

GRUPO DE ESTUDO DE OPERAÇÃO DE SISTEMAS ELÉTRICOS - GOP

PROCESSAMENTO DE ARQUIVOS DE ÁUDIOS E TEXTO RELACIONADOS A COMUNICAÇÃO VERBAL DA OPERAÇÃO DO SETOR ELÉTRICO

**MATHEUS NASCIMENTO SOARES MARQUES DE LIMA(1);BRUNA HAMES COELHO(1);FABRICIO YUTAKA KUWABATA TAKIGAWA(1)
IFSC(1)**

RESUMO

Na operação do setor elétrico, o Operador Nacional do Sistema se comunica com os agentes por meio de ligações telefônicas. Toda comunicação é gravada e armazenada para o setor de pós-operação realizar oitivas quando necessário. No entanto, trata-se de um procedimento rotineiro e que demanda tempo na consulta de informações. Neste sentido, o trabalho proposto propõe uma metodologia para o processamento dos arquivos de áudios e de textos, buscando uma transcrição que preserve as principais informações, explorando soluções gratuitas. A metodologia proposta foi analisada por meio de métricas de quantificação de perda de palavras de forma geral e de palavras-chaves.

PALAVRAS-CHAVE

Operação do Setor Elétrico; Comunicação Verbal; Processamento e Transcrição de Áudios; Processamento e Análise de Textos; Solução Gratuita.

1.0 INTRODUÇÃO

A operação centralizada do Sistema Interligado Nacional (SIN) é uma tarefa complexa desempenhada pelo Operador Nacional do Sistema Elétrico (ONS). O ONS opera o SIN por meio de centros regionais ao longo do território brasileiro, cada centro se comunica com os agentes do setor elétrico para a realização da operação coordenada de diversos ativos do setor [1]. De maneira geral, os centros regionais de operação se comunicam com os agentes por meio de ligações telefônicas. Toda comunicação efetuada nas salas de operação é gravada (armazenada em um servidor) e, posteriormente, é disponibilizada para o setor de pós-operação, para ser consultada quando for necessária a validação de informações. Para manter uma padronização na comunicação, o ONS desenvolveu um documento específico que descreve como o diálogo deve ocorrer [2]. A Figura 1 (a esquerda) exhibe a sequência de comunicação a ser respeitada visando minimizar a ocorrência de possíveis equívocos.

Por outro lado, o setor de pós-operação do ONS efetua a ponderação das ações tomadas na operação, assim como a confirmação de informações da operação realizada (validação de dados horários, montantes, tipo de serviço, tipo de ocorrências, entre outros). O escopo das informações abordadas pelo operador é consideravelmente extenso, considerando particularidades operativas da geração, da transmissão e da fronteira com a distribuição de energia. A Figura 1 (a direita) ilustra alguns assuntos tratados pelo operador do ONS.

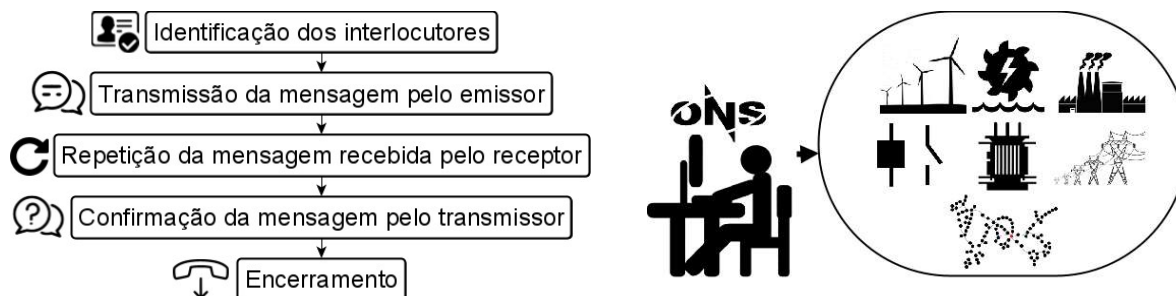


FIGURA 1 - Etapas para a comunicação verbal na operação, baseado em [2] (esquerda) e Comunicação e operação do setor elétrico (direita).

O procedimento de consultas de informações, normalmente, é realizado por meio de oitivas das ligações efetuadas, ou seja, são consultados os dados por meio da escuta de áudios. Pode-se afirmar que esse procedimento é um aspecto rotineiro e ilustra uma lacuna que pode ser aprimorada, visto que um dos principais problemas é o tempo

gasto na realização dessa atividade pelos profissionais. Adicionalmente, dependendo da ocorrência sistêmica podem ser necessárias diversas oitivas a serem realizadas.

Neste sentido, o presente trabalho propõe uma metodologia para o processamento dos arquivos de áudios e, posteriormente, de texto de forma gratuita, buscando uma transcrição automatizada que consiga compilar as principais informações da oitiva. No processamento proposto são abordados: o tratamento dos áudios, considerando o controle no consumo, limites de requisições e/ou de minutos transcritos, assim como, os ajustes e tratamentos no texto transcrito, para que a informação necessária seja evidente e de fácil acesso ao responsável pela oitiva. A Figura 2 ilustra o processo proposto.

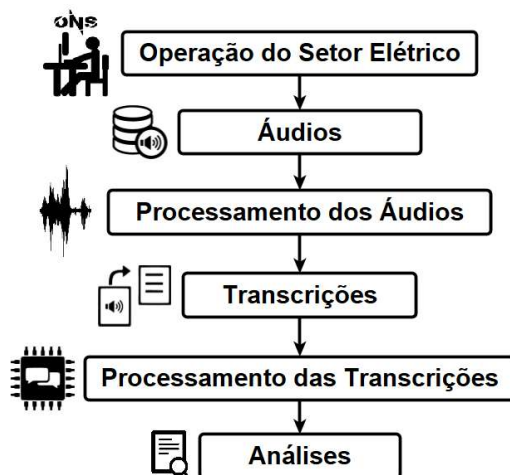


FIGURA 2 - Processo geral da metodologia proposta.

Pode-se observar no processo proposto que os áudios da operação do setor elétrico devem ser, inicialmente, processados e transcritos. Posteriormente a transcrição, ocorre o tratamento da mesma e, finaliza-se com a sua análise. Todo o processamento dos áudios é efetuado utilizando bibliotecas gratuitas em *Python* e o transcritor explorado foi selecionado baseado na qualidade e nas limitações impostas em relação ao uso gratuito dos serviços. Na etapa de processamento do texto, foi realizado o agrupamento das palavras e o treinamento de um algoritmo capaz de representar palavras com vetores para análise de similaridade.

Este artigo está organizado conforme descrito na sequência. A Seção 2.0 descreve os materiais e métodos, a Seção 3.0 aborda a metodologia em relação ao processamento dos áudios, a Seção 4.0 aborda transcrição dos áudios e a Seção 5.0 aborda o processamento dos textos transcritos. Posteriormente, a Seção 6.0 compila as principais conclusões obtidas.

2.0 - MATERIAIS E MÉTODOS

A principal ferramenta utilizada é a linguagem de programação *Python* [3], que possibilita o acesso a uma grande gama de bibliotecas relacionadas ao processamento de áudio e de processamento de linguagem natural, que podem ser acessadas usando o site *PyPI* [4]. Em relação ao material (*dataset* utilizado), foi disponibilizado pelo ONS uma determinada quantidade de arquivos de áudio no formato WAV com um único canal e taxa de amostragem de 8.000 kHz. Neste sentido, realizando a escuta dos mesmos, foi selecionado um grupo de 110 arquivos de áudios para a realização das transcrições manuais e análises. A Figura 3 ilustra a caracterização dos áudios originais selecionados, em relação a hora do dia (a esquerda) e ao comprimento dos áudios (a direita).

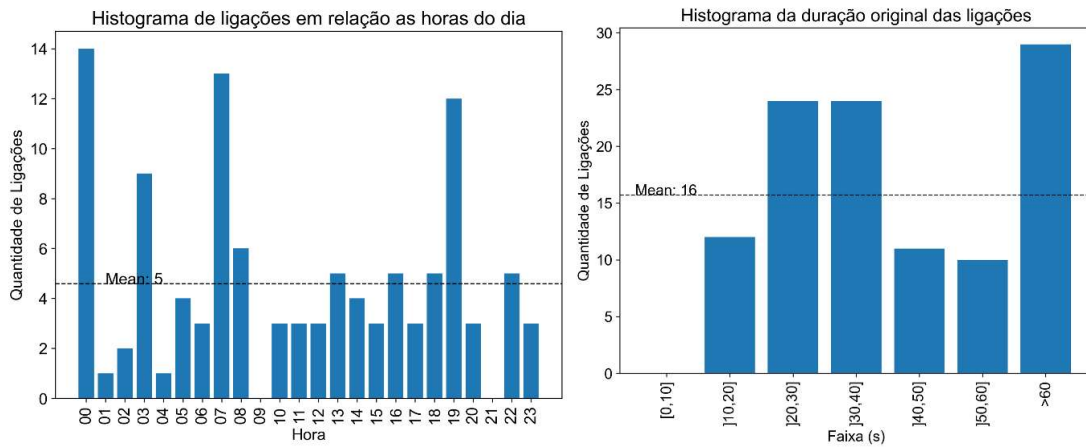


FIGURA 3 - Características do grupo de áudios analisado.

Pode-se observar na Figura 3 que as ligações selecionadas estão distribuídas em relação as horas do dia, representando a operação diária desempenhada pelo ONS. Outra observação, relacionada com a duração das ligações, pode-se observar que a média da duração das ligações é de 49 segundos, com maior concentração nas ligações entre 20 e 40 segundos e acima de 60 segundos.

Outra característica que pode ser extraída do *dataset* baseia-se na relação de palavras-chaves. Após a oitava efetuada manualmente, foram elencadas trinta palavras-chave: máquina, horário, desligar, reduzir, geração, mw, sgi, kv, chuva, disjuntor, usina, tensão, conversor, intervenção, potência, barra, milímetro, elevar, nível, metro, manobra, reservatório, gerar, compensa, vazão, watt, montante, documento, prorrogar e vertimento juntamente com suas variações (ex. gerar, gera, geração, gerei, gerando, entre outros). A Figura 4 ilustra a frequência das palavras-chave no *dataset* (word cloud).

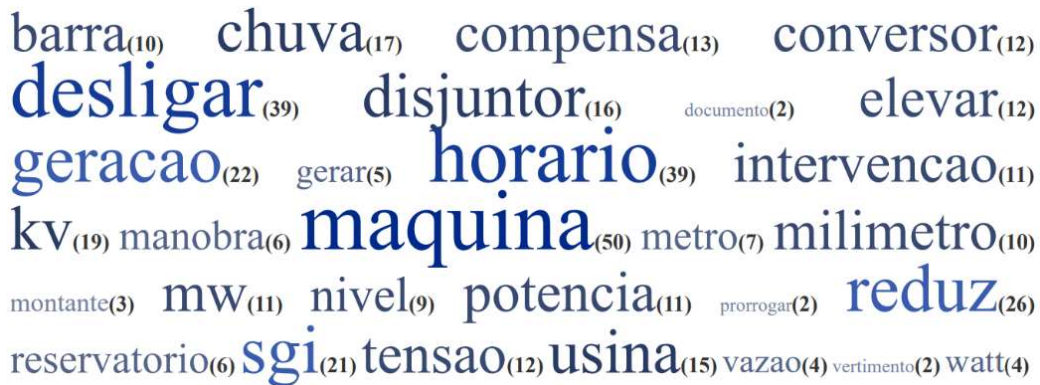


FIGURA 4 - Frequência de aparições das palavras-chave no *dataset* analisado.

Para a etapa de validação da metodologia, foram utilizadas métricas relacionadas ao Processamento de Linguagem Natural (PLN) [5] para realizar uma comparação entre as transcrições manuais efetuadas com os textos transcritos e processados automaticamente. Neste sentido, a métrica comumente adotada na literatura é o *Word Error Rate* (WER), que quantifica a taxa de erro entre dois textos, ilustrada em (1).

$$WER = \frac{S + D + I}{N} \quad (1)$$

Sendo que:

- S Quantidade de substituições;
- D Quantidade de deleções;
- I Quantidade de inserções;
- N Quantidade de palavras no texto de referência.

Outra métrica abordada no trabalho é a *Keyword Error Rate* (KER), que quantifica a taxa de erro em relação a palavras-chaves, usando a equação (2).

$$KER = \frac{F + M}{N} \quad (2)$$

Sendo que:

- F Quantidade de palavras-chave falsamente identificadas;
- M Quantidade de palavras-chave não identificadas.

Dessa forma, neste trabalho, as duas métricas serão utilizadas na avaliação do desempenho do processo proposto, sendo o WER utilizado para avaliar a qualidade geral da transcrição e o KER para o acerto de palavras específicas. Destaca-se que como são métricas que representam erro, quanto menor o valor melhor será a transcrição efetuada, representando maior similaridade entre as amostras comparadas.

3.0 - PROCESSAMENTO DOS ÁUDIOS

O objetivo principal do trabalho é aproveitar os serviços gratuitos oferecidos por empresas para o processamento dos áudios. Algumas empresas oferecem seus serviços para a realização da conversão de fala para texto, como a *Microsoft (Azure)*, *IBM (Watson)*, *Amazon Transcribe (AWS)*, *Google* e *Wit.ai*, que são empresas consolidadas e que disponibilizam seus modelos de transcrição para o mercado, com algumas limitações para o plano gratuito. A Tabela 1 exibe as principais características limitantes desses serviços [6].

TABELA 1 - Limites dos serviços de transcrição.

	Azure	Watson	AWS	Google	Wit.ai
Online/pagos	Sim	Sim	Sim	Sim	Sim
Limite/mês (min)	300	500	60	60	∞
Limite/requisição	1 min	100 MB	240 min	1 min	20 seg
Personalização	Sim	Sim	Sim	Sim	Não

Observando a Tabela 1 é possível constatar que o *Wit.ai* não apresenta limite mensal de uso, sendo uma grande vantagem entre os serviços elencados e que possibilita a realização de várias amostras de áudio. Por outro lado, o *Wit.ai* apresenta o valor mais restritivo em relação ao limite por requisição, limitando a amostra em, no máximo, 20 segundos de duração, assim como, restringe o número de requisições em 60 por minuto [7]. Entretanto, após uma análise de desempenho entre os serviços [6], foi selecionado o *Wit.ai* considerando principalmente pelo melhor desempenho e pela possibilidade de realizar as transcrições de forma completamente gratuita.

Desta forma, no processo proposto, deve-se efetuar a segmentação dos áudios baseado na atividade de voz, a fim de reduzir o tempo dos áudios (para o *Wit.ai* a duração não deve ultrapassar 20 segundos) e reter somente as informações de voz dos operadores. Para a segmentação dos áudios, foram analisadas a performance das bibliotecas em Python: *audiotk* [8], *webrtcvad* [9] e *pyAudioAnalysis* [10]. Em cada uma dessas bibliotecas foi realizada a variação de parâmetros a fim de gerar diferentes *datasets* e compará-los para avaliar a forma mais atrativa de segmentação. A Tabela 2 exibe os parâmetros (e sua descrição) e os valores estipulados.

TABELA 2 - Bibliotecas e seus parâmetros para segmentação de áudios.

Biblioteca	Parâmetro	Descrição	Valor
<i>audiotk</i>	<i>min_dur</i>	Mínima duração dos áudios	0.2
	<i>max_dur</i>	Máxima duração dos áudios	20
	<i>max_silence</i>	Máxima duração de silêncio tolerado	0.3
	<i>energy_threshold (et)</i>	Limiar de energia para detecção de atividade de áudio	30, 35, 40, 45, 50, 55
<i>webrtcvad</i>	<i>aggressiveness (ag)</i>	Agressividade da remoção do silêncio	0, 1, 2, 3
<i>pyAudioAnalysis</i>	<i>st_win</i>	Tamanho da janela de curto prazo	0.05
	<i>st_step</i>	Passo da janela de curto prazo	0.05
	<i>smooth_window (sw)</i>	Valor para suavização a sequência probabilística do algoritmo	0.3 0.5
	<i>weight (w)</i>	Fator entre 0 e 1 que especifica quão rigoroso é o limiar	0.1, 0.2, 0.3
	<i>plot</i>	Booleano associado ao plotagem dos resultados	false

Nesta análise, foram geradas dezesseis variações do *dataset* original, cada um com a variação dos parâmetros ilustrados na Tabela 2. Após o processamento dos áudios foi possível comparar as características do *dataset* original com os segmentados. A Tabela 3 exibe os dados de tempo e a quantidade de arquivos de cada variação de *dataset* criado.

TABELA 3 - Variações dos parâmetros das bibliotecas utilizadas para segmentação dos áudios.

Biblioteca	Parâmetro	Quantidade Total de Áudios	Média de Áudios por Ligação	Duração Total (segundos)	Média de Duração dos Áudios (segundos)	Quantidade de Áudios > 20 Segundos	Tempo total de requisições (segundos)
<i>original</i>	-	110	-	5.405,16	49,14	98	-
<i>audiotk</i>	<i>et: 30</i>	681	6,16	5.108,60	7,50	0	369,05
	<i>et: 35</i>	964	8,72	4.916,91	5,10	0	452,28

	<i>et: 40</i>	1.316	11,91	4.661,70	3,54	0	481,67
	<i>et: 45</i>	1.552	14,05	4.444,05	2,86	0	622,31
	<i>et: 50</i>	1.779	16,10	4.213,95	2,37	0	647,79
	<i>et: 55</i>	2.041	18,47	3.894,21	1,91	0	660,62
<i>webrtcvad</i>	<i>ag: 0</i>	663	6,00	4.842,66	7,30	41	289,64
	<i>ag: 1</i>	702	6,35	4.812,36	6,86	36	354,84
	<i>ag: 2</i>	869	7,86	4.691,40	5,40	24	376,11
	<i>ag: 3</i>	1.266	11,46	4.338,54	3,43	6	539,14
<i>pyAudioAnalysis</i>	<i>sw: 0.3 w: 0.1</i>	1.834	16,60	4.263,90	2,32	1	646,80
	<i>sw: 0.3 w: 0.2</i>	2.239	20,26	3.824,60	1,71	0	670,42
	<i>sw: 0.3 w: 0.3</i>	2.526	22,86	3.412,15	1,35	0	744,29
	<i>sw: 0.5 w: 0.1</i>	1.211	10,96	4.493,80	3,71	8	530,00
	<i>sw: 0.5 w: 0.2</i>	1.539	13,93	4.093,45	2,66	0	566,54
	<i>sw: 0.5 w: 0.3</i>	1.780	16,11	3.673,40	2,06	0	638,14

Observa-se que, em relação ao *auditok* conforme varia-se o parâmetro *energy_threshold* (*et: 30* a *et: 55*), se elevam a média de áudios por ligação, a quantidade total de áudios e tempo de requisição, no entanto, a duração total e média de duração dos áudios se reduz. Ou seja, ocorre um aumento na quantidade de áudios e uma redução no comprimento deles. Adicionalmente, pontua-se que quanto maior a quantidade de áudios maior o tempo de requisições necessário, em média leva-se 536,82 segundos (aproximadamente 9 minutos) para transcrever em média de 1485 segmentos de áudios que representam 110 ligações. Adicionalmente, buscando facilitar a visualização e a comparação dos dados representados na tabela, a Figura 5 exibe os dados normalizados de cada *dataset* de áudios criado, para cada biblioteca.

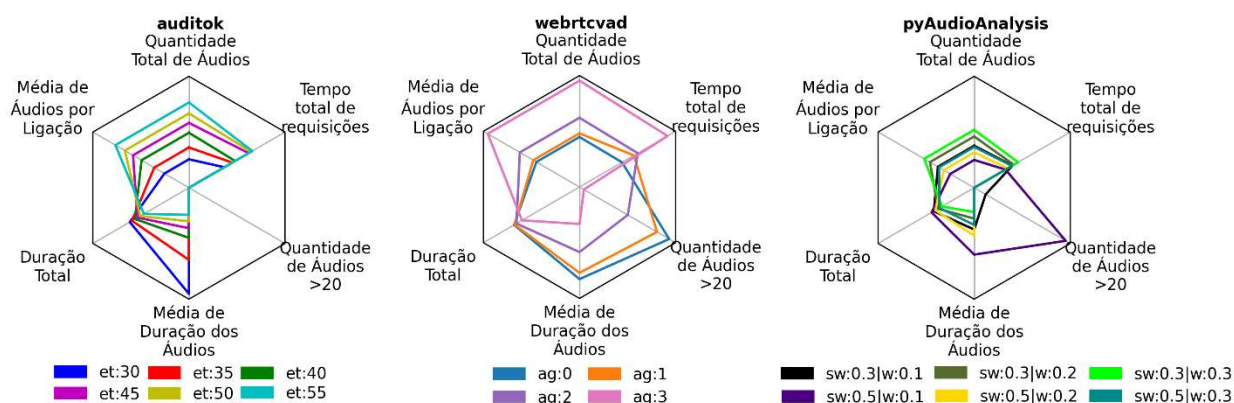


FIGURA 5 - Características dos dados com a variação dos parâmetros.

Com a variação dos parâmetros das funções, pode-se criar *datasets* com diferentes características. Pode-se observar que no *auditok* não existem áudios maiores que 20 segundos, pois o argumento *max_dur* permite limitar a duração dos áudios segmentados. Isso é um ponto positivo pois áudios com mais de 20 segundos serão ignorados pela API de transcrição, podendo existir uma perda de informação relevante em relação a essa característica. Em relação a *webrtcvad* e *pyAudioAnalysis*, observa-se que existem áudios maiores que 20 segundos por não existir a limitação de tempo como parâmetro, isso é um ponto negativo pois pode representar diretamente perda de informação.

Por outro lado, elevar o parâmetro de agressividade do *webrtcvad* resulta principalmente na redução da média de duração dos áudios e, consequentemente, na elevação da quantidade total dos áudios, apesar de que a duração total não apresentou grandes variações. Para a biblioteca *pyAudioAnalysis*, aumentar o *smooth_window* acarreta na elevação da duração média dos áudios criados e, consequentemente, na redução da quantidade de segmentos de áudios. O outro parâmetro dessa biblioteca é o peso, com a variação do mesmo é possível modificar o limiar de segmentação.

Para presente contexto, utilizando o *Wit.ai*, não existe a limitação em relação ao tempo de transcrição, o fator limitante (fora o tempo máximo de 20 segundos) seria a quantidade de arquivos, pois mais áudios representam mais requisições, logo mais tempo de processamento. Caso fosse contratado um serviço de transcrição, seria mais atrativo *datasets* com menor duração total, pois impacta diretamente no custo do serviço, nesse sentido existe a necessidade da realização de um *trade-off* entre custo e qualidade, visto que quanto menor o tempo total mais informação deve ser perdida.

4.0 - TRATAMENTO DAS TRANSCRIÇÕES

Após a transcrição dos áudios é possível realizar o tratamento do texto buscando melhorar o acerto de palavras que são transcritas equivocadamente pelo serviço utilizado, criando um dicionário com o *Python* no formato “de” “para” nesta etapa de tratamento das transcrições. Foram abordadas, neste trabalho, duas formas de realizar esse tratamento, utilizando algoritmos de aprendizado não supervisionados, realizando agrupamento (*clustering*) e o *Word Embeddings*, que se refere a uma representação de palavras por meio de vetores.

Com a *clusterização* (agrupamento) é possível identificar palavras morfologicamente similares. Por exemplo, a palavra “falha” é comumente transcrita como “palha”, essa similaridade pode ser identificada e corrigida por meio do agrupamento visto que a palavra “palha” é dificilmente dita no contexto da operação do setor elétrico. Para o estudo de agrupamento, utilizou-se o *software* de código aberto *OpenRefine*, que consegue realizar a limpeza, filtragem, transformação customizadas, agrupamento (*clustering*), *web scraping* e escrita de expressões [11]. Para a opção de *clustering*, é disponibilizado os seguintes algoritmos: *key collision* com as variações: *fingerprint*, *ngram-fingerprint*, *metaphone3*, *cologne-phonetic*, *Daitch-Mokotoff* e *Beider-Morse* ou *nearest neighbor* com as variações: *levenshtein* e *ppm* [12]. A Figura 6 (a esquerda) exibe a tela de *clustering* do *software*, com a opção da seleção do método e seus parâmetros e exibe os grupos (*clusters*) encontrados, possibilitando a combinação (*merge*) ou a exportação dos resultados no formato JSON.

Por outro lado, com *Word Embedding* é possível inferir palavras similares com base no padrão de aparição de palavras nos documentos. Por exemplo, a palavra “kv” é comumente transcrita como “caveira”, como são diferentes morfologicamente o estudo de agrupamento não identificaria esse equivoco, entretanto baseado no uso dessa palavra em sentenças com *Word Embedding* é possível identificar e corrigir esse equívoco. Novamente, a palavra “caveira” não faz parte do vocabulário da operação do setor elétrico. A abordagem de *Word Embeddings* foi efetuada utilizando a biblioteca *Python* de código aberto *Gensim*, que busca representar documentos como vetores semânticos. *Gensim* foi desenvolvido para processar documentos de texto não estruturados, usando algoritmos de aprendizado não supervisionado como *Word2Vec*, *FastText*, *Latent Semantic Indexing* (LSI, LSA, *LsiModel*), *Latent Dirichlet Allocation* (LDA, *LdaModel*), entre outros. Com base em um *corpus* (dados de texto) é possível identificar padrões nos documentos e correlacionar os dados [13]. Para o presente estudo foi utilizado o algoritmo *Word2Vec*, que apresenta duas opções de algoritmo de treino *skip-gram* ou *continuous bag of words* (CBOW). A Figura 6 (a direita) exibe a lógica desses algoritmos de treino [14].

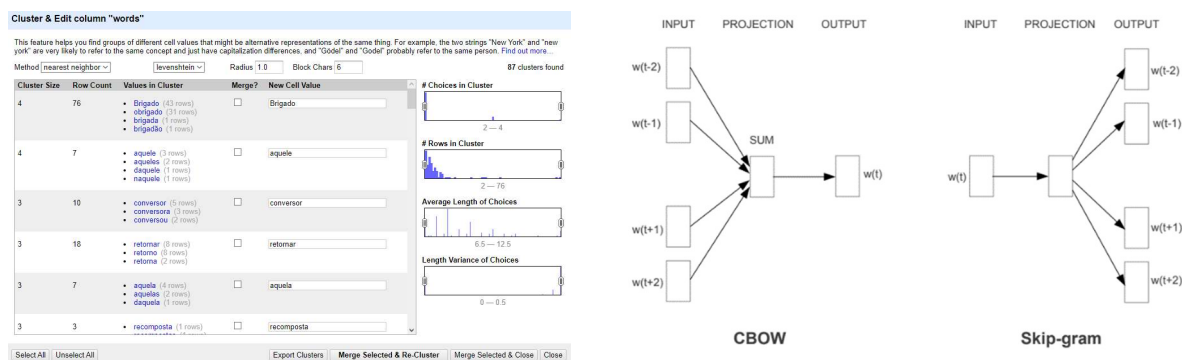


FIGURA 6 - Interface para agrupamento de palavras do *software* *OpenRefine* (esquerda) e Exemplo dos algoritmos de treino do algoritmo *Word2Vec* (direita).

O algoritmo de treino padrão do *Gensim* é o CBOW, que infere a palavra alvo pelo contexto, enquanto o *skip-gram* aprende a prever o contexto de acordo com a palavra alvo. Tem-se como resposta, um algoritmo capaz de transformar palavras em vetores, possibilitando a verificação de similaridade semântica de palavras, por exemplo, ao realizar as operações algébricas vetor("Rei") - vetor("Homem") + vetor("Mulher") resulta em um vetor próximo do vetor das palavras “Rainha” [15]. Para o desenvolvimento do estudo foi realizado o treino com ambos os algoritmos com o auxílio da função que retorna as palavras mais similares “*most_similar*”.

Finalmente, após o mapeamento dos padrões de texto, foi possível criar o dicionário com o *Python* (formato “de” “para”) para aprimorar o processo automático dos textos transcritos. Importante destacar que para realizar o *clustering* e *Word Embeddings* de forma confiável é necessário o uso de uma quantidade considerável de amostras, logo, o dicionário foi desenvolvido usando uma quantidade maior do que as 110 amostras transcritas deste trabalho.

5.0 - RESULTADOS

Após as etapas de transcrição e de processamento é possível realizar o cálculo das métricas WER e KER, buscando evidenciar a qualidade geral da transcrição e o acerto das palavras-chaves. Foram calculadas as métricas de WER e KER para a transcrição de cada um dos dezesseis *datasets* tendo como referência a transcrição manual efetuada (referência). Neste sentido, foram efetuadas duas comparações relacionadas com a transcrição sem o tratamento textual e com o tratamento textual proposto (seção anterior). A Figura 7 exibe os valores de WER e KER comparando

as métricas relacionadas ao texto transcrito (azul) e comparando com as métricas relacionadas ao texto transcrito utilizando o tratamento proposto (laranja). Pode-se observar também na figura a redução percentual obtida.

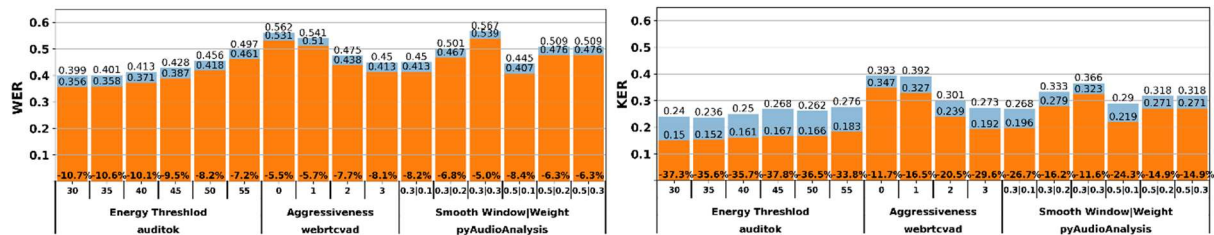


FIGURA 7 - Valores de WER, KER e a redução percentual sem e com o processamento proposto para o texto transcrito.

Pode-se observar que o melhor tipo de segmentação foi com a utilização da biblioteca *auditok* e com o parâmetro *energy_threshold* igual a 30 para a métrica WER (0,399) e para a métrica KER o *energy_threshold* que atingiu o menor valor foi igual a 35 (0,236), considerando as transcrições sem processamento. Nota-se que os valores de 30 e 35 são similares, logo supõem-se que a partir do *threshold* igual a 35 não existe grande ganho de informação.

De forma geral, pode-se afirmar que o processo de tratamento de texto para as transcrições realizadas com *Wit.ai* são vantajosos para qualquer *dataset* criado pois sempre existe uma redução dos valores. Para o WER a maior redução foi de 10,7% (0,399 para 0,356, com *et*: 30) e para o KER a maior redução foi de 37,8% (0,268 para 0,167, com *et*: 45), entretanto o *energy_threshold* igual a 30 apresentou os melhores valores finais.

Segundo [16], usando o modelo da *Microsoft* áudios com a taxa de amostragem baixa (8 kHz) atingem no máximo 30% de WER. Em [17], é realizada uma comparação entre IBM, *Google* e *Wit.ai* e constata-se que o IBM atingiu 26,67%, o *Google*, 43,33% e o *Wit.ai*, 36,67% de WER. Nesse sentido, comparando os valores das referências com o valor de WER obtido, conclui-se que foi possível atingir valores de WER próximos ou melhores que as soluções comerciais apresentadas.

6.0 - CONCLUSÕES

Considera-se que o objetivo de obter uma transcrição automática dos áudios e que consiga compilar as principais informações relacionadas à operação foi atingido com sucesso pela metodologia proposta (por meio da segmentação, transcrição e tratamento do texto). Após passar pelas etapas de tratamento de áudio e de processamento de texto propostas neste trabalho, foi possível reduzir consideravelmente os erros de transcrição geral (WER) e de palavras-chaves (KER), tendo como resultado uma transcrição que proporciona um aproveitamento próximo das oitivas realizadas na pós-operação.

Comparando os valores de WER das soluções disponíveis no mercado [16] e das referências [17], o valor final de 35,6% atingido pela metodologia apresentada é atrativo e viável. Adicionalmente, a métrica WER que avalia o acerto de palavras de maneira geral, foi proposta a métrica KER capaz de quantificar o acerto de palavras consideradas mais relevantes para o contexto estudado e conclui-se que o valor de 15% de erro é promissor e tem potencial para viabilizar soluções relacionadas a inteligência operacional do setor elétrico.

Segundo a metodologia proposta, avaliando três bibliotecas capazes de segmentar áudio (*webrtcvad*, *auditok* e *pyAudioAnalysis*), a melhor abordagem encontrada foi com o *auditok* com o valor de 30 para o parâmetro *energy_threshold*, atingindo os valores finais de 0,356 para a métrica WER e 0,15 para a métrica KER. Para o consumo da API do *Wit.ai* é recomendado o uso da biblioteca *auditok* com os parâmetros *min_dur* = 0.2, *max_dur* = 20, *max_silence* = 0.3 e *energy_threshold* = 30, pois como o *Wit.ai* não cobra pelo uso da API, pode-se explorar a metodologia com maior valor de duração total possível, visto que quanto menor a duração total mais informação é perdida. Como recomendação futura, caso fosse realizada contratação de um serviço (como *Azure*, *IBM*, *Google* e *Amazon*), seria necessário avaliar qual valor de *energy_threshold* usar levando em consideração o quanto de capital financeiro o contratante se dispõe a economizar em troca da perda de informação atrelada a remoção de silêncio.

Por fim, a metodologia proposta possibilita a transcrição automática, gratuita e com qualidade para a comunicação verbal dos operadores e acredita-se ter grande potencial para otimizar as atividades do setor de pós-operação, facilitando suas atividades e possibilitando desenvolver análises quantitativas e qualitativas a respeito da operação e das ações tomadas pelos operadores. Destaca-se que, utilizando a metodologia proposta, os autores desenvolveram um protótipo *web*, denominado de Sistema para Classificação da Comunicação Verbal na Operação do Setor Elétrico¹, que possibilita a utilização dos usuários de forma gratuita.

8.0 - REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ONS. Operador Nacional do Sistema Elétrico. 2021. Disponível em: <http://www.ons.org.br>. Acesso em: 12 jul. 2021.
- [2] ONS. Submódulo 5.13 - Comunicação Verbal na Operação. Operador Nacional do Sistema Elétrico, 2020.

¹<https://gese.florianopolis.ifsc.edu.br/classificador-operacao/>

- [3] PYTHON SOFTWARE FOUNDATION. Python 3.9.2. 2021. Disponível em: <https://docs.python.org/3/>. Acesso em: 12 jul. 2021.
- [4] PYPI. The Python Package Index. 2021. Disponível em: <https://pypi.org>. Acesso em: 12 jul. 2021.
- [5] PARK, Y., PATWARDHAN, S., VISWESWARIAH, K., GATES, S. C. An empirical analysis of word error rate and keyword error rate. INTERSPEECH. ISCA. p. 2070-2073. 2008.
- [6] LIMA, M. N. S. M., COELHO, B. H., TAKIGAWA, F. Y. K. Ferramentas e recursos disponíveis para reconhecimento de fala em Português Brasileiro. Anais do Computer on the Beach, 12, 475-479. 2021.
- [7] Wit.ai. Retrieve the meaning of an audio wave, 2021. Disponível em: https://wit.ai/docs/http/20200513#post__speech_link. Acesso em: 01 set. 2021.
- [8] SEHILI A. auditok, an AUDIo TOKenization tool. 2021. Disponível em: <https://auditok.readthedocs.io/en/latest/>. Acesso em: 01 set. 2021.
- [9] WEBRTCvad. Python interface to the WebRTC Voice Activity Detector. 2021. Disponível em: <https://github.com/wiseman/py-webrtcvad>. Acesso em: 01 set. 2021.
- [10] GIANNAKOPOULOS, T. pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis. PLoS ONE 10(12): e0144610. 2015.
- [11] OPEN REFINE. Open Refine. 2021. Disponível em: <https://openrefine.org/>. Acesso em: 14 jul. 2021.
- [12] OPEN REFINE. Open Refine: Clustering Methods. 2021. Disponível em: <https://docs.openrefine.org/manual/cellediting#clustering-methods>. Acesso em: 16 jul. 2021.
- [13] GENSIN. GENSIN. 2021. Disponível em: <https://radimrehurek.com/gensim/intro.html>. Acesso em: 14 jul. 2021.
- [14] MIKOLOV, T. et al. Efficient Estimation of Word Representations in Vector Space, 2013. Disponível em: <http://arxiv.org/abs/1301.3781>. Acesso em: 16 jul. 2021.
- [15] MIKOLOV T., CHEN K., CORRADO G., DEAN J. Efficient Estimation of Word Representations in Vector Space. 2013. Disponível em: <http://arxiv.org/abs/1301.3781>. Acesso em: 01 set. 2021.
- [16] MICROSOFT. Avaliar e aprimorar a precisão da Fala Personalizada. 2021. Disponível em: <https://docs.microsoft.com/pt-br/azure/cognitive-services/speech-service/how-to-custom-speech-evaluate-data>. Acesso em: 09 set. 2021.
- [17] FILIPPIDOU F, MOUSSIADES L. A Benchmarking of IBM, Google and Wit Automatic Speech Recognition Systems. Artificial Intelligence Applications and Innovations. 2020;583:73-82. 2020. DOI: 10.1007/978-3-030-49161-1_7.

DADOS BIOGRÁFICOS



Formação Técnica em Programação de Jogos Digitais (2014 – 2015), pelo Grupo Educacional Opet. Estudante de Engenharia Elétrica (2016-2021), no Instituto Federal de Santa Catarina (IFSC). Pesquisador do Grupo em Estudos de Sistemas de Energia (GESE) (2016-atual) e estagiou no setor de pós-operação no Operador Nacional do Sistema Elétrico (ONS) (2019-2020). Formado em Desenvolvimento Mobile Android (2020 - 2021) pela Digital House. Atualmente atua como Assistente de Projetos P&D na ENGIE Brasil (2021-atual).

(2) BRUNA HAMES COELHO

Formada no curso técnico de Refrigeração e Climatização no Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina (IFSC). Estudante de Engenharia Elétrica no Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina (IFSC). Bolsista de iniciação científica no Grupo de Estudos de Sistemas de Energia (GESE). Estagiária na empresa Intelbras atuando no setor de Engenharia de Qualificação de Fornecedores (EQF).

(3) FABRICIO YUTAKA KUWABATA TAKIGAWA

Possui graduação em Engenharia Elétrica pela Universidade Estadual Paulista Júlio de Mesquita Filho (2003), mestrado e doutorado em Engenharia Elétrica pela Universidade Federal de Santa Catarina (2006 e 2010). Atualmente é professor do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina. Tem experiência na área de sistemas de energia elétrica, com ênfase em sistemas hidrotérmicos, planejamento da operação energética e otimização matemática aplicada.