.256	25.983
PLDA	i-vector PLDA (III)	3.226	79.171	24.431

of 0.5 seconds. Our proposal obtains the best MDR performance (0.168% in Table 1) also using the Euclidean distance and the window length of 0.5 seconds (Table 1), which means 8.2% of improvement compared to the i-vector MDR performance. The best FAR and F_1 measure performance are obtained for the cosine distance with the window length of 0.5 seconds for both i-vector (78.621% and 30.165% in Table 1) and MCAF (78.605% and 27.964% in Table 1).

Analyzing the PLDA approaches for both i-vector and MCAF, we observe the improvement of the FAR and F_1 measure for most of the methods as the MDR performance deteriorates (Tables 1-3).

Table 4 Best parameters found for the development set using the minimum ERR criterion to choose both window length and α.

Distance	Features	α	Window (secs)
Cosine	MCAF	0.098	1
Cosine	i-vector	0.138	1.25
Euclidean	MCAF	0.016	0.5
Euclidean	i-vector	0.062	1.25
PLDA	MCAF PLDA (I)	0.238	0.5
PLDA	i-vector PLDA (I)	1.504	1
PLDA	MCAF PLDA (II)	0.256	0.5
PLDA	i-vector PLDA (II)	0.804	1
PLDA	MCAF PLDA (III)	0.002	0.5
PLDA	i-vector PLDA (III)	0.274	0.5

Table 5 Best results for the development set using the minimum ERR criterion. The overall best results are shown in bold.

Distance	Features	MDR (%)	FAR (%)	F_1 measure (%)
Cosine	MCAF	4.984	78.654	26.814
Cosine	i-vector	1.911	78.669	23.299
Euclidean	MCAF	**0.168**	79.381	20.921
Euclidean	i-vector	0.963	79.041	22.113
PLDA	MCAF PLDA (I)	4.281	78.941	26.051
PLDA	i-vector PLDA (I)	35.759	**66.620**	**50.918**
PLDA	MCAF PLDA (II)	4.495	78.862	26.400
PLDA	i-vector PLDA (II)	15.716	74.853	43.562
PLDA	MCAF PLDA (III)	0.841	79.497	28.105
PLDA	i-vector PLDA (III)	3.914	78.917	30.798

Table 6 Results for the evaluation set. The best results are shown in bold.

Distance	Features	MDR (%)	FAR (%)	F_1 measure (%)
Cosine	MCAF	6.577	82.664	23.093
Cosine	i-vector	1.525	82.376	19.620
Euclidean	MCAF	**0.165**	83.124	17.640
Euclidean	i-vector	1.286	82.681	18.722
PLDA	MCAF PLDA (I)	5.383	82.726	22.189
PLDA	i-vector PLDA (I)	33.640	**71.928**	**45.975**
PLDA	MCAF PLDA (II)	5.989	82.688	22.454
PLDA	i-vector PLDA (II)	14.881	79.134	37.720
PLDA	MCAF PLDA (III)	2.278	83.214	20.386
PLDA	i-vector PLDA (III)	3.546	82.463	22.203

Table 6 shows the evaluation results. We observe that the MCAF and i-vector using the Euclidean distance obtain the best MDR performance (0.165% and 1.286%, respectively in Table 6), using the window length of 0.5 and 1.25 seconds, respectively (Table 4). The MDR for our proposal has 1.1% of difference compared to the best result of the i-vector (Table 6). The processing time of MCAF extraction is at least 3 times faster (top of Figure 5) than the processing time of i-vector extraction (the time spent on the training phase is not considered on both features in this analysis).

6 Discussion

In general, the lowest MDRs are obtained using the Euclidean distance, showing that the magnitude information from both i-vector and MCAF is important to segment the speakers for fast style conversations. The MDR obtained when we use the i-vector is higher than the observed for our proposal. A higher MDR implies in more heterogeneous segments, which demands an additional re-segmentation step in the speaker diarization systems. The MDR and FAR obtained for our proposal become worst when the window length is above 0.5 seconds (Tables 2 and 3), which is expected, considering that the most different correlation behavior happens when the window length is 0.5 seconds (Figure 4). The 0.5 seconds window presents the lowest processing time in the experiments (first plot in Figure 5). The window length used to obtain the best results for the i-vector is 1.25 seconds, which presents the highest processing time in the experiments (third plot in Figure 5). We also observe the deterioration of the i-vector performance with the mismatch of speakers in the evaluation set, comparing the best result for the development set of MDR (0.963%) and FAR (66.620%) in Table 5 to the best result for the evaluation set of MDR (1.286%) and FAR (71.928%) in Table 6. For our approach, the mismatch of speakers does not deteriorates the best MDR and FAR results (Table 5 and 6).

Modeling the i-vector and MCAF using the PLDA does not improve the segmentation performance as we would expect. The GPLDA utilize a linear dimensionality reduction method to extract latent features and a generative model for classification. The linearity premise to extract more features from both i-vector and MCAF can not be useful to the space of classes “homogeneous” (speakers as classes, or just one class) and “heterogeneous” speech.

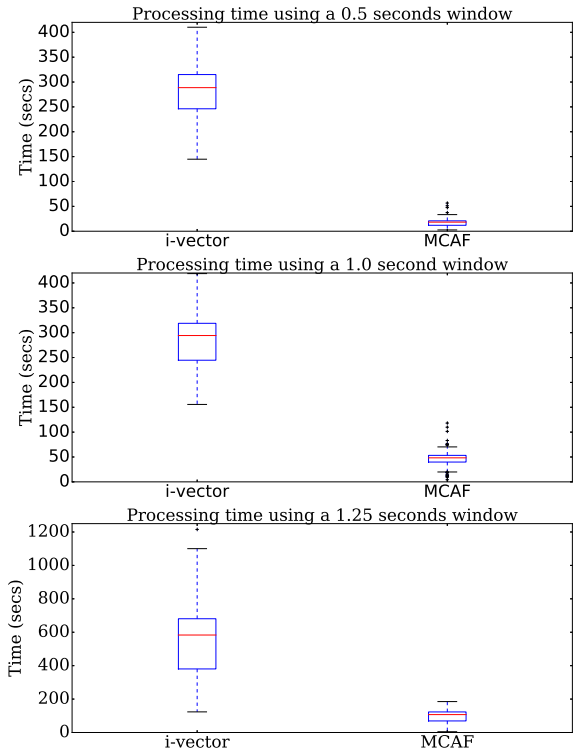

Fig. 5 Processing time for features extraction from the AMI Corpus. We collect the processing time for each conversation.

7 Conclusion

The proposed segmentation approach detects the speaker transitions when two adjacent segments have different types of speech: homogeneous and heterogeneous speech. We present a feature to represent the discriminant information between homogeneous and heterogeneous speech. The affinity features space derived from MFCCs shows the types of speech have distinct behavior. Experiment results show the proposed approach achieves comparable performance to the state-of-the-art i-vector as front-end feature with lower computational cost.

For future work, we experiment with model-based segmentation approaches [9, 8] using the MCAF. We also plan to investigate the use of the MCAF in other tasks, such as speaker clustering and verification. More conversation domains may be considered, like telephone speech, and broadcast news, to evaluate the channel effects on the MCAF. Further researches to discriminate overlap speech and homogeneous speech is also of interest since this problem remains a challenge in the field of speaker diarization.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

LVN designed the features of the study, carried out the implementation and partially the experiments, and he drafted the manuscript. HNBP and IRT participated in the study, helped in the experiments and the draft. GDCC and AGD participated in the study, read and helped to write the final manuscript. All authors approved the final manuscript.

Funding

This work was partially funded by CAPES, CNPq and FACEPE.

References

1. X.A. Miro, S. Bozonnet, Speaker diarization: A review of recent research, IEEE Transactions on Audio, Speech, and Language Processing 20 (2) (2012) 356–370.
2. N. Evans, S. Bozonnet, D. Wang, C. Fredouille, R. Troncy, A Comparative Study of Bottom-Up and Top-Down Approaches to Speaker Diarization, IEEE Transactions on Acoustics, Speech, and Signal Processing 20 (2) (2012) 382–392.
3. M. Kotti, V. Moschou, C. Kotropoulos, Speaker segmentation and clustering, Signal Processing, 88 (5), 2008, pp. 1091–1124
4. S. Chen, P. Gopalakrishnan, Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion, in Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, Lansdowne, 1998, number 6, pp. 67–72.
5. M.A. Siegler, U. Jain, R. Bhiksha, R.M. Stern, Automatic segmentation, classification and clustering of broadcast news audio, in Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, Chantilly, 1997, pp. 97– 99
6. P. Delacourt, C. Wellekens, DISTBIC: A speaker-based segmentation for audio data indexing, Speech Communication, vol. 32, no. 1-2, 2000, pp. 111–126
7. Y. Lavner, D. Ruinskiy, A Decision-Tree-Based Algorithm for Speech/Music Classification and Segmentation, in EURASIP Journal on Audio, Speech, and Music Processing, 1 (2009), pp. 1–14, 2009
8. T. Butko, C. Nadeu, Audio segmentation of broadcast news in the Albayzin-2010 evaluation: overview, results, and discussion, in EURASIP Journal on Audio, Speech, and Music Processing, 1 (2011), pp. 1–10, 2011
9. D. Castán, A. Ortega, A. Miguel, E. Lleida, Audio segmentation-by-classification approach based on factor analysis in broadcast news domain, in EURASIP Journal on Audio, Speech, and Music Processing, 1 (2014), pp. 1–13, 2014
10. D. Castán, D. Tavarez, P. Lopez-Otero, J. Franco-Pedroso, H. Delgado, E. Navas, L. Docio-Fernández, D. Ramos, J. Serrano, A. Ortega, E. Lleida, Albayzín-2014 evaluation: audio segmentation and classification in broadcast news domains, in EURASIP Journal on Audio, Speech, and Music Processing, 1 (2015), pp. 1–9, 2015
11. N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification, IEEE Transactions on Audio, Speech, and Language Processing 19 (4) (2011) 788–798.
12. P. Kenny, Bayesian speaker verification with heavy-tailed priors, in:The speaker and language recognition workshop (Odyssey),p. 14, 2010
13. F. Castaldo, D. Colibro, E. Dalmasso, P. Laface, C. Vair, Stream-based speaker segmentation using speaker factors and eigenvoices, in IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4133—4136, 2008

14. P. Kenny, D. Reynolds, F. Castaldo, Diarization of Telephone Conversations using Factor Analysis, in IEEE Journal of Selected Topics in Signal Processing, 6 (4), pp. 1–7, 2010
15. P. Kenny, P. Dumouchel, Disentangling speaker and channel effects in speaker verification, In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP, pp. 37–40, 2004
16. S. Shum, D.A. Reynolds, J.R. Glass, N. Dehak, E. Chuangsuwanich, J. Glass, Exploiting Intra-Conversation Variability for Speaker Diarization, IEEE International Conference on Speech and Language Processing, Florence, pp. 945–948, 2011
17. N. Dehak, P.A. Torres-Carrasquillo, D. Reynolds, R. Dehak, Language recognition via Ivectors and dimensionality reduction, in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pp. 857–860, 2011
18. M. Senoussaoui, P. Kenny, T. Stafylakis, P. Dumouchel, A study of the cosine distance-based mean shift for telephone speech diarization, in IEEE Transactions on Audio, Speech and Language Processing, 1 (22), pp. 217–227, 2014
19. B. Desplanques, K. Demuynck, J.-P. Martens, Factor analysis for speaker segmentation and improved speaker diarization, in: INTERSPEECH, pp. 3081–3085, 2015
20. L.V. Neri, H.N.B. Pinheiro, I.R. Tsang, G.D.C. Cavalcanti, A.G. Adami, Speaker segmentation using i-vectors in meetings domain, in: International Conference on Acoustics, Speech and, Signal Processing, pp. 5455–5459, 2017
21. Z. Zajíc, M. Kunešová, V. Radová, Investigation of Segmentation in i-Vector Based Speaker Diarization of Telephone Speech, in: International Conference on Speech and Computer, pp 411–418, 2016
22. A. Poddar, M. Sahidullah, G. Saha, Improved i-vector extraction technique for speaker verification with short utterances, International Journal of Speech Technology, pp. 1–16, 2017
23. A. Poddar, Md, Sahidullah, G, Saha, Speaker verification with short utterances: a review of challenges, trends and opportunities, IET Biometrics, 7 (2), 2017
24. V. Hautamäki, Y.C. Cheng, P. Rajan, and C.H. Lee, Minimax i-vector extractor for short duration speaker verification, in: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, pp. 3708–3712, 2013.
25. A. Ng, M.I. Jordan, Y. Weiss, On Spectral Clustering: Analysis and an Algorithm, Advances in Neural Information Processing Systems, pp. 849–856, 2002.
26. J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kaldec, V. Karaiskos, W. Kraaij, M. Kronenthal The AMI meeting corpus: A pre-announcement, International workshop on machine learning for multimodal interaction, pp. 28–39, 2005
27. C. Cieri, D. Miller, K. Walker, The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text., LREC, 4, pp. 69–71, 2004