AVALIAÇÃO DO IMPACTO DAS EXPRESSÕES MULTIPALAVRAS NOS MODELOS DE LINGUAGEM

Fábio Corrêa Cordeiro
Escola de Matemática Aplicada
Fundação Getúlio Vargas

Eduardo Fonseca Mendes
Escola de Matemática Aplicada
Fundação Getúlio Vargas

Thiago Ramos
Instituto de Matemática Pura e Aplicada

1 de setembro de 2020

RESUMO

Modelos de linguagem servem de base para uma série de tarefas de processamento de linguagem natural, entre elas a geração automática de textos. Em muitos casos as palavras são usadas como tokens, ou seja, como unidades básicas para o processamento do texto. Para documentos de domínios específicos, onde existem muitas expressões e jargões próprios, as expressões multipalavras (ou *multiword expressions*) podem impactar no processamento dos texto. Buscando subsidiar a construção de modelos de linguagem, neste trabalho avaliamos o impacto das expressões multipalavras nacriação destes modelos. Criamos dois modelos de linguagem estatísticos usando Cadeias de Markov, o primeiro utilizou apenas palavras únicas como tokens, já o segundo modelo juntou as expressões multipalavras em um *token* único. As expressões multipalavras do segundo modelo foram identificadas usando a métrica Informação Mútua (*Mutual Information*). Por fim, comparamos os dois modelos avaliando as suas perplexidades.

Após os nossos testes, encontramos que o modelo que utilizou apenas palavras únicas teve o melhor resultado. Portanto, considerar as expressões multipalavras não melhorou os modelos de linguagem de domínio específico. Esse trabalho foi limitado a modelos de linguagem estocástico, portanto é necessário avaliar se os mesmos resultados se mantém para modelos baseados em redes neurais. Também é importante avaliar o comportamento dos modelos de linguagem ao usar subpalavras - tokenização de elemento com tamanho inferior ao tamanho das palavras.

***P*alavras-chave** Modelos de Linguagem · Expressões Multipalavras · Cadeias de Markov · Informação Mútua · Perplexidade

1 Introdução

Os modelos de linguagens são modelos estatísticos que, ao observar uma sequência de palavras ou caracteres, atribuem uma distribuição de probabilidade para as próximas palavras ou caracteres [1]. Eles são essenciais para diversas tarefas de processamento de linguagem natural. Alguns exemplos dessas tarefas são classificação de textos, tradução automática, reconhecimento de discurso, análise de sentimentos, entre outros.

Para que o texto seja processado por um modelo matemático, é necessário segmentá-lo em unidades básicas, os *tokens*. Há diversas estratégias para a segmentação do texto, mas o mais comum é utilizar as palavras como *tokens*. O princípio por trás de se usar as palavras como unidade básica é que cada palavra representaria um conceito. No entanto, algumas expressões compostas por múltiplas palavras (*Multiwords Expressions*) podem encapsular um único conceito diferente dos conceitos atribuídos às palavras que a compõem. Por exemplo, a palavras "rio"representa um acidente geográfico, bem como "janeiro"representa um mês do ano; no entanto a expressão "Rio de Janeiro"remete a uma cidade e não está relacionado aos conceitos de "rio"e "janeiro". Essas expressões são comuns em todas as línguas, mas são mais impactantes em documentos técnicos e de domínio específico, pois estes costumam usar jargões e expressões pouco conhecidas fora do seu contexto.

Neste trabalho avaliamos o impacto das expressões multipalavras nos modelos de linguagem. Para identificar as expressões multipalavras utilizamos a métrica de Informação Mútua (*Mutual Information*) para comparar as probabilidades

de ocorrência de duas palavras. Em seguida, criamos dois modelos estatísticos de linguagem baseados em Cadeia de Markov, um utilizando apenas palavras simples como *token* e outro utilizando as expressões multipalavras. Por fim, comparamos os dois modelos avaliando a sua perplexidade, uma medida comumente usada para verificar se a distribuição de probabilidade prevista pelos modelos de linguagem está ajustada a uma amostra de teste.

Além desta introdução, neste trabalho apresenta uma breve fundamentação dos métodos utilizados. Em seguida, descrevemos o corpus utilizado bem como a metodologia para a criação e avaliação dos modelos. Por fim, os resultados seguidos por uma breve conclusão.

2 Fundamentação

Nesta seção, apresentamos os conceitos e fundamentos teóricos utilizados para a construção e avaliação dos modelos de linguagem. Inicialmente, introduzimos os modelos de linguagem, detalhamos as Cadeias de Markov e como esses modelos são utilizados para o processamento de linguagem natural. Em seguida, falamos sobre tokenização dos textos e identificação das expressões multipalavras. Finalmente, discutimos os conceitos de entropia, Informação Mútua e perplexidade, e suas aplicações na identificação das expressões multipalavras e na avaliação de distribuição de probabilidades.

2.1 Modelos de Linguagem e Cadeias de Markov

Um bom exemplo para ilustrar o que são os modelos de linguagem é um jogo de adivinhação proposto por Claude Shannon. Neste jogo, ele fornecia um texto em inglês e solicitava ao jogador que adivinhasse a próxima letra. Após um número de jogadas suficientemente grande, seria possível calcular a distribuição empírica das frequências das letras. É de se esperar que no início das sentenças, quando há pouco texto prévio para se basear, o jogador gastasse mais tentativas até acertar a letra correta. Por outro lado, quanto mais texto prévio disponível mais assertivo ficavam as respostas [2]. De uma maneira empírica, ao olhar para uma sequência de letras, o jogador formava uma hipótese de qual era a distribuição de probabilidade da próxima letra.

De uma maneira mais formal, podemos dizer que um modelo estatístico de linguagem é uma distribuição de probabilidades para todas as palavras de vocabulário que é atribuído após se observar uma sequência de caracteres ou palavras [1]. Seja ν um vocabulário finito e ν^* um conjunto de caracteres ou palavras definido em um subconjunto de ν. A função de distribuição de probabilidade p dessa linguagem é tal que:

$$\sum_{x \in \nu^*} p(x) = 1, \text{ e } p(x) > 0 \text{ para todo } x \in \nu^*$$

Portanto, para se construir os modelos de linguagem, é necessário extrair a função de distribuição de probabilidades de um conjunto de textos que seja representativo da linguagem que se deseja modelar. Estes conjuntos de textos são conhecidos como corpora (ou no singular corpus). Consideremos que $x_1x_2...x_n$ é uma sentença de tamanho n, e que x_1 é a primeira palavra, x_2 a segunda palavra, e assim por diante. Podemos considerar a construção de uma sentença como um processo estocástico onde $X_1, X_2, ...X_n$ são variáveis aleatórias. O nosso modelo terá que encontrar $p(X_1 = x_1, X_2 = x_2, ..., X_n = x_n)$, que pela regra da cadeia

$$p(X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = \prod_i p(x_i|x_1x_2...x_{i-1})$$

Como as sentenças do corpus de treinamento não são exatamente as sentenças que serão inferidas pelo modelo, é improvável que a sequência de palavras que se deseja inferir tenha aparecido no treinamento. Como não conseguimos calcular $p(x_1, x_2, ...x_n)$ para sentenças relativamente longas a suposição de Markov se mostra útil, no qual a probabilidade de uma palavra depende apenas de algumas poucas palavras anteriores. Ou seja,

$$p(x_i|x_1x_2...x_{i-1}) \approx p(x_i|x_{(i-k)}x_{(i-k+1)}x_{(i-k+2)}...x_{(i-1)}), \text{ para } 1 < k < i$$

Quando escolhemos o $k = 0$, o modelo não depende do contexto, ou seja, a probabilidade de ocorrência da próxima palavra é a probabilidade dela ocorrer em qualquer texto dessa linguagem. Esse é o modelo *unigram*. Os modelos *bigram*, *trigram* e *tetragram* consideram respectivamente uma, duas ou três palavras anteriores para calcular a probabilidade da palavra seguinte, ou seja usam $k = 1$, $k = 2$ e $k = 3$.

$$\text{Modelo Unigram } p(x_i|x_1x_2...x_n) \approx \prod_i p(x_i)$$

$$\text{Modelo Bigram } p(x_i|x_1x_2...x_n) \approx \prod_i p(x_i|x_{i-1})$$

$$\text{Modelo Trigram } p(x_i|x_1x_2...x_n) \approx \prod_i p(x_i|x_{i-1}x_{i-2})$$

$$\text{Modelo Tretragram } p(x_i|x_1x_2...x_n) \approx \prod_i p(x_i|x_{i-1}x_{i-2}x_{i-3})$$

Portanto, para se montar uma Cadeia de Markov que represente a linguagem que se deseja modelar é necessário percorrer o corpus de teste contando os pares de palavra (ou trio no caso de trigram). Desta forma, é possivel calcular $p(x_i|x_{i-1})$ (ou $p(x_i|x_{i-1}x_{i-2})$).

Os modelos de linguagem são utilizados para muitas tarefas de processamento de linguagem natural. Mais diretamente são aplicados para a geração automática de textos ou completar sentenças. Também são utilizados em conjunto com outras técnicas para classificação de textos, tradução automática, identificação de entidades nomeadas, entre outras tarefas. Por muito tempo, essas tarefas eram realizadas por modelos puramente estatísticos, mas na última década os modelos baseados em redes neurais ganharam espaço.

2.2 Tokenização do texto e expressões multipalavras

A primeira atividade necessária antes de se criar um modelo de linguagem é preprocessar os textos que compõem o corpus. Esse preprocessamento pode ser composto por diversas subtarefas, e uma das mais importantes é a segmentação do texto em unidades básicas, os *tokens*. Essas unidades podem ser letras, fonemas, subpalavras, palavras ou expressões. Para algumas tarefas pode ser interessante segmentar o texto em unidades maiores como sentenças, parágrafos e até documentos. Essas unidades básicas serão codificadas e incorporadas ao modelo. Por exemplo, a expressão em português "Expressão Multipalavra"é escrita usando duas palavras, em alemão ("*Mehrwortausdruck*") apenas uma e em chinês "多字表达" quatro ideogramas. A decisão de como tokenizar o corpus depende da língua e dos objetivos do modelo de linguagem.

Para muitas tarefas é importante que a tokenização separe conceitos diferentes em *tokens* diferentes. Logo, a escolha das palavras como unidade básica parece natural, e na prática é largamente utilizado. No entanto, vários conceitos podem ser expresso por múltiplas palavras, como no exemplo anteriormente apresentado do nome da cidade "Rio de Janeiro".

A identificação das expressões multipalavras pode ser feita utilizando três abordagens [3]. A primeira abordagem usa medidas de associação estatísticas, sendo a mais comum a Informação Mútua (*Mutual Information*). Essa estratégia tem a vantagem de ser fácil de implementar, uma vez que somente é necessário conhecer as frequências das palavras no corpus. Outra opção é a substituição e inserção de sinônimos, ou seja, é necessário possuir uma lista de sinônimos e comparar a frequência dos pares de palavras originais e os pares de palavras substituídas pelos sinônimos. Por fim, existe as abordagem utilizando aprendizado de máquina supervisionado. Neste trabalho, utilizamos as a a métrica de Informação Mútua para a identificar as expressões multipalavras.

2.3 Entropia, Informação Mútua e Perplexidade

No desenvolvimento da Teoria da Informação, Shannon trabalhou com modelos de linguagem de textos em inglês [4] [5]. Ele criou alguns modelos estatísticos de linguagem baseados em Cadeias de Markov e usou esses modelos para calcular a entropia da língua inglesa. Segue abaixo alguns exemplos apresentados em seu artigo original de textos gerados automaticamente:

- Aproximação de zero-ordem (caracteres independentes com a mesma probabilidade).
  XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZLHJQD.
- Aproximação de primeira ordem (caracteres independentes mas com a frequência em que aparecem na língua inglesa).
  OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL
- Aproximação de segunda ordem (a probabilidade dos caracteres depende do caractere imediatamente anterior).
  ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

- Aproximação de primeira ordem das palavras (as palavras são escolhidas independentemente mas respeitando a frequência em que aparecem na língua inglesa).
  REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.
- Aproximação de segunda ordem das palavras (a transição entre as palavras respeitam as frequências da língua inglesa).
  THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

Antes de calcular a Informação Mútua e a perplexidade precisamos definir o que é entropia [2]. Dizemos que a entropia $H(X)$ é a medida de incerteza de uma variável aleatória X, e é definida como:

$$H(X) = -\sum_{x \in X} p(x) \log p(x)$$

Se temos X com distribuição $p(x)$, e $E_p g(X)$ a esperança de uma variável aleatória $g(X)$, temos que:

$$E_p g(X) = \sum_{x \in X} g(x) p(x)$$

Logo, substituindo $g(x)$ por $\log p(X)$, podemos dizer que

$$H(X) = -E_p \log p(X)$$
$$H(X) = E_p \log \frac{1}{p(X)}$$

Também podemos estender o conceito de entropia para um par de variáveis aleatórias (X, Y). A entropia conjunta das duas variáveis $H(X, Y)$ é uma extensão direta da definição:

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y)$$

ou

$$H(X, Y) = -E \log p(X, Y)$$

Finalmente, quando queremos comparar duas distribuições podemos usar o conceito de entropia relativa. Dizemos que a entropia relativa $D(p||q)$ é uma medida de ineficiência quando assumimos que uma distribuição é q (quando essa distribuição é gerada por um modelos, por exemplo) quando de fato a distribuição é p. A entropia relativa - também conhecida como distância de Kullback-Leibler - é definida para duas distribuições de probabilidade $p(x)$ e $q(x)$, tal que

$$D(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$
$$D(p||q) = E_p \log \frac{p(x)}{q(x)}$$

Dado duas variáveis aleatórias X e Y, a Informação Mútua $I(X; Y)$ é o caso especial de entropia relativa quando comparamos as probabilidade conjuntas $p(x, y)$ e o produto das distribuições $p(x)p(y)$.

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$
$$I(X; Y) = D(p(x, y)||p(x)p(y))$$

$$I(X;Y) = E_p \log \frac{p(x,y)}{p(x)p(y)}$$

A Informação Mútua é uma medida bastante usada para identificar expressões multipalavras [3]. Comparamos a probabilidade de duas palavras aparecerem juntas no texto ($p(x_1, x_2)$) com a probabilidade de ocorrerem independentemente, ou seja, ao acaso ($p(x_1)p(x_2)$). Portanto, se duas palavras são completamente independentes, então

$$p(x_1, x_2) = p(x_1)p(x_2) \text{ e } I(X_1, X_2) = 0$$

Por outro lado, se duas palavras sempre ocorrem juntas, ou seja, poderíamos tratá-las como uma palavra única, então $p(x_1, x_2) = p(x_1) = p(x_2)$, e

$$I(X_1; X_2) = E_p \log \frac{p(x_1, x_2)}{p(x_1)p(x_2)}$$

$$I(X_1; X_2) = E_p \log \frac{1}{p(x_1, x_2)}$$

Logo, quanto maior a Informação Mútua, maior a chance de um par de palavras ser uma expressão multipalavras.

Após ter definido o conceito de entropia e de conseguir usar a medida de Informação Mútua para identificar as expressões multipalavras, queremos avaliar o quão bem uma distribuição prevista por um modelo se ajusta à distribuição real. Primeiramente, consideremos a linguagem ν^* no qual as suas palavras se distribuem de acordo com a distribuição $X \sim p(x)$. Após treinado, o nosso modelo prevê uma distribuição $m(x)$ e queremos compará-lo com a distribuição real usando um conjunto de teste de tamnho n. A entropia cruzada $H(p, m)$ é definida como [6]:

$$H(p, m) = -\lim_{n\to\infty} \frac{1}{n} \sum_X p(x) \log m(x)$$

Em situações em que não temos o valor real da distribuição $p(x)$, como é o nosso caso, assumimos uma versão simplificada da entropia cruzada. O modelo é treinado usando um conjunto de dados e a entropia é calculado em um conjunto de teste que não entrou no treinamento do modelo. A versão simplificada da entropia cruzada $H'(x)$ é dada por:

$$H'(p, m) = -\lim_{n\to\infty} \frac{1}{n} \sum_X \log m(x)$$

E para um conjunto de teste suficientemente grande temos que:

$$H'(p, m) \approx -\frac{1}{n} \sum_X \log m(x)$$

Finalmente, a perplexidade é a dada simplesmente por

$$\text{perplexidade}(p, m) = 2^{H(p,m)}$$

Manning e Schütze [6] fazem uma piada sobre a diferença entre a perplexidade e entropia cruzada. Eles dizem que a grande diferença entre as duas medidas é que a perplexidade permitiria aos pesquisadores causarem uma melhor impressão aos seus financiadores. Para eles seria mais impactante apresentar uma redução de perplexidade de 950 para 540, do que uma redução de entropia de 9,9 para 9,1 bits.

Apesar da brincadeira de Manning e Schütze a perplexidade continua sendo uma medida largamente usada para avaliar modelos de linguagem. Uma recente pesquisa publicada pela equipe de pesquisas do Google [7], apresentou uma forte correlação entre a perplexidade e avaliações de modelos de linguagem realizadas por voluntários. Os voluntários receberam textos gerados por *chatbot* e avaliaram o quão parecido eram com respostas humanas. Foram avaliados se os textos eram coerentes e se eram razoavelmente específicos - respostas vagas eram penalizadas. Ao final da pesquisa, os pesquisadores encontraram uma correlação $R^2 > 0,9$ entre as avaliações dos voluntários e a perplexidade, reforçando a ideia de usar a perplexidade como uma métrica automática para avaliação de modelos e linguagem.

3 Recursos e Métodos

Nesta seção, apresentamos o corpus utilizado e as etapas realizadas para a criação e avaliação dos modelos de linguagem. Utilizamos o Petrolês, um corpus público com textos em português no domínio da indústria de óleo e gás. Calculamos a informação mútua dos pares de palavras contidas nesse corpus para identificar as expressões multipalavras. Por fim, geramos e avaliamos os modelos de linguagem.

O Petrolês[1] é um corpus formado por teses e dissertações nas áreas de interesse da indústria do petróleo [8]. Os documentos que compõem o Petrolês estão disponíveis na Biblioteca Digital de Teses e Dissertações (BDTD)[2], um acervo do Instituto Brasileiro de Informação de Ciência e Tecnologia que reuni os trabalhos de conclusão de mestrado e doutorado das principais universidades brasileiras. Os textos das teses foram extraídos dos seus arquivos originais e unificados em um corpus único que serve de referência para os trabalhos de processamento de linguagem natural para o setor de óleo e gás. Neste trabalho, utilizamos um subconjunto do Petrolês composto por 154 documentos (teses ou dissertações) de 14 universidades diferentes (Tabela 1). Destes, 140 documentos foram utilizados para o treinamento dos modelos e 14 para o teste da perplexidade.

Tabela 1: Corpus Petrolês

	Número de documentos	Número de Palavras
Treino	140	3.681.623
Teste	14	293.401

Realizamos um preprocessamento simples do texto, no qual separamos o texto em sentenças, colocamos todas as letras em minúsculas e retiramos os espaços no início e final das sentenças. Também excluímos sentenças pequenas, com menos de 100 caracteres. Do conjunto de treino, contabilizamos a frequência de palavras e bigramas para então calcular a Informação Mútua. Com isso, escolhemos um limiar para a Informação Mútua e consideramos que todo bigram acima desse limiar seria uma expressão multipalavra. Fizemos o experimento com três limiares para a Informação Mútua: 8, 10 e 12.

Treinamos os modelos de linguagem usando cadeias de Markov através da biblioteca markovify[3]. Testamos modelos com dois e três estados, ou seja, modelos *trigram* e *tetragram*. Por fim, calculamos a perplexidade nas sentenças dos 14 documentos do conjunto de teste (Figura 1). Os experimentos estão disponíveis no repositório deste trabalho[4].

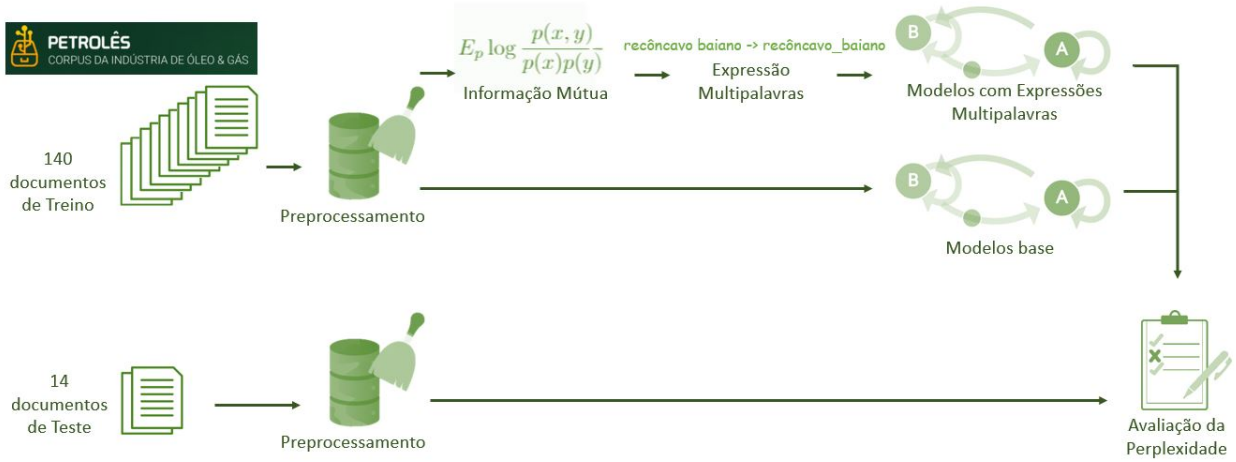

Figura 1: Etapas do experimento: separação do conjunto de treino e teste; preprocessamento; cálculo da informação mútua; unificação da expressões multipalavras; treinamento dos modelos de linguagem; e avaliação da perplexidade.

[1] petroles.ica.ele.puc-rio.br
[2] bdtd.ibict.br
[3] `https://github.com/jsvine/markovify`
[4] `https://github.com/fabiocorreacordeiro/MWE_MarkovChain`

4 Resultados e Discussões

Como esperado, os valores da perplexidade dos modelos tetragram são menores do que os trigram. Ou seja, ao usar uma probabilidade condicional com uma cadeia mais longa de palavras, os modelos tetragram modelaram melhor a linguagem. No entanto, utilizar a Informação Mútua para identificar as expressões multipalavras e uni-las em um único *token* não melhorou as avaliações dos modelos de linguagem.

Tabela 2: Avaliação da perplexidade. Os casos base consideraram apenas palavras únicas, os demais usaram os bigramas com Informação Mútua (IM) acima de um limiar.

	Caso Base	IM>12	IM>10	IM>8
Modelo Trigram	16,5496	17,9663	17,9683	17,9085
Modelo Tetragram	4,5635	5,3498	5,3501	5,3459

Neste trabalho nos limitamos a avaliar modelos de linguagem estocásticos. Portanto, uma continuação natural é ampliar a avaliação para os modelos de linguagem baseados em redes neurais e verificar se os resultados se mantém. Outra abordagem que pode ser testada, é a avaliação dos modelos de linguagem quando a tokenização é feita no domínio das subpalavras, ou seja, na direção inversa da que tratamos aqui. Por fim, para a avaliação das expressões multipalavras consideramos apenas as expressões encontradas automaticamente usando a métrica de Informação Mútua. É possível que encontremos melhores resultados se essas expressões forem curadas por especialista e enriquecidas por termos extraídos de glossários, tesauros e outras fontes de termos especializados.

5 Conclusão

Neste trabalho, identificamos as expressões multipalavras do corpus Petrolês, um conjunto de textos em português, do domínio específico da indústria do petróleo composto por teses e dissertações. Geramos dois modelos de linguagem baseados em Cadeias de Markov com o objetivo de avaliar a influência das expressões multipalavras. Ao avaliar a perplexidade dos dois tipos de modelos, encontramos que a união das expressões multipalavras em um único *token* não melhorou as avaliações dos modelos. Portanto, considerar as expressões multipalavras não ajudou na geração de modelos de linguagem de domínio específico. Para reforçar o que encontramos, em trabalhos futuros pretendemos ampliar as avaliações para abranger modelos baseados em redes neurais, avaliar o comportamento dos modelos de linguagem usando uma tokenização no nível de subpalavras e identificar as expressões multipalavras através da curadoria de especialistas e fontes de referência.

6 Anexo - Exemplo de expressões multipalavras indetificadas

IM>12	IM>10	IM>18
boosted-discharge hollow	curava tracejada	carbonílicos alifáticos
education program/adult	etileno diamina	espontânea persiste
engine manufactures	fischer tropsch	exame espectroscópico
espectroscópico transverso	flame furnace	fluem livremente
fundente lecocel	furnace atomic	fluorescente rotacionar
heterocíclico tiofênico	iron chip	horizontalmente treze
psa millennium	pressure swing	radiante transmitida
quarteto mortal	pudéssemos dispor	revestimento termorrígido
relaxações rotovibracionais	recôncavo baiano	revestimentos antiabrasivos
swing adsorption	sofram revoluções	sub-níveis vibracionais

Referências

[1] Venkat N. Gudivada. Natural Language Core Tasks and Applications. In *Handbook of Statistics*, volume 38, pages 403–428. Elsevier, 2018.

[2] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 1 edition, September 2005.

[3] Mathieu Constant, Gülşen Eryiğit, Johanna Monti, Lonneke van der Plas, Carlos Ramisch, Michael Rosner, and Amalia Todirascu. Survey: Multiword expression processing: A Survey. *Computational Linguistics*, 43(4):837–892, December 2017.

[4] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, July 1948.

[5] C. E. Shannon. Prediction and Entropy of Printed English. *Bell System Technical Journal*, 30(1):50–64, January 1951.

[6] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, Mass, 1999.

[7] Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. Towards a Human-like Open-Domain Chatbot. *arXiv:2001.09977 [cs, stat]*, February 2020. arXiv: 2001.09977.

[8] Fábio Corrêa Cordeiro. *Petrolês - Como Construir um Corpus Especializado em Óleo e Gás em Português*. Monografia, PUC-Rio, Rio de Janeiro, 2020.