FREITAS NETO, A. M.; LAZZARINI-CYRINO, J. P. A segmentação morfológica não-supervisionada como ferramenta para documentação linguística: comparando duas abordagens. *ReVEL,* edição especial n. 21, 2024. [www.revel.inf.br].

A SEGMENTAÇÃO MORFOLÓGICA NÃO-SUPERVISIONADA COMO FERRAMENTA PARA DOCUMENTAÇÃO LINGUÍSTICA: COMPARANDO DUAS ABORDAGENS

Unsupervised Morphological Segmentation as a tool for Language Documentation: a comparison between two approaches

Antonio Morais de Freitas Neto[1]
João Paulo Lazzarini Cyrino[2]

tonemfn@gmail.com
jpcyrino@gmail.com

RESUMO: Recentemente temos visto grande desenvolvimento nas áreas de processamento de linguagem natural. Tal desenvolvimento, no entanto, ainda precisa atingir os mais de dois terços das línguas do mundo que ainda não foram devidamente documentados. Neste artigo, apresentamos duas abordagens de segmentação morfológica não-supervisionada, que, acreditamos, podem vir a ser de ajuda para trabalhos de documentação linguística: uma baseada em *Minimum Description Length* (Rissanen, 1978; DeMarcken, 1995) e outra em *Byte Pair Encoding* (Gage, 1994; Sennrich et al. 2016). Comparamos o quanto cada abordagem consegue analisar morfologicamente as palavras de uma lista de palavras de forma idêntica à segmentação feita por um linguista, sem necessitar de nenhuma informação prévia para além das próprias palavras.
PALAVRAS-CHAVE: aprendizado de máquina; morfologia; documentação linguística.

ABSTRACT: A strong development in the fields of natural language processing has been taking place. Such a development, nevertheless, still needs to reach the more than two thirds of the world's languages that remain undocumented. This article introduces two approaches of unsupervised morphological segmentation, which we believe could be helpful to the field of language documentation: one is based in the Minimum Description Length optimization method (Rissanen, 1978; DeMarcken, 1995) and another in the Byte Pair Encoding algorithm (Gage, 1994; sennrich et al. 2016). We compare how much each approach can analyze morphologically the words within a word list in a manner identical to the segmentation done by a linguist, without the need for any other previous information than the words themselves.
KEYWORDS: machine learning; morphology; language documentation.

INTRODUÇÃO

No campo do aprendizado de máquina, reconhecem-se dois tipos de algoritmos: os de aprendizagem supervisionada e os de aprendizagem não-supervisionada. Os primeiros são treinados por um conjunto de dados que inclui as variáveis explanatórias juntamente com a variável resposta. O algoritmo então relaciona essas variáveis a fim de estimar a resposta mais apropriada de acordo com os novos dados que lhe serão fornecidos após esse treinamento. Os segundos recebem apenas um conjunto de dados e, sem depender de nenhuma outra informação, realizam alguma tarefa que pode ser de vários tipos. Em termos de linguística computacional, podemos ter, por exemplo,

[1] Graduando em Letras – Inglês. Universidade Federal da Bahia – UFBA.
[2] Doutor. Universidade Federal da Bahia – UFBA.

algoritmos que realizam tarefas de análise linguística de forma supervisionada ou não-supervisionada.

O presente artigo compara dois tipos de algoritmos de segmentação morfológica não-supervisionada. A partir de apenas uma lista de palavras de uma língua qualquer, esses algoritmos são capazes de inferir – por critérios estatísticos – uma possível análise morfológica para cada palavra da lista. Sendo assim, eles fornecem a segmentação morfológica de cada palavra além de um léxico de possíveis morfemas da língua.

Os objetivos de um segmentador morfológico não-supervisionado, no entanto, contrastam com os de sua contraparte supervisionada. Segundo Clark & Lappin (2010), algoritmos supervisionados conseguem maior precisão com poucos dados, dependendo, contudo, de um árduo trabalho de anotação dos *corpora* envolvidos. Para além do tempo envolvido nessa atividade, há que se considerar também que ela depende de especialistas no campo de interesse: no caso, estudiosos da morfologia das línguas para as quais se desenvolve o algoritmo. O segmentador não-supervisionado, apesar de ser menos preciso, opera sem o trabalho prévio de um linguista. Inclusive, ele pode atuar como ferramenta para o linguista que está estudando a morfologia de uma língua ainda não descrita.

Existem cerca de 7.000 línguas sendo faladas no mundo (cf. Mithun, 1998) e ao menos dois terços delas carecem de gramáticas descritivas. Afora o iminente risco de extinção de algumas dessas línguas, essa lacuna nos estudos linguísticos, além de representar uma perda irreparável para a ciência, contribui para o apagamento social, cultural e econômico dos povos falantes dessas línguas. Embora tenha havido grande desenvolvimento da área de documentação linguística nas últimas décadas, fazendo eco a Bird (2009), qual tipo de contribuição a linguística computacional pode dar a esse trabalho?

Nessa esteira podemos tomar como exemplo o trabalho de Figueiredo (2024), que desenvolveu o primeiro Treebank para o Nheengatu, abrindo um novo capítulo na pesquisa dessa língua. A partir de um corpus anotado como um Treebank, é possível identificar padrões linguísticos recorrentes de forma objetiva e reprodutível, ampliando a confiabilidade dos resultados.

Acreditamos que uma ferramenta que forneça possíveis análises para as palavras de uma língua que ainda está sendo documentada pode, para além de acelerar o trabalho de documentação em si, auxiliar o linguista na formulação de hipóteses e auxiliá-lo a verificar se as análises que ele está propondo são consistentes. Dada a menor precisão dos algoritmos não-supervisionados, no entanto, é importante que o usuário entenda como eles operam e em quais contextos eles apresentam maiores limitações.

Tendo isso em conta, comparamos dois tipos de abordagens algorítmicas que podem ser utilizadas para a segmentação morfológica não-supervisionada. Ambas as abordagens possuem pressupostos simples, de forma que é possível compreender, de forma geral, o que o algoritmo faz em cada etapa. O primeiro algoritmo é baseado na abordagem estatística denominada *Minimum Description Length* (MDL), proposta por Rissanen (1978). A aplicabilidade dessa abordagem para tarefas de análise linguística é

discutida, inicialmente, em De Marcken (1995). O segundo algoritmo é denominado *Byte Pair Encoding* (BPE) e foi introduzido em Gage (1994), com aplicação para segmentação linguística não-supervisionada introduzida em Senrich et al. (2016). Os dois algoritmos podem ser acessados via uma ferramenta que desenvolvemos na linguagem de programação Python, denominada UParser[3].

Testamos ambos os algoritmos com dados de oito línguas, extraídos de gramáticas descritivas, o arawete, da família Tupi-Guarani; o bathari, da família Afro-Asiática na região do Omã; o cheke holo, pertencente ao ramo Oceânico; o daw, língua Nadahup, subdivisão da família Maku; o kanoe, um idioma isolado; o khwarshi, classificado como pertencente à família Caspiana; o Pite Saami da família Urálica; e o yakkha, pertencente ao tronco Sino-Tibetano. A diversidade estrutural presente nesse conjunto de línguas contextualiza uma análise abrangente do desempenho do algoritmo em diferentes cenários linguísticos. Para avaliar o desempenho de cada abordagem, utilizamos uma métrica de cobertura de segmentações corretas, que mede a proporção de palavras corretamente segmentadas em relação a uma referência padrão, a segmentação apresentada pelo autor da gramática descritiva.

Organizamos o texto da seguinte forma. A primeira seção aborda a fundamentação teórica dos modelos utilizados para cada algoritmo: MDL e BPE. Na segunda seção, discutimos os contornos a metodológicos da pesquisa, como os objetivos, os dados utilizados, a implementação do UParser e os testes realizados. Na terceira seção, apresentamos e analisamos os resultados obtidos, considerando as características de cada algoritmo e os desafios específicos de cada língua. Por fim, discutimos as implicações da pesquisa para a documentação linguística e as perspectivas para o desenvolvimento de ferramentas computacionais mais robustas e eficientes para a análise de línguas com poucos recursos disponíveis.

1. FUNDAMENTAÇÃO TEÓRICA

Conforme apresentado anteriormente, existem dois tipos de modelos de aprendizagem de máquina, a depender do tipo de dados com os quais trabalham. Modelos que trabalham com dados previamente analisados, rotulados, para estimar análises de dados novos são conhecidos como *modelos de aprendizagem supervisionada*. Modelos que trabalham diretamente com dados não-analisados, não-rotulados, são conhecidos como *modelos de aprendizagem não-supervisionada*.

Podemos descrever o de aprendizado de máquina supervisionado como um modelo estatístico de dependência. Temos dois tipos de variáveis: as variáveis independentes e a variável dependente. Para cada conjunto de valores de variáveis independentes existe uma variável dependente correspondente. Em outras palavras, as variáveis independentes atuam como preditivas de um valor da variável dependente. Em termos matemáticos, o aprendizado supervisionado parte da suposição de que existe

[3] Disponível em: https://github.com/Netoamf/UParser.

uma função desconhecida que mapeia valores de entrada a um valor de saída, e cabe ao algoritmo encontrar essa função.

Na aprendizagem não-supervisionada, não há variável dependente, de forma que não há a suposição de uma função que relacione valores de entrada a um valor de saída. O que se explora, na verdade, são as interdependências entre as variáveis fornecidas, de forma que o objetivo é encontrar estruturas subjacentes aos dados apresentados. Se pensarmos no contexto da aquisição da linguagem, crianças não recebem um *input* estruturado ou palavras segmentadas e rotuladas. Modelos que empregam heurísticas associadas ao aprendizado não-supervisionado tendem a ter maior proximidade ao aprendizado da criança (Reddington et al, 1998).

A depender dos objetivos, existem várias técnicas de aprendizagem não-supervisionada. Para o trabalho de segmentação morfológica, concentramo-nos em técnicas de compressão de dados. Ambos os algoritmos de MDL e BPE que discutiremos adiante são baseados na ideia de encontrar um conjunto de sequências de símbolos que se repetem diversas vezes nos dados e podem ser, consequentemente, substituídos por outros símbolos. Um exemplo simples é considerar que várias palavras em português são terminadas em *-mos*, que indica um verbo na terceira pessoa do plural. A alta frequência da sequência *mos* pode levar o algoritmo a interpretar que ela pode ser considerada como um único símbolo ao invés de uma sequência de símbolos. Trocar uma sequência de três símbolos por um único símbolo representa um encurtamento do *corpus*, ou seja, uma compressão. O novo símbolo é adicionado a uma lista, que intitulamos léxico, que faz a correspondência entre sequências de símbolos e símbolos únicos. Novos elementos são acrescentados ao léxico de acordo com técnicas de otimização que têm por objetivo chegar ao menor léxico que atinja a maior compressão do *corpus*.

1.1. *MINIMUM DESCRIPTION LENGTH* (MDL)

A primeira abordagem de segmentação não-supervisionada que testamos é baseada na técnica de otimização denominada *Minimum Description Length* (Rissanen, 1978). Trata-se de uma técnica baseada em Estatística Bayesiana na Teoria da Informação de Shannon, 1949. A Estatística Bayesiana é uma escola de inferência estatística desenvolvida inicialmente por Laplace nos fins do século XVIII com base no teorema de Bayes.

O teorema permite encontrar a probabilidade de um evento A dado um evento B quando temos as probabilidades dos eventos A e B individualmente e a probabilidade do evento B dado o evento A. Aplicando para a pesquisa científica, podemos substituir A pela hipótese e B pelos dados. Sendo assim, podemos estabelecer qual a propriedade de uma hipótese estar correta considerando os dados apresentados. Para isso, precisamos ter uma probabilidade prévia de a hipótese estar correta, uma probabilidade dos dados e uma probabilidade de os dados ocorrerem considerando a hipótese. Sendo H a hipótese e D os dados, a equação é a seguinte:

(1) $$P(H|D) = \frac{P(D|H)P(H)}{P(D)}$$

Alguns componentes de equação recebem nomes específicos dada a sua importância: (i) P(H), a probabilidade da hipótese é denominada ***priori*;** (ii) P(H|D), a probabilidade da hipótese considerando os dados é denominada ***posteriori*** e (iii) P(D|H), a probabilidade dos dados considerando a hipótese é denominada **verossimilhança**. A ideia da Estatística Bayesiana é observar o quanto a *posteriori* aumenta quando aplicamos uma hipótese H de probabilidade P(H) para explicar os dados D. Na realidade, no entanto, não se trabalha com valores de probabilidade simplesmente, mas com funções de probabilidade.

As funções de probabilidade são funções que dão um determinado valor entre 0 e 1, representando uma probabilidade, a depender de parâmetros inseridos. Embora existam diversos tipos de funções de probabilidade já consagrados no estudo da estatística (e a Estatística Bayesiana tenha diversos métodos de como combinar essas diferentes funções), essa abordagem dá ainda mais flexibilidade. Por exemplo, é possível o estudioso utilizar funções mais específicas para o tipo de problema com que se está lidando.

No caso de um segmentador morfológico, podemos entender que estamos em busca de encontrar um léxico de morfemas da língua. Sendo assim, nossa *priori* e *posteriori* correspondem a funções de probabilidade para propriedades desse léxico. A proposta do MDL é a de que o léxico mais provável é um que tenha o menor número de elementos que atinjam maior compressão do *corpus*. A verossimilhança seria a probabilidade de um *corpus* de palavras segmentadas terem sido segmentadas de acordo com o léxico em questão.

Para lidar com dados de *corpora*, que são sequências de símbolos, a Teoria da Informação oferece alguns recursos que são cruciais. Para os objetivos de um segmentador morfológico não-supervisionado, o principal deles é tratar as probabilidades em termos de custos. Na equação a seguir, o custo *c* de uma probabilidade *p* calcula-se pelo negativo do logaritmo de *p* na base 2:

(2) $$c(\mathrm{p}) = -log_2 p$$

Um outro nome para custo comumente utilizado na literatura é ***plog***, referente a ***positive logarithm*** ou logaritmo positivo. Isso porque o resultado do negativo do logaritmo de uma probabilidade será sempre um número maior que 0. Lidar com custos ao invés de probabilidades é especialmente vantajoso. As probabilidades relativas à ocorrência de itens lexicais em *corpora* linguísticos são normalmente muito baixas, e isso dificulta os cálculos computacionalmente. Logaritmos apresentam os números em uma escala mais adequada, e a multiplicação de probabilidades é a soma dos logaritmos dessas probabilidades, ou seja, o cálculo é mais fácil.

Voltando ao funcionamento da Estatística Bayesiana, para muitas aplicações não há a necessidade de se obter o valor exato da *posteriori*. Para o algoritmo de

segmentação morfológica, o que se deseja obter é a melhor *posteriori* possível: aquela com probabilidade mais alta ou custo mais baixo. Ou seja, queremos otimizar a *posteriori*. Calcular o denominador – P(D), probabilidade dos dados – do teorema de Bayes pode ser especialmente desafiador. Quando não precisamos do valor exato da *posteriori*, simplesmente ignoramos esse componente sabendo que a combinação da *priori* com a verossimilhança é proporcional à *posteriori*. Considerando os dados em termos de custos agora, e não de probabilidades, temos que o custo da hipótese (*priori*) somado ao custo dos dados considerando a hipótese (verossimilhança) é proporcional ao custo da hipótese considerando os dados (*posteriori*):

(3) $c(h|d) \propto c(d|h) + c(h)$

A segmentação de dados linguísticos seguindo esse tipo de proposta foi introduzida em De Marcken (1995). Baseando-se na técnica de MDL, o autor propôs tratar o custo da hipótese em termos de comprimento do léxico e a verossimilhança em termos de comprimento do *corpus*. A ideia é que o tamanho do léxico deve crescer na medida em que o comprimento do *corpus* diminui.

O comprimento do léxico é calculado com base na seguinte premissa: cada caractere do *corpus* possui um custo. Esse custo é calculado em termos do tamanho do alfabeto. Um *corpus* com, por exemplo, 35 caracteres distintos, terá um alfabeto de tamanho 35. Sendo assim, o custo de cada caractere é o logaritmo positivo de 1/35. Em termos gerais, sendo *c*(*a)* o custo de um caractere de um alfabeto de tamanho *a* temos:

(4) $c(a) = -log_2\left(\frac{1}{a}\right)$

Conforme itens lexicais com mais de um caractere são adicionados ao léxico, o comprimento do léxico aumenta de acordo com o número de caracteres de cada novo item. Um léxico que contenha somente o alfabeto terá um comprimento dado pela multiplicação do tamanho **a** do alfabeto pelo custo de cada caractere **c**, ou seja **ac**. Se um item de 3 caracteres for adicionado a esse léxico, ele passa a ter o comprimento **ac + 3c**. Mais um item de 5 caracteres é adicionado e o comprimento passa a ser **ac + 3c + 5c**, ou **ac + 8c**. Ou seja, o comprimento do léxico cresce linearmente a partir do tamanho do alfabeto e do número de caracteres de cada palavra que é inserida nele.

O comprimento do *corpus* é calculado de acordo com o custo de cada item lexical que o compõe. O custo do item lexical é uma função sua frequência de ocorrência. Sendo **w** o número total de itens lexicais presentes no *corpus* e **l** a frequência de um determinado item, o custo **c(l)** do item é dado por:

(5) $c(l) = -log_2\left(\frac{l}{w}\right)$

Tendo o custo de cada item lexical, calcula-se o comprimento do *corpus* somando o custo de cada item lexical que ali ocorre. Sendo assim, uma frase como "um cão e um gato", em que **um** teria custo 2, **cão** teria custo 12, **gato** teria custo 14 e **e** teria custo 3, o comprimento seria *2 + 12 + 3 + 2 + 14 = 33*.

Se o comprimento do léxico cresce conforme o tamanho de cada item lexical que é acrescentado, o comprimento do *corpus* diminui conforme menos itens lexicais o constituem. Um *corpus* analisado com um léxico constituído apenas pelo alfabeto da língua terá o comprimento proporcional ao número de caracteres do *corpus* e ao custo de cada caractere. Se o léxico passa a ganhar mais itens lexicais constituídos por mais caracteres, a expectativa é que o *corpus* passe a ter um comprimento menor pois será constituído de menos itens lexicais.

Considerando tudo isso, um algoritmo para segmentar morfologicamente um *corpus* que consiste em uma lista de palavras deve partir de um léxico inicial que analise o *corpus* de forma a obter um valor de comprimento de *corpus*. Somando-se o comprimento do *corpus* ao comprimento do léxico, obtém-se um valor proporcional ao custo da hipótese. Esse valor será inicialmente alto e o objetivo é reduzi-lo ao máximo. Isso se dá por adicionar ao léxico combinações de caracteres que sejam frequentes ao longo do *corpus*. Com a adição de determinado número de novos itens lexicais obtidos a partir das sequências de caracteres mais frequentes do *corpus*, a expectativa é justamente que o comprimento do *corpus* diminua enquanto o comprimento do léxico aumente. Diminuições substanciais do comprimento do *corpus* são esperadas com um pequeno aumento do número de elementos do léxico, otimizando o custo da hipótese.

1.2 BYTE PAIR ENCODING (BPE)

A segunda abordagem para segmentação não-supervisionada é baseada no algoritmo denominado *Byte Pair Encoding* (Gage, 1994). Trata-se de um algoritmo de compressão que não possuía, a época de seu desenvolvimento, uma fundamentação matemática tão elaborada quanto o MDL. O nome **byte** origina-se do tamanho que um caractere tradicionalmente ocupa na memória dos computadores: um caractere ocupa um byte de memória. O algoritmo consiste em substituir pares de símbolos frequentes por símbolos únicos não utilizados no conjunto de dados original. Esse processo iterativo continua até que nenhuma substituição adicional seja possível, pela ausência de pares frequentes ou pela exaustão dos símbolos disponíveis para substituição.

Uma maneira de visualizar esse processo é com a seguinte cadeia de dados 'ABABCABCD'. O algoritmo identifica o par 'AB' como o mais frequente e substitui por um novo símbolo 'X', o que resulta na cadeia 'XXCXCD'. A seguir o par a ser substituído é o par 'XC' por 'Y', resultando na cadeia final e mais compacta 'XYYD'. Uma lista guarda as correspondências, em que X = AB e XC = Y. Nesse caso, se 'ABABCABCD' for uma palavra, XYYD representa seus morfemas: AB-ABC-ABC-D.

Sennrich et al (2016) reconheceram o potencial do BPE para segmentação de palavras, adaptando o algoritmo para tarefa de tradução automática. Sua abordagem

baseia-se na ideia de que a tradução de diversas classes de palavras, como nomes próprios, palavras compostas e cognatos, pode ser realizada através da tradução de unidades menores do que a palavra em si.

A escolha do BPE por Sennrich et al (2016) se justifica por diversas razões. O BPE, pela natureza do algoritmo, permite modelar uma linguagem aberta, ou seja, um sistema com um vocabulário potencialmente infinito, por meio de um vocabulário finito de subunidades. Além disso, o BPE gera representações de comprimento variável, o que permite capturar a estrutura interna das palavras de maneira mais adequada que um algoritmo que use representações de comprimento fixo. Essas subunidades geradas pelo BPE permite um grau de interpretabilidade linguística razoável, permitindo que a partir delas o algoritmo aprenda os padrões e generalizações de uma língua.

A relevância linguística das subunidades geradas pelo BPE se manifesta em diversos níveis de análise. Em línguas com processos de formação de palavras produtivos, como a aglutinação e a composição, o BPE consegue capturar morfemas e afixos relevantes, facilitando a tradução e a geração de palavras complexas. Em línguas com alta incidência de cognatos e empréstimos, o BPE pode aprender transformações fonológicas e morfológicas regulares, auxiliando na tradução dessas palavras (Gutierrez-Vasques et al, 2023).

Para a segmentação morfológica a luz da natureza do algoritmo e seu comportamento na tarefa, Gutierrez-Vasques et al (2023) aponta que línguas com um alto índice de síntese, como o turco e o finlandês que tendem a ter palavras com múltiplos morfemas, facilitam o aprendizado de subunidades relevantes pelo BPE, enquanto línguas com baixo índice de síntese como o vietnamita que tendem a ter palavras monomorfêmicas acabam dificultando o processo pelo algoritmo.

2. METODOLOGIA

Implementamos, em linguagem Python, dois segmentadores morfológicos baseados nas abordagens discutidas anteriormente: *Minimum Description Length* (MDL) e *Byte Pair Encoding* (BPE).

Para o MDL, adaptamos o algoritmo proposto por De Marcken (1995) para a descoberta lexical não supervisionada. De Marcken (1995) aplicou o algoritmo a sequências como "umcãoeumgato", segmentando-as corretamente em "um cão e um gato". Nesse sentido, inferimos que, se o algoritmo consegue delimitar palavras em uma sequência sem espaços, ele também seria capaz de identificar morfemas dentro de uma palavra, adaptando-se à tarefa de segmentação morfológica.

Concomitantemente, para avaliarmos a eficácia do MDL em relação a outras possibilidades de segmentação morfológica não-supervisionada, decidimos implementar um segundo algoritmo, para avaliar sua real eficácia. O algoritmo Byte Pair Encoding (BPE), também originário do campo da compressão de dados e adaptado para a tarefa de segmentação morfológica foi o escolhido.

2.1. *CORPORA*

Os *corpora* analisados consistiram em gramáticas descritivas de oito línguas tipologicamente diversas, cada uma com estruturas distintas. As línguas estudadas foram Araweté (Solano, 2009), Bathari (Gasparini, 2018), Cheke Holo (Boswell, 2018), Daw (Carvalho, 2016), Kanoê (Bacelar, 2004), Khwarshi (Khalilova, 2009), Pite Saami (Wilbur, 2014) e Yakkha (Schackow, 2015), todas elas com número reduzido de falantes e recursos disponíveis.

Por exemplo, o Araweté é falado por apenas 467 pessoas. O Bathari conta com apenas 16 falantes nativos e está em risco de extinção, não sendo mais ensinada nas escolas. Já o Kanoê, ou Kapixaná, é uma língua polissintética e aglutinante e não possui mais falantes nativos (Eberhard et al.,2024).

As gramáticas descritivas costumam apresentar seus dados com glosas interlineares, como se apresenta abaixo:

(6) maj-ahu ø-ha mɨde r-e ʔe we mɨde ø-ʔu
Cobra-INTS R-ir gente R-CR esse TOP gente R-comer
"A cobra grande vai comer pessoa"

(Solano, 2009: 4)

Na primeira linha, a sentença na língua Araweté é segmentada em seus morfemas constituintes. Essa segmentação é realizada pelo linguista, que separa cada unidade morfológica, evidenciando a estrutura interna da língua. A frase é decomposta em suas partes menores, permitindo uma análise precisa da formação das palavras e das relações entre os morfemas. A segunda linha apresenta uma glosa morfêmica, onde cada segmento identificado na primeira linha recebe seu devido rótulo, essencial para atribuição das funções gramaticais de cada componente e suas relações na sentença. Na terceira linha vemos a tradução livre da sentença para a língua de destino. Para os fins desse trabalho, consideramos apenas a primeira linha.

Os dados das gramáticas são processados de duas maneiras. Na primeira, as linhas de glosa e tradução são removidas, e os hifens que segmentam as palavras em morfemas são eliminados. As palavras são, então, concatenadas e separadas em linhas individuais, criando o *corpus* alvo da análise. Na segunda, os hifens são removidos, mas as palavras permanecem segmentadas (por espaços em branco) conforme a análise do linguista. Essa abordagem constitui o *corpus* referência, que permite comparar a segmentação feita pelo linguista com a segmentação automatizada pela ferramenta.

A proporção de palavras segmentadas pelos algoritmos que correspondem à segmentação dada no *corpus* referência constitui uma métrica denominada **cobertura**. Quanto maior a cobertura de uma segmentação feita pelos algoritmos, mais próxima ela está da segmentação feita por um linguista. O quadro a seguir ilustra como cada frase é transformada em lista de palavras.

FRASE ORIGINAL	*CORPUS* ALVO	*CORPUS* REFERÊNCIA
mũmũ ∅ puty	Mũmũ ∅puty	Mũmũ ∅ puty
padydy-i r-awe	padydyi rawe	padydy i r awe
akaju-i r-atʃĩ	Akajui ratʃĩ	akaju i r atʃĩ
a-maʔẽ ku he tajahu r-ehe	amaʔẽ ku he tajahu rehe	a maʔẽ ku he tajahu r ehe

Tabela 1: Processamento inicial dos dados obtidos das gramáticas descritivas.

2.2. O ALGORITMO MDL

Para implementar o segmentador morfológico baseado em MDL, desenvolvemos um algoritmo que segue um processo iterativo de busca da análise que minimize a soma do comprimento do *corpus* e do comprimento do léxico.

O algoritmo inicia por estabelecer o comprimento inicial do léxico, que é calculado a partir do tamanho do conjunto de caracteres que constitui o *corpus*, multiplicando tal quantitativo pelo custo de cada caractere conforme a equação apresentada anteriormente em (4). Obtém-se posteriormente a frequência de cada item lexical no *corpus*, cada um constituído somente por um caractere neste momento. A partir destas frequências calcula-se o comprimento do *corpus*, seguindo a equação em (5). A soma do comprimento do léxico inicial e do comprimento do *corpus* de acordo com os itens presentes nesse léxico será proporcional à posteriori da hipótese. Esse número deverá ser reduzido ao longo das iterações subsequentes.

Estabelecido o léxico inicial, constituído de itens de um único caractere, cada iteração consiste nas seguintes etapas. Primeiramente, toma-se o *corpus* segmentado em morfemas pela iteração anterior e concatena-se cada par de morfemas. Em seguida, observa-se a frequência desses pares concatenados e selecionam-se os *n* pares mais frequentes, que serão acrescentados ao léxico juntamente com sua frequência de ocorrência. Após isso, segmenta-se o *corpus* utilizando o léxico acrescido dos novos pares.

É importante levar em conta que podem ocorrer diferentes formas de segmentação para cada sequência de caracteres. por exemplo, uma sequência como “andávamos” pode ser segmentada como “a-n-dá-vamos” ou “andá-va-mos” se o léxico que contiver os itens **a, n, dá, vamos, andá, va** e **mos**. A segmentação escolhida pelo algoritmo é a mais provável, ou seja, aquela cuja soma dos custos de cada item lexical será a menor. Essa forma de segmentação, obtida por meio de técnicas de programação dinâmica, é crucial para o funcionamento correto do algoritmo.

Após essa primeira segmentação, obtém-se a frequência de cada item lexical no *corpus* e atualiza-se o léxico com essas frequências. Calcula-se o comprimento desse

novo léxico e segmenta-se o *corpus* novamente, obtendo seu comprimento. Caso a soma do comprimento do léxico com o comprimento do *corpus* seja menor que uma dada margem de diferença para com a soma da iteração anterior, procede-se para mais uma iteração. Caso contrário, descarta-se essa iteração e encerra-se o algoritmo na iteração anterior que tem uma probabilidade maior.

O algoritmo requer obrigatoriamente o parâmetro do número **n** de concatenações mais frequentes de morfemas que deverão ser mantidas no léxico. Além disso, permite-se optar por trabalhar com um número **i** de iterações ao invés de aguardar a interrupção do algoritmo quando a soma dos comprimentos do *corpus* e léxico ser maior ou muito próxima da soma da iteração anterior.

Para este estudo, diferentes combinações de parâmetros **n** e **i** foram testadas. Com relação ao parâmetro **i,** cada língua teve 5 rodadas de testes, cada uma para um número de iterações: 1, 2, 4, 8 e 16. Em cada rodada o algoritmo foi executado 10 vezes para diferentes valores do parâmetro **n**: 1, 2,4,8,16,32,64,128,256 e 512. Os valores selecionados para os parâmetros variaram, portanto, em escala logarítmica na base de 2. Para cada língua e registrou-se os valores dos parâmetros com os quais se obteve a maior cobertura. O gráfico abaixo ilustra, para o *corpus* da língua Araweté, os resultados das coberturas obtidos com os diferentes valores dos parâmetros:

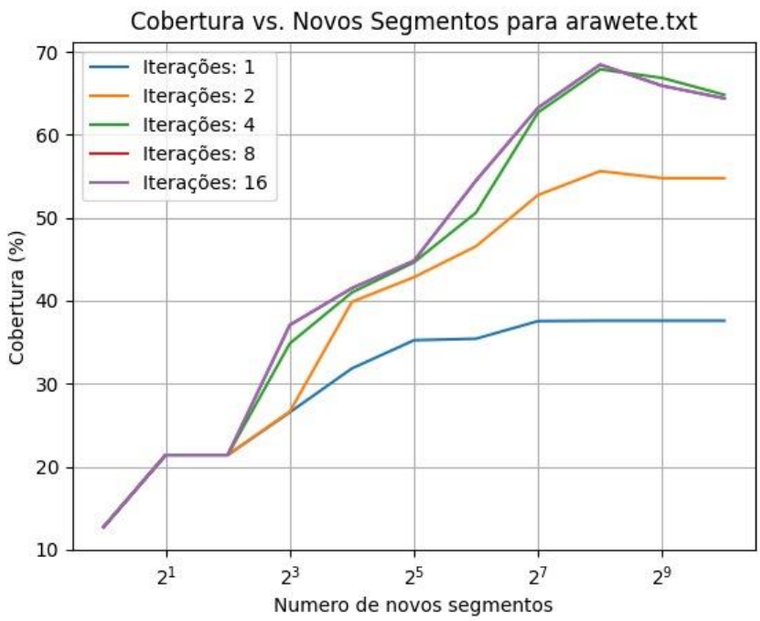

Gráfico 1: Relação entre número de iterações (i), novos segmentos (n) e cobertura na segmentação do *corpus* da língua Araweté.

2.3. O ALGORITMO BPE

Para o BPE, optamos pela implementação de Sennrich et al. (2016) disponibilizada na biblioteca **subword-nmt.** A escolha desta implementação se justifica por sua robustez e aceitação acadêmica, garantindo a reprodutibilidade e a comparabilidade de nossos resultados.

O algoritmo pode ser descrito da seguinte forma. Da mesma forma que no MDL, iniciamos o léxico com os caracteres individuais do *corpus* com sendo os itens lexicais.

Cada palavra do *corpus* é uma sequência de símbolos presentes no léxico finalizada por um símbolo especial que marca o fim de palavra. Concatenam-se todos os pares adjacentes de símbolos presentes no *corpus* e contam-se suas frequências. O par mais frequente é adicionado ao léxico como um único símbolo e substituído no *corpus*. Cada iteração acrescenta um novo par no *corpus*, operação a que denominamos mesclagem.

Neste estudo implementamos rodadas de 100 em 100 iterações e medimos a cobertura resultante. Para a maior parte das línguas a cobertura máxima foi atingida entre 200 e 400 iterações. O gráfico a seguir ilustra os diferentes valores de cobertura atingidos para cada língua de acordo com os diferentes números de iterações:

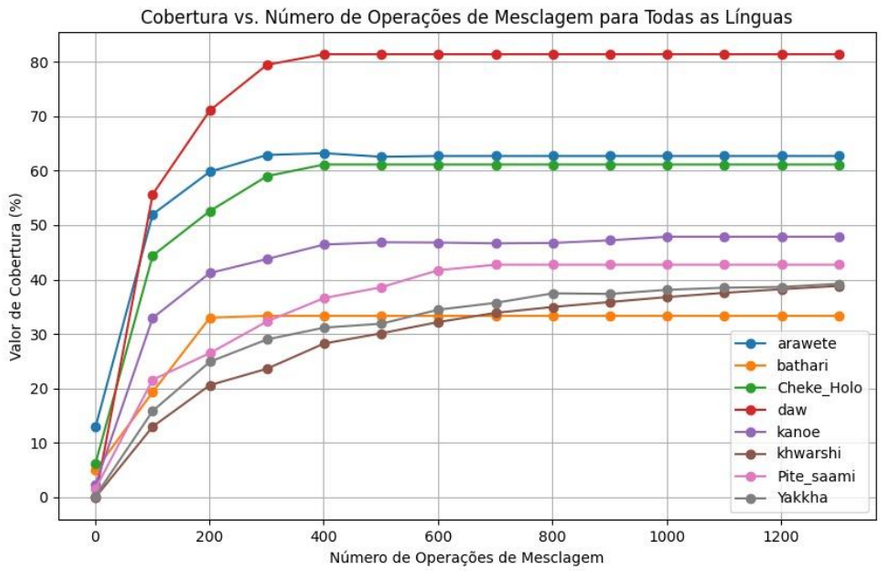

Gráfico 2: Coberturas obtidas com diferentes números de iterações (mesclagens) em cada língua estudada.

3. RESULTADOS

O tamanho do *corpus* de cada língua analisada, em termos de número de palavras, foi diferente e encontra-se relacionado na **Tabela 2** a seguir:

Língua	Tamanho
Araweté	2.120
Bathari	300
Cheke Holo	795
Dâw	1.085
Kanoê	4.133
Khwarshi	6.366
Pite Saami	1.306
Yakkha	2.634

Tabela 2: Quantidade de palavras em cada *corpus*.

As coberturas máximas para cada língua obtidas via MDL estão apresentadas na Tabela 3. A tabela também mostra a quantidade de iterações (parâmetro i) e de novos itens a cada iteração (parâmetro n) que resultaram na referida cobertura.

Língua	Iterações (i)	Novos itens (n)	Cobertura
Araweté	8	256	68,44%
Bathari	8	256	51,00%
Cheke Holo	8	512	87,42%
Dâw	8	256	94,01%
Kanoê	16	512	61,26%
Khwarshi	16	1.024	46,69%
Pite Saami	8	512	57,81%
Yakkha	8	512	47,72%

Tabela 3: Coberturas máximas atingidas pelo segmentador MDL e os respectivos parâmetros utilizados.

Como é possível observar, das aplicações do segmentador nas 8 línguas, 7 atingiram a cobertura máxima com i = 8, destas, 3 a atingiram com n = 256 e 4 com n = 512. Com o kanoê o segmentador atingiu a cobertura máxima com i = 16 e n = 512 e com o khwarshi, com i = 16 e n = 1024. A cobertura média obtida com a abordagem MDL foi de 64,29% com 0,18% de desvio-padrão.

Os resultados de cobertura máxima e respectivos números de iterações (mesclagens) obtidos via BPE estão apresentados na Tabela 4.

Língua	Mesclagens	Cobertura
Araweté	400	63,21%
Bathari	400	33,33%
Cheke Holo	400	61,13%
Dâw	400	81,38%
Kanoê	900	47,21%
Khwarshi	2.200	41,75%
Pite Saami	700	42,73%
Yakkha	600	37,47%

Tabela 4: Coberturas máximas atingidas pelo segmentador BPE e os respectivos números de mesclagens utilizadas.

Conforme podemos observar, para a segmentação baseada em BPE, para as 8 línguas, 4 obtiveram cobertura máxima com 400 mesclagens. A segmentação dos dados do Kanoê atingiu a cobertura máxima com 900 mesclagens, dos dados do Khwarshi com 2200 mesclagens, dos dados do Pite Saami com 700 mesclagens e dos dados do Yakkha com 600 mesclagens. A média das coberturas máximas atingidas foi de 51,03% com desvio-padrão de 0,17%.

Com desvios-padrões semelhantes entre os dois métodos, é de se supor que as diferenças de desempenho que ambos os métodos apresentam para os dados de cada língua esteja relacionado ao tamanho do *corpus* ou a propriedades da morfologia de cada língua. Apesar de termos testado com um número pequeno de línguas, a hipótese da influência do tamanho do *corpus* não parece ser provável uma vez que o resultado do teste de Pearson para a correlação entre o tamanho dos *corpora* e o respectivo desempenho de cada segmentador não se mostrou significativo ($p = 0.232$ para o segmentador MDL e $p = 0.533$ para o segmentador BPE). Sendo assim, propriedades relacionadas à morfologia de cada língua são as candidatas mais apropriadas para melhor ou pior desempenho dos segmentadores e investigações a esse respeito estão em andamento.

Aplicando o teste-t para comparar a diferença de desempenhos entre cada segmentador, obtivemos o valor-p = 0.074. Apesar de serem poucas as línguas observadas, o teste mostra que, mesmo o segmentador baseado em MDL tendo maior média de cobertura que o baseado em BPE (13,26% de diferença), essa diferença ainda não é significativa o suficiente (valor-p > 0.05) para se afirmar que o segmentador baseado em MDL será sempre superior ao baseado em BPE. Ainda assim, o MDL apresentou resultados superiores ao BPE para os dados de todas as línguas. A diferença de desempenho segue apresentada na Tabela 5.

Língua	MDL	BPE	Diferença
Araweté	68,44%	63,21%	5,23%
Bathari	51,00%	33,33%	17,67%
Cheke Holo	87,42%	61,13%	26,29%
Dâw	94,01%	81,38%	12,63%
Kanoê	61,26%	47,21%	14,05%
Khwarshi	46,69%	41,75%	4,94%
Pite Saami	57,81%	42,73%	15,08%
Yakkha	47,72%	37,47%	10,25%

Tabela 5: Comparação das coberturas máximas das segmentações baseadas em MDL e BPE.

4. Discussão e Considerações Finais

Os resultados apresentados mostram que os algoritmos de segmentação morfológica não-supervisionada testados aqui analisam as palavras de forma idêntica ao linguista em 46,69% a 94,01% dos casos para o método MDL e de 33,33% a 81,38% para o método que segue o BPE. Apesar de serem números que variam bastante e que evidenciam baixa precisão em muitos dos casos, deve-se ter em conta que os testes se basearam um número bastante reduzido de dados, com listas de que variam em quantidade de 300 a 6.366 palavras. Além disso, sendo os dados vindos de gramáticas descritivas, eles tendem a favorecer a diversidade de estruturas linguísticas e não a representar o uso cotidiano da linguagem. Dessa forma, a frequência das palavras

normalmente não corresponde à frequência que ocorreria em outros tipos de *corpora,* como jornais ou obras literárias, que são – muitas vezes – indisponíveis para essas línguas.

Ainda assim, mais importante do que a própria cobertura da segmentação, é que o usuário de segmentadores como os testados neste trabalho tenham noção de como funcionam e como reagem a determinados tipos de dados linguísticos. Nesse sentido, a pesquisa deve se concentrar doravante em compreender o que causa a variação nas taxas de cobertura de língua para língua. Quais características tipológicas são favorecidas ou desfavorecidas por esse tipo de método e qual o impacto da natureza dos dados de gramática descritiva, selecionados em termos de diversidade estrutural, sobre esses resultados.

Mesmo diante da baixa precisão em alguns casos, é importante ressaltar que esse tipo de segmentador pode ser útil para o trabalho de documentação. Uma vantagem presente no segmentador baseado em MDL, por exemplo, é a possibilidade de funcionar em modo semi-supervisionado. Dada sua fundamentação em estatística bayesiana, é permitido que itens lexicais sejam fornecidos pelo linguista a qualquer tempo e que novas palavras sejam descobertas a partir de análises que já considerem a descrição prévia feita manualmente. A eficácia desse tipo de estratégia ainda precisa ser estudada, mas métodos semi-supervisionados têm se mostrado promissores de forma geral para aplicações em linguística (cf. Abney, 2007; Clark & Lappin, 2010).

REFERÊNCIAS

ABNEY, Steven. **Semi-supervised Learning for Computational Linguistics.** Chapman & Hall. 2007.

BACELAR, Laercio Nora. **Gramática da Língua Kanoê.** Nijmegen: Katoliek Universiteit Nijmegen. 2004.

BIRD, Steven. Natural Language Processing and Linguistic Fieldwork. **Computational Linguistics, v. 35, n. 3.** 2009.

BOSWELL, Fredrik Alvin. **A Grammar of Cheke Holo.** Utrecht: LOT Publications. 2018.

CARVALHO, Maurício Oliveira Pires. **Aspecto Verbal na Língua Dâw.** Dissertação de mestrado apresentada ao programa de pós-graduação em Linguística da Universidade de São Paulo. 2016.

CLARK, Alexander; LAPPIN, Shalom. Unsupervised Learning and Grammar Induction. *In:* CLARK, Alexander; Fox, CHRIS; LAPPIN, Shalom. **The Handbook on Computational Linguistics and Natural Language Processing.** Hoboken: Wiley-Blackwell. 2010.

De MARCKEN, Carl G. **Unsupervised Language Acquisition.** Tese de doutorado apresentada no Massachusetts Institute of Technology. 1995.

EBERHARD, David M.; Simons, Gary F.; Fennig, Charles D. **Ethnologue: Languages of the World. Twenty-seventh edition.** Dallas: SIL International. 2024. Disponível em: http://www.ethnologue.com

FIGUEIREDO, L. A Universal Dependencies Treebank for Nheengatu. **ACL Anthology**, p. 37–54, mar. 2024.

GAGE, Philip. A New Algorithm for Data Compression. **The C User Journal.** 1994.

GASPARINI, Fabio. **The Bathari Language of Oman: Towards a Descriptive Grammar.** Tese de doutorado apresentada na Università degli Studi di Napoli "L'Orientale". 2018

GUTIERREZ-VASQUES, Ximena; BENTZ, Christian; SAMARDŽIĆ, Tanja. Languages Through the Looking Glass of BPE Compression. **Computational Linguistics, v. 49, n 4.** 2023.

KHALILOVA, Zaira. **A Grammar of Khwarshi**. Utrecht: LOT Publications. 2009

MITHUN, Marianne. The significance of diversity in language endangerment and preservation. In: Grenoble, Leonore; Whaley, Lindsay. **Endangered Languages Language Loss and Community Response**. Cambridge University Press. 1998.

REDDINGTON, Martin; CHATER, Nick; FINCH, Steven. Distributional information: A powerful cue for acquiring syntactic categories. **Cognitive Science,** v. 22, n. 4. pp. 425-469**.** 1998

RISSANEN, Jorma. Modeling by shortest data description. **Automatica**, v. 14 n.5. pp. 465-471. 1978

SCHACKOW, Diana. **A Grammar of Yakkha.** Leipzig: Language Science Press. 2015

SENNRICH, Rico; HADDOW, Barry; BIRCH, Alexandra. Neural Machine Translation of Rare Words with Subword Units. https://doi.org/10.48550/arXiv.1508.07909. 2016.

SHANNON, Claude; WEAVER, Warren. **The Mathematical Theory of Communication**. Urbana: The University of Illinois Press. 1949

SOLANO, Eliete Bararuá. **Descrição Gramatical da Língua Araweté**. Tese de Doutorado apresentada ao programa de Pós-Graduação em Linguística da UnB. 2009

WILBUR, Joshua. **A Grammar of Pite Saami.** Leipzig: Language Science Press. 2014