Nanodegree Engenheiro de Machine Learning

Projeto Final

Diarização de tradução alternada com retreinamento incremental

Definição

Este é um projeto de classificação de trechos de áudio de discursos traduzidos. Nos discursos específicos que desejamos classificar um palestrante e uma tradutora falam alternadamente. Cada pequeno trecho de fala do palestrante é seguido de sua respectiva tradução. Nosso objetivo é fragmentar o áudio do discurso e classificar cada fragmento indicando quem fala em cada um.

O problema de separar vozes em áudio é conhecido pelo nome de **diarização** (em inglês *diarization*) e possui algumas soluções implementadas. Entretanto nossa situação possui algumas particularidades, o que enseja uma solução específica. Na verdade estas particularidades tornam nosso cenário mais simples que o usual. As seguintes condições são válidas em nosso caso e servem de **base** para nossa solução:

- O áudio possui apenas **vozes**: não inclui música ou outros tipos de som.
- Somente **duas** pessoas falam.
- Falam de forma **alternada**
- Em geral falam **sem sobreposição**.

Também identificamos outras características incidentais. Elas podem variar em amostras futuras e não são usadas na solução:

- Os falantes falam em **línguas diferentes** (português e inglês).
- Falam em **ritmos diferentes** (o palestrante fala de forma mais pausada).
- São de **sexos** diferentes.

Temos algumas motivações para esta solução. Ela pode servir, por exemplo, para construir um áudio em que somente o palestrante fala, retirando a tradução. Também pode funcionar como parte de um processo de conversão de fala em texto, com marcação dos momentos de fala no áudio. Isso, por sua vez, pode servir de base para busca em áudio, em que o instante de ocorrência de termos é indicado nos resultados de busca.

Nossa estratégia de solução utiliza aprendizagem supervisionada baseada em uma máquina de suporte de vetores (SVM). Modelamos o áudio como uma série pontos no tempo, que são janelas de alguns milisegundos. A SVM irá classificar cada ponto com uma de duas classes: *palestrante* ou *tradutor*.

O processo consiste dos seguintes passos:

1. **Fragmentar o áudio por silêncios**

   Usando pausas, dividimos o áudio em trechos em que somente uma pessoa fala. Cada fragmento é constituído por uma série de pontos no tempo.

2. **Pré-classificar alguns fragmentos**

   Pedimos a um operador que ouça e indique quem fala em alguns poucos trechos. Esses trechos tem duração total de poucos segundos.

3. **Treinar** o classificador e **classificar** todos os pontos de todos os fragmentos.

4. Calcular a **proporção** de pontos de **cada classe** em cada trecho.

   A **classe do fragmento** será a classe com maior proporção de pontos no trecho. Diremos que esta proporção é a **probabilidade** da classificação.

5. Tomar certo número dos fragmentos melhor classificados (com **maior probabilidade**), agregá-los à base anterior de treinamento e, com essa nova base, **retreinar** o classificador e **reclassificar** todos os pontos e fragmentos.

   Isto é feito sucessivamente, certo número de vezes ou de forma exaustiva.

O efeito esperado desse processo é obter classificações cada vez melhores para os fragmentos.

As métricas de avaliação que utilizaremos são o **percentual de número de fragmentos** classificados incorretamente e o **tempo** percentual desses fragmentos, no áudio de um discurso. Ou seja os **erros percentuais de classificação** em relação ao número e tempo dos fragmentos.

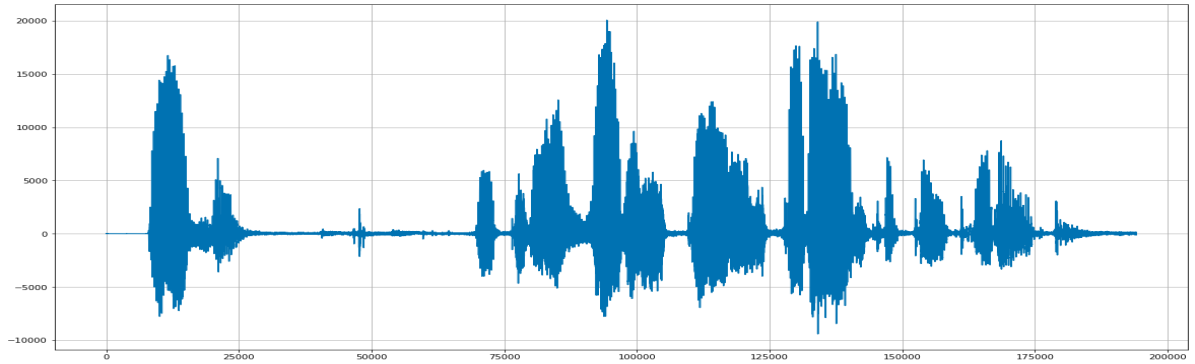
Faremos esta avaliação através de um experimento que registra estes percentuais de erro, em diferentes níveis, a cada iteração do processo. Estes erros devem convergir para valores baixos como medida de sucesso.

Análise

Os dados que utilizamos como base são áudios de 4 discursos, de aproximadamente uma hora cada. Para tratar cada áudio como uma sequência temporal e extrair características de cada momento, fracionamos o áudio em janelas de 25 milissegundos de largura, tomando uma nova janela a cada 10 ms. Por simplicidade chamaremos essas janelas de **pontos**. Para cada ponto calculamos duas características:

- o **volume**, em decibéis relativos à escala máxima (*decibels relative to full scale - dBFS*), e
- os **coeficientes mel-cepstrais** (*Mel Frequency Cepstral Coefficients - MFCC*).

Volume

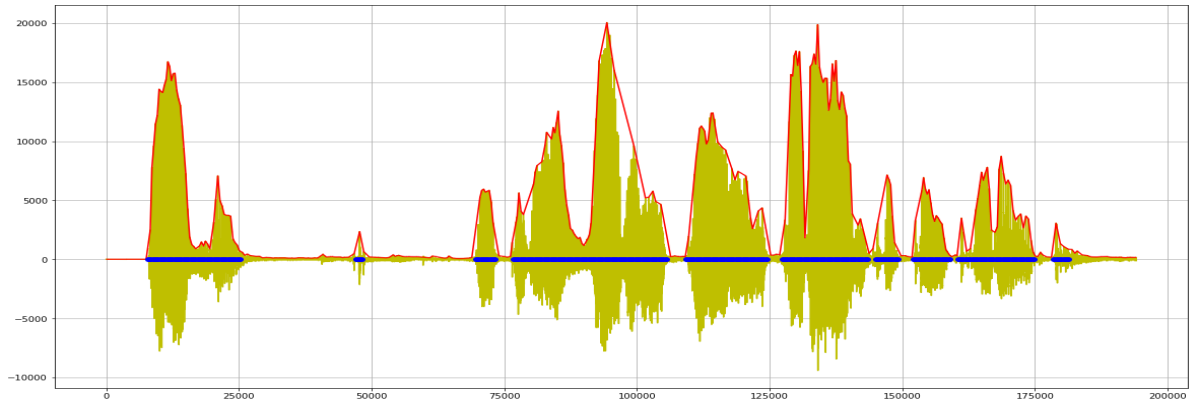


Onda sonora de uma amostra de 4.4 segundos de um discurso

Ao analisar o volume dos pontos (em dBFS), percebemos uma grande variabilidade. Alguns pontos inclusive, de ausência total de som, têm valores de infinito negativo. Esses são pontos provavelmente de transição da edição de áudio, pois, mesmo em ambiente muito silencioso, sempre existe som residual. Descartando estes pontos de exceção podemos calcular algumas estatísticas, tanto sobre um discurso completo e como sobre um pequeno trecho de silêncio (sem falas):

Volume em um discurso completo		Volume em um trecho de silêncio	
mean	-35.831240	mean	-54.289840
std	17.098029	std	1.521404
min	-90.308999	min	-58.713327
25%	-53.787503	25%	-55.345238
50%	-34.659546	50%	-54.322188
75%	-18.410104	75%	-53.283832
max	-5.914165	max	-49.170902

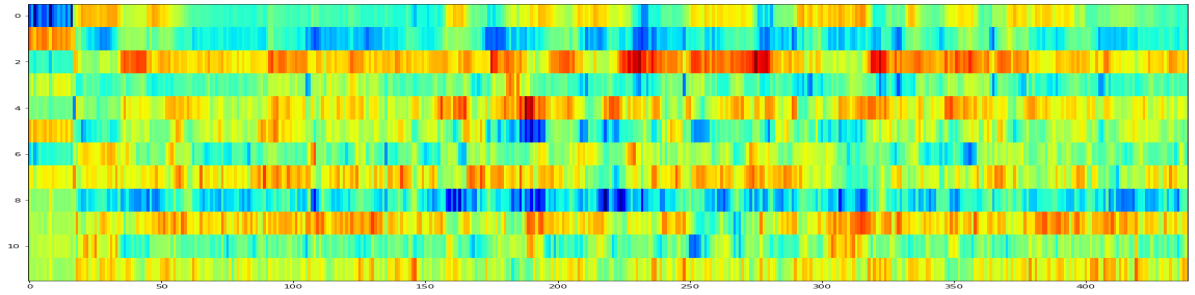
É importante ressaltar que a medida *dBFS* é relativa e afetada por fatores ambientais (como o equipamento de gravação). Além disso, o nível de ruído basal durante o discurso é variável. De forma empírica, observamos que o volume de **-42 dBFS** é um limite razoável para distinguir trechos de fala nos discursos. Consideramos que os pontos com volume acima dessa medida são **audíveis**. Tomamos este número como um parâmetro de entrada da classificação. A figura a seguir destaca trechos audíveis segundo esse critério.



Volume máximo a cada 300 ms (em vermelho) sobre a onda de uma amostra e pontos audíveis (acima de -42 dBFS, marcados em azul)

Coeficientes mel-cepstrais (MFCC)

Os coeficientes mel-cepstrais (MFCC) são comumente utilizados como base para processos de reconhecimento de voz, desde a diarização até a conversão de voz em texto. Tomamos, para cada ponto do áudio, os coeficientes de 2 a 13, que é a configuração usual neste tipo de problema. Desta forma, para cada ponto de nossa série temporal teremos um vetor de 12 valores reais, que são os MFCC.



Mapa de calor dos coeficientes mel-cepstrais (MFCC) de uma amostra

Os MFCC têm a propriedade de capturar a articulação vocal, ou seja, seus valores determinados por aspectos físicos do aparelho fonador humano. Por isso entendemos que sua dispersão será afetada pelo timbre da voz.

Support Vector Machine

Uma vez que optamos por uma abordagem de aprendizagem supervisionada, um candidato natural para a fazer a classificação destes vetores é uma máquina de suporte de vetores (Support Vector Machine). Em particular, utilizamos um Support Vector Classifier, da biblioteca *scikit-learn*, que se mostrou muito adequado já nos primeiros testes do estudo. A classificação obtida para pontos isolados é **agregada** para classificar fragmentos. Sob a hipótese de que os fragmentos contém apenas uma voz, podemos "corrigir" pontos incorretamente classificados, usando o algoritmo que detalharemos mais à frente.

Fragmentação por pausas

O método de fragmentação que desenvolvemos foi bastante eficaz em gerar fragmentos com apenas uma voz. Após classificarmos manualmente os trechos de 4 discursos observamos que, em geral, menos de 1% dos fragmentos gerados pelo método contiveram mais de uma voz.

arquivo	proporção fragmentos com dois falantes
data/2014_01_20_F.wav	0,68%
data/2014_01_21_F.wav	1,11%
data/2014_01_22_F.wav	0,67%
data/2014_01_23_F.wav	0,78%

Metodologia

Ajustes de formato

Os áudios originais dos discursos foram fornecidos no formato MP3, 192 kbps, 44.1 kHz, stereo. Para uso da biblioteca de extração de MFCC os convertemos para WAV 16 bit, mono, mantendo a frequência de 44.1 kHz. Além disso retiramos os primeiros 33 segundos de cada arquivo, que continham uma mensagem padrão anterior ao início dos discursos. A conversão para *mono* foi necessária para a extração de atributos do áudio.

Fragmentação de áudio por pausas

O objetivo deste processo é dividir o áudio de um discurso em uma sequência de trechos com apenas um falante.

A técnica básica é a fragmentação por silêncios. O método consiste em tomar uma janela móvel de **largura** base, ao longo de todo o áudio, medindo seu **volume**. Em todos os pontos onde o volume for inferior a determinado nível, dividimos o áudio.

Nosso método fragmenta o áudio por essa técnica, de forma recursiva, tentando níveis crescentes de volume (de -42 a -34 dBFS) e, para cada volume, larguras decrescentes de janela, variando de 500 a 200 ms. A fragmentação é interrompida quando os fragmentos são menores que um valor alvo ou exaurimos as faixas de volume/largura.

O algoritmo dá preferência à fragmentação em silêncios mais profundos e mais longos, porém de maneira flexível. O relaxamento de volume e largura é uma forma de obter fragmentos suficientemente pequenos, mesmo em condições variáveis de ruído e transição entre falas.

Ground truth

De posse dos fragmentos dos áudios, assinalamos manualmente quem fala em cada um deles. Para isso usamos um processo assistido que, na verdade, utiliza o próprio método de classificação (que detalharemos a seguir). Após atribuirmos a classificação real para alguns trechos (10 segundos são suficientes), o processo, de forma recorrente:

- classifica todos os trechos
- agrupa alguns trechos por falante, segundo a classificação
- reproduz todo o grupo, sugerindo a classificação como ground-truth (os valores possíveis são: *"palestrante"*, *"tradutor"* ou *"ambos"*)
- pede a confirmação e permite mudança da sugestão

Aprendizado

Com o áudio fragmentado, utilizamos um *Support Vector Classifier* (SVC) para classificar todos os pontos do áudio. Inicialmente o classificador é treinado com base na *ground truth* de poucos trechos, que somados dão em torno de 10 segundos para cada falante (palestrante ou tradutor). A partir daí o algoritmo refina sucessivamente o classificador, expandindo a base de treinamento a partir das "bordas" das melhores classificações. Ele itera realizando os seguintes passos:

1. Calcula a **proporção** de pontos de **cada classe** em cada fragmento de áudio, definindo assim:
    a. Uma **classe do fragmento** é a classe com maior proporção de pontos, e
    b. a **probabilidade da classificação**, que é proporção obtida

2. Seleciona certo número dos fragmentos **melhor classificados** (com maior probabilidade) de fora da base de treinamento e os agrega à ela.

   Nessa seleção tomamos quantidades aproximadamente iguais de fragmentos de cada classe.

3. Com a base de treinamento expandida, **retreina** o classificador e **reclassifica** todos os pontos do áudio.

4. Retorna ao passo (1).

Uma dificuldade de implementação encontrada foi o estouro de memória ao calcular os MFCCs para áudios grandes. Contornamos isso dividindo o áudio em segmentos de até 10 min. Foi importante reservar uma pequena margem ao redor de cada segmento, ou seja, tomar segmentos com sobreposição de alguns milisegundos, pois o cálculo dos MFCCs é afetado pelas bordas. Um teste unitário demonstra que esta adaptação não afeta o cálculo final.

Resultados

Realizamos experimentos para verificar a eficácia do método de classificação descrito. Na reclassificação e previsão exaustiva de um trecho de 10 minutos obtivemos as taxas de erro a seguir. Os conjuntos mencionados são:

- ***training****:* a base usada para treinamento da iteração
- ***remaining****:* o restante dos fragmentos
- ***all****:* a lista completa de fragmentos

Medimos os erros proporcionais quanto ao **número de fragmentos** *(N)* e **tempo total** *(Time)* dos fragmentos, para cada um desses grupos.

iteração	trainning (N)	trainning (Time)	remaining (N)	remaining (Time)	all (N)	all (Time)
0	0	0	5,67%	4,53%	5,31%	4,28%
1	0	0	3,28%	2,50%	2,90%	2,24%
2	0	0	2,31%	1,92%	1,93%	1,58%
3	0	0	2,45%	2,24%	1,93%	1,73%
4	0	0	1,95%	1,77%	1,45%	1,28%
5	0	0	2,74%	3,10%	1,93%	2,09%
6	0	0	2,17%	2,08%	1,45%	1,28%
7	0	0	3,08%	3,64%	1,93%	2,09%
8	0	0	2,44%	2,34%	1,45%	1,28%
9	0	0	2,59%	2,54%	1,45%	1,28%

10	0	0	2,75%	2,81%	1,45%	1,28%
11	0	0	2,94%	3,00%	1,45%	1,28%
12	0	0	4,17%	5,59%	1,93%	2,09%
13	0	0	4,40%	5,98%	1,93%	2,09%
14	0	0	3,49%	4,14%	1,45%	1,28%
15	0	0	2,47%	3,41%	0,97%	0,97%
16	0	0	2,60%	3,71%	0,97%	0,97%

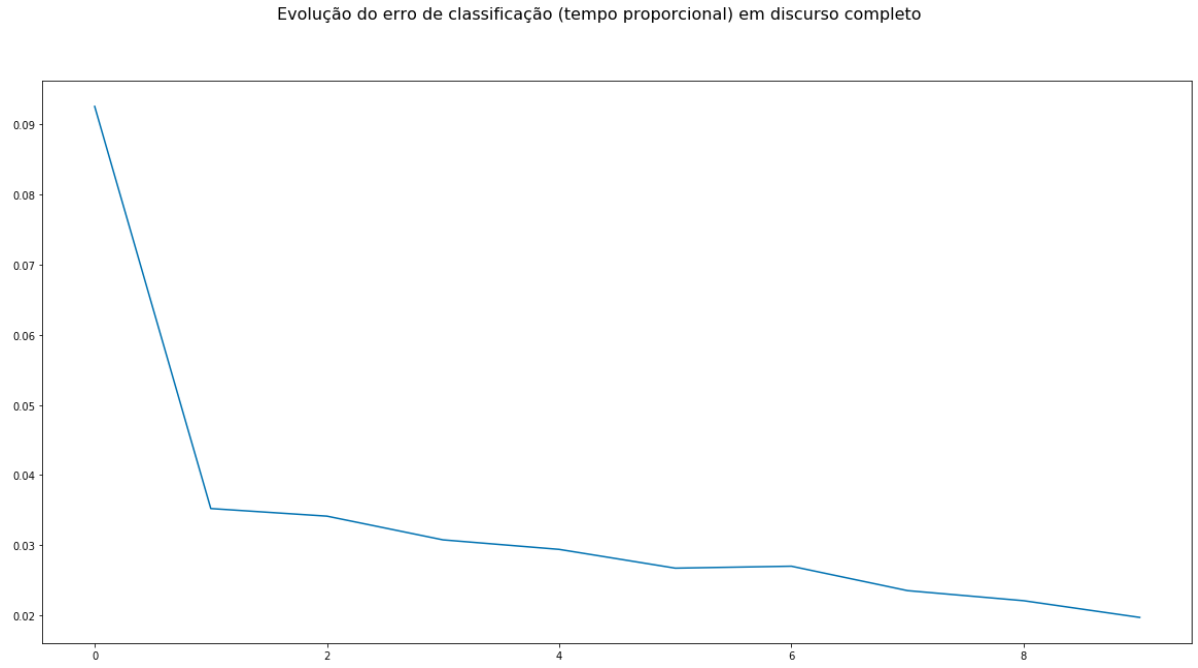
Realizamos experimentos com áudios de 4 discursos completos, de aproximadamente uma hora de duração. Em todos os cenários o método de reclassificação incremental de descrevemos exibiu o mesmo tipo de comportamento, com taxa de erro convergente para valores menores do que 2%.

Isso é especialmente interessante considerando o fato de que o treinamento foi feito com apenas cerca de 10 segundos de cada voz. O treinamento pede então menos de 0,6% do tempo total de um discurso para ser eficaz.

Conclusão

Na abordagem que apresentamos, consideramos que o problema de separação de discursos traduzidos foi resolvido, utilizando aprendizagem supervisionada, com baixa margem de erro. Conseguimos uma forma adequada de fragmentar os áudios por falantes e classificar cada trecho com uma Support Vector Machine. Além disso melhoramos a precisão das previsões explorando a segregação das falas por fragmentos. Essa foi uma forma de explorar as "bordas" das melhores previsões como forma de generalizá-las.

De fato, percebemos que as taxas de erro, embora decrescentes, convergem rapidamente para uma constante,e melhoram pouco a partir de certo número de iterações. Isso é natural devido ao fato de que existem fragmentos residuais com duas vozes, que não podem ser corretamente identificados.



Importância da amostragem equilibrada

Um aspecto fundamental do método foi a escolha de, ao selecionar os fragmentos melhor classificados para o retreinamento, tomar **quantidades equilibradas de ambos os falantes**. Em experimentos intermediários, que fracassaram totalmente, não levamos esse detalhe em consideração: retreinamos o classificador pelo critério único de selecionar os melhores fragmentos. Por alguma razão os fragmentos com melhor classificação eram todos de uma das vozes, especificamente, da tradutora. Ao retreinar o classificador com mais e mais fragmentos classificados como da tradutora (corretos ou não) a taxa de erros divergia, chegando a passar de 60%. A cada iteração mais e mais fragmentos, e com maior probabilidade, eram atribuídos a essa classe, em geral incorretamente.

Melhorias futuras

Identificamos algumas melhorias futuras possíveis para o processo.

Desempenho

O processo atual está longe de ter um bom desempenho. A parte de fragmentação utiliza uma biblioteca de alto nível, não adequada ao grande número de operações de corte de áudio que precisamos fazer ao longo do processo. Certamente melhorará muito se trabalharmos em nível mais baixo, diretamente com arrays da onda sonora.

Também podemos dividir o processamento em múltiplas tarefas assíncronas, explorando o uso de mais de um processador.

Busca de fragmentos irregulares

Podemos melhorar o processo como um todo identificando previamente fragmentos suspeitos de conter mais de uma voz e separá-los. Isso pode ser feito, por exemplo, apontando fragmentos muito grandes. Também podemos explorar a subdivisão por classificação, ou seja, observando a classe atribuída a cada ponto do fragmento e identificando agrupamentos.

Revisão pós-classificação

Também podemos melhorar a utilidade do processo oferecendo ao final uma lista dos fragmentos com piores classificações. Eles tendem a ser aqueles onde ocorre colisão de vozes ou que simplesmente foram classificados de forma errada.