Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Validação Multilingual de métricas de complexidade das línguas humanas

Um TCC sobre a análise de métricas de complexidade de línguas naturais baseadas em algoritmos de compressão

Gabriel Ferreira de Souza Araujo
Lucas Irineu Rebouças Guimarães

Monografia Final

mac 499 — Trabalho de Formatura Supervisionado

Supervisor: Marcelo Finger
Cossupervisor: Felipe Ribas Serras

São Paulo
2024

Agradecimentos

Gostaríamos de agradecer às nossas famílias e amigos pelo apoio que nos deram durante a graduação. Agradecemos também ao professor Marcelo Finger, doutorando Felipe Serras e mestrando Miguel Carpi por nos ter ajudado durante o desenvolvimento deste projeto

Resumo

Gabriel Ferreira de Souza Araujo
Lucas Irineu Rebouças Guimarães
. **Validação Multilingual de métricas de complexidade das línguas humanas: *Um TCC sobre a análise de métricas de complexidade de línguas naturais baseadas em algoritmos de compressão*** . Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2024.

Em Processamento de Linguagem Natural (PLN), a área de complexidade de linguagem está crescendo, já que ela possibilita uma análise quantitativa de línguas naturais. Dita análise é feita baseada em duas métricas, a complexidade morfológica e a sintática. A complexidade morfológica representa a quantidade de informações que cada língua carrega dentro das palavras, já a complexidade sintática as informações fora das palavras. Este trabalho testa como estas métricas funcionam para línguas de diferentes origens. O método para testá-las é fazer uma degradação de textos destas línguas e analisar a quantidade de informação perdida durante o processo de degradação.

Palavras-chave: Processamento de Linguagem Natural. Complexidade de Linguagem. Complexidade Morfológica. Complexidade Sintática.

Abstract

Gabriel Ferreira de Souza Araujo
Lucas Irineu Rebouças Guimarães
. **Multilingual validation of human language complexity metrics:** ***A Final Paper about complexity metrics analysis of natural languages based on compression algorithms***. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2024.

In Natural Language Processing, language complexity is an emerging area, because of its capability of quantitative analyze natural languages. Said analysis can be made through two metrics, morphological and syntactic complexity. Morphological complexity represents how much information each language carries inside words, whereas syntactic complexity represents how much information each language carries outside words. This paper test how these metrics works for languages of different origins. The method to test them is by degrading texts of each language and analyze the quantity of lost information during the degrading process.

Keywords: Natural Language Processing. Language Complexity. Morphological Complexity. Syntactic Complexity.

Lista de Abreviaturas

PLN	Processamento de Linguagem Natural
TCC	Trabalho de Conclusão de Curso
SEM	Erro Padrão da Média

Lista de Figuras

Sumário

Capítulo 1

Introdução

Desde as sociedades da antiguidade, o debate sobre o significado das palavras e línguas é um tema que sempre é abordado pelos pensadores da época, devido à importância da linguagem para a convivência humana. Na grécia antiga, o filósofo Platão explora o sentido das palavras, questionando se as palavras possuem uma relação intrínseca com as coisas que a representam ou se essa relação é arbitrária. Já na tradição chinesa, Confúcio discute a importância das palavras corretas para a harmonia social, já que apenas nomes corretos podem expressar a verdade e, sem a verdade, os assuntos não podem ser bem-sucedidos. Da mesma forma, filosofos árabes como Al-Kindi desenvolvem teorias sobre a relação entre as palavras e seu significado. A convergência de culturas diferentes para o estudo da linguagem e seus significados mostra como o entendimento de línguas é algo necessário para a prosperidade das sociedades.

Da mesma forma, para a área de computação, o interesse por compreender línguas também é um tema importante. No entanto, ao invés do interesse por ter um entendimento da comunicação entre pessoas, a computação se preocupa com a comunicação entre pessoas e computadores. No começo da área, essa comunicação só era possível através da linguagem de máquina, mas com a evolução da computação foram criadas diversas ferramentas para funcionar como intermediários entre a máquina e o humano, como linguagens de programação, até chegar na criação da área de Processamento de Linguagem Natural (PLN), que permite ao computador processar linguagens naturais, ou seja, linguagens utilizadas na comunicação entre humanos. O PLN é importante para que programas sejam capazes de realizar atividades na língua natural como humanos fazem, além de facilitar pessoas que não entendem computação a utilizar programas que necessitam de um entendimento computacional.

Adentrando PLN, o desenvolvimento de métodos para analisar línguas de forma quantitativa é util para estudos linguístico-computacionais, pois permite entender como línguas carregam informações, além de esclarecer o processo cognitivo usado nas línguas humanas. O conceito de complexidade de línguas é algo intuitivo para pessoas, por exemplo, um texto não formal parece menos complexo do que um formal, ou algumas línguas parecem ser mais fáceis de se aprender do que outras. Para ir além da intuição, é necessário estabelecer métricas rigorosas, teóricas e práticas. Esse trabalho pretende examinar métricas de complexidade derivadas da Teoria da Informação, contribuindo para a robustês da biblioteca

em Python "*lang-complexity*"[16], capaz de analisar e estimar a complexidade de textos em qualquer língua.

Capítulo 2

Trabalhos Relacionados

A ideia de utilizar métricas quantitativas para avaliar a complexidade de línguas naturais é recente. Um trabalho pioneiro dessa ideia foi Nichols [13], que propôs analisar a complexidade de linguagem a partir do número de pontos de inflexão em uma sentença.

Por outro lado, Patrick Juola [10] [9] [8] propôs métricas de complexidade de línguas a partir da compressão de textos, baseadas na Teoria da Informação. A partir dessa família de métricas foram desenvolvidos diversos trabalhos, que propuseram melhorias e analisaram a sua sensibilidade ([3], [4], [5], [6], [7], [11], [12], [14], [15]), em especial o *Analysing and Validating Language Complexity Metrics Across South American Indigenous Languages*[16], que analisa se as métricas mantém-se robustas para línguas indígenas sul-americanas, validando as métricas além do cenário linguístico europeu.

A partir da analise feita no artigo [16], a biblioteca *lang-complexity* https://github.com/frserras/lang_complexity foi criada, para fazer a análise de complexidade de línguas a partir da compressão de textos para qualquer texto. O objetivo deste TCC é de adicionar robustês a essa biblioteca, refinando seu funcionamento e contribuindo para análises dos resultados da biblioteca.

Capítulo 3

Métricas de Complexidade

A métrica de Complexidade de Linguagem abordada nesse TCC é uma métrica quantitativa baseada em uma abordagem Teleológica de línguas humanas, ou seja, focada na missão das línguas de transmitir informações. De acordo com essa métrica, toda sentença carrega informação e a complexidade é a quantidade de informação que cada mensagem carrega. Para mensagens longas, a quantidade de informações pode ser aproximada pelo tamanho da mensagem quando comprimida por um bom algoritmo de compressão.

Ainda assim, a complexidade de um texto vai além da quantidade de informação transmitida, já que linguas naturais possuem meios diferentes para transmitir informações. Por exemplo, em Inuktitut, a língua dos povos Inuit [1], toda a informação de uma frase costuma ser codificada em uma única palavra, ou seja, está no aspecto morfológico da frase, enquanto que em Português a informação está dividida entre diversas palavras, isto é, ela transmite informação tanto pela morfologia das palavras quanto pelos padrões sintáticos entre elas. Para demonstrar isso, a figura 3.1 mostra a mesma frase escrita nas duas línguas, com as cores indicando significados equivalentes.

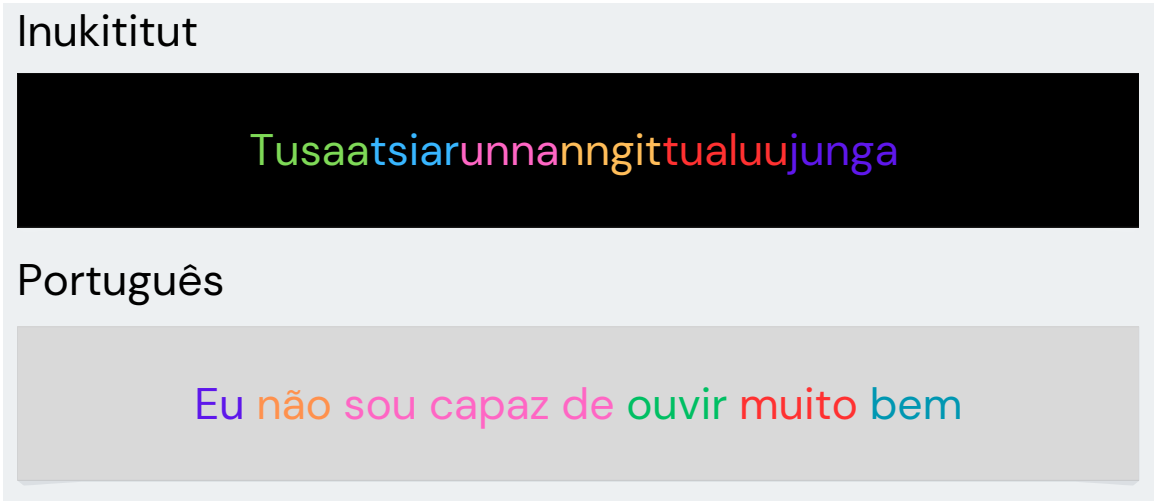


Figura 3.1: *Comparação entre Inuktitut e Português.*

Como pode ser visto, em Inuktitut a frase inteira está contida em uma palavra, enquanto que em Português a frase está dividida entre palavras diferentes. Ou seja, apesar de ambas as frases carregarem a mesma informação, o nível linguístico através do qual ela é transmitida é diferente. Isto evidencia como não é possível usar apenas uma métrica para medir a complexidade de línguas naturais.

Em termos de Teoria da Informação, as informações codificadas em linguas naturais são carregadas em diferentes níveis, sendo que cada texto possui uma complexidade diferente em cada nível e, portanto, deve ser medido em todos os níveis para se entender sua complexidade total. Para medir a complexidade geral de uma língua natural, Juola [8] propôs métricas para avaliar três níveis linguísticos, a Morfologia, Sintaxe e Pragmática. As métricas para avaliar estes três níveis consistem no mesmo princípio, degradar apenas o nível desejado e calcular a razão entre o tamanho do texto original comprimido e o texto degradado comprimido, para então saber quanta informação está sendo transmitida neste nível. A degradação experimentalmente selecionada sugerida é de 10% ([16]), pois é uma porcentagem em que a variação entre as línguas é mais notável, já que uma porcentagem menor resulta em uma degradação pequena demais, que pode afetar pouco a compressão, e uma porcentagem maior resulta em um texto degradado demais, em que o algoritmo de compressão não consegue achar padrões para comprimir.

Quanto mais dependente uma língua for de um nível específico, mais a degradação deste nível afeta a quantidade de informação perdida. Consequentemente, o algoritmo de compressão vai ter mais dificuldade em encontrar padrões para seu processo de compressão, resultando em uma compressão com tamanho maior e portanto uma métrica de complexidade maior.

A imagem 3.2 mostra o que são os três níveis de complexidade propostos e as informações que são carregadas em cada nível para a mesma frase vista em 3.1.

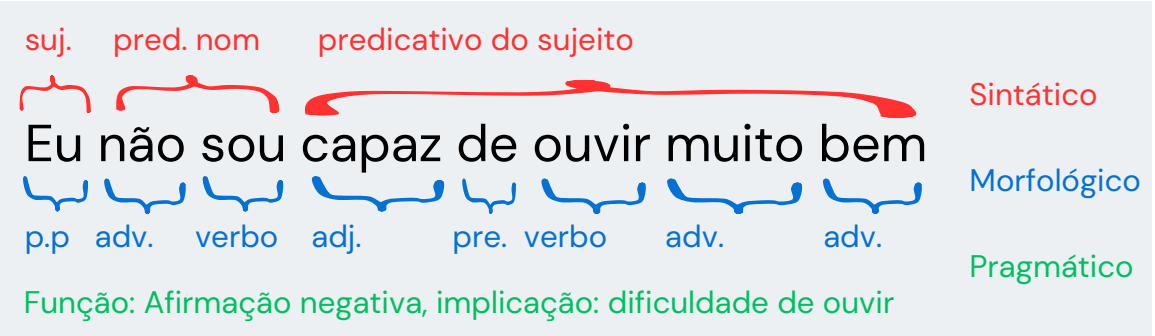


Figura 3.2: *Diferentes níveis linguísticos.*

3.1 Complexidade Morfológica

A complexidade morfológica se refere à analise de complexidade no nível morfológico de textos.

Para degradar um texto apenas no nível morfológico, o escopo da degradação deve ser a alteração de caracteres dentro de uma palavra, já que morfologia se refere ao estudo da

estrutura de palavras e suas partes, os morfemas. Logo, alterar caracteres dentro de uma palavra significa alterar a estrutura morfológica da mesma. Sendo assim, línguas que se utilizam mais de estruturas como afixos tem uma compressão de tamanho maior quando são degradadas morfologicamente, pois o sentido das palavras foi mais alterado pela perda de morfemas.

A complexidade morfológica tem uma peculiaridade, de que ao retirar caracteres de uma palavra, é possível que a palavra resultante seja também válida para a língua, o que pode fazer com que a compressão mantenha um tamanho menor de texto. Essa peculiaridade é especialmente comum para línguas mais dependentes de sistemas morfológicos. Para representar essa peculiaridade, é adicionado um sinal negativo nos cálculos de complexidade morfológica.

3.2 Complexidade Sintática

A complexidade sintática se refere à analise de complexidade no nível sintático de textos.

Para degradar um texto apenas no nível sintático, o escopo da degradação deve ser a alteração de palavras dentro de uma frase, já que sintaxe se refere à forma como as palavras se combinam para formar uma frase com sentido, ou seja, alterar palavras dentro de uma frase significa alterar a estrutura sintática da mesma. Sendo assim, línguas que se utilizam mais da relação entre palavras para expressar informações tem uma compressão de tamanho maior quando são degradadas sintaticamente, pois o sentido das frases foi mais alterado pela perda de palavras.

3.3 Complexidade Pragmática

A complexidade pragmática se refere à analise de complexidade no nível pragmático de textos.

Para degradar um texto apenas no nível pragmático, o escopo da degradação deve ser a alteração de frases dentro de um parágrafo, já que a pragmática se refere a como o contexto e a intenção do autor afeta a interpretação de um texto, ou seja, alterar frases dentro de um parágrafo significa alterar a coesão do texto, degenerando suas relações contextuais, e portanto a estrutura pragmática do mesmo. Sendo assim, línguas que dependem da associação entre elementos de diferentes sentenças para a transmissão de significado tem uma compressão de tamanho maior quando são degradadas pragmaticamente, pois o sentido dos textos foi mais alterado pela perda de frases.

Capítulo 4

A biblioteca lang-complexity

A biblioteca *lang-complexity* [2] implementa as métricas de complexidade linguística baseadas em compressão e degradação. A partir de um arquivo com o mesmo texto para varias línguas, a biblioteca mede as três complexidades de cada língua para estes textos. Além disso, é possível escolher outros parâmetros como a porcentagem de degradação, a codificação dos textos e o número de repetições que o algoritmo vai fazer. A figura 4.1 demonstra o processo que a biblioteca faz para medir a complexidade das línguas. O exemplo consiste em uma degradação morfológica, mas o processo é análogo para os outros tipos de degradação também. Em específico, a degradação vista na imagem, representada pelo texto *degrade text*, substitui letras pelo caractere _, degradando o sentido das palavras.

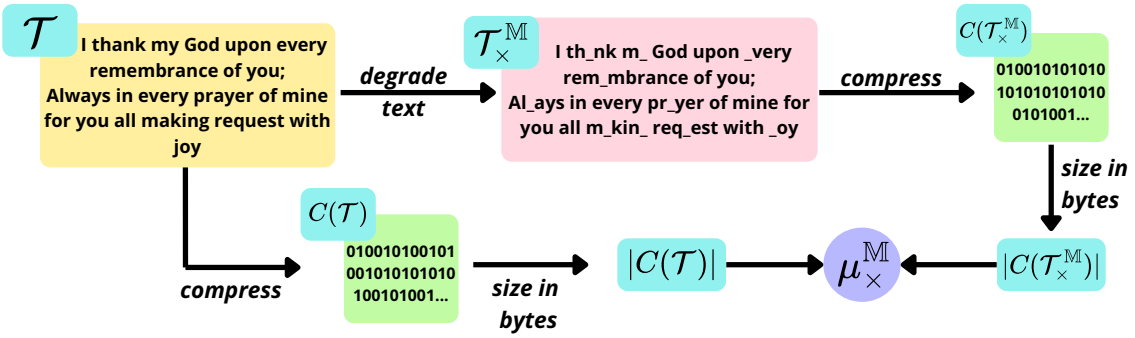


Figura 4.1: *Diagrama representando o processo da biblioteca. Imagem retirada de [16]*

Antes do começo deste trabalho, a biblioteca possuia 3 métricas para medir complexidade local, uma para cada nível, além de 4 kernels para calcular o resultado das degradações. A figura 4.2 mostra o funcionamento interno da biblioteca. O usuário da biblioteca chama a classe `DegradeAndCompress` com as classes `Degrader` e `Kernel` que deseja para fazer a degradação, além do algoritmo de compressão desejado. Então, a classe `DegradeAndCompress` chama as classes `Degrader`, `Compress` e `Kernel` em sequência, para degradar o texto, comprimir tanto o texto original quanto o degradado e comparar o tamanho dos dois textos degradados. Dentro de `Degrader`, a classe chama `Unit` para separar o texto em grupos de caracteres/palavras/frases, que ficam armazenados na clase

`ParseResult`, e em seguida `Degrader` chama `Strategy` para degradar o texto armazenado em `ParseResult` e chamar a função de `ParseResult` para reconstruir o texto sem a separação em grupos.

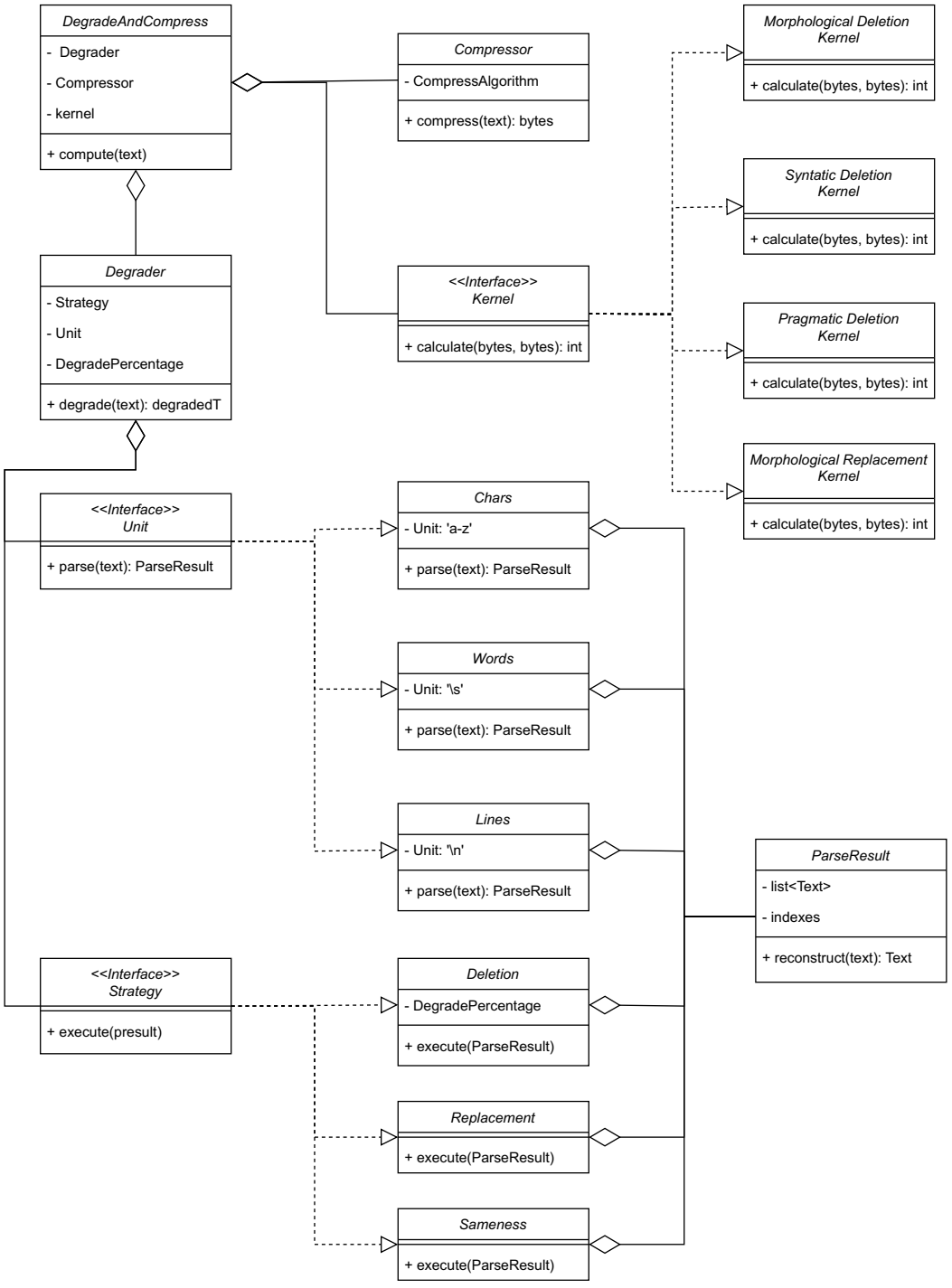


Figura 4.2: *Diagrama de classes da biblioteca inicialmente.*

Após as mudanças feitas durante o trabalho, o diagramada de classes da biblioteca ficou como na figura 4.3. Além das mudanças visíveis de a classe ter mais implementações de `Strategy` e `Kernel`, foram feitas outras mudanças que não estão visíveis no diagrama de classes. A lógica que `Unit` e `ParseResult` usa para separar o texto em caracteres/palavras/frases foi alterado para lidar com caracteres especiais e com diferentes espaçamentos (espaços, tabs ou quebras de linha) de forma mais consistente.

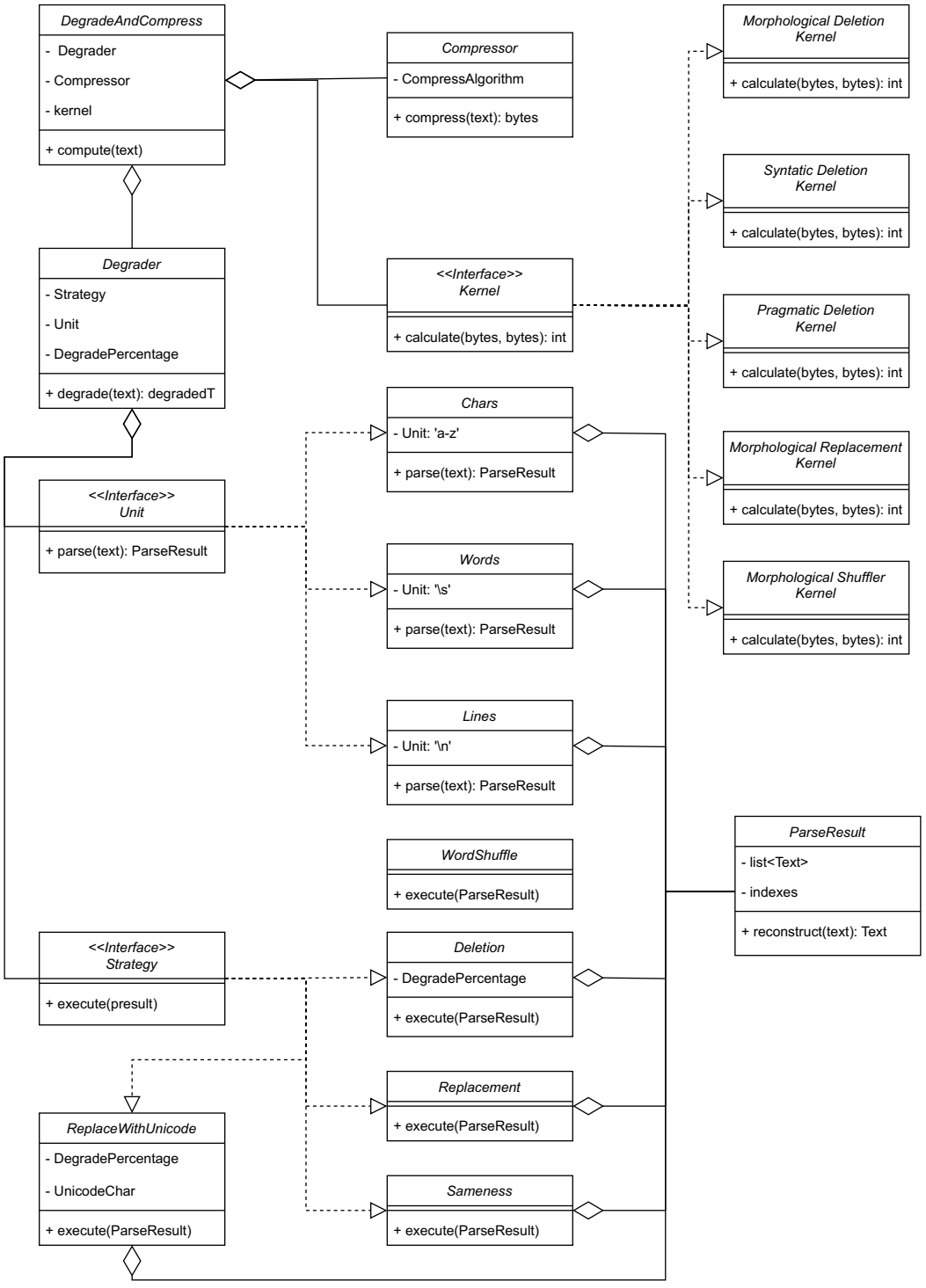


Figura 4.3: *Diagrama de classes da biblioteca no fim.*

Além dessas alterações, para garantir a consistência dos resultados da biblioteca, foram feitos maior número de testes, a geração de gráficos de regressão linear das línguas, gráficos que exibem as linguas por relações entre métricas desejadas ao invez de apenas o trade-off, o cálculo do erro padrão da média (SEM), a geração de csv que contém informações estatísticas do SEM e do erro absoluto entre a biblioteca original e com as modificações. Esses incrementos mencionados serão discutidos com mais detalhes nos próximos capítulos.

Capítulo 5

Medindo Complexidade

Para medir complexidade linguística, a biblioteca *lang-complexity* utiliza 7 métricas derivadas da degradação morfológica, sintática e pragmática. As três primeiras métricas fazem a degradação por deleção, cada uma para um nível diferente. Então, as próximas duas fazem a degradação por substituição para o nível morfológico e sintático. A sexta métrica mede a complexidade sintática embaralhando palavras para alterar o sentido das frases. Enfim, a sétima métrica mede complexidade sintática ao retirar o significado das palavras trocando elas por um número inteiro. A figura 5.1 demonstra como cada métrica funciona.

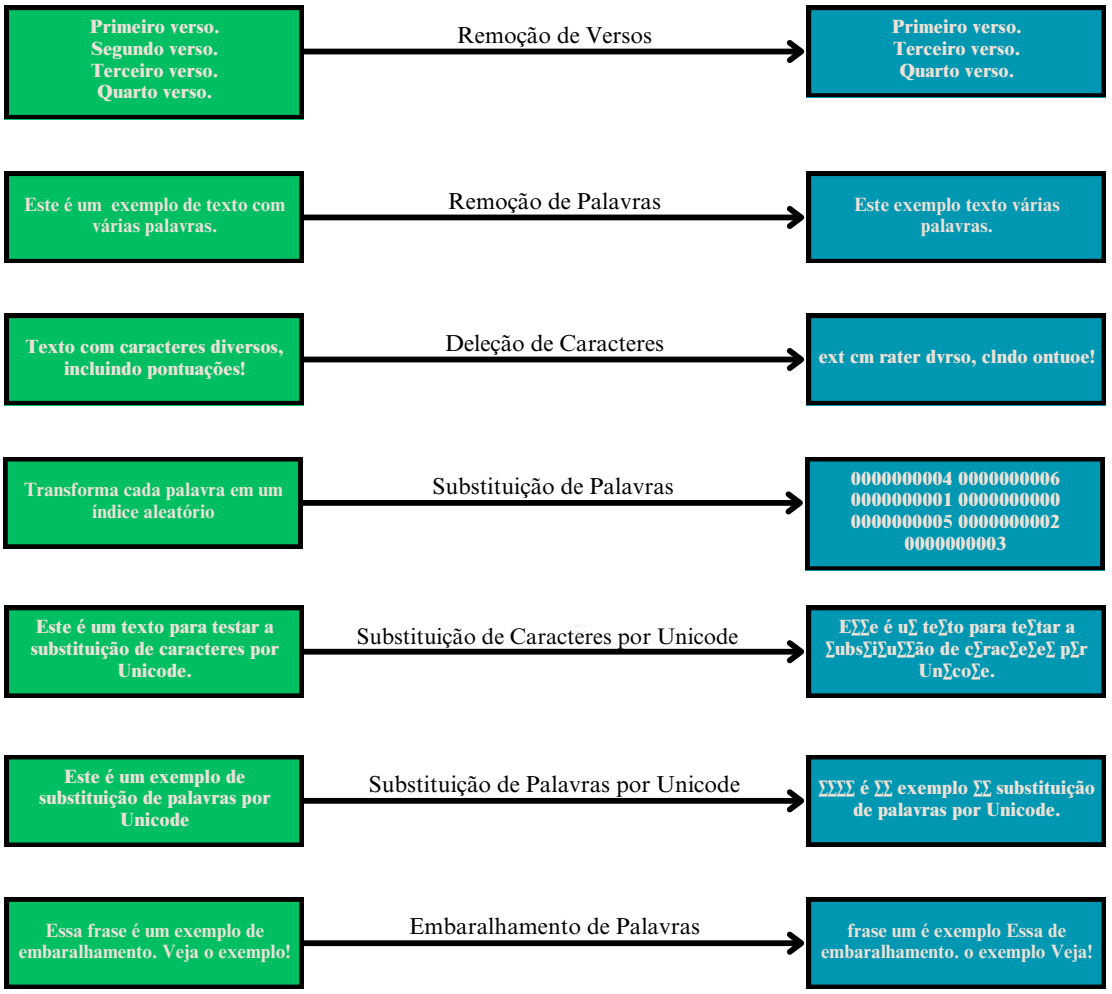


Figura 5.1: *Demonstração de como cada algoritmo degrada textos*

A existência de 7 métricas para apenas 3 níveis de complexidade leva a pergunta de por que testar mais de uma métrica para o mesmo nível. Existem dois motivos para cada nível ter mais de uma métrica. O primeiro é consistência, pois métricas diferentes medem a complexidade de um nível de uma forma ligeiramente diferente, resultando em uma cobertura maior de cada nível e portanto resultados mais precisos para a análise. O segundo motivo é analisar a diferença entre os resultados obtidos por métricas do mesmo nível e entender se existem casos em que uma métrica pode funcionar melhor para um grupo linguístico no mesmo nível.

Por exemplo, a palavra "Casar", ao sofrer uma deleção morfológica, pode ser degradada para "Casa", que é uma outra palavra existente na língua portuguesa, enquanto que ao sofrer uma substituição por unicode a mesma degradação resulta em "Casa*", que não é uma palavra. Essa diferença pode ter implicações na degradação, mas para entender melhor elas seria preciso ter um entendimento de todas as línguas degradadas e de um estudo linguístico próprio, que vai além do escopo desse TCC.

5.1 Deleção de versos

Este método visa medir a complexidade pragmática do texto. A ideia é que a remoção de versos afeta a coerência e o contexto do discurso, permitindo avaliar como a língua depende de informações contextuais para transmitir significado.

```
Exemplo de entrada:
Verso 1
Verso 2
Verso 3
Verso 4

Saída:
Verso 1
Verso 3
Verso 4
```

Para fazer uma degradação utilizando a métrica de deleção de versos, a biblioteca descrita na figura 4.3 usa a implementação de estratégia `Deletion`, de unidade `Lines` e de kernel `Pragmatic Deletion Kernel`.

5.2 Deleção de palavras

Esta técnica é usada para medir a complexidade sintática. Isso perturba a estrutura sintática das frases, permitindo avaliar como a língua depende da ordem das palavras e de elementos gramaticais para transmitir significado.

```
Exemplo de entrada: "Este é um exemplo de texto com várias palavras."
             Saída: "Este um texto várias palavras."
```

Para fazer uma degradação utilizando a métrica de deleção de palavras, a biblioteca descrita na figura 4.3 usa a implementação de estratégia `Deletion`, de unidade `Words` e de kernel `Syntactic Deletion Kernel`.

5.3 Deleção de caracteres

Esta técnica é utilizada para medir a complexidade morfológica. Isso perturba a estrutura interna das palavras, permitindo avaliar como a língua depende da morfologia para transmitir significado.

```
Exemplo de entrada: "Texto com caracteres diversos, incluindo pontuação!"
             Saída: "txt cm rateres dverso, clndo ontuo!"
```

Para fazer uma degradação utilizando a métrica de deleção de caracteres, a biblioteca descrita na figura 4.3 usa a implementação de estratégia `Deletion`, de unidade `Chars` e de kernel `Morphological Deletion Kernel`.

5.4 Substituição de palavras por sequência de caracteres unicode

Nesta técnica, cada palavra inteira é substituída por um único caractere unicode. O propósito é medir o impacto da remoção de informação no nível das palavras na compressibilidade do texto, mantendo o mesmo número de "tokens". Isso permite uma avaliação mais precisa da complexidade linguística, isolando o efeito da perda de informação lexical das mudanças no tamanho bruto do texto.

```
Exemplo de entrada: "Este é um exemplo de substituição."
             Saída: "Este é ** exemplo ** ************"
```

Para fazer uma degradação utilizando a métrica de substituição de palavras por unicode, a biblioteca descrita na figura 4.3 usa a implementação de estratégia `ReplaceWithUnicode`, de unidade `Words` e de kernel `Syntactic Deletion Kernel`.

5.5 Substituição de caracteres por caracteres unicode

Este método envolve a substituição de caracteres individuais por caracteres unicode específicos. O objetivo é isolar o efeito da remoção de informação no nível de caracteres na compressibilidade do texto, sem alterar seu tamanho bruto.

```
Exemplo de entrada: "Este é um texto para testar."
             Saída: "Es⋆e ⋆ ⋆m te⋆⋆⋆ ⋆ara tes⋆a⋆."
```

Para fazer uma degradação utilizando a métrica de substituição de caracteres por unicode, a biblioteca descrita na figura 4.3 usa a implementação de estratégia `ReplaceWithUnicode`, de unidade `Words` e de kernel `Morphological Deletion Kernel`.

5.6 Embaralhamento de palavras

Este método envolve a reorganização aleatória da ordem das palavras dentro de cada frase ou verso. Ele visa medir como a ordem das palavras afeta a complexidade e a compreensibilidade do texto, permitindo avaliar a importância da ordem das palavras na estrutura da língua.

```
Exemplo de entrada: "Essa frase é um exemplo de embaralhamento."
             Saída: "frase um é exemplo Essa o embaralhamento."
```

Para fazer uma degradação utilizando a métrica de embaralhamento de palavras, a biblioteca descrita na figura 4.3 usa a implementação de estratégia `WordShuffle`, de unidade `Words` e de kernel `Morphological Shuffler Kernel`.

5.7 Troca de palavras

Este método envolve a substituição de cada palavra única no texto por um número inteiro único de 10 dígitos, mantendo a consistência ao longo do texto. O processo usa um dicionário para mapear cada palavra para um número específico. Palavras idênticas são substituídas pelo mesmo número, preservando a estrutura de repetição do texto. Esta técnica mantém a ordem e a distribuição das palavras, mas remove o conteúdo lexical. Isso permite avaliar como a língua depende da estrutura sintática e da ordem das palavras para transmitir informação, independentemente do significado específico de cada palavra.

```
Exemplo de entrada: "Transforma cada palavra em um índice."
Saída: "0000000005 0000000002 0000000001 0000000003 0000000004 0000000006"
```

Para fazer uma degradação utilizando a métrica de troca de palavras, a biblioteca descrita na figura 4.3 usa a implementação de estratégia `Replacement`, de unidade `Words` e de kernel `Morphological Replacement Kernel`.

5.8 Degradação Nula

Esse método passa por todo o processo da biblioteca sem fazer nenhuma degradação. Ele serve de controle para os experimentos, como uma garantia de que os resultados são os esperados.

Para fazer uma degradação utilizando a métrica de não fazer nada, a biblioteca descrita na figura 4.3 usa a implementação de estratégia `Sameness`, com a implementação de unidade e kernel que o usuário queira testar.

Capítulo 6

Dados

Para aumentar a robustês do projeto, a primeira etapa do processo foi a coleta de dados para testar a biblioteca. A seleção dos textos seguiu alguns critérios para garantir a qualidade e a relevância dos dados para os objetivos do projeto:

1. **Variedade de línguas:** Foi priorizada a escolha por textos que estavam disponíveis em múltiplos idiomas. Isso permitiria a análise comparativa entre diferentes línguas e possibilitou a aplicação das métricas em um conjunto diversificado e representativo. Em específico, foi buscado abrangir línguas originadas de todos os continentes, para testar se a análise da biblioteca se mantém válida para línguas com origens diferentes e entender se as métricas de complexidade linguística se mantém válidas além das línguas indo-europeias sobre as quais as métricas foram pensadas.
2. **Existência de traduções equivalentes:** Um dos critérios essenciais foi identificar textos que existissem em versões equivalentes em diversas línguas. Essa característica é importante para validar propriedades universais, eliminando interferências causadas por diferenças no gênero textual ou no propósito de comunicação.
3. **Estrutura textual:** Textos com estrutura sequencial bem definida foram a prioridade. Essa escolha se justifica por facilitar a análise da coesão e da progressão de informações, elementos importantes para avaliar a complexidade linguística, em especial as métricas de complexidade pragmática. Também foram escolhidos textos que não possuem uma estrutura textual clara, pois seriam interessantes para verificar como esse tipo de texto afeta a complexidade.
4. **Variedade de tamanhos de arquivos:** A diversidade no tamanho dos textos também foi considerada, uma vez que diferentes tamanhos podem impactar as métricas de compressão utilizadas. Esse aspecto ajuda a garantir que os resultados sejam robustos e não enviesados por textos extremamente curtos ou longos, além de possibilitar um estudo do efeito do tamanho dos textos sobre a robustez das métricas.
5. **Licença aberta:** Os dados precisam ter uma licença aberta de uso para que possam ser utilizados para testar a biblioteca com a autorização dos autores ou detentores dos textos.

Coletar dados que sigam todas estas especificações é algo complicado, pois os *datasets*

que possuem o mesmo texto para diversas línguas são poucos. Os dados que seguiam essas especificações estavam principalmente no corpus OPUS (https://opus.nlpl.eu), uma das maiores coleções de corpora multilíngues disponíveis para a comunidade científica. O OPUS organiza textos em diversos formatos, de forma padronizada e com licença aberta para uso acadêmico. Os dados do OPUS são amplamente utilizados em PLN, pois oferecem textos traduzidos em várias línguas, facilitando análises comparativas e o desenvolvimento de modelos computacionais, em especial modelos de tradução automática.

A justificativa para o uso do OPUS e desses critérios de seleção está na ampla gama de diferentes famílias linguísticas e graus de complexidade estrutural. A presença de traduções paralelas garante homogeneidade nos conteúdos analisados, permitindo que os resultados reflitam diferenças em nível de linguagem e não de conteúdo. O foco em textos estruturados e variados em extensão assegura que as métricas aplicadas sejam consistentes e generalizáveis.

Ao decorrer do desenvolvimento do projeto, foi notória a necessidade de tratamento extensivo nos dados do OPUS, que demanda um esforço significativo para atender aos requisitos do projeto. Com isso, foi utilizado o conjunto privado fornecido pela IBM que também foi usado na biblioteca original *lang-complexity*, composto por traduções da Bíblia, incluindo um grande número de idiomas indígenas sul-americanos. A Bíblia apresenta traduções consistentes em várias línguas, além de uma divisão clara e bem estruturada em livros, capítulos e versículos, facilitando sua manipulação e análise. Apesar do conjunto ser privado, o uso dele na biblioteca *lang-complexity* foi explicitamente autorizado, desde que os textos não sejam compartilhados, apenas os resultados obtidos a partir deles.

Essa escolha permitiu expandir as análises realizadas, abrangendo novas perspectivas sobre as métricas de complexidade linguística. É importante ressaltar que muitas das línguas indígenas incluídas foram historicamente traduzidas para que fossem compreensíveis ao "homem branco", sem levar em consideração a cultura destes povos, uma vez que essas línguas eram predominantemente orais e não possuíam um sistema de escrita próprio. Essa característica pode influenciar significativamente a forma como essas línguas se comportam em termos de complexidade textual. Embora esse fenômeno seja interessante e digno de maior exploração, ele remete a questões de linguística que vão além do escopo deste TCC.

Capítulo 7

Testes

Este capítulo apresenta a estratégia adotada para validar e comparar as biblioteca *lang-complexity* com o programa do qual ela foi originada. As validações foram realizadas por meio de diferentes tipos de testes: unitários, funcionais e A/B, cujos detalhes são descritos nas seções a seguir.

7.1 Testes Unitários

Os testes unitários foram projetados para validar o comportamento correto de componentes fundamentais das bibliotecas, como parsers que operam em caracteres, palavras e linhas. Cada teste segue uma lógica geral que envolve três etapas principais: primeiro, a string de entrada é analisada pelo parser correspondente, que divide o texto com base na unidade alvo (caracteres, palavras ou linhas). Em seguida, a string processada é reconstruída, unindo novamente as unidades. Por fim, são realizadas verificações para garantir que o texto reconstruído possui o mesmo comprimento e conteúdo do texto original, confirmando a preservação da integridade durante o processamento.

Por exemplo, no teste do parser `Chars`, a string de entrada é composta apenas por caracteres alfabéticos. O objetivo é garantir que todos os caracteres sejam processados individualmente e que a reconstrução do texto mantenha o conteúdo original. Já no caso do parser `NotChar`, utilizado para ignorar delimitadores como espaços ou quebras de linha, os testes verificam se as unidades delimitadas por esses caracteres são tratadas corretamente e se a reconstrução da string preserva sua estrutura.

Esses testes cobrem cenários essenciais para avaliar a robustez dos parsers, incluindo a manipulação de diferentes tipos de delimitadores e a consistência no processamento de textos com diversas estruturas. A abordagem sistemática garante que as operações fundamentais da biblioteca sejam confiáveis, estabelecendo uma base sólida para os testes funcionais e comparativos realizados posteriormente.

7.2 Testes Funcionais

Os testes funcionais foram importantes para identificar e corrigir possíveis desvios ou inconsistências, assegurando que as estratégias de degradação implementadas fossem aplicáveis ao objetivo do projeto, com maior precisão. Implementados no arquivo `test_degrader.py`, esses testes verificam estratégias de degradação de texto, como remoção, substituição e embaralhamento de unidades textuais. Cada teste foi estruturado para avaliar a consistência, robustez e aderência às expectativas de comportamento das bibliotecas. De forma geral, os testes funcionais seguem um fluxo padrão: primeiro, um texto de entrada é submetido a uma estratégia específica de degradação. Em seguida, os resultados são analisados para garantir que as alterações aplicadas ao texto respeitem os parâmetros definidos, como percentuais de remoção ou padrões de substituição. Por último, verificações adicionais asseguram que elementos fundamentais, como o comprimento ou a estrutura do texto, permaneçam consistentes com o esperado.

As estratégias avaliadas incluem a remoção de caracteres, palavras e linhas, onde o teste verifica se o percentual de remoção aplicado condiz com o configurado. Por exemplo, no caso de remoção de caracteres, se um texto contém 100 caracteres e o percentual de remoção configurado é de 10%, espera-se que o texto resultante tenha aproximadamente 90 caracteres. Testes semelhantes foram realizados para palavras e linhas, garantindo que os resultados mantivessem uma correspondência direta com os parâmetros fornecidos.

Além disso, a substituição de caracteres e palavras foi testada para validar a aplicação de estratégias como a substituição por caracteres Unicode ou índices aleatórios. Esses testes verificam se os elementos substituídos correspondem aos critérios definidos, como manter o comprimento das palavras originais ou preservar a pontuação do texto. Em um exemplo, palavras aleatórias em uma frase foram substituídas por caracteres Unicode, e verificou-se se o número de palavras substituídas correspondia ao percentual configurado, enquanto a estrutura geral do texto era mantida. Outra funcionalidade avaliada foi o embaralhamento de palavras em um texto. Nesse caso, os testes confirmaram que todas as palavras originais estavam presentes no texto embaralhado, mas em uma ordem diferente, garantindo que a integridade do conteúdo fosse preservada.

7.3 Testes A/B

Testes A/B são testes feitos para comparar o funcionamento de dois programas similares. Os testes A/B foram realizados para comparar diretamente o desempenho da biblioteca *lang-complexity* com o do programa do qual ela foi originada. Esses testes focaram na equivalência funcional e na análise de melhorias específicas introduzidas na biblioteca, garantindo que ela mantivesse a funcionalidade do programa e incorporasse avanços consistentes. Os testes foram conduzidos com base em um conjunto padronizado de textos que incluíam diferentes tipos de estruturas e características, como espaçamentos irregulares, tabs, caracteres especiais e combinações de quebra de linha. Para garantir a reprodutibilidade dos experimentos, utilizamos sementes fixas na geração de valores aleatórios, permitindo que ambas as bibliotecas fossem testadas sob as mesmas condições.

O fluxo dos testes consistiu em aplicar as mesmas estratégias de degradação em ambas

as bibliotecas e comparar os resultados obtidos. Estratégias como remoção de versos, palavras e caracteres, conversão de palavras para índices aleatórios e embaralhamento foram cuidadosamente analisadas. Por exemplo, ao testar a remoção de palavras, foi verificado se ambas as versões removiam o mesmo número de palavras e se os textos resultantes apresentavam estruturas equivalentes. Similarmente, no teste de conversão de palavras para índices, os índices gerados pelas duas versões foram comparados para garantir consistência.

Os resultados dos testes A/B demonstraram que a biblioteca manteve o comportamento esperado do programa original na maioria dos casos, garantindo compatibilidade. Além disso, a biblioteca apresentou melhorias em situações específicas, como no tratamento de textos com espaçamentos ou tabs no início e no final, onde o programa original poderia apresentar inconsistências. Por fim, a biblioteca também demonstrou maior robustez ao lidar com textos contendo caracteres especiais e estruturas mais complexas.

Capítulo 8

Hipótese de Complexidade Linguística

A partir das métricas de complexidade de línguas medidas pela compressão de textos, Juola [8] analisa 3 hipóteses sobre como as métricas deveriam comportar, além de 4 observações sobre os resultados obtidos com suas métricas. Identificar como o algoritmo da biblioteca *lang-complexity* se comporta para essas hipóteses e observações é importante para medir a corretude do algoritmo e para verificar a validade das métricas de Juola para além das línguas indo-europeias.

Como citado em 6, o *dataset* utilizado para testar a biblioteca e medir sua corretude foi o *dataset* da bíblia para línguas indigenas. O *dataset* foi escolhido porque possui uma quantidade grande de línguas em que o mesmo texto foi escrito, o que permite uma comparação mais rica e facilita a localização de erros durante o desenvolvimento da biblioteca. Além disso, o trabalho original que originou a biblioteca *lang-complexity* [16] foi feito a partir destes dados, ou seja, já se tem os resultados esperados destes textos, o que serve para garantir que as mudanças feitas no desenvolvimento estão de acordo com o esperado. O *dataset* possui dois grupos de texto, o D90 e o Dall. O D90 são os textos com as línguas que possuem pelo menos 90% dos versículos da bíblia traduzidos, enquanto que o Dall são os textos com todas as línguas que possuem algum versículo da bíblia traduzidos.

8.1 Hipótese 1: Complexidade menor para línguas originais

A primeira hipótese se refere a comparação da complexidade de um texto em sua língua original com o mesmo texto em traduções. A hipótese alega que a complexidade deve ser menor na língua original, pois as traduções adicionam clarificações culturais para manter o sentido do texto, como a explicação de expressões idiomáticas.

Como o *dataset* utilizado foi o da bíblia, em que não se tem um exemplar original, a análise sobre esta hipótese não foi feita.

8.2 Hipótese 2: Equicomplexidade das línguas

A segunda hipótese se refere a ideia de que todas as línguas possuem uma complexidade parecida, alterando apenas em qual nível a complexidade está. Esta hipótese tem relevância em mostrar como as línguas possuem um nível de riqueza parecido e que não existe uma língua inerentemente com mais capacidade comunicacional do que outras.

Ao fazer a análise desta hipótese utilizando o *dataset* da bíblia, os resultados corroboram a hipótese na comparação entre diferentes famílias linguísticas indígenas e línguas indo-europeias, o que indica que as línguas possuem uma complexidade geral parecida.

8.3 Hipótese 3: *Trade-off* Sintático-Morfológico

A terceira hipótese se refere ao *Trade-off* entre a sintaxe e a morfologia das línguas. Essa hipótese sugere que grande parte da diferença sobre em qual nível cada língua carrega cada informação está nos níveis sintático e morfológico. Isso deve acontecer porque a diferença entre as línguas deve estar na forma como ela é escrita (complexidade sintática e morfológica), não na mensagem que ela transmite (complexidade pragmática), e como a seção 8.3 mostra, as línguas possuem uma complexidade total parecida, então o *Trade-off* sintático-morfológico deve acontecer. A figura 8.1 mostra esse *Trade-off* para o *dataset* da bíblia para a biblioteca *lang-complexity* atual.

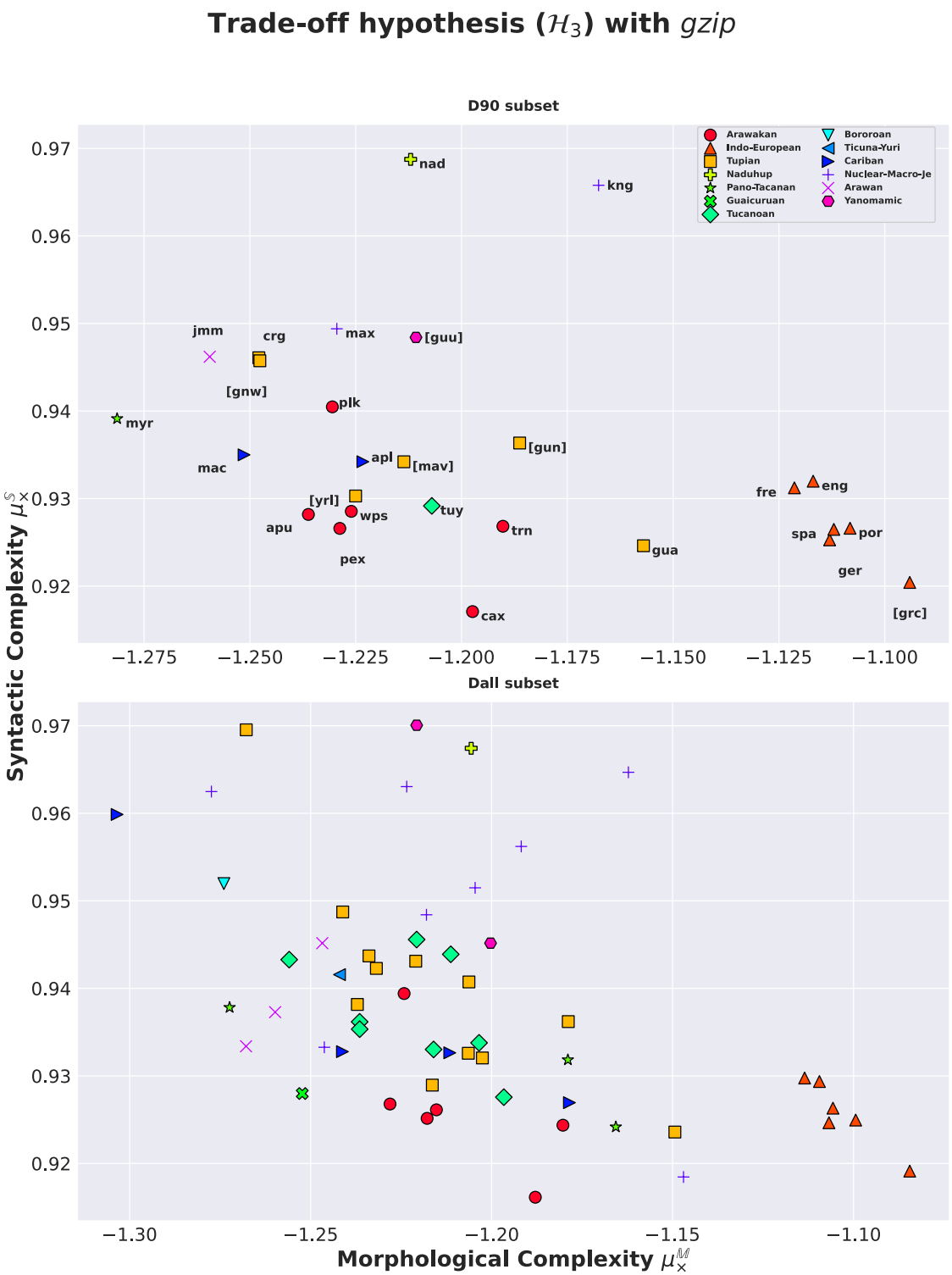


Figura 8.1: *Trade-off Sintático-Morfológico para o algoritmo de compressão gzip e estratégia de deleção*

Como pode ser visto na figura 8.1, a diferença na complexidade sintática e morfológica das línguas é pequena e, de modo geral, as línguas com maior complexidade sintática possuem menor complexidade morfológica e o contrário também é verdade. Além disso, em grande maioria as famílias linguísticas possuem maior proximidade, como pode ser visto nas línguas indo-europeias (legenda de um triângulo vermelho na figura 8.1), que são as com maior complexidade morfológica e menores complexidades sintáticas. Outro ponto interessante é como as línguas indígenas possuem maior proximidade entre si do que com as línguas indo-europeias, o que faz sentido ao analisar as culturas das quais as línguas vieram, pois os povos indígenas interagiram entre si por milhares de anos antes da vinda dos povos europeus às Americas, logo esse contato que os povos tem entre si é refletido na construção de suas línguas e portanto no seu *Trade-off* sintático-morfológico. Mesmo que dois povos não tenham tido contato direto entre si, durante os milhares de anos eles tiveram contato indireto, o que justifica a sua proximidade no *Trade-off*.

8.4 Observações

Juola fez quatro observações sobre as métricas após fazer seus experimentos, dos quais duas são interessantes para o contexto da biblioteca. A primeira observação é que todas as línguas são aproximadamente iguais em sua complexidade pragmática, ou seja, a variação entre línguas na complexidade pragmática é significativamente menor que nos outros níveis,o que está de acordo com a seção 8.3. A figura 8.2 mostra a diferença na variação de valores entre Complexidade Sintática e Complexidade Pragmática. Note que o eixo X e o Y possuem o mesmo tamanho, mas a variação dos pontos no eixo Y é muito maior do que no X, ou seja, a observação se mantém válida para as línguas indígenas.

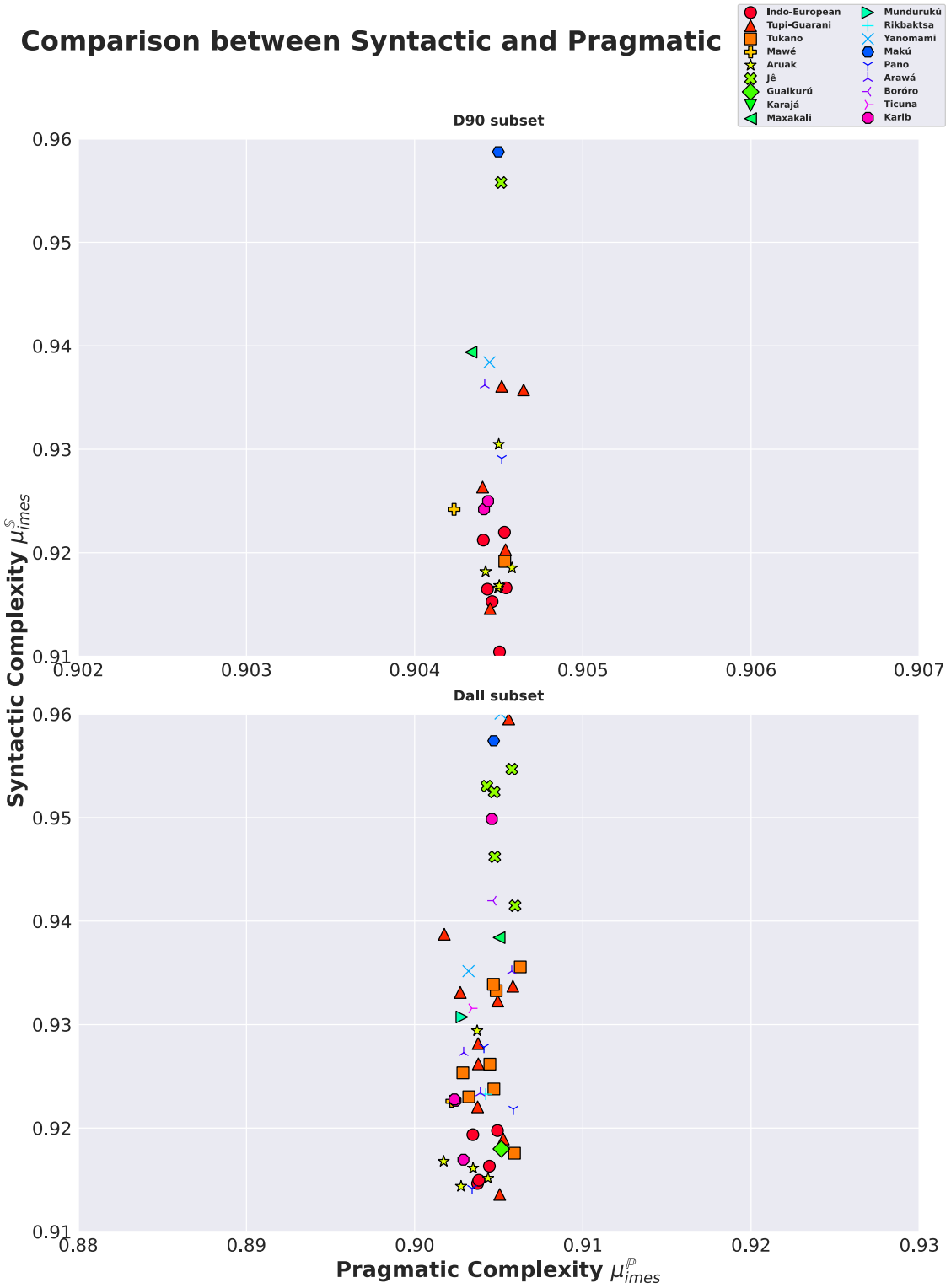


Figura 8.2: *Comparação entre a variação na Complexidade Sintática e Morfológica.*

A segunda observação é que os resultados obtidos são equivalentes ao mudar o algoritmo de compressão utilizado, o que valida a métrica, pois indica que a métrica é válida independentemente do método como os arquivos são comprimidos. A figura 8.3 mostra como o *Trade-off* ficou para o algoritmo de compressão bz2. É possível ver que, embora o valor numérico das complexidades tenha mudado em comparação com 8.1, o que é de ser esperado já que o algoritmo comprime de uma forma diferente, a distribuição dos pontos de cada língua e as localizações relativas das línguas se mantiveram.

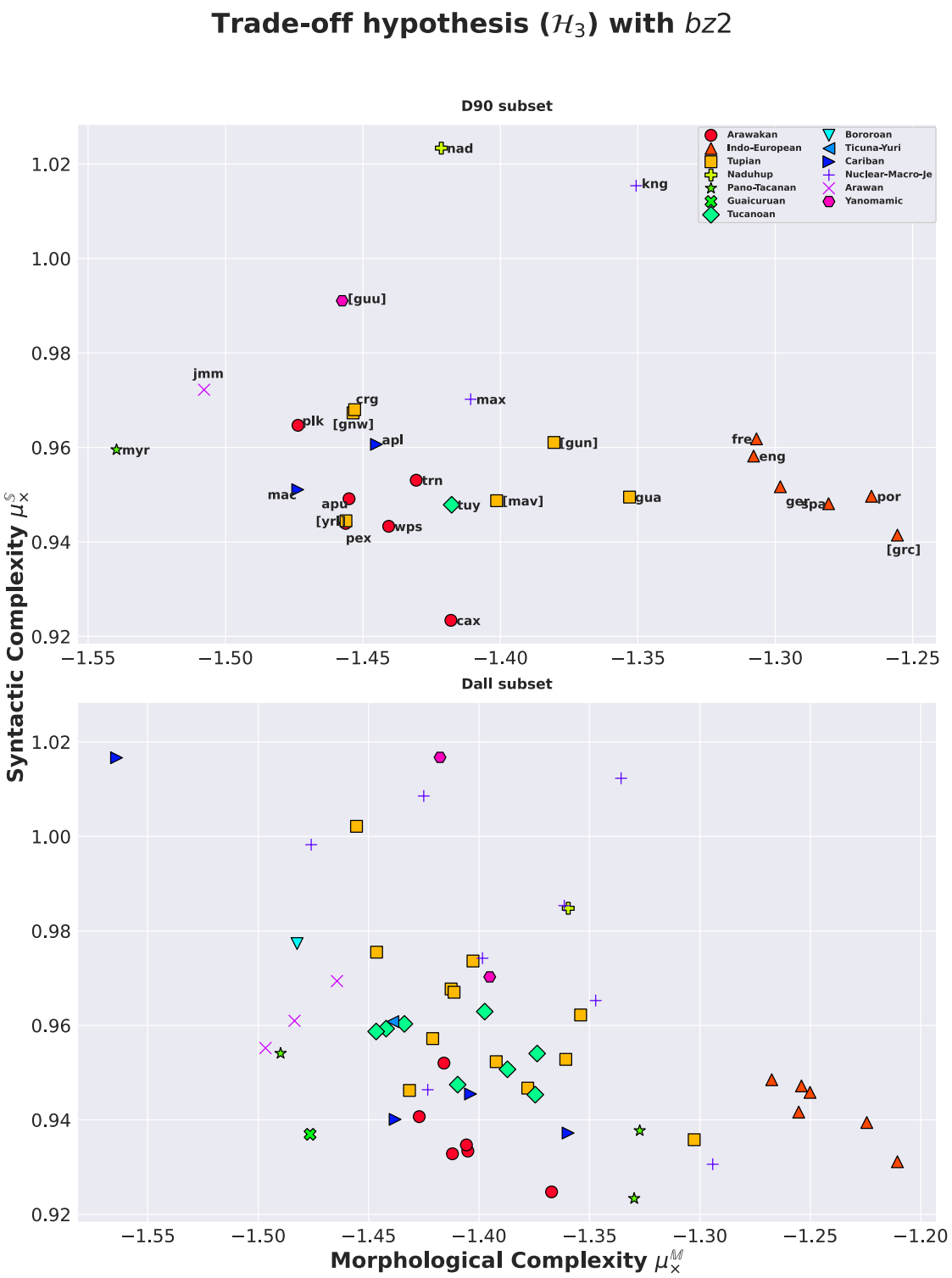


Figura 8.3: *Trade-off Sintático-Morfológico para o algoritmo de compressão bz2*

8.5 Comparações entre métricas

Como a biblioteca *lang-complexity* testa métricas diferentes para o mesmo nível de complexidade, como explicado no capítulo 5, pode-se observar resultados interessantes a partir das comparações entre as métricas.

Primeiramente, a figura 8.1 calcula a hipótese de *Trade-off* utilizando as métricas que usam a estratégia de deleção. A figura 8.4 mostra a mesma hipótese para a estratégia de substituição por unicode. Embora os valores absolutos tenham mudado, pois a degradação por unicode comprimida tem um tamanho diferente da degradação por deleção comprimida, as posições relativas se mantiveram, o que indica que a hipótese de *Trade-off* não é dependente de uma estratégia de degradação especifica.

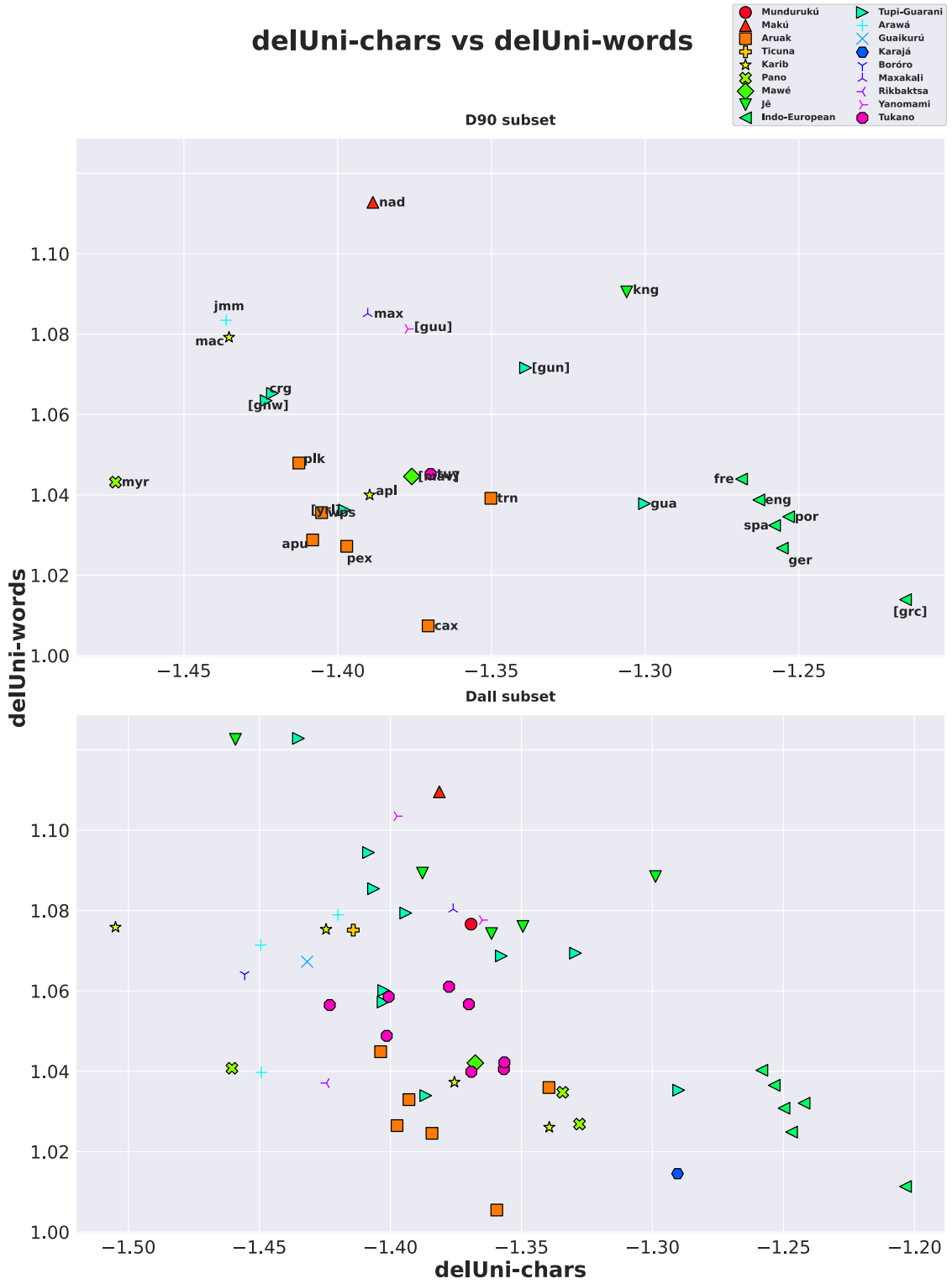


Figura 8.4: *Trade-off Sintático-Morfológico para o algoritmo de compressão gzip e estratégia de substituição por unicode*

Aprofundando, a comparação entre os resultados da mesma degradação utilizando estratégias diferentes evidencia como os resultados obtidos não são dependentes da estratégia de degradação utilizada. A figura 8.5 mostra a semelhança dos resultados para estratégias diferentes de degradação morfológica e a figura 8.6 faz o mesmo para degradação sintática.

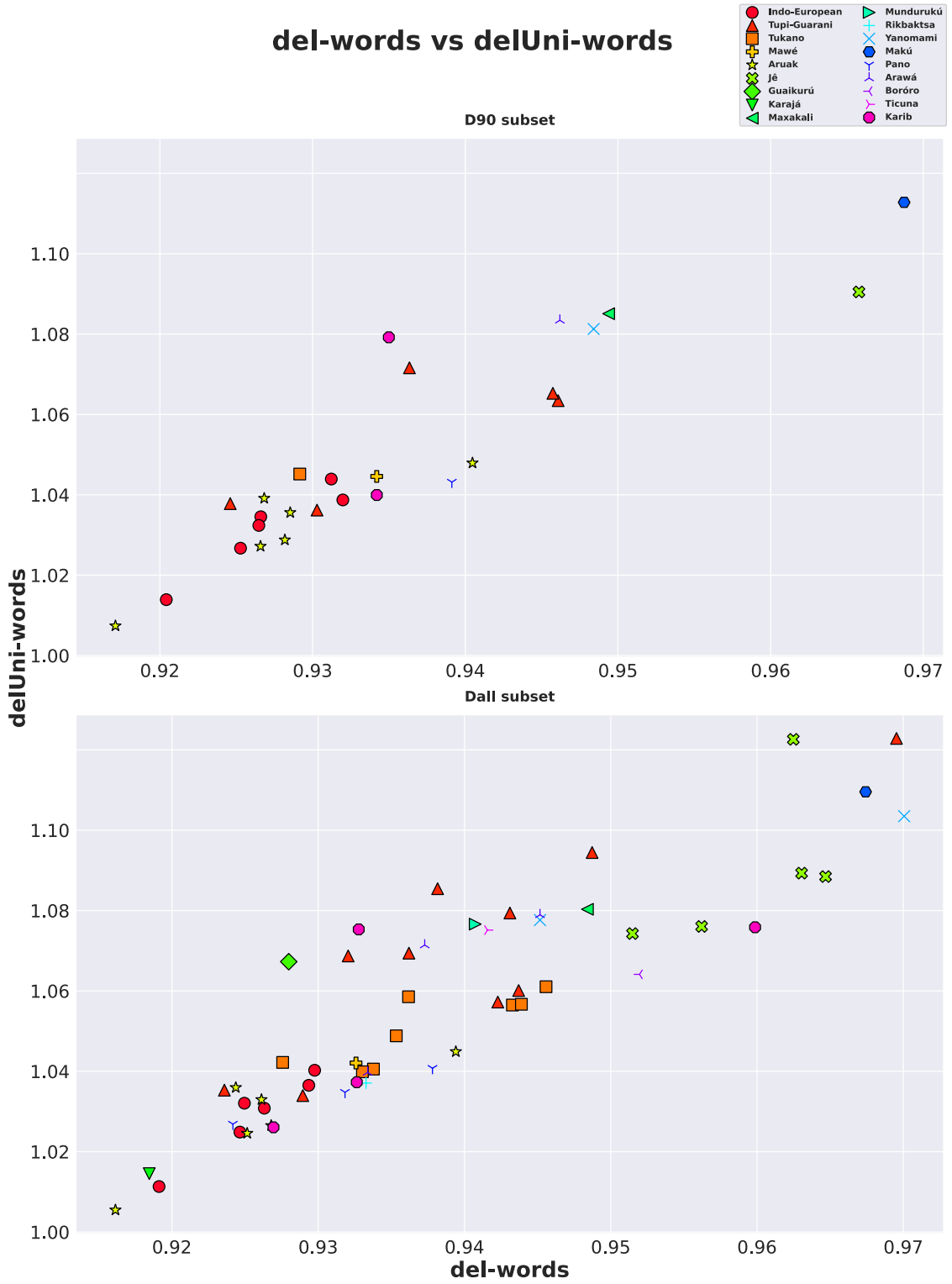


Figura 8.5: *Comparação dos resultados de degradação sintática para estratégias de deleção e substituição por unicode*

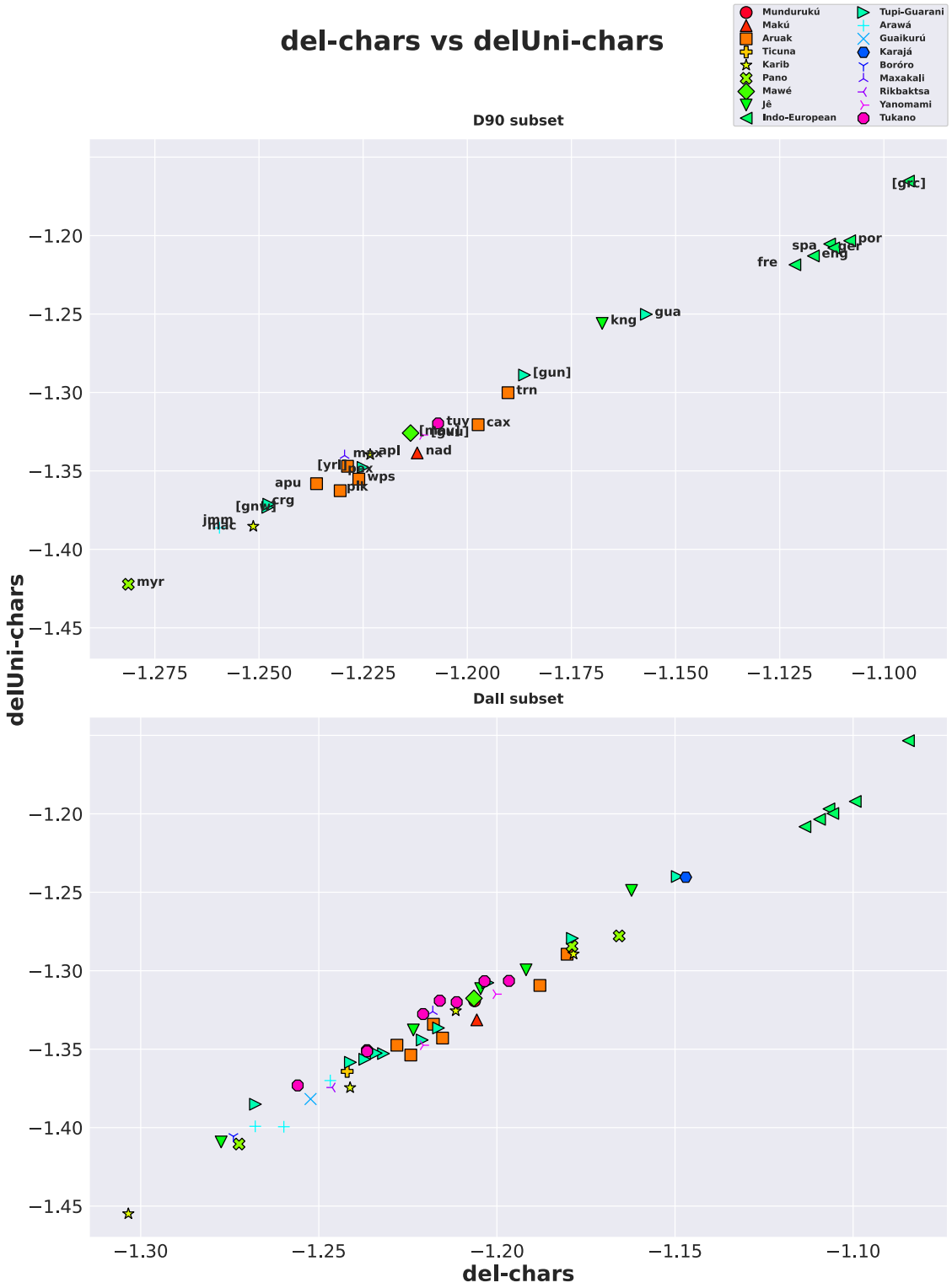


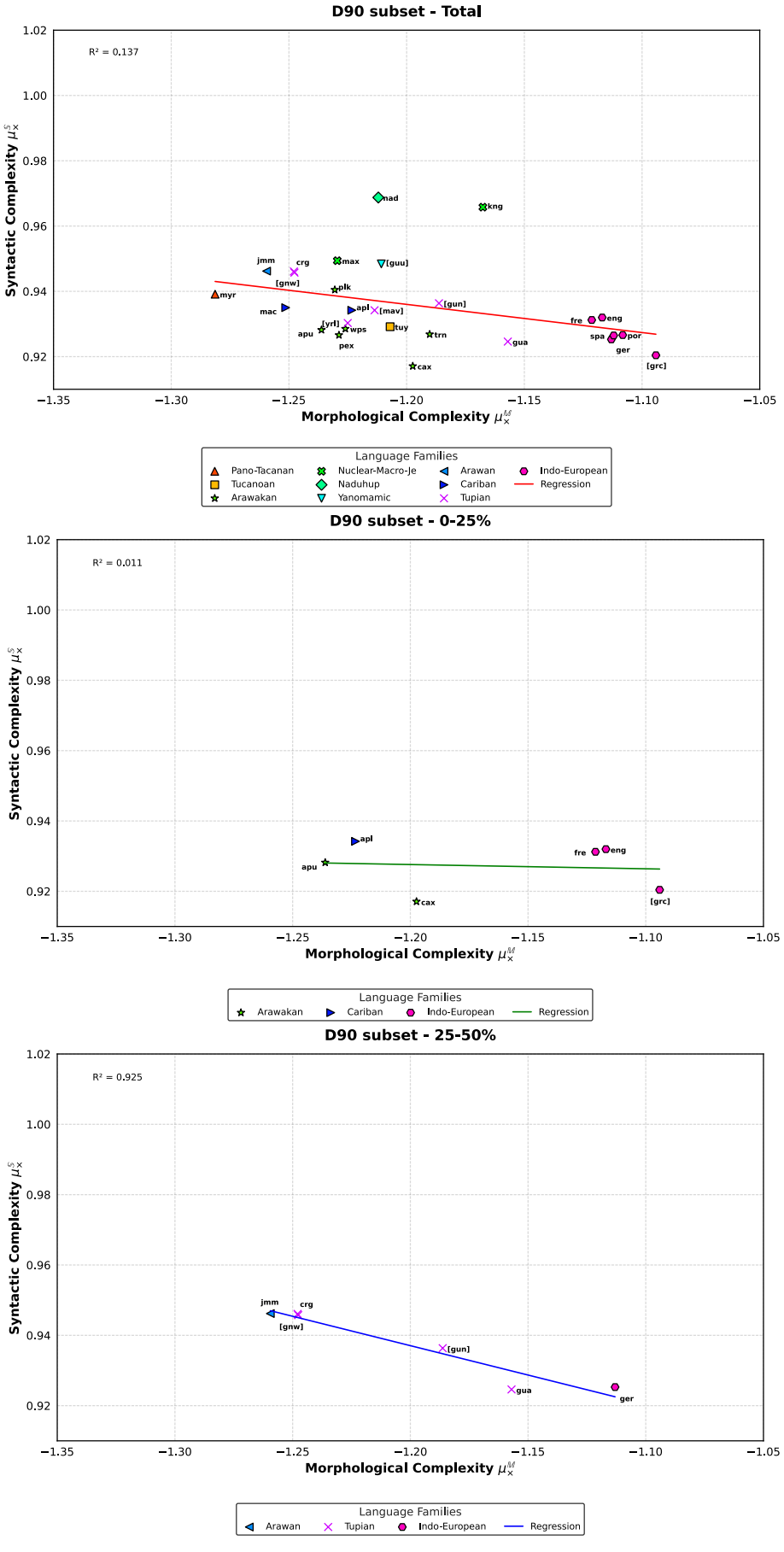
Figura 8.6: *Comparação dos resultados de degradação morfológica para estratégias de deleção e substituição por unicode*

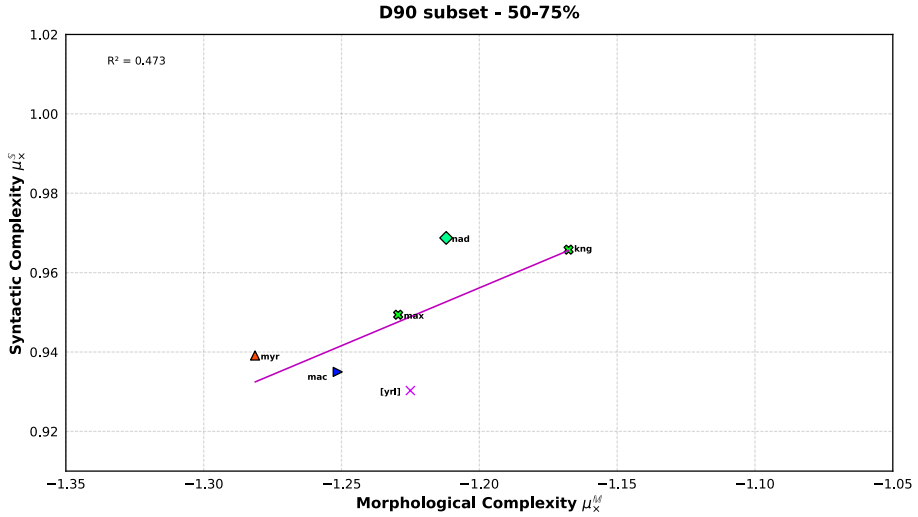
Capítulo 9

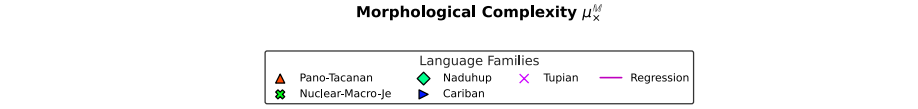
Trade-off Sintático-Morfológico para famílias linguísticas

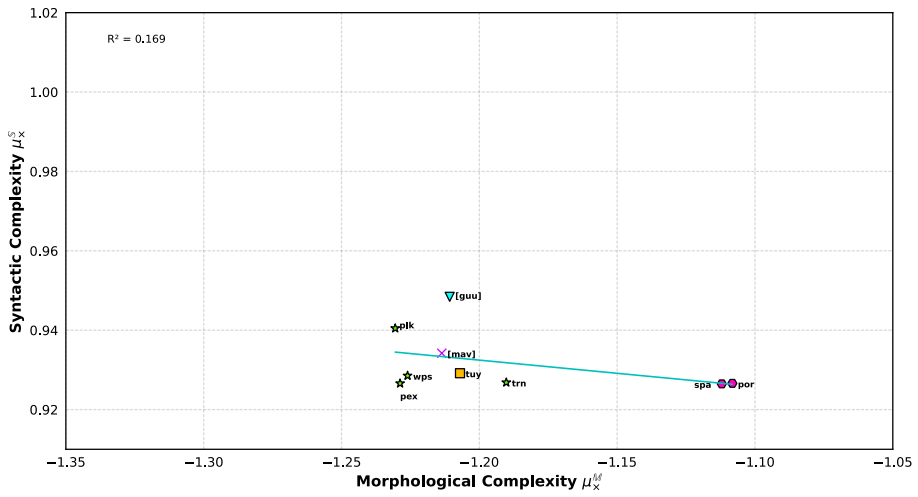
A hipótese de *trade-off* apresentada na seção 8.3 mostrou que existe uma certa correlação entre a complexidade sintática e morfológica que se aplica a todas as línguas. No entanto, essa correlação ainda é apenas uma tendência, existindo línguas com maior complexidade tanto sintática quanto morfológica que outras. Logo, a ideia de identificar subgrupos para os quais essa correlação seja mais forte foi testada.

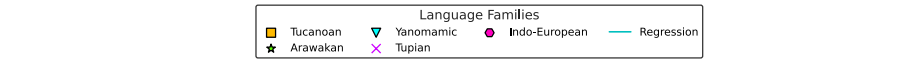
A primeira hipótese levantada foi de separar as línguas baseadas no tamanho original de cada texto. Essa hipótese foi levantada pensando em entender se o comportamento dos textos quando comprimidos está relacionado com o tamanho original dos textos. Alguns dos resultados se mostraram inconsistentes, demonstrando em alguns casos que não existe uma correlação entre o tamanho original de um texto e seu *trade-off*. As figuras abaixo representam a regressão linear para cada quartil dos textos quando separados pelos seus tamanhos. O valor de R^2 para alguns casos foi baixo, indicando que a variação no tamanho do texto não explica bem a correlação entre as variáveis.

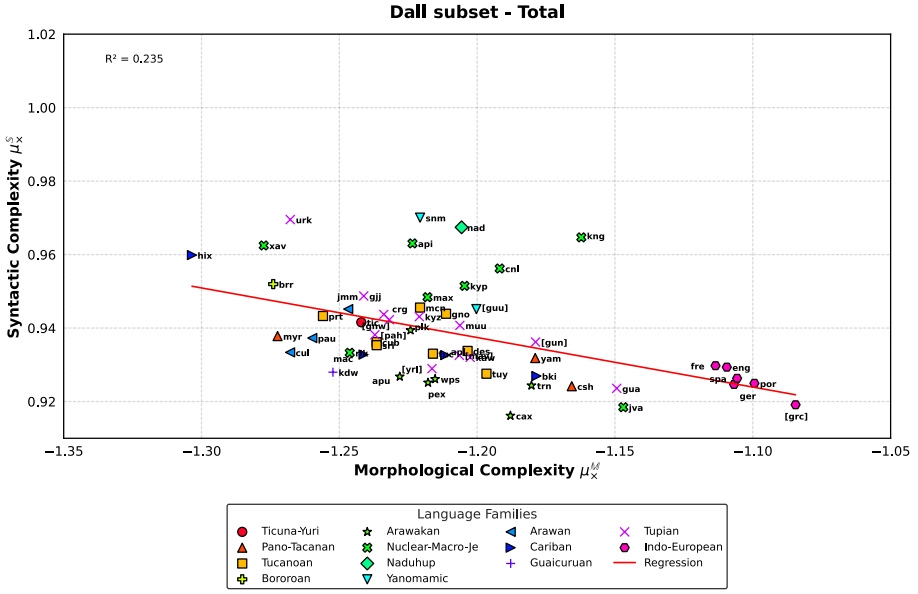

D90 subset - Total
R² = 0.137
Syntactic Complexity $\mu_\times^S$
Morphological Complexity $\mu_\times^M$
nad
kng
jmm
crg
max
[guu]
[gnw]
plk
myr
mac
apl
[mav]
[gun]
[yrl]
wps
tuy
apu
pex
trn
gua
fre
eng
spa
por
ger
[grc]
cax
Language Families
Pano-Tacanan
Tucanoan
Arawakan
Nuclear-Macro-Je
Naduhup
Yanomamic
Arawan
Cariban
Tupian
Indo-European
Regression
D90 subset - 0-25%
R² = 0.011
Syntactic Complexity $\mu_\times^S$
Morphological Complexity $\mu_\times^M$
apl
apu
fre
eng
cax
[grc]
Language Families
Arawakan
Cariban
Indo-European
Regression
D90 subset - 25-50%
R² = 0.925
Syntactic Complexity $\mu_\times^S$
Morphological Complexity $\mu_\times^M$
jmm
crg
[gnw]
[gun]
gua
ger
Language Families
Arawan
Tupian
Indo-European
Regression


D90 subset - 50-75%
R² = 0.473
Syntactic Complexity $\mu_\times^S$
Morphological Complexity $\mu_\times^M$
1.02
1.00
0.98
0.96
0.94
0.92
−1.35
−1.30
−1.25
−1.20
−1.15
−1.10
−1.05
nad
kng
max
myr
mac
[yrl]

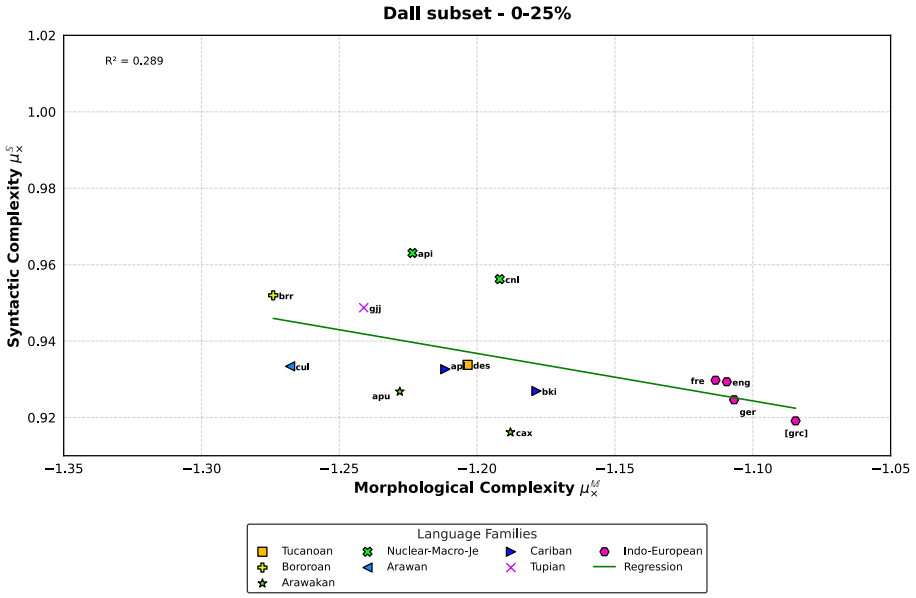

Language Families
Pano-Tacanan
Nuclear-Macro-Je
Naduhup
Cariban
Tupian
Regression

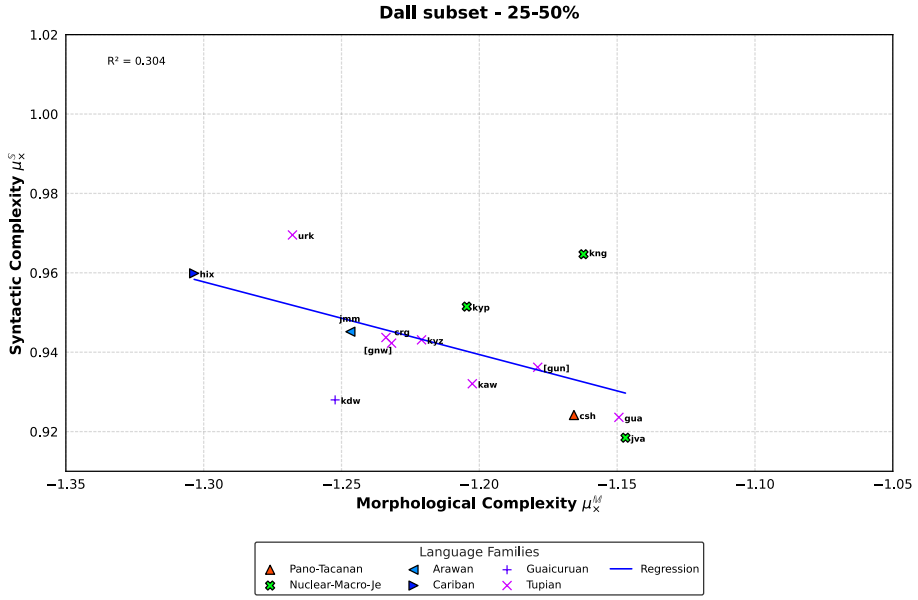

D90 subset - 75-100%
R² = 0.169
Syntactic Complexity $\mu_\times^S$
Morphological Complexity $\mu_\times^M$
1.02
1.00
0.98
0.96
0.94
0.92
−1.35
−1.30
−1.25
−1.20
−1.15
−1.10
−1.05
[guu]
plk
[mav]
wps
tuy
trn
pex
spa
por

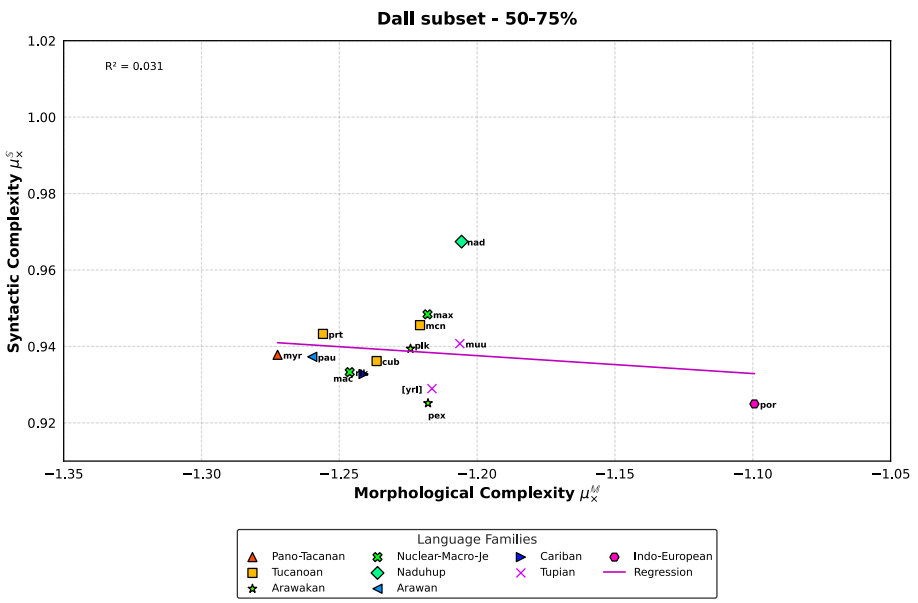

Language Families
Tucanoan
Arawakan
Yanomamic
Tupian
Indo-European
Regression

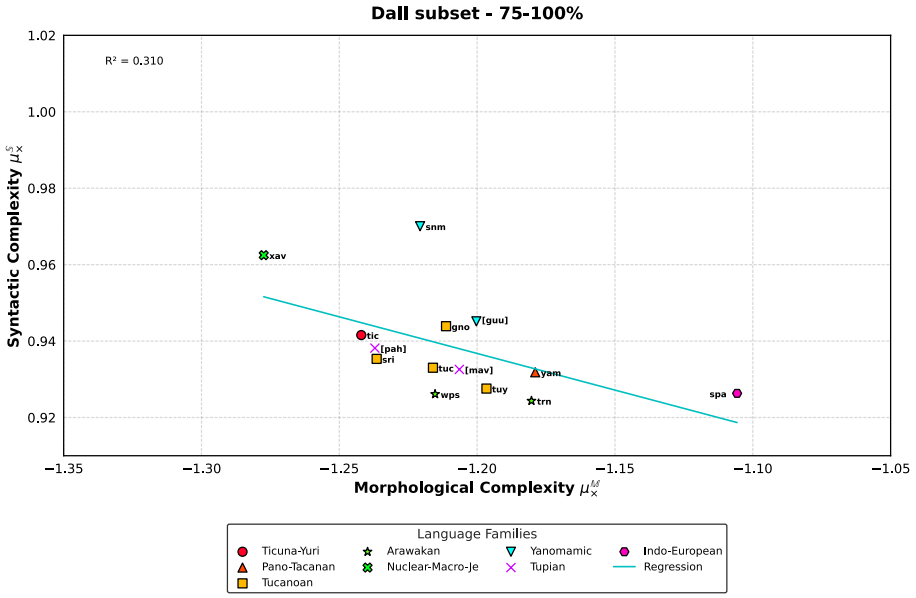

Dall subset - Total
R² = 0.235
Syntactic Complexity $\mu_\times^S$
Morphological Complexity $\mu_\times^M$
1.02
1.00
0.98
0.96
0.94
0.92
−1.35
−1.30
−1.25
−1.20
−1.15
−1.10
−1.05
urk
snm
nad
kng
xav
api
hix
cnl
brr
kyp
jmm
qjj
max
prt
crg
[guu]
gno
muu
plk
myr
pau
cul
[gun]
yam
kdw
apu
[yrl]
wps
tuy
pex
bki
trn
csh
gua
jva
cax
fre
eng
spa
por
ger
[grc]

Language Families
Ticuna-Yuri
Pano-Tacanan
Tucanoan
Bororoan
Arawakan
Nuclear-Macro-Je
Naduhup
Yanomamic
Arawan
Cariban
Guaicuruan
Tupian
Indo-European
Regression


Dall subset - 0-25%
R² = 0.289
Syntactic Complexity $\mu_\times^S$
Morphological Complexity $\mu_\times^M$
1.02
1.00
0.98
0.96
0.94
0.92
−1.35
−1.30
−1.25
−1.20
−1.15
−1.10
−1.05
api
cnl
brr
gjj
cul
apu
bki
cax
fre
eng
ger
[grc]
Language Families
Tucanoan
Bororoan
Arawakan
Nuclear-Macro-Je
Arawan
Cariban
Tupian
Indo-European
Regression


Dall subset - 25-50%
R² = 0.304
Syntactic Complexity $\mu_\times^S$
Morphological Complexity $\mu_\times^M$
1.02
1.00
0.98
0.96
0.94
0.92
−1.35
−1.30
−1.25
−1.20
−1.15
−1.10
−1.05
urk
kng
hix
kyp
jmm
ecg
[gnw]
kyz
[gun]
kaw
kdw
csh
gua
jva
Language Families
Pano-Tacanan
Nuclear-Macro-Je
Arawan
Cariban
Guaicuruan
Tupian
Regression

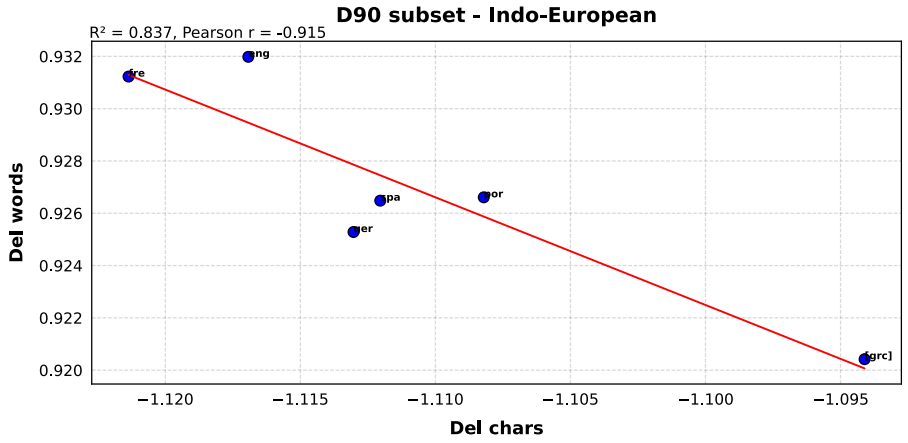

Dall subset - 50-75%
R² = 0.031
Syntactic Complexity $\mu_\times^S$
Morphological Complexity $\mu_\times^M$
1.02
1.00
0.98
0.96
0.94
0.92
−1.35
−1.30
−1.25
−1.20
−1.15
−1.10
−1.05
nad
max
mcn
prt
plk
muu
myr
pau
cub
mac
[yrl]
pex
por
Language Families
Pano-Tacanan
Tucanoan
Arawakan
Nuclear-Macro-Je
Naduhup
Arawan
Cariban
Tupian
Indo-European
Regression

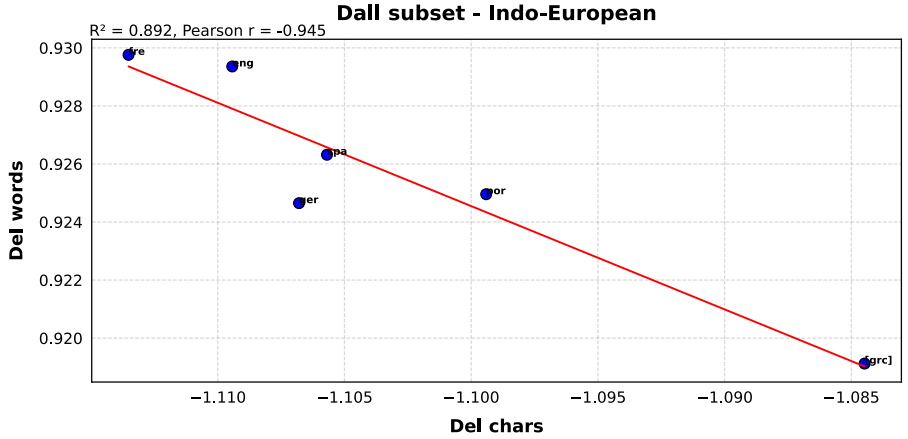

Dall subset - 75-100%
R² = 0.310
Syntactic Complexity $\mu_\times^S$
Morphological Complexity $\mu_\times^M$
1.02
1.00
0.98
0.96
0.94
0.92
−1.35
−1.30
−1.25
−1.20
−1.15
−1.10
−1.05
snm
xav
gno
[guu]
tic
[pah]
sri
tuc
[mav]
yam
tuy
wps
trn
spa
Language Families
Ticuna-Yuri
Pano-Tacanan
Tucanoan
Arawakan
Nuclear-Macro-Je
Yanomamic
Tupian
Indo-European
Regression

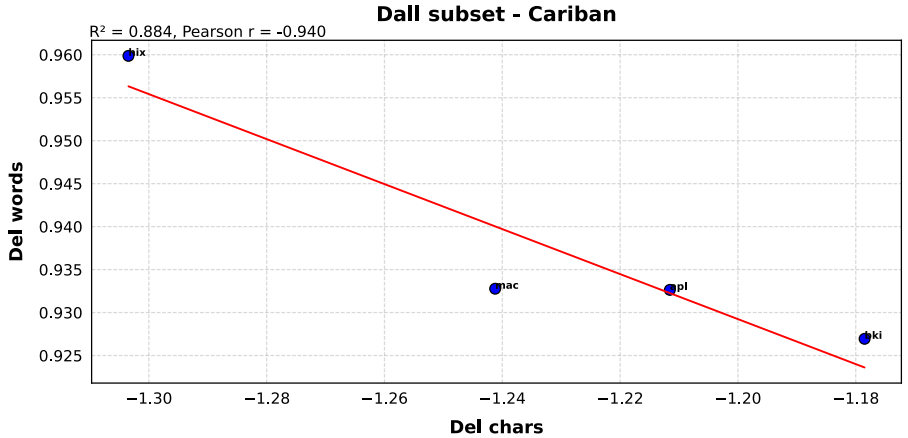
A linha da regressão linear que melhor se aproxima dos pontos de todas as linguas (dall subset - Total), confirma a relação entre morfologia e sintaxe, onde quanto maior a complexidade morfológica, a complexidade sintátia tende a ser menor. Como o agrupamento por famílias não mostrou bons resultados, foi testado o subgrupo de famílias linguísticas, que obtiveram resultados melhores. Famílias linguísticas são um grupo de línguas que acredita-se que compartilham uma língua originária comum. Sendo assim, de modo geral, existe uma proximidade maior entre as línguas da mesma família, portanto a existência da correlação mais forte faz sentido. Ao agrupar os dados por família linguística, o R^2 das regressões lineares frequentemente aumentava, mostrando uma explicação mais robusta da variabilidade das línguas dentro das famílias. Isso sugere que a relação entre complexidade sintática e morfológica é mais coerente dentro dos grupos linguísticos relacionados.

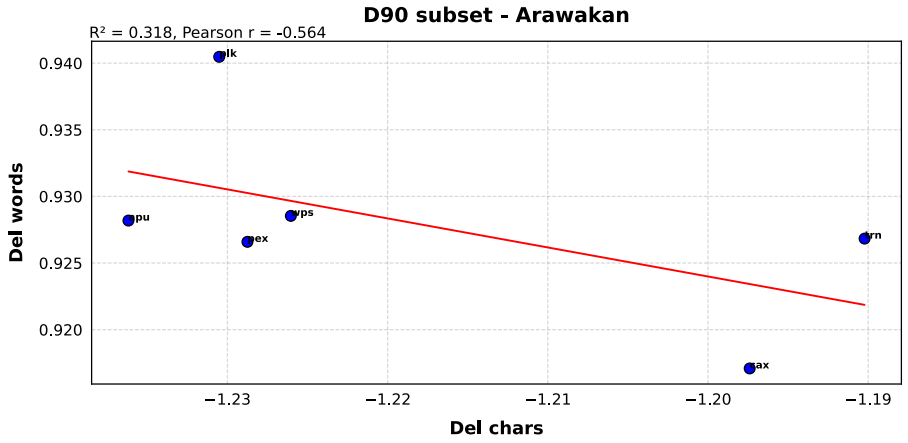
As figuras abaixo representam a regressão linear para cada família linguística. Note que, para a grande maioria, a correlação se aplica, mas para a família **Nuclear-Macro-Je**, a correlação não é significativa devido às línguas Xavante (xva) e Javaé (jva), que são exceções e afetam o cálculo da regressão linear. Como essa exceção só ocorreu para essa família, foi assumido que o comportamento atípico é devido a características específicas da família, não à hipótese de que há uma correlação maior para famílias linguísticas, mas para ter certeza é preciso fazer um estudo histórico/linguístico sobre a família **Nuclear-Macro-Je**, o que vai além do escopo deste projeto. Para lidar com isso, calculamos o coeficiente de correlação de Pearson, que é menos sensível a *outliers*, para confirmar a força da relação entre as variáveis. O coeficiente de Pearson oferece uma visão complementar, ajudando a validar se a correlação observada é estatisticamente significativa apesar da presença de valores atípicos.

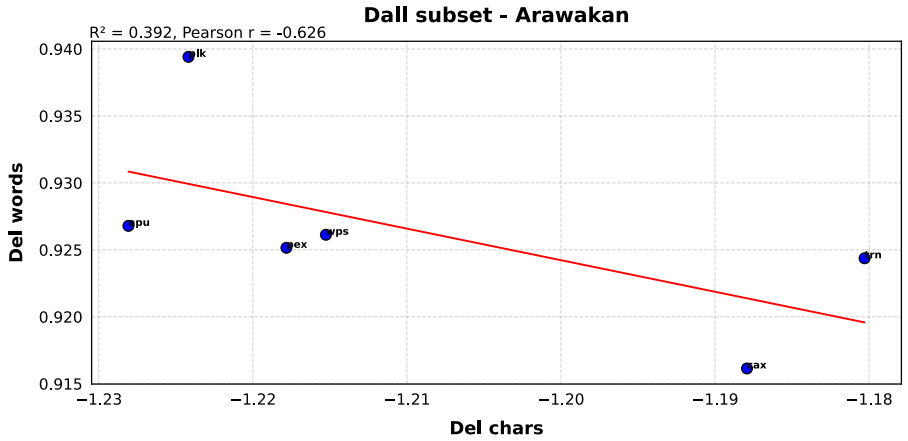
Nos gráficos abaixo, existem casos em que a regressão linear apontou que a complexidade sintática e morfológica cresceram ao mesmo tempo dentro de uma família, mas isso ocorre porque a família possui poucos membros no *dataset*, portanto a regressão linear não possui elementos o suficiente para chegar no resultado esperado.

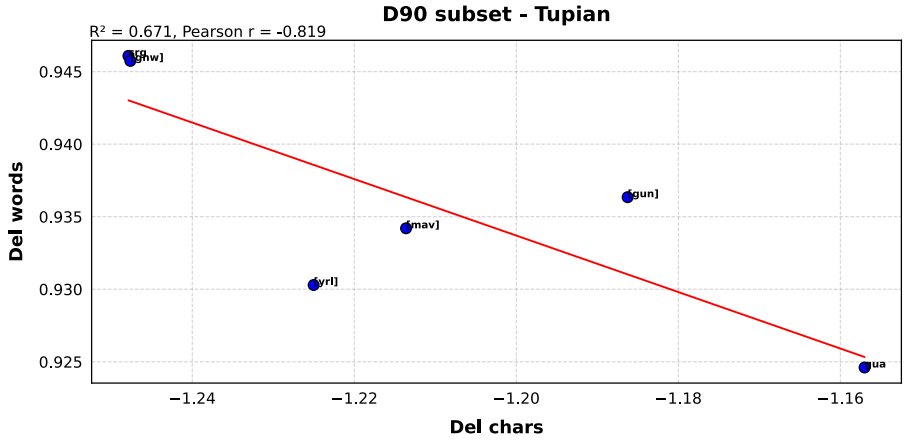

D90 subset - Indo-European
R² = 0.837, Pearson r = -0.915
Del words
Del chars
0.932
0.930
0.928
0.926
0.924
0.922
0.920
−1.120
−1.115
−1.110
−1.105
−1.100
−1.095
eng
fre
spa
gor
ger
[grc]

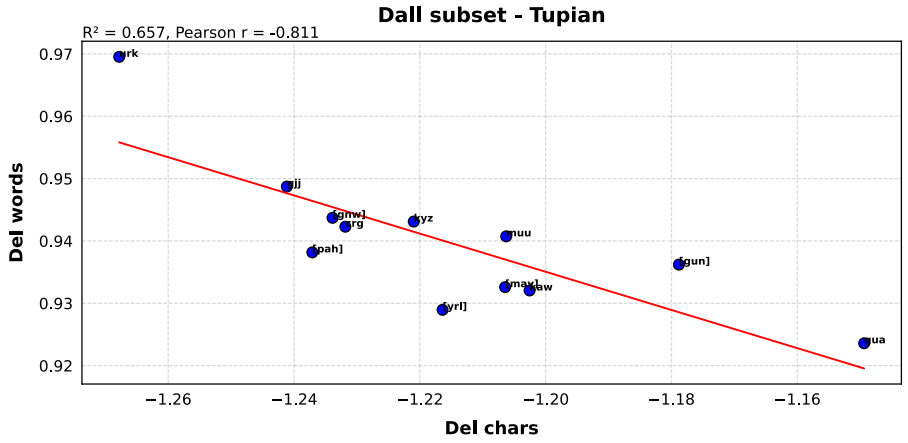

Dall subset - Indo-European
R² = 0.892, Pearson r = -0.945
Del words
Del chars
0.930
0.928
0.926
0.924
0.922
0.920
−1.110
−1.105
−1.100
−1.095
−1.090
−1.085
fre
eng
spa
ger
gor
[grc]

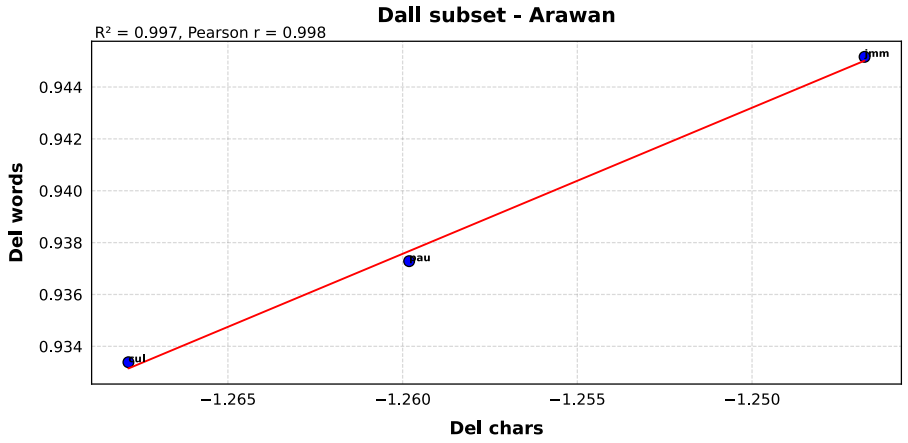

Dall subset - Cariban
R² = 0.884, Pearson r = -0.940
Del words
Del chars
0.960
0.955
0.950
0.945
0.940
0.935
0.930
0.925
−1.30
−1.28
−1.26
−1.24
−1.22
−1.20
−1.18
hix
mac
apl
yki

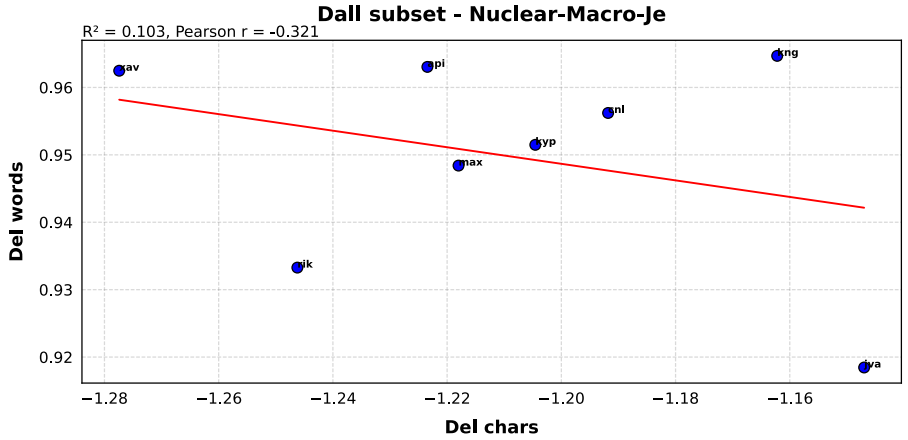

D90 subset - Arawakan
R² = 0.318, Pearson r = -0.564
Del words
Del chars
0.940
0.935
0.930
0.925
0.920
−1.23
−1.22
−1.21
−1.20
−1.19
ilk
apu
wps
nex
yrn
sax

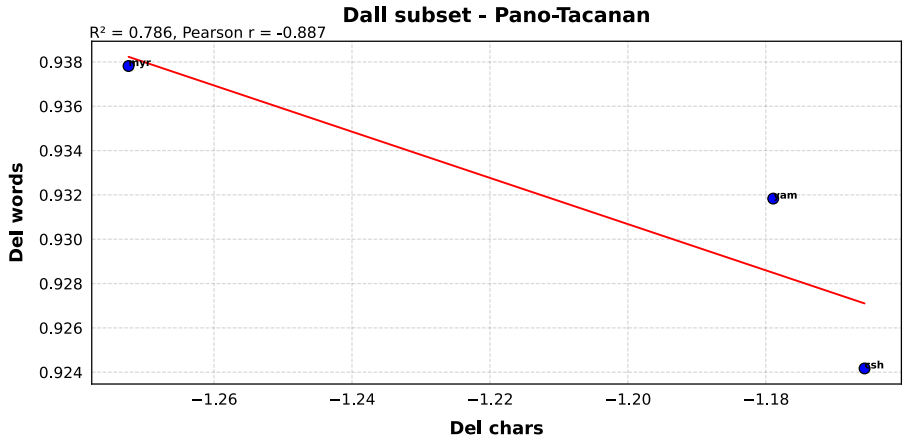

Dall subset - Arawakan
R² = 0.392, Pearson r = -0.626
Del words
Del chars
0.940
0.935
0.930
0.925
0.920
0.915
−1.23
−1.22
−1.21
−1.20
−1.19
−1.18
ilk
apu
wps
nex
yrn
sax

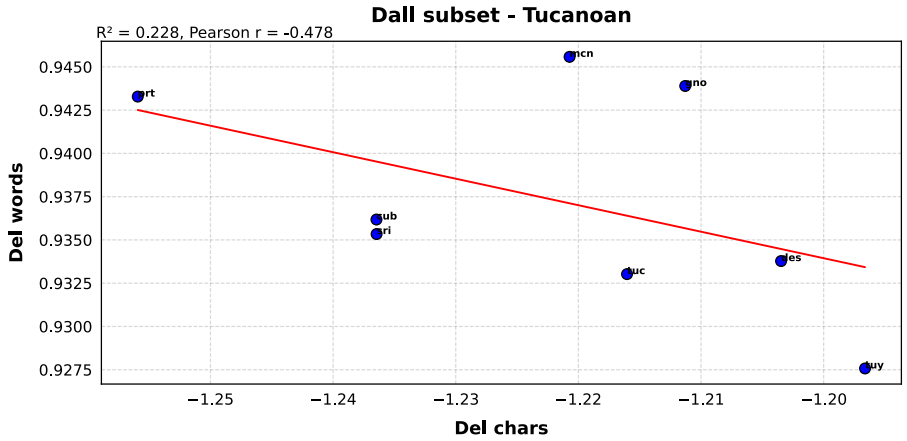

D90 subset - Tupian
R² = 0.671, Pearson r = -0.819
Del words
Del chars
0.945
0.940
0.935
0.930
0.925
−1.24
−1.22
−1.20
−1.18
−1.16
gun]
mav]
yrl]
gua


Dall subset - Tupian
R² = 0.657, Pearson r = -0.811
Del words
Del chars
0.97
0.96
0.95
0.94
0.93
0.92
−1.26
−1.24
−1.22
−1.20
−1.18
−1.16
[rk
[jj
[gnw]
[rg
[xyz
[nuu
[pah]
[gun]
[maxl
[aw
[yrl]
[ua


Dall subset - Arawan
R² = 0.997, Pearson r = 0.998
Del words
Del chars
0.944
0.942
0.940
0.938
0.936
0.934
−1.265
−1.260
−1.255
−1.250
[mm
[au
[ul


Dall subset - Nuclear-Macro-Je
R² = 0.103, Pearson r = -0.321
Del words
Del chars
0.96
0.95
0.94
0.93
0.92
−1.28
−1.26
−1.24
−1.22
−1.20
−1.18
−1.16
[av
[pl
[ng
[nl
[yp
[nax
[ik
[ya


Dall subset - Pano-Tacanan
R² = 0.786, Pearson r = -0.887
Del words
Del chars
0.938
0.936
0.934
0.932
0.930
0.928
0.926
0.924
−1.26
−1.24
−1.22
−1.20
−1.18
tyr
sam
ssh


Dall subset - Tucanoan
R² = 0.228, Pearson r = -0.478
Del words
Del chars
0.9450
0.9425
0.9400
0.9375
0.9350
0.9325
0.9300
0.9275
−1.25
−1.24
−1.23
−1.22
−1.21
−1.20
ncn
rt
ino
ub
ri
uc
es
uy

Capítulo 10

Erro Padrão da Média *Trade-off*

Para aumentar a robustez da biblioteca, abordamos o conceito estatístico conhecido como Erro Padrão da Média (SEM, do inglês *Standard Error of the Mean*). O SEM é uma medida que indica a precisão com que a média de uma amostra estima a média de uma população. Ele é calculado através da fórmula abaixo, onde s representa o desvio padrão da amostra e n o tamanho da amostra.

$$\text{SEM} = \frac{s}{\sqrt{n}}$$

O SEM é fundamental para avaliar a variabilidade da média estimada em relação à média verdadeira da população. Um SEM pequeno sugere que a média amostral é uma estimativa precisa, enquanto um SEM grande indica maior incerteza. À medida que o tamanho da amostra aumenta, o SEM diminui, reforçando a precisão da estimativa.

Na comparação entre dados da biblioteca *lang-complexity* e o programa do qual ela foi originada, este cálculo é útil para determinar se as médias das amostras são significativamente diferentes. SEMs baixos e semelhantes nas duas indicam que os dados são comparáveis.

Visualmente, barras de erro baseadas no SEM são usadas para transmitir a precisão das médias em gráficos, facilitando a visualização da confiabilidade dos dados apresentados. Assim, o SEM é uma ferramenta estatística valiosa para garantir que comparações de médias sejam baseadas em análises robustas.

Para a biblioteca *lang-complexity*, a variância está em quais os elementos que foram degradados. Por exemplo, para a métrica de substituição de palavras por unicode, dependendo de quais palavras foram escolhidas para a degradação, a saída pode ser a Saída 1 vista abaixo, a Saída 2 ou qualquer outra combinação de 3 palavras que foram substituidas. Independentemente de quais palavras foram degradadas, o resultado precisa ser o mesmo.

```
Exemplo de entrada: "Este é um exemplo de substituição."
           Saída 1: "Este é ** exemplo ** ************"
           Saída 2: "**** é um ******* ** substituição."
```

Os resultados obtidos pela biblioteca mostram que a consistência esperada foi obtida. O SEM teve ordem de grandeza de 10^{-7}, para resultados de complexidades de ordem de 10^{0}.

A figura 10.1 é um gráfico de barras que mostra o SEM para o algoritmo de substituição de palavras por caracteres unicode. Nele, é possível ver como o SEM (parte em preto) é infímo quando comparado com o valor médio dos resultados (parte em verde).

A figura 10.2, é um gráfico de dispersão de erro, nele as barras indicam a variabilidade dos dados, permitindo avaliar a precisão das medições. A distribuição dos pontos possibilita a análise de padrões e correlações entre complexidades sintáticas e morfológicas das famílias linguísticas estudadas, facilitando a comparação entre a biblioteca e o programa que a originou. No caso da figura, as barras não são visíveis, pois a medida do erro é muito baixa.

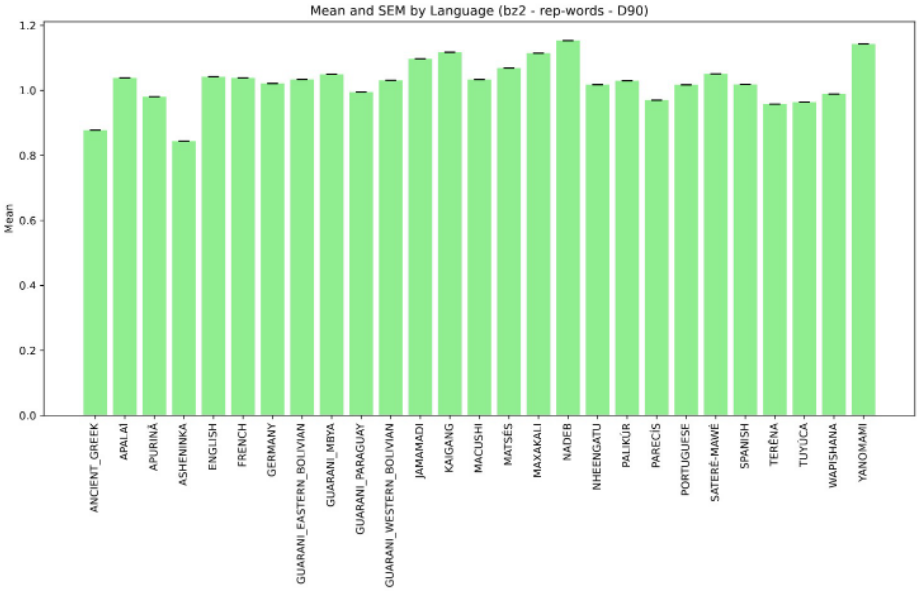


Figura 10.1: *Gráfico de barras com o SEM do algoritmo de substituição de palavras por caracteres Unicode, com a compressão por bz2.*

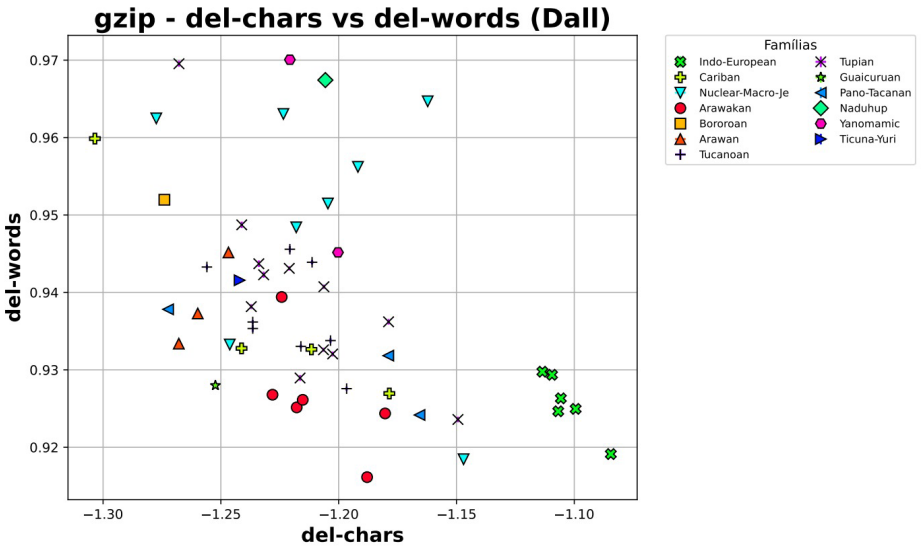
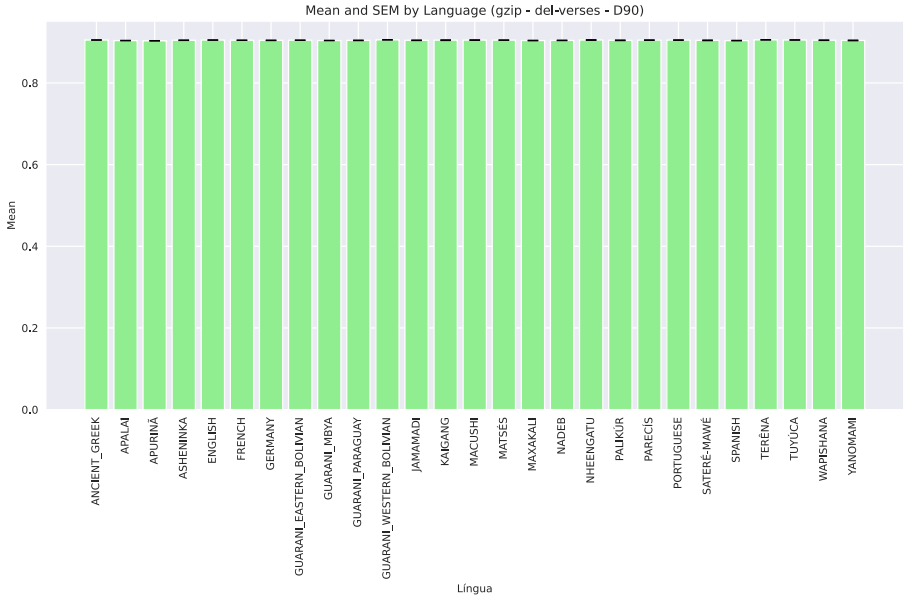


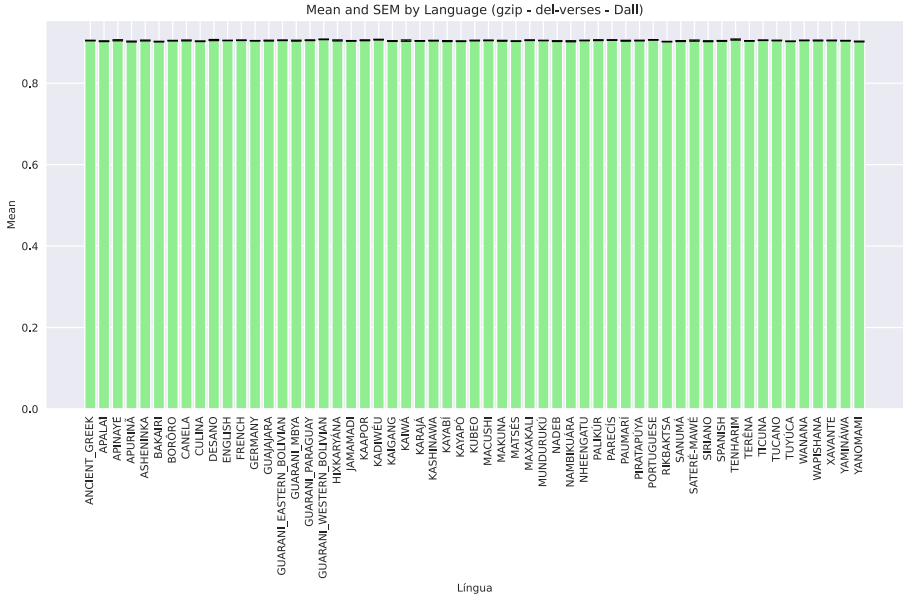
Figura 10.2: *Gráfico de dispersão de erro SEM para cada língua em relação morfológica (`del-chars`) e sintática (`del_words`).*

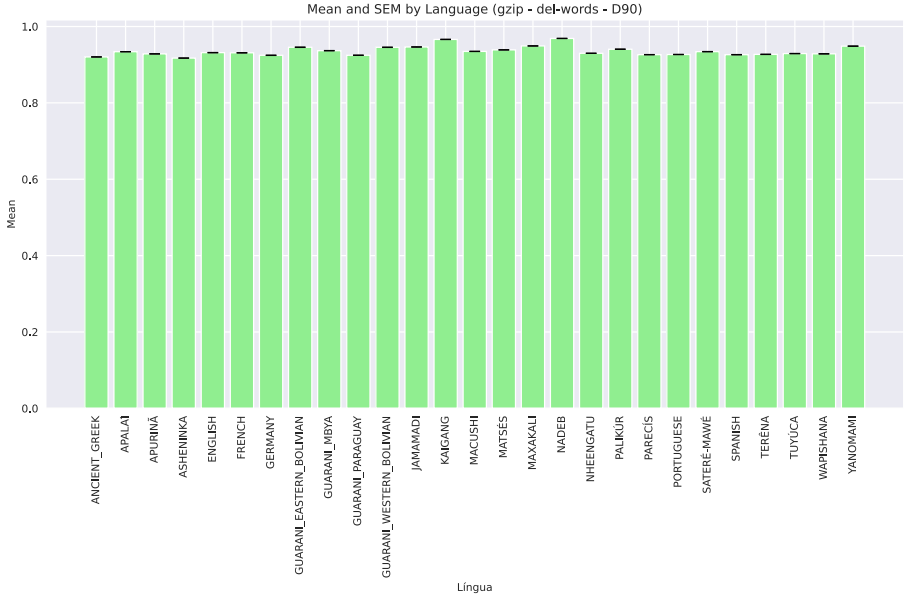
Mesmo com a variância sendo mínima, a implementação do cálculo de SEM é importante para a robustês da biblioteca. Além de verificar que a variância é minima, o SEM pode se mostrar interessante para extensões futuras da biblioteca, como por exemplo uma análise de amostragens aleatórias de cada texto poderia ter um SEM maior com resultados mais complexos do que a confirmação de que o algoritmo está correndo como desejado.

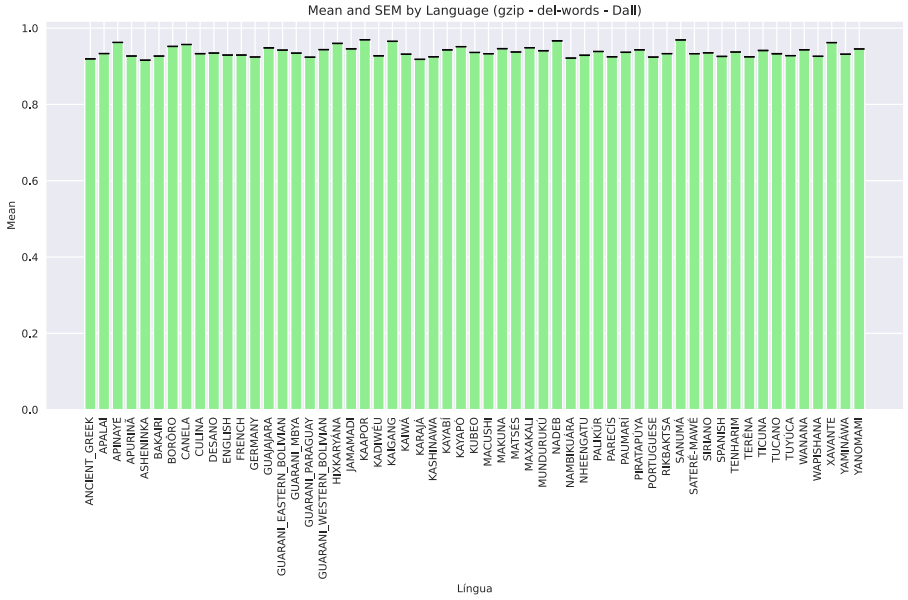
10.1 Resultados com Gráficos de Barras

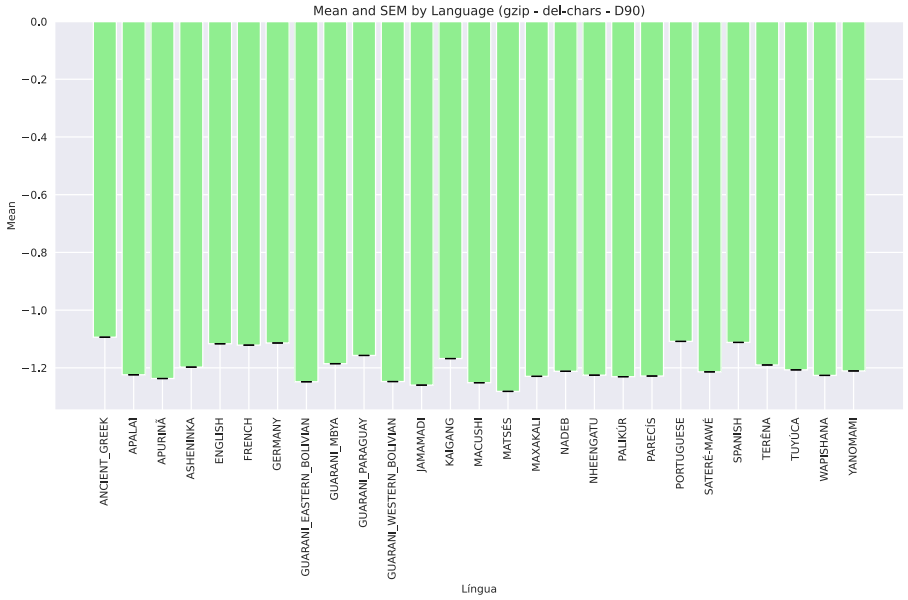
Uma característica a se notar é que todos os gráficos de compressão morfológica tem um valor negativo, assim como descrito na seção 3.1. Isso significa que a base do gráfico está na parte superior do gráfico e ele se extende para baixo. O título de cada gráfico explica o seu conteúdo. Ele está ordenado por (algoritmo de compressão - algoritmo de degradação - base de texto).

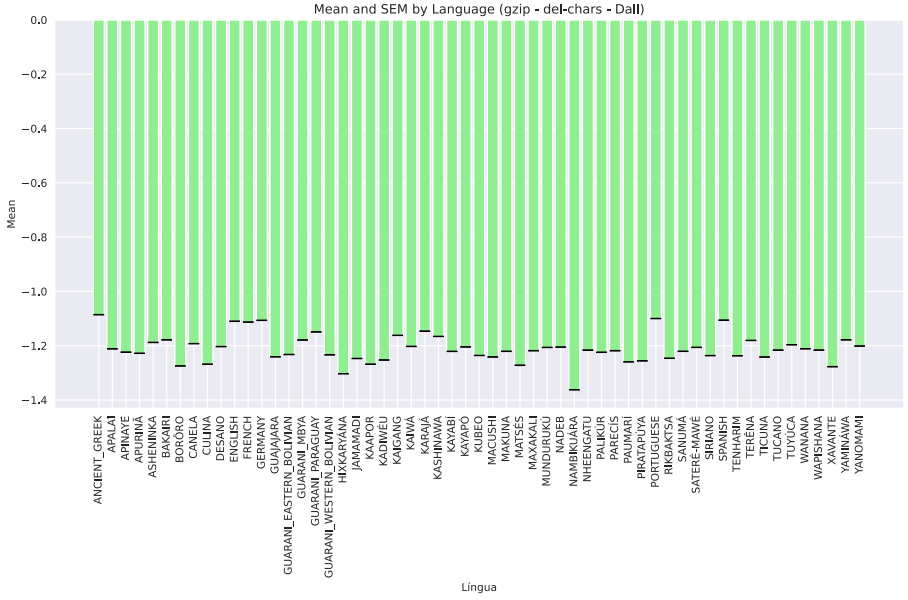

Mean and SEM by Language (gzip - del-verses - D90)
Mean
0.0
0.2
0.4
0.6
0.8
ANCIENT_GREEK
APALAI
APURINÃ
ASHENINKA
ENGLISH
FRENCH
GERMANY
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
JAMAMADI
KAIGANG
MACUSHI
MATSÉS
MAXAKALI
NADEB
NHEENGATU
PALIKÚR
PARECÍS
PORTUGUESE
SATERÉ-MAWÉ
SPANISH
TERÊNA
TUYÚCA
WAPISHANA
YANOMAMI
Língua

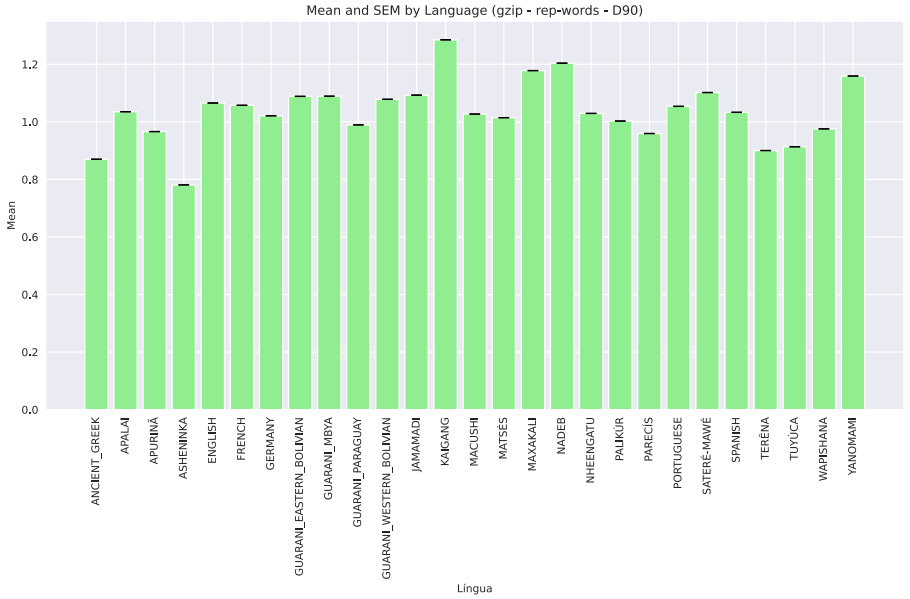

Mean and SEM by Language (gzip - del-verses - Dall)
Mean
0.0
0.2
0.4
0.6
0.8
ANCIENT_GREEK
APALAI
APINAYE
APURINÃ
ASHENINKA
BAKAIRI
BORORO
CANELA
CULINA
DESANO
ENGLISH
FRENCH
GERMANY
GUAJAJARA
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
HIXKARYANA
JAMAMADI
KAAPOR
KADIWEU
KAIGANG
KANWA
KARAJA
KASHINAWA
KAYABI
KAYAPO
KUBEO
MACUSHI
MAKUNA
MATSÉS
MAXAKALI
MUNDURUKU
NADEB
NAMBIKUARA
NHEENGATU
PALIKUR
PARECÍS
PAUMARI
PIRATAPUYA
PORTUGUESE
RIKBAKTSA
SANUMÁ
SATERÉ-MAWÉ
SIRIANO
SPANISH
TENHARIM
TERÊNA
TICUNA
TUCANO
TUYÚCA
WANANA
WAPISHANA
XAVANTE
YAMINAWA
YANOMAMI
Língua

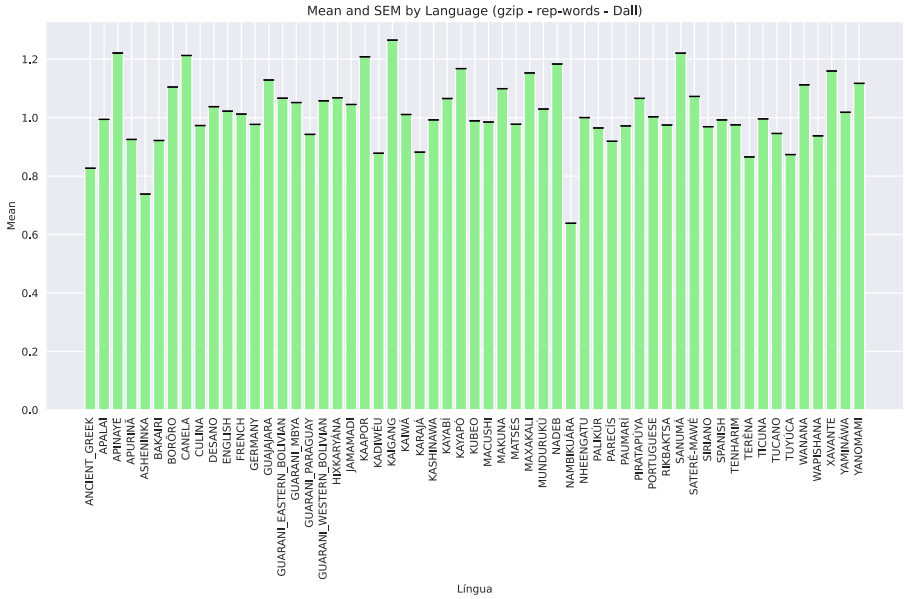

Mean and SEM by Language (gzip - del-words - D90)
Mean
0.0
0.2
0.4
0.6
0.8
1.0
ANCIENT_GREEK
APALAI
APURINÃ
ASHENINKA
ENGLISH
FRENCH
GERMANY
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
JAMAMADI
KAIGANG
MACUSHI
MATSÉS
MAXAKALI
NADEB
NHEENGATU
PALIKÚR
PARECÍS
PORTUGUESE
SATERÉ-MAWÉ
SPANISH
TERÊNA
TUYÚCA
WAPISHANA
YANOMAMI
Língua

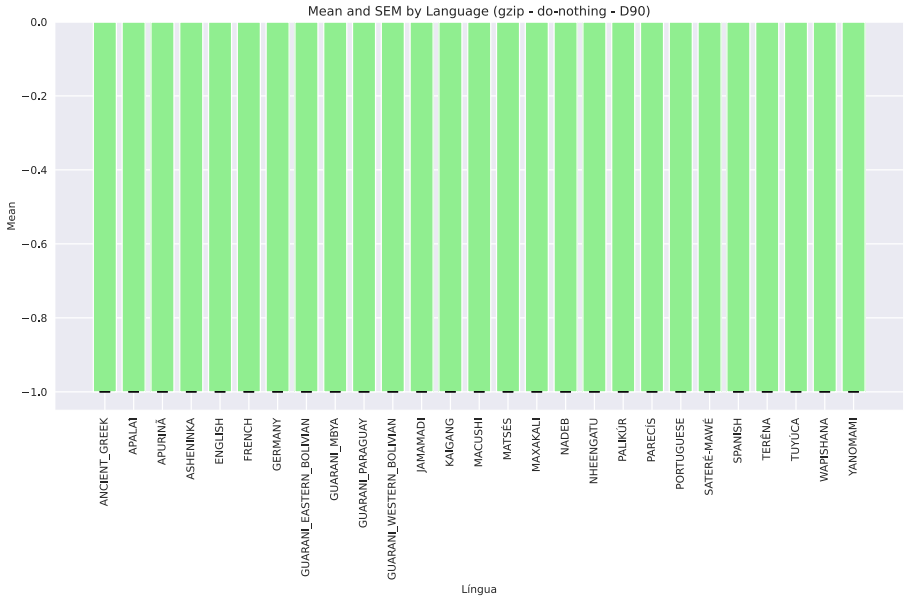

Mean and SEM by Language (gzip - del-words - Dall)
Mean
Língua

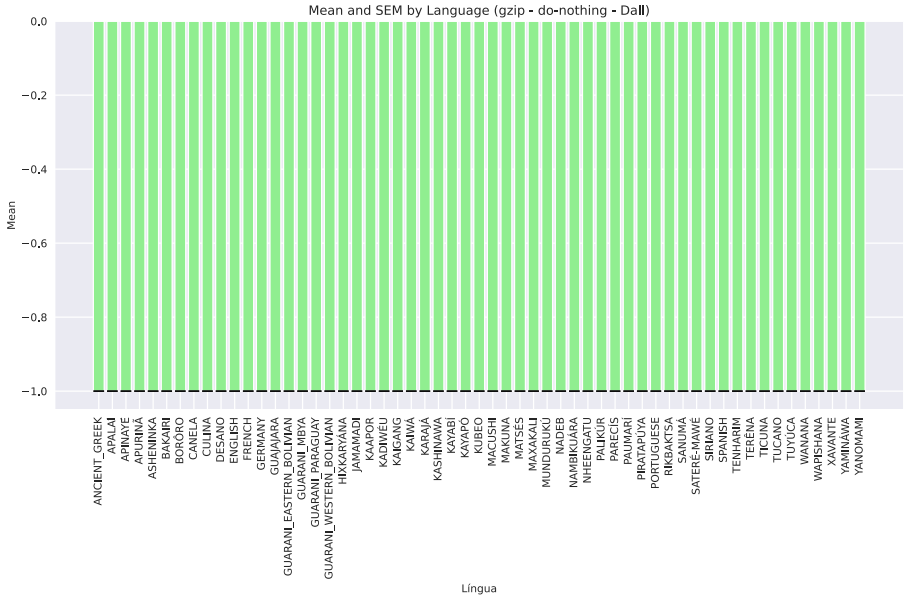

Mean and SEM by Language (gzip - del-chars - D90)
Mean
Língua

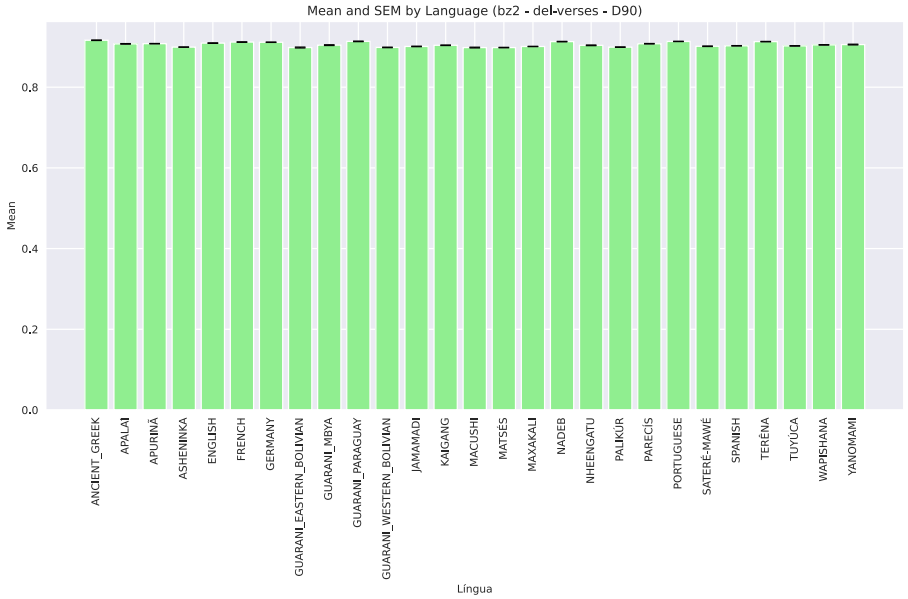

Mean and SEM by Language (gzip - del-chars - Dall)
Mean
Língua

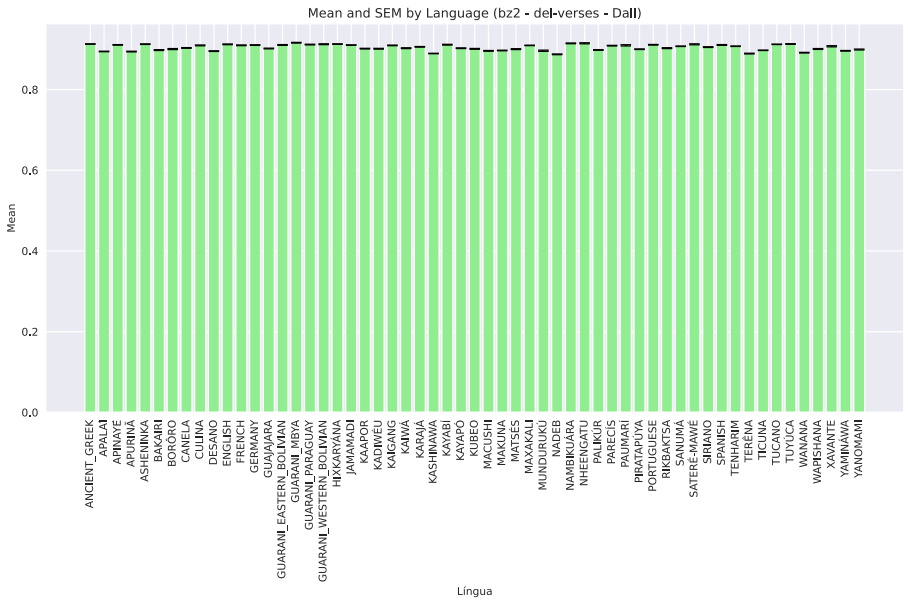

Mean and SEM by Language (gzip - rep-words - D90)
Mean
Língua

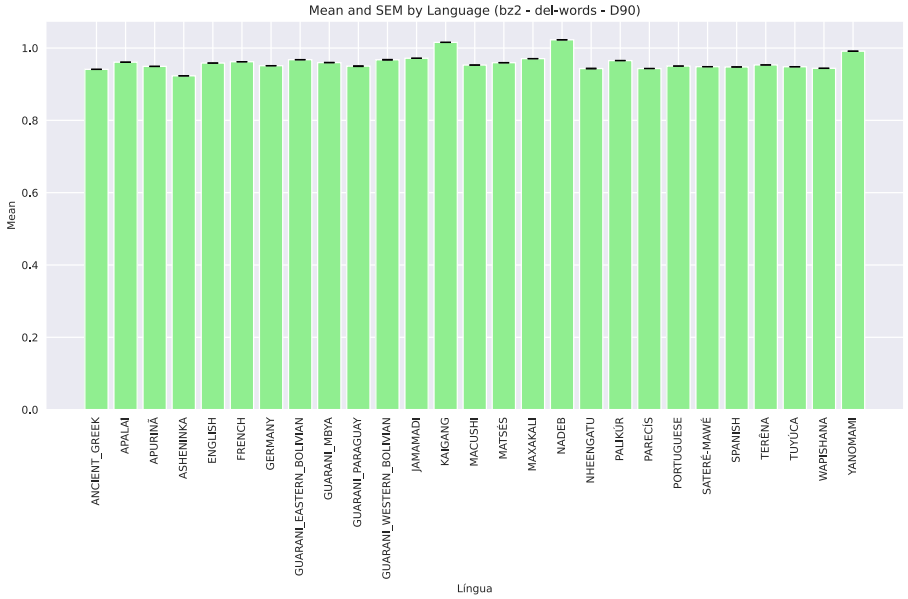

Mean and SEM by Language (gzip - rep-words - Dall)
Mean
Língua

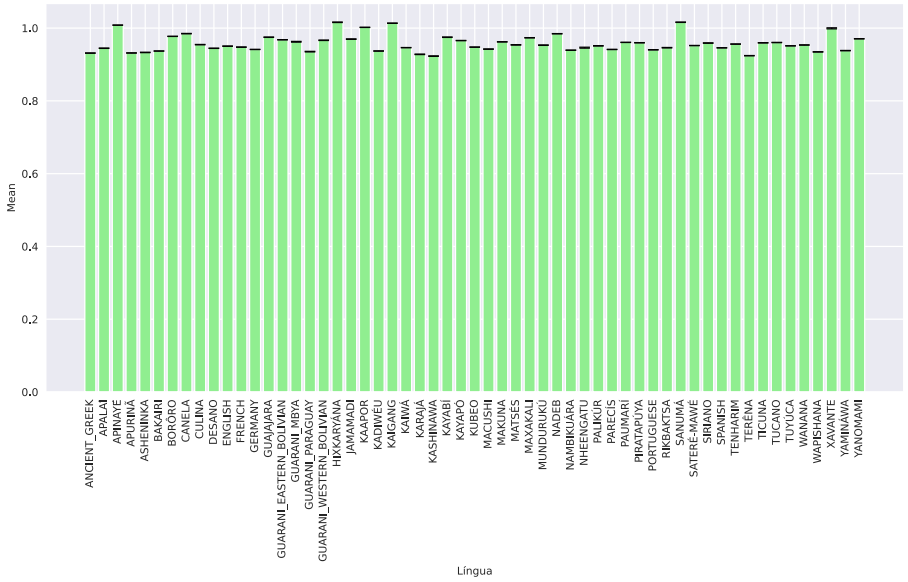

Mean and SEM by Language (gzip - do-nothing - D90)
Mean
Língua

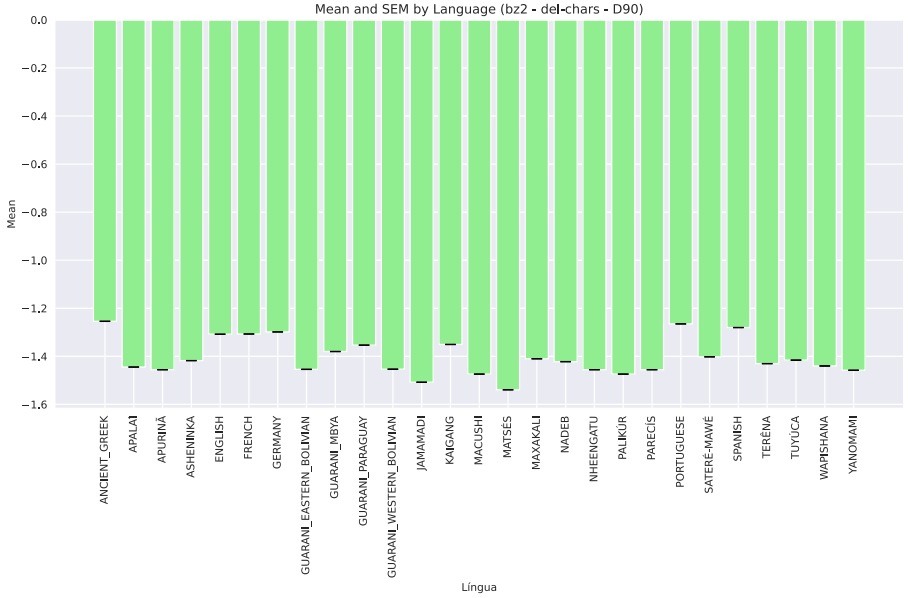

Mean and SEM by Language (gzip - do-nothing - Dall)
Mean
Língua

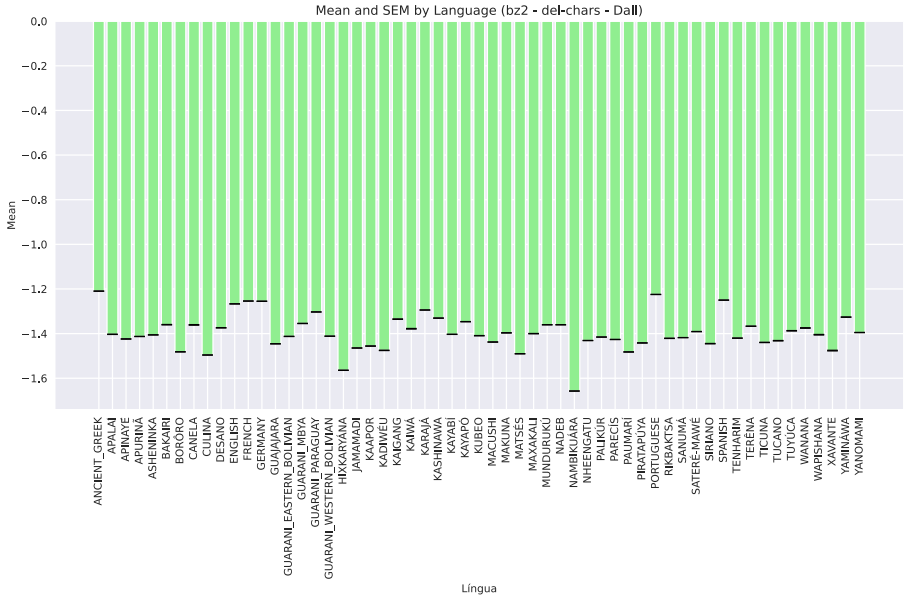

Mean and SEM by Language (bz2 - del-verses - D90)
Mean
Língua

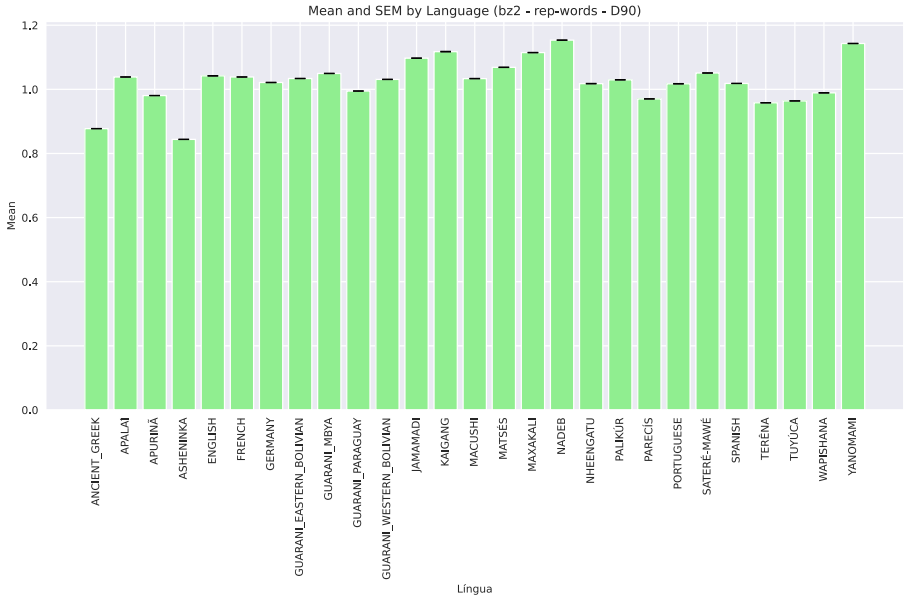

Mean and SEM by Language (bz2 - del-verses - Dall)
Mean
Língua

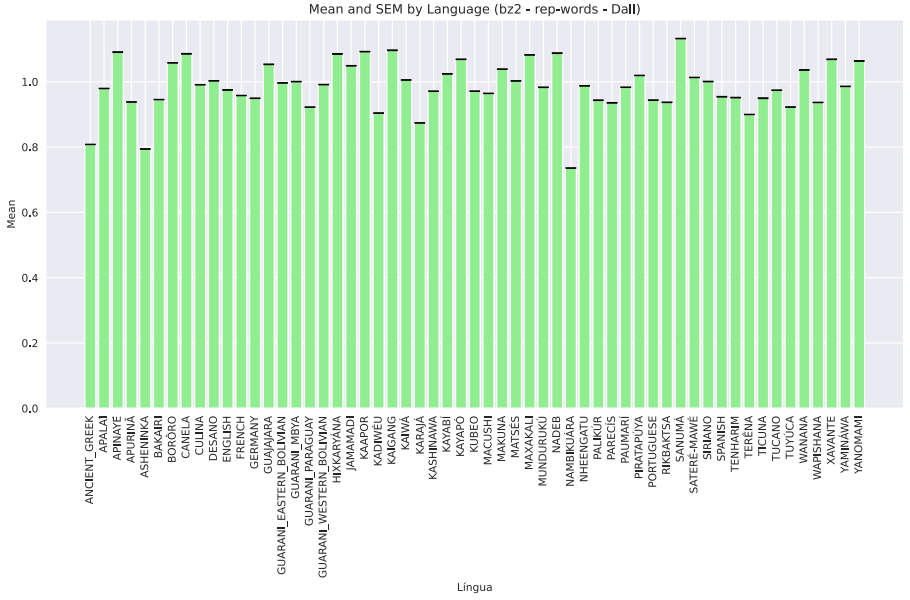

Mean and SEM by Language (bz2 - del-words - D90)
Mean
Língua

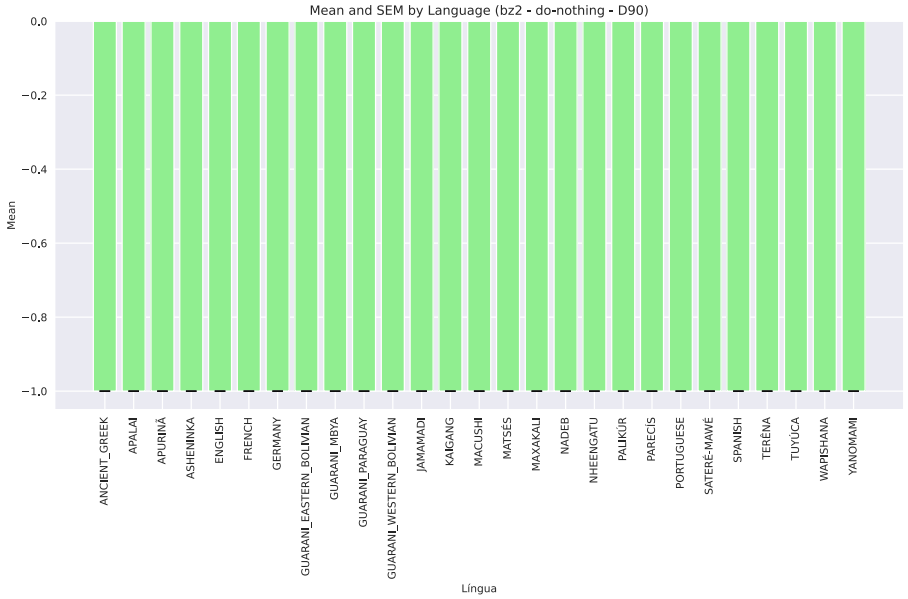

Mean and SEM by Language (bz2 - del-words - Dall)
Mean
Língua

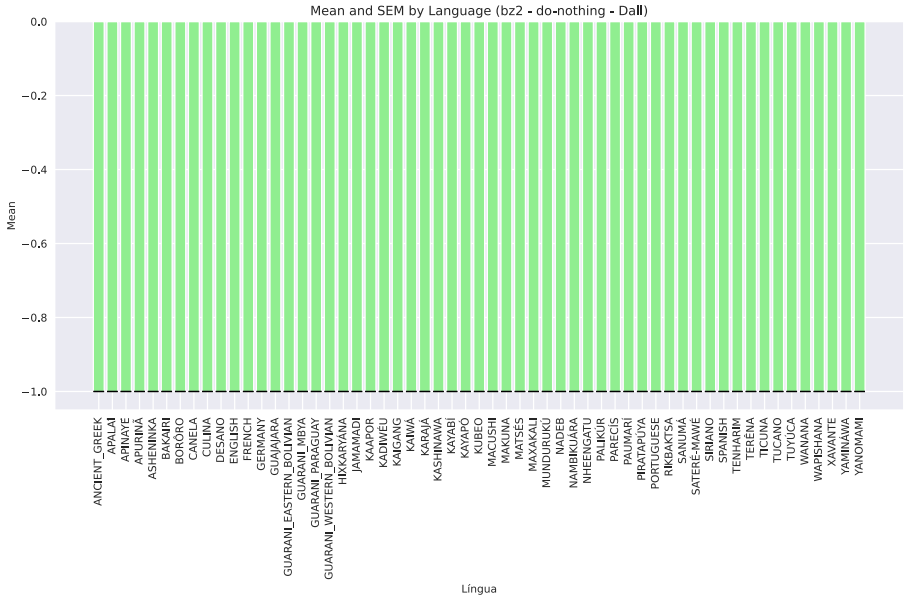

Mean and SEM by Language (bz2 - del-chars - D90)
Mean
Língua

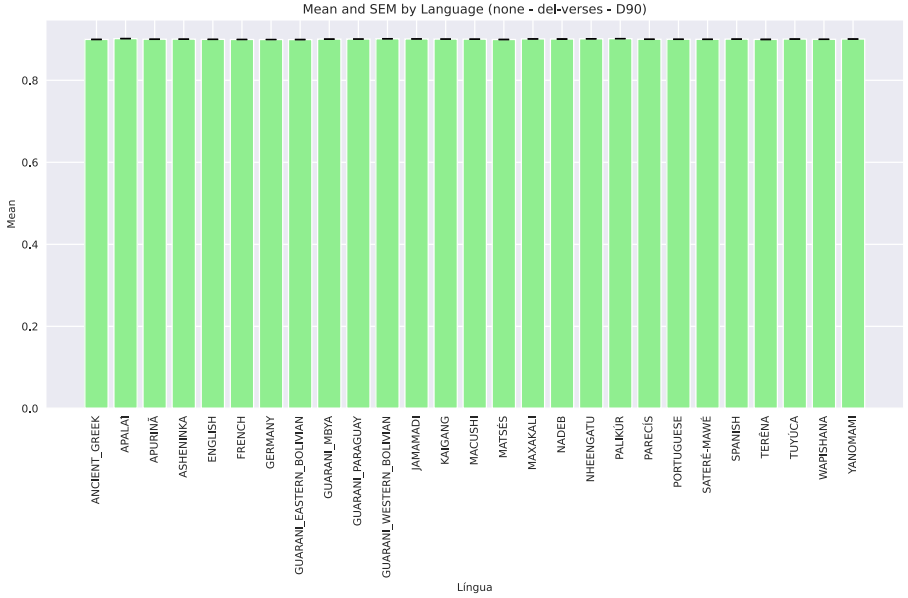

Mean and SEM by Language (bz2 - del-chars - Dall)
Mean
Língua

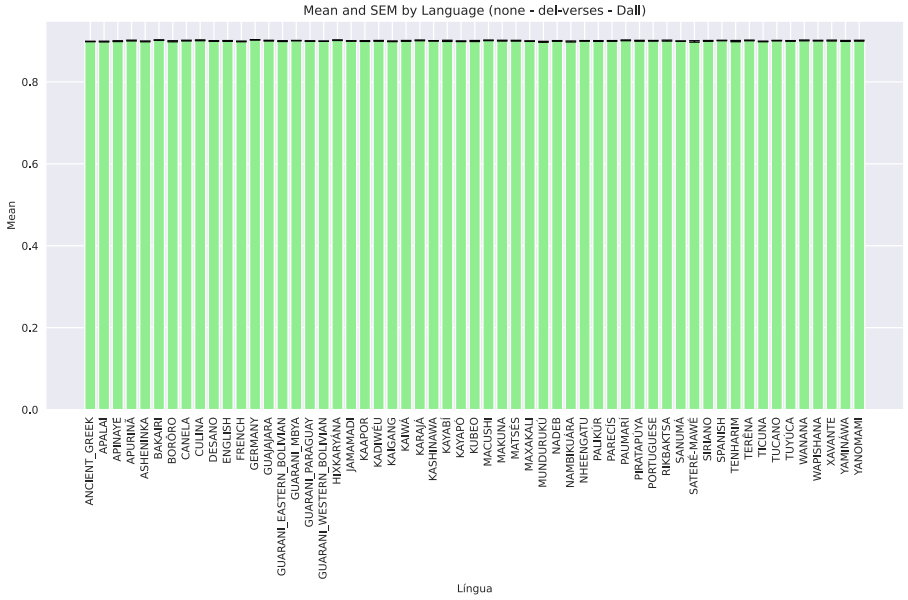

Mean and SEM by Language (bz2 - rep-words - D90)
Mean
Língua

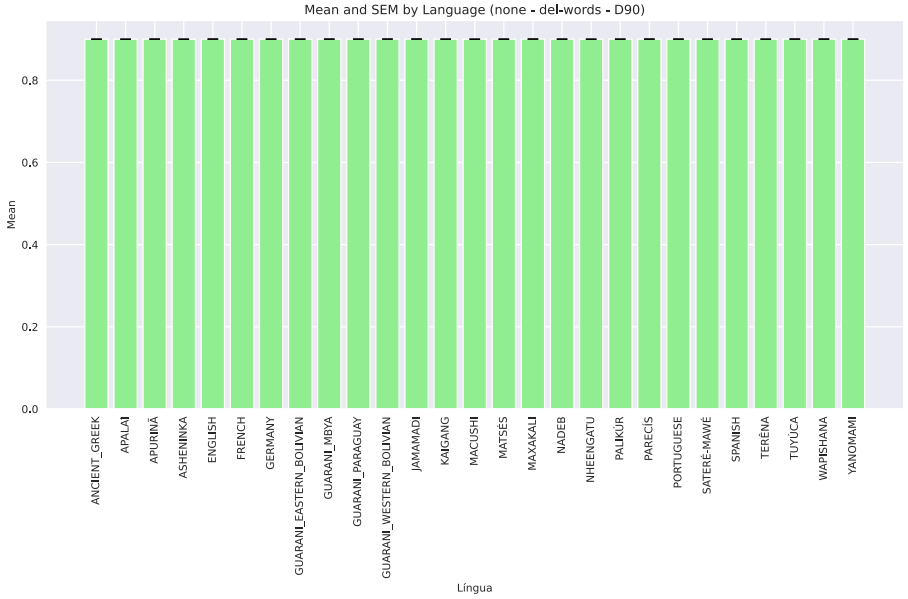

Mean and SEM by Language (bz2 - rep-words - Dall)
Mean
Língua

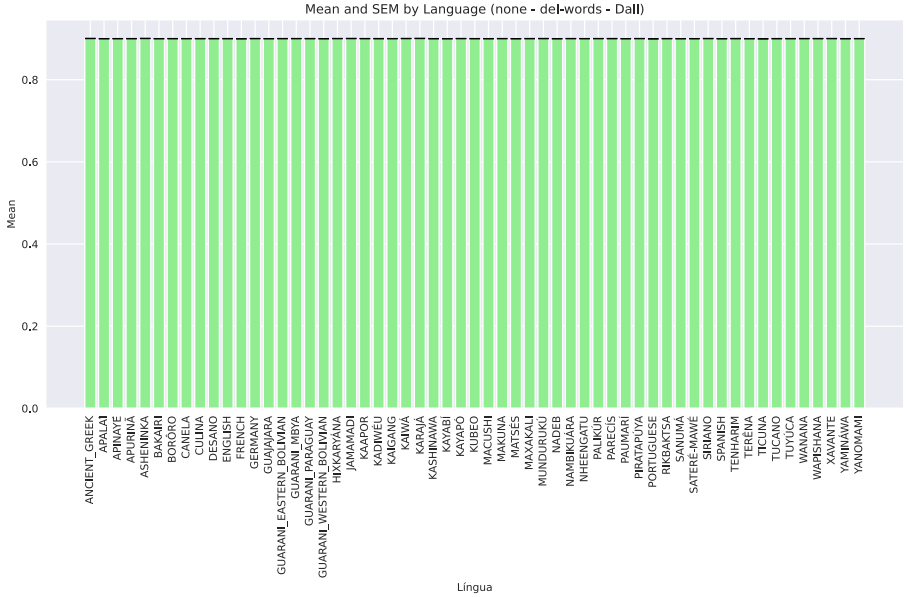

Mean and SEM by Language (bz2 - do-nothing - D90)
Mean
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
ANCIENT_GREEK
APALAI
APURINÃ
ASHENINKA
ENGLISH
FRENCH
GERMANY
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
JAMAMADI
KAIGANG
MACUSHI
MATSÉS
MAXAKALI
NADEB
NHEENGATU
PALIKÚR
PARECÍS
PORTUGUESE
SATERÉ-MAWÉ
SPANISH
TERÊNA
TUYÚCA
WAPISHANA
YANOMAMI
Língua

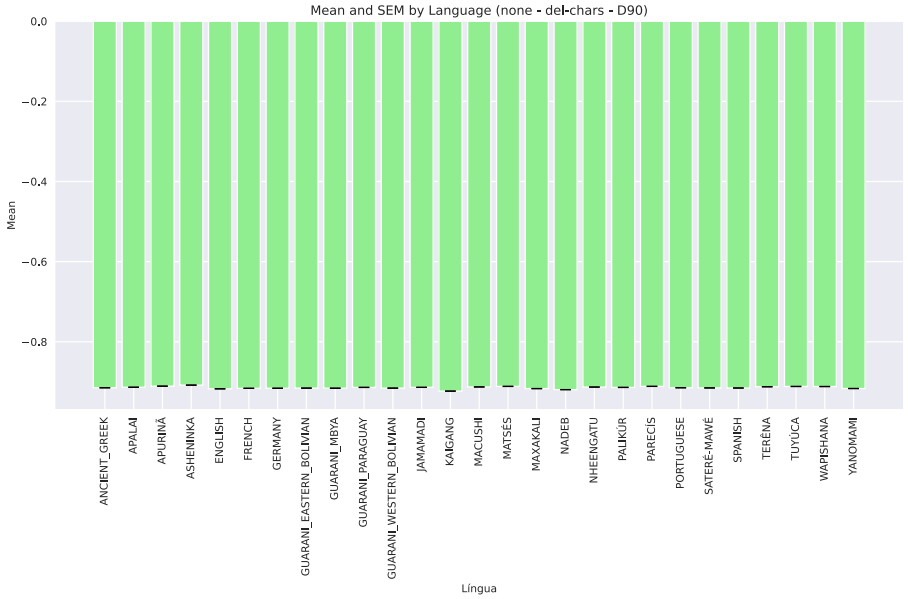

Mean and SEM by Language (bz2 - do-nothing - Dall)
Mean
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
ANCIENT_GREEK
APALAI
APINAYE
APURINÃ
ASHENINKA
BAKAIRI
BORORO
CANELA
CULINA
DESANO
ENGLISH
FRENCH
GERMANY
GUAJAJARA
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
HIXKARYANA
JAMAMADI
KAAPOR
KADIWEU
KAIGANG
KANWÁ
KARAJÁ
KASHINAWA
KAYABI
KAYAPÓ
KUBEO
MACUSHI
MAKUNA
MATSÉS
MAXAKALI
MUNDURUKÚ
NADEB
NAMBIKUARA
NHEENGATU
PALIKÚR
PARECÍS
PAUMARÍ
PIRAHÃ
PORTUGUESE
RIKBAKTSA
SANUMÁ
SATERÉ-MAWÉ
SIRIANO
SPANISH
TENHARIM
TERÊNA
TICUNA
TUCANO
TUYÚCA
WANANA
WAPISHANA
XAVANTE
YAMINAWA
YANOMAMI
Língua

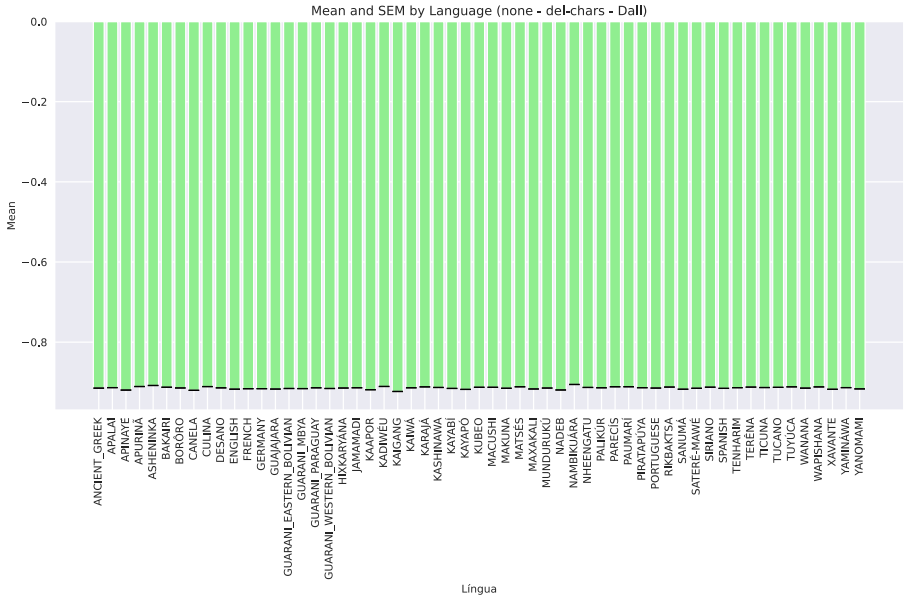

Mean and SEM by Language (none - del-verses - D90)
Mean
0.8
0.6
0.4
0.2
0.0
ANCIENT_GREEK
APALAI
APURINÃ
ASHENINKA
ENGLISH
FRENCH
GERMANY
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
JAMAMADI
KAIGANG
MACUSHI
MATSÉS
MAXAKALI
NADEB
NHEENGATU
PALIKÚR
PARECÍS
PORTUGUESE
SATERÉ-MAWÉ
SPANISH
TERÊNA
TUYÚCA
WAPISHANA
YANOMAMI
Língua

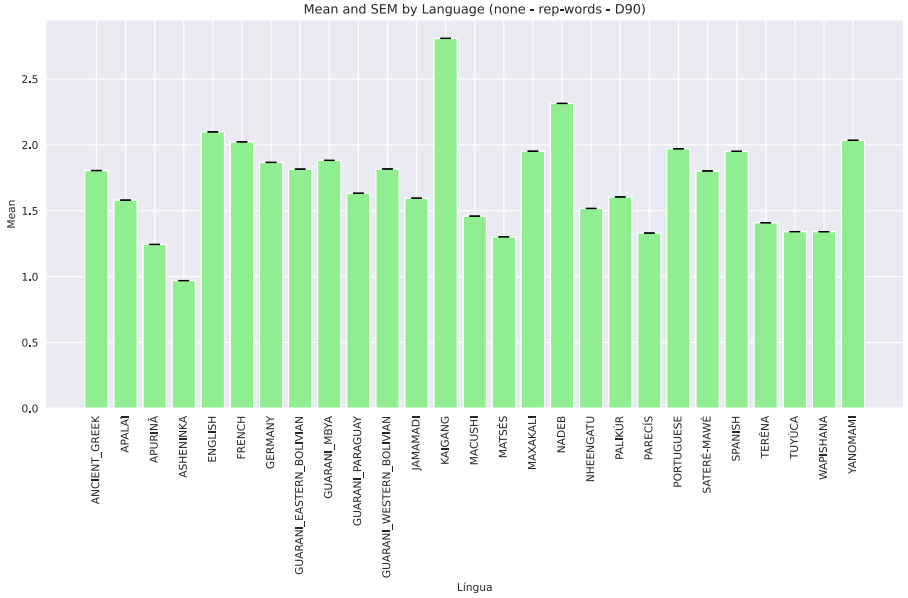

Mean and SEM by Language (none - del-verses - Dall)
Mean
0.0
0.2
0.4
0.6
0.8
ANCIENT_GREEK
APALAI
APINAYE
APURINÃ
ASHENINKA
BAKAIRI
BORÓRO
CANELA
CULINA
DESANO
ENGLISH
FRENCH
GERMANY
GUAJAJARA
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
HIXKARYANA
JAMAMADI
KAAPOR
KADIWEU
KAIGANG
KAIWÁ
KARAJÁ
KASHINAWA
KAYABI
KAYAPÓ
KUBEO
MACUSHI
MAKUNA
MATSÉS
MAXAKALI
MUNDURUKU
NADEB
NAMBIKUARA
NHEENGATU
PALIKÚR
PARECÍS
PAUMARI
PIRATAPUYA
PORTUGUESE
RIKBAKTSA
SANUMÁ
SATERÉ-MAWÉ
SIRIANO
SPANISH
TENHARIM
TERÊNA
TICUNA
TUCANO
TUYÚCA
WANANA
WAPISHANA
XAVANTE
YAMINÁWA
YANOMAMI
Língua

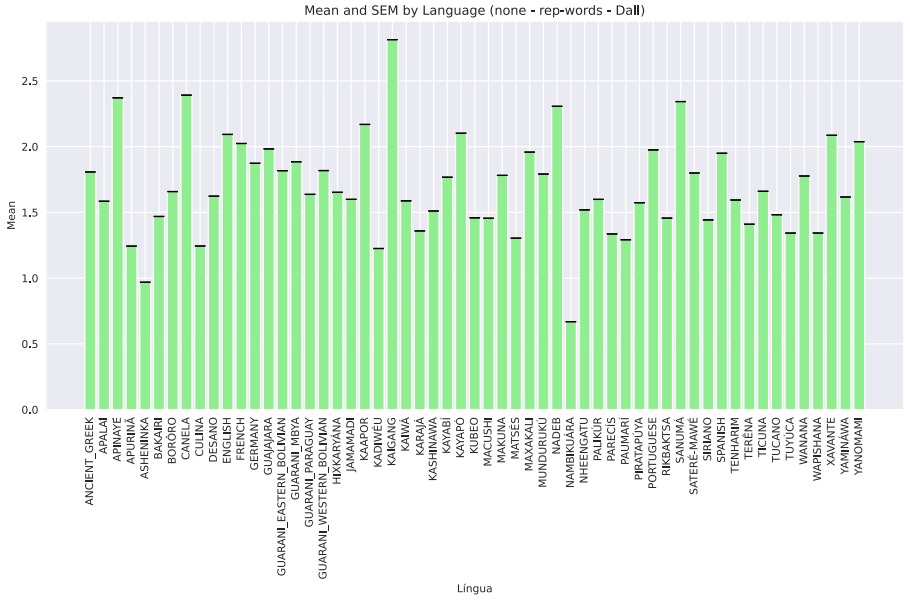

Mean and SEM by Language (none - del-words - D90)
Mean
0.0
0.2
0.4
0.6
0.8
ANCIENT_GREEK
APALAI
APURINÃ
ASHENINKA
ENGLISH
FRENCH
GERMANY
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
JAMAMADI
KAIGANG
MACUSHI
MATSÉS
MAXAKALI
NADEB
NHEENGATU
PALIKÚR
PARECÍS
PORTUGUESE
SATERÉ-MAWÉ
SPANISH
TERÊNA
TUYÚCA
WAPISHANA
YANOMAMI
Língua

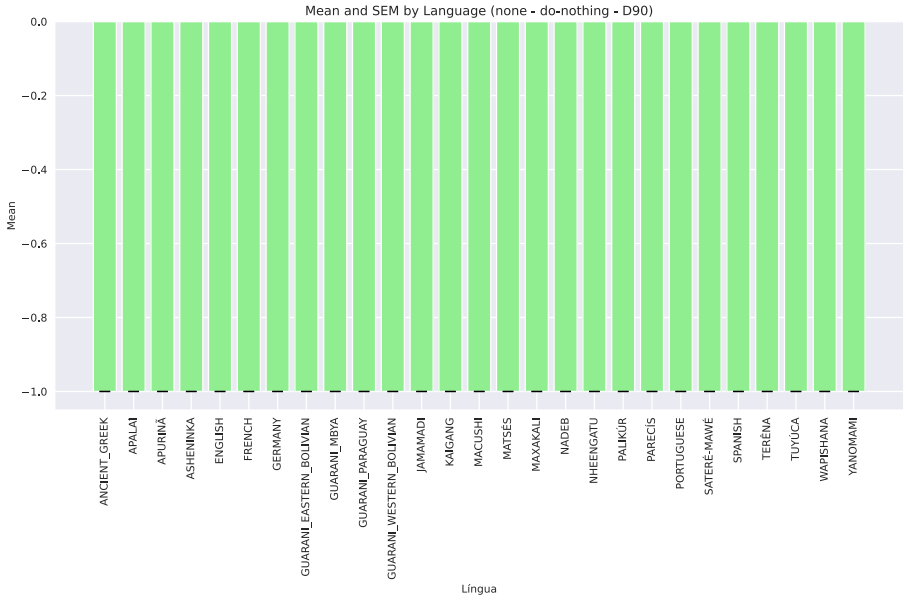

Mean and SEM by Language (none - del-words - Dall)
Mean
0.0
0.2
0.4
0.6
0.8
ANCIENT_GREEK
APALAI
APINAYE
APURINÃ
ASHENINKA
BAKAIRI
BORÓRO
CANELA
CULINA
DESANO
ENGLISH
FRENCH
GERMANY
GUAJAJARA
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
HIXKARYANA
JAMAMADI
KAAPOR
KADIWEU
KAIGANG
KAIWÁ
KARAJÁ
KASHINAWA
KAYABI
KAYAPÓ
KUBEO
MACUSHI
MAKUNA
MATSÉS
MAXAKALI
MUNDURUKU
NADEB
NAMBIKUARA
NHEENGATU
PALIKÚR
PARECÍS
PAUMARI
PIRATAPUYA
PORTUGUESE
RIKBAKTSA
SANUMÁ
SATERÉ-MAWÉ
SIRIANO
SPANISH
TENHARIM
TERÊNA
TICUNA
TUCANO
TUYÚCA
WANANA
WAPISHANA
XAVANTE
YAMINÁWA
YANOMAMI
Língua

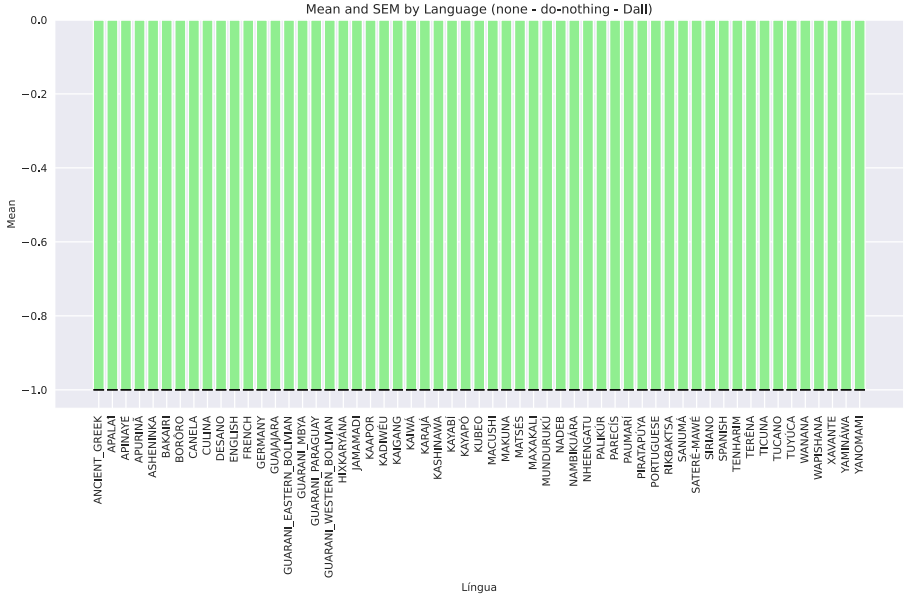

Mean and SEM by Language (none - del-chars - D90)
Mean
0.0
−0.2
−0.4
−0.6
−0.8
ANCIENT_GREEK
APALAI
APURINÃ
ASHENINKA
ENGLISH
FRENCH
GERMANY
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
JAMAMADI
KAIGANG
MACUSHI
MATSÉS
MAXAKALI
NADEB
NHEENGATU
PALIKÚR
PARECÍS
PORTUGUESE
SATERÉ-MAWÉ
SPANISH
TERÊNA
TUYÚCA
WAPISHANA
YANOMAMI
Língua


Mean and SEM by Language (none - del-chars - Dall)
Mean
0.0
−0.2
−0.4
−0.6
−0.8
ANCIENT_GREEK
APALAI
APINAYE
APURINÃ
ASHENINKA
BAKAIRI
BORORO
CANELA
CULINA
DESANO
ENGLISH
FRENCH
GERMANY
GUAJAJARA
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
HIXKARYANA
JAMAMADI
KAAPOR
KADIWEU
KAIGANG
KANWÁ
KARAJÁ
KASHINAWA
KAYABI
KAYAPÓ
KUBEO
MACUSHI
MAKUNA
MATSÉS
MAXAKALI
MUNDURUKÚ
NADEB
NAMBIKUARA
NHEENGATU
PALIKÚR
PARECÍS
PAUMARÍ
PIRAHÃ
PORTUGUESE
RIKBAKTSA
SANUMÁ
SATERÉ-MAWÉ
SIRIANO
SPANISH
TENHARIM
TERÊNA
TICUNA
TUCANO
TUYÚCA
WANANA
WAPISHANA
XAVANTE
YAMINAWA
YANOMAMI
Língua


Mean and SEM by Language (none - rep-words - D90)
Mean
2.5
2.0
1.5
1.0
0.5
0.0
ANCIENT_GREEK
APALAI
APURINÃ
ASHENINKA
ENGLISH
FRENCH
GERMANY
GUARANI_EASTERN_BOLIVIAN
GUARANI_MBYA
GUARANI_PARAGUAY
GUARANI_WESTERN_BOLIVIAN
JAMAMADI
KAIGANG
MACUSHI
MATSÉS
MAXAKALI
NADEB
NHEENGATU
PALIKÚR
PARECÍS
PORTUGUESE
SATERÉ-MAWÉ
SPANISH
TERÊNA
TUYÚCA
WAPISHANA
YANOMAMI
Língua

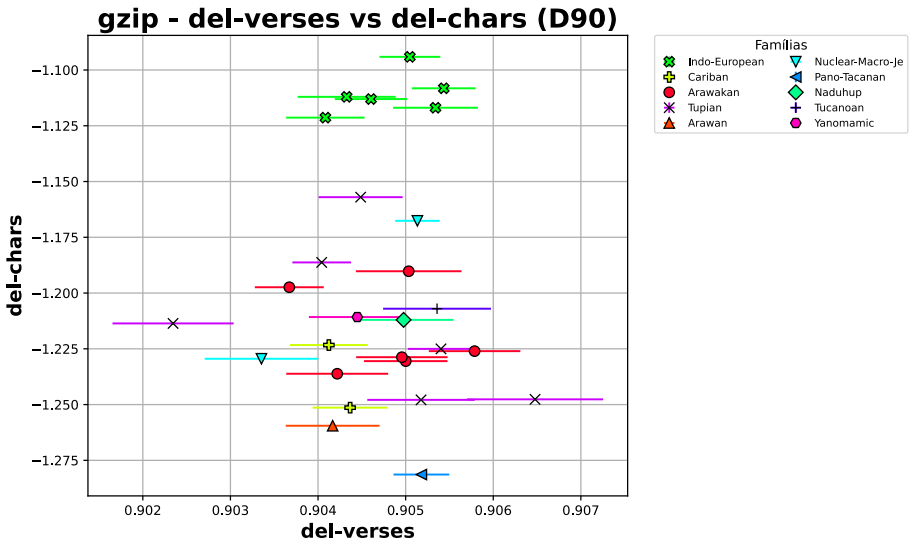

Mean and SEM by Language (none - rep-words - Dall)
Mean
Língua

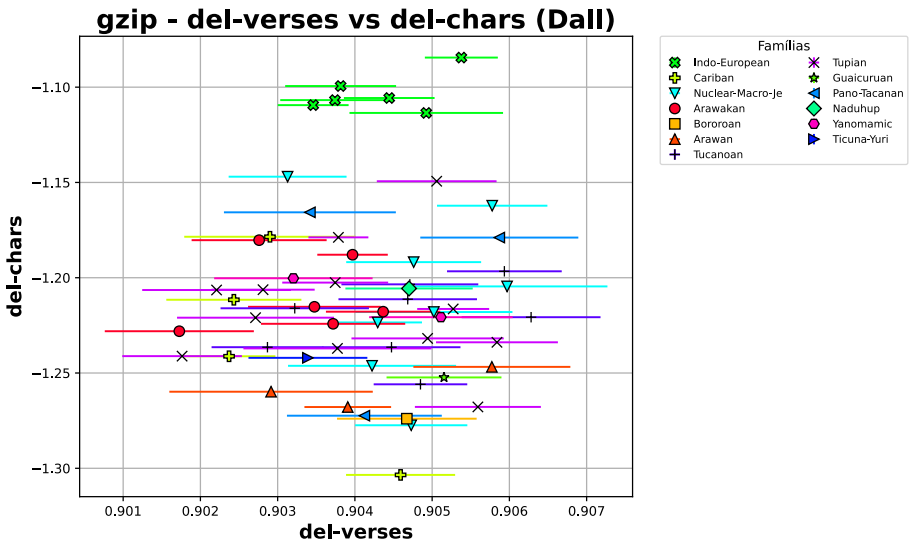

Mean and SEM by Language (none - do-nothing - D90)
Mean
Língua

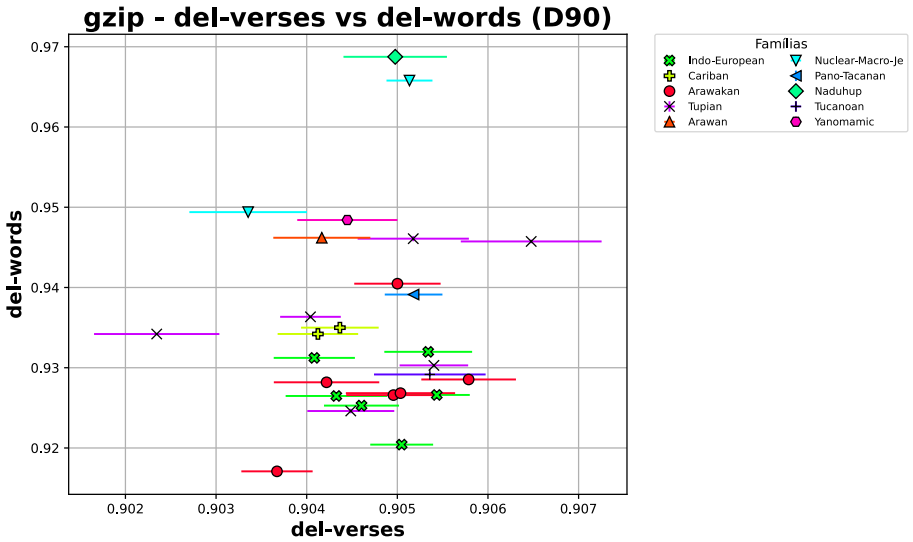

Mean and SEM by Language (none - do-nothing - Dall)
Mean
Língua

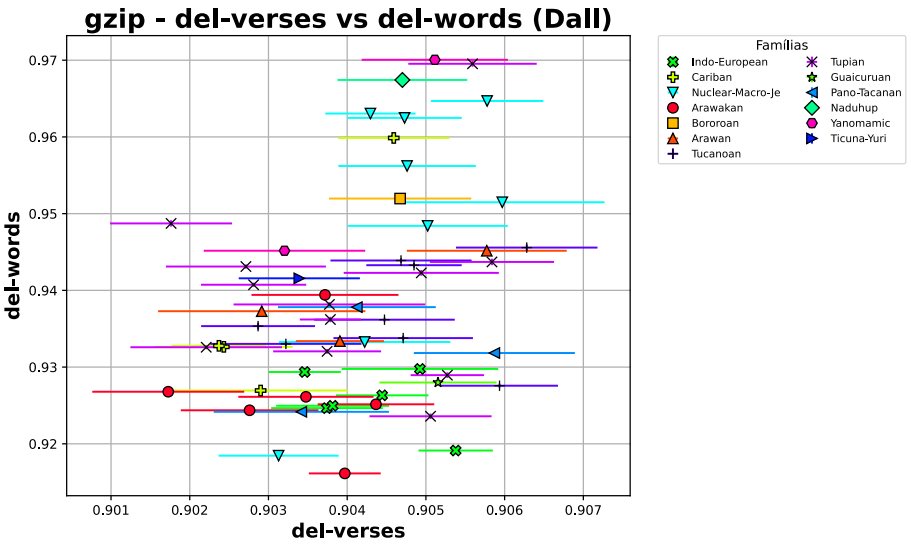
10.2 Resultados com Gráficos de Dispersão

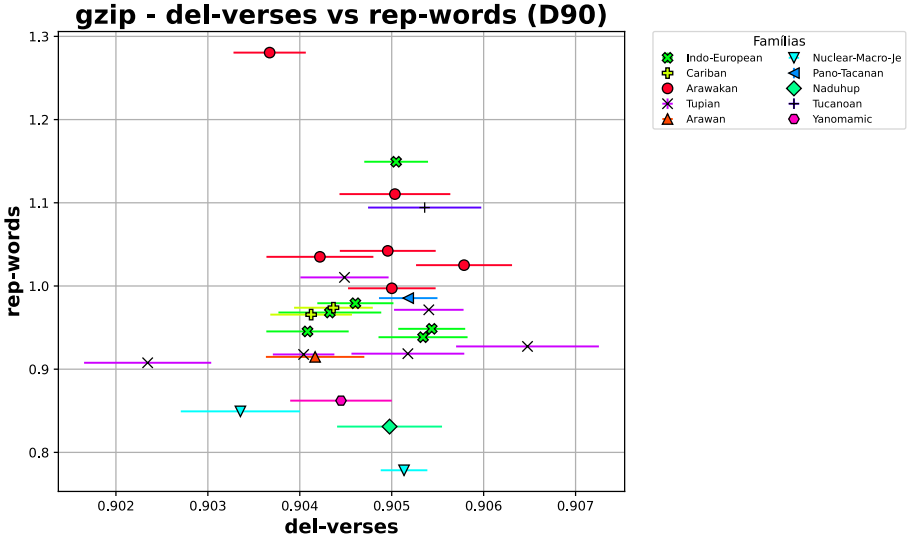
Nas figuras a seguir, nota-se que a dispersão do erro se torna visível (de maior ordem) para métricas sintáticas e pragmáticas, mas nunca visível para morfológica. Vale observar que para a maioria das relações, a ordem do erro foi significativamente baixa. Concluindo, portanto, a consistência do processamento dos dados após as modificações da biblioteca.

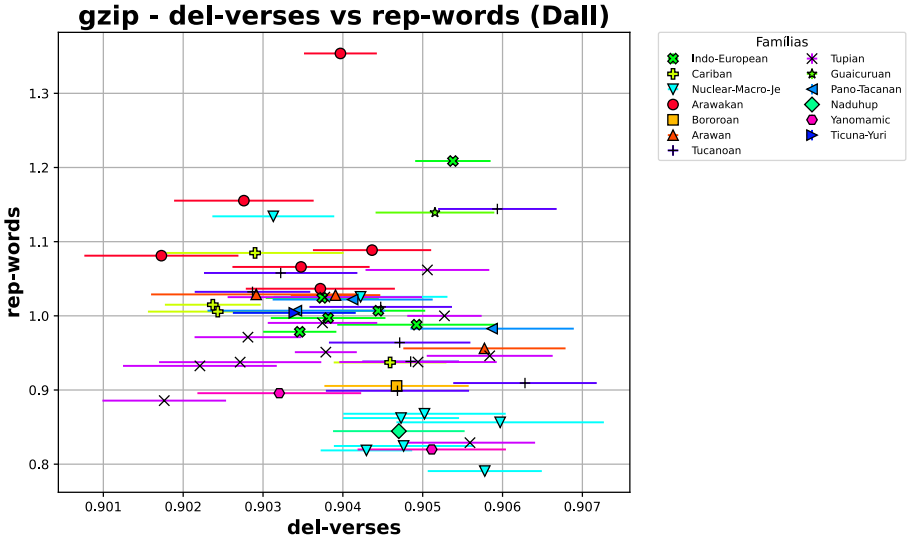

gzip - del-verses vs del-chars (D90)
del-chars
del-verses
−1.100
−1.125
−1.150
−1.175
−1.200
−1.225
−1.250
−1.275
0.902
0.903
0.904
0.905
0.906
0.907
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

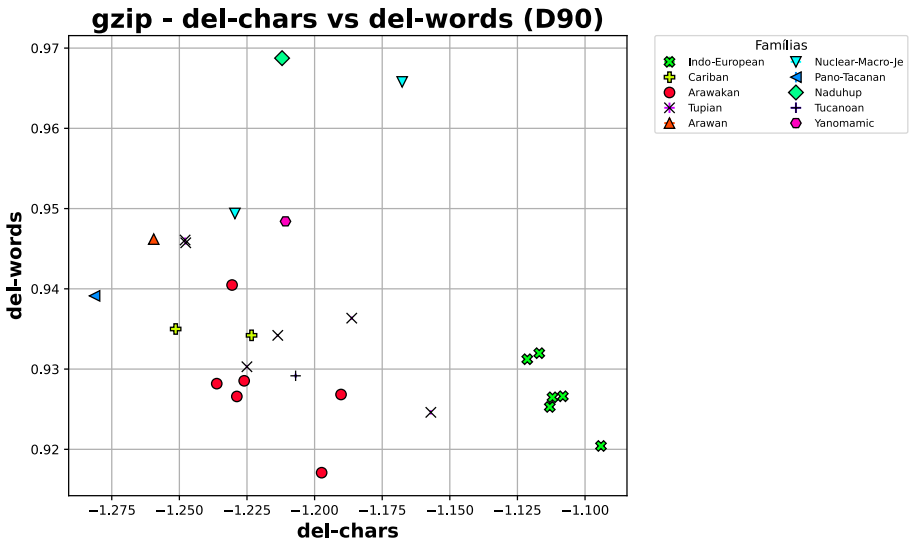

gzip - del-verses vs del-chars (Dall)
del-chars
del-verses
−1.10
−1.15
−1.20
−1.25
−1.30
0.901
0.902
0.903
0.904
0.905
0.906
0.907
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

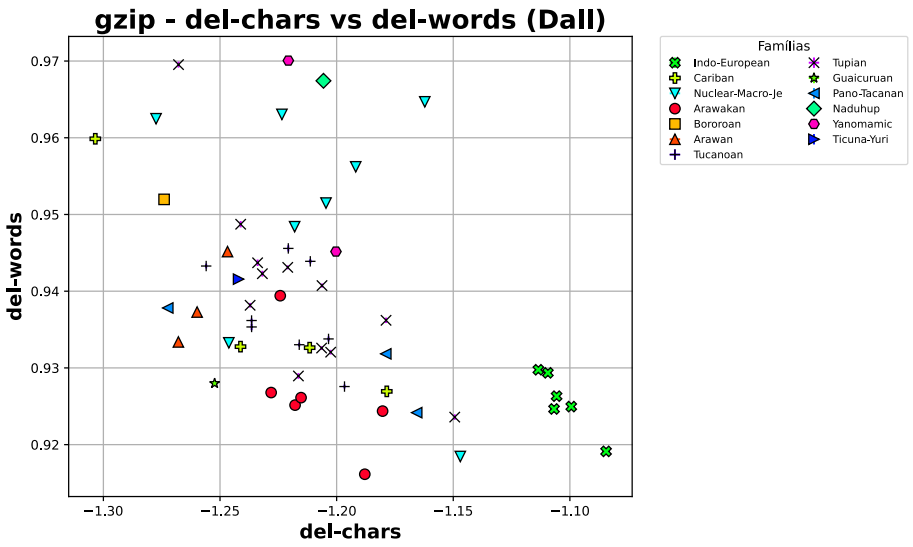

gzip - del-verses vs del-words (D90)
del-words
del-verses
0.97
0.96
0.95
0.94
0.93
0.92
0.902
0.903
0.904
0.905
0.906
0.907
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

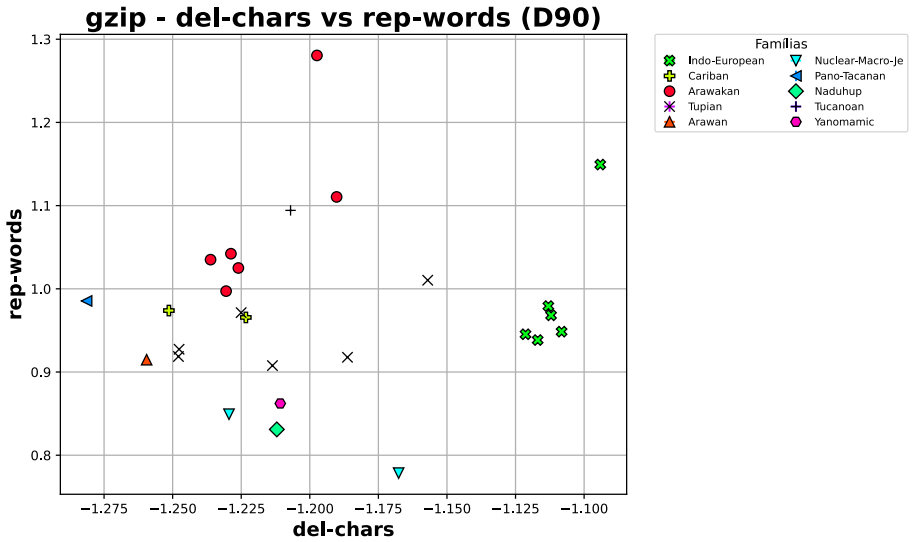

gzip - del-verses vs del-words (Dall)
del-words
del-verses
0.97
0.96
0.95
0.94
0.93
0.92
0.901
0.902
0.903
0.904
0.905
0.906
0.907
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

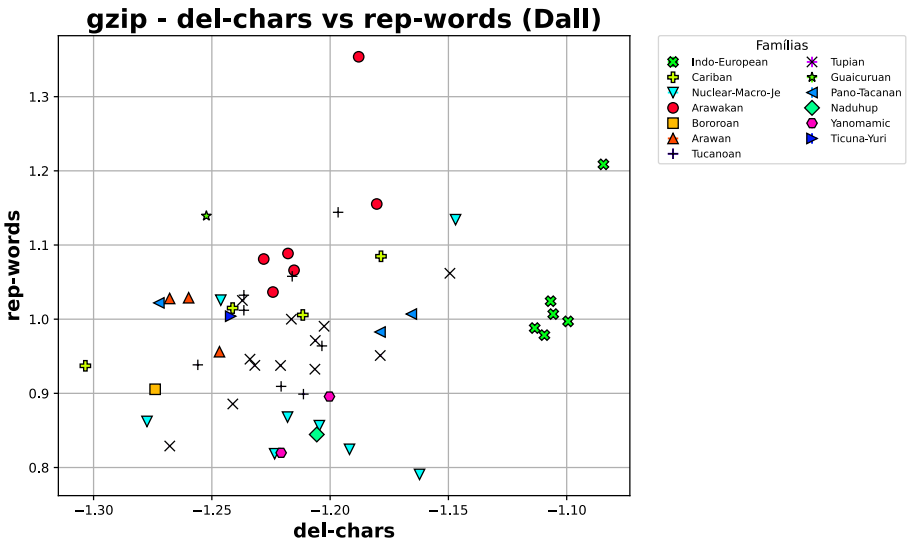

gzip - del-verses vs rep-words (D90)
rep-words
del-verses
1.3
1.2
1.1
1.0
0.9
0.8
0.902
0.903
0.904
0.905
0.906
0.907
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

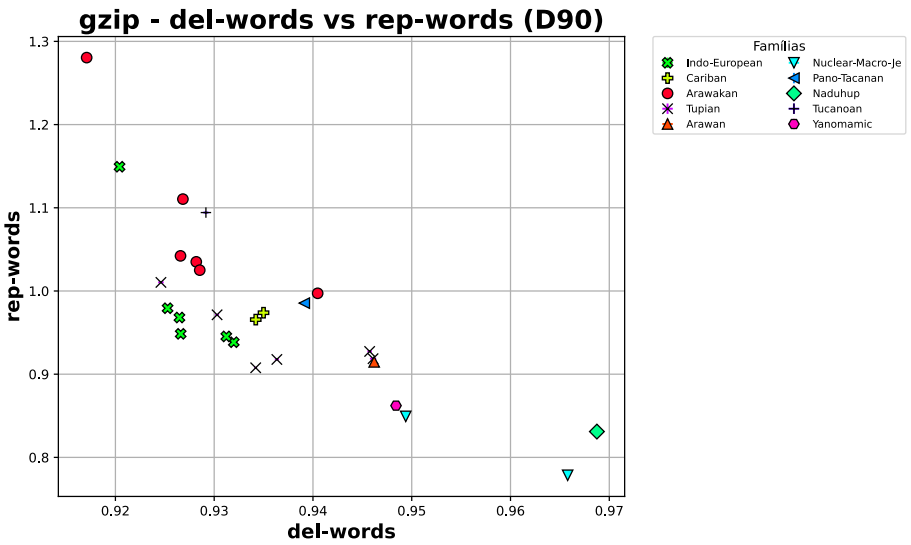

gzip - del-verses vs rep-words (Dall)
rep-words
del-verses
1.3
1.2
1.1
1.0
0.9
0.8
0.901
0.902
0.903
0.904
0.905
0.906
0.907
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

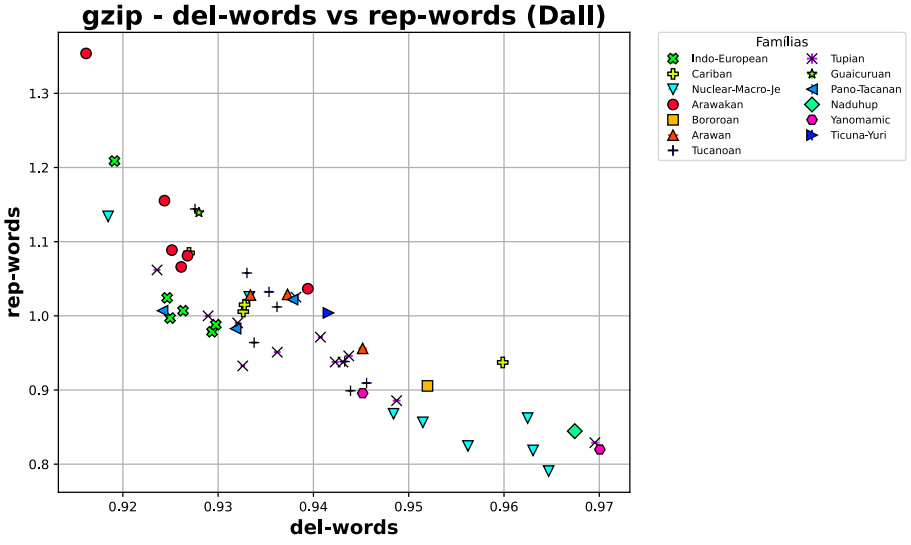

gzip - del-chars vs del-words (D90)
del-words
del-chars
0.97
0.96
0.95
0.94
0.93
0.92
−1.275
−1.250
−1.225
−1.200
−1.175
−1.150
−1.125
−1.100
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

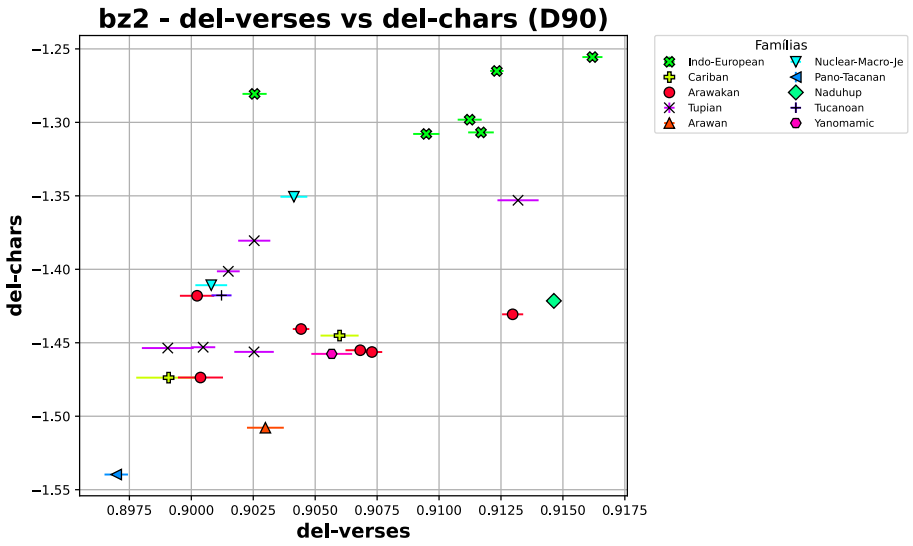

gzip - del-chars vs del-words (Dall)
del-words
del-chars
0.97
0.96
0.95
0.94
0.93
0.92
−1.30
−1.25
−1.20
−1.15
−1.10
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

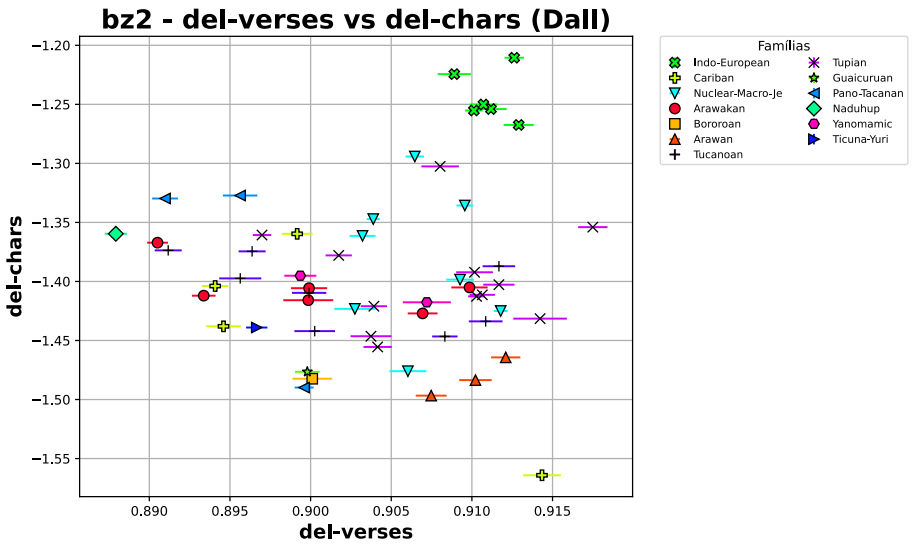

gzip - del-chars vs rep-words (D90)
rep-words
del-chars
1.3
1.2
1.1
1.0
0.9
0.8
−1.275
−1.250
−1.225
−1.200
−1.175
−1.150
−1.125
−1.100
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

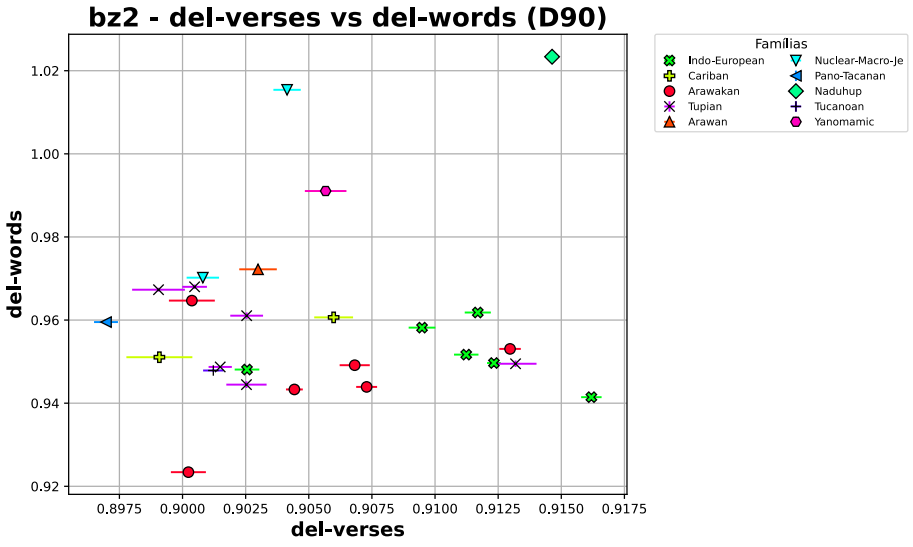

gzip - del-chars vs rep-words (Dall)
rep-words
del-chars
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

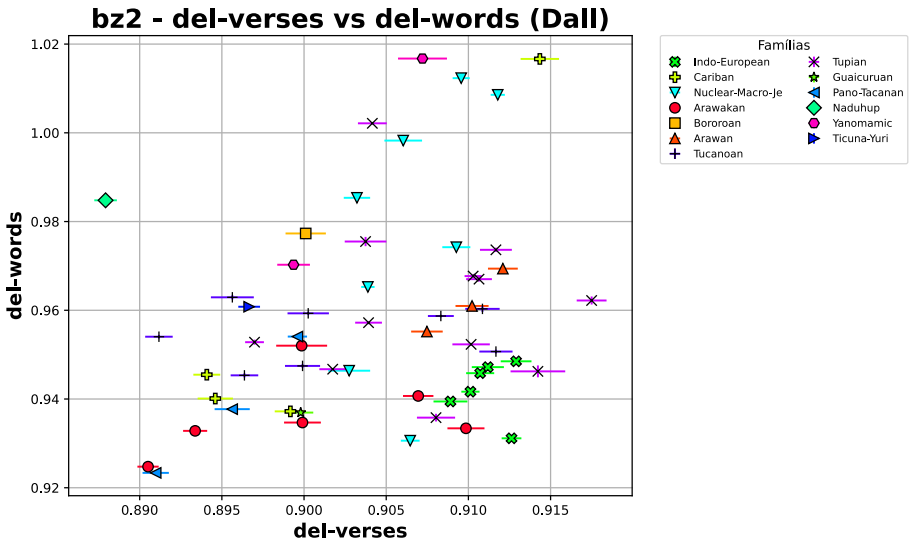

gzip - del-words vs rep-words (D90)
rep-words
del-words
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

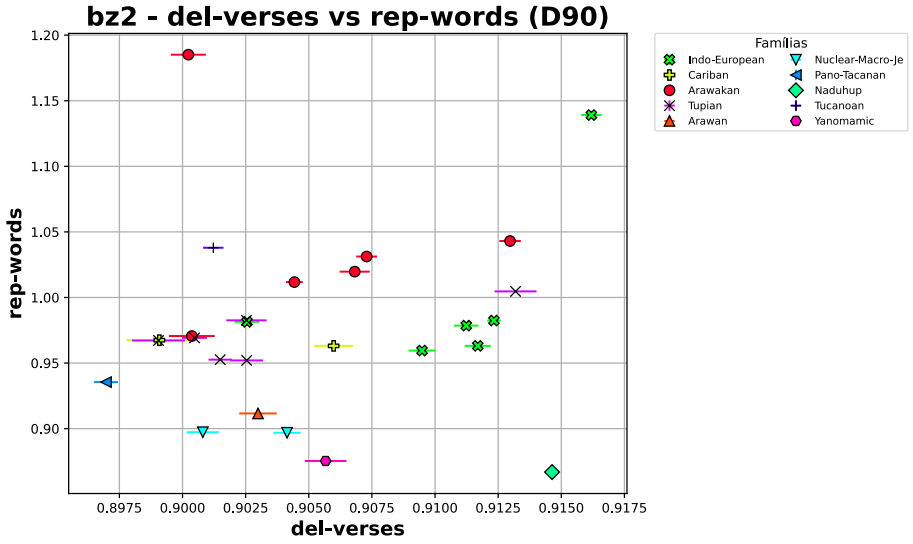

gzip - del-words vs rep-words (Dall)
rep-words
del-words
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

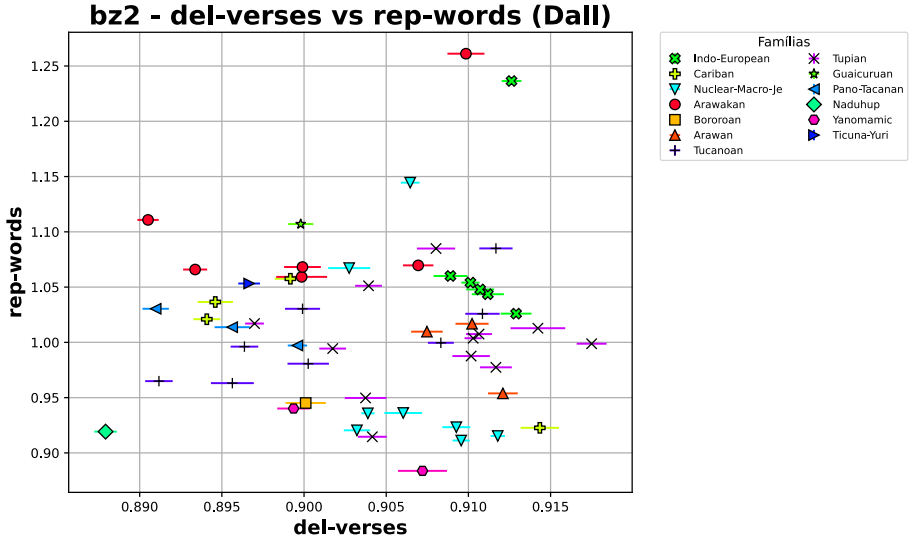

bz2 - del-verses vs del-chars (D90)
del-chars
del-verses
−1.25
−1.30
−1.35
−1.40
−1.45
−1.50
−1.55
0.8975
0.9000
0.9025
0.9050
0.9075
0.9100
0.9125
0.9150
0.9175
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

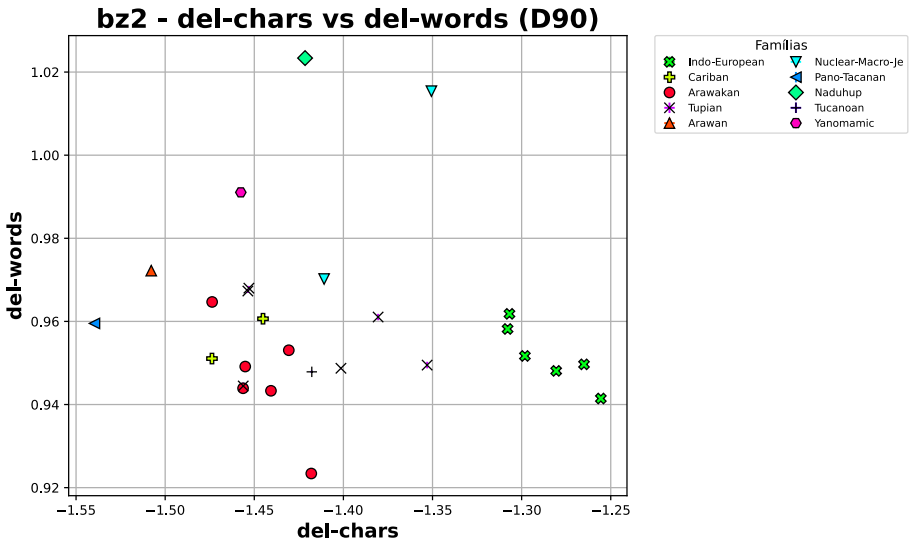

bz2 - del-verses vs del-chars (Dall)
del-chars
del-verses
−1.20
−1.25
−1.30
−1.35
−1.40
−1.45
−1.50
−1.55
0.890
0.895
0.900
0.905
0.910
0.915
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

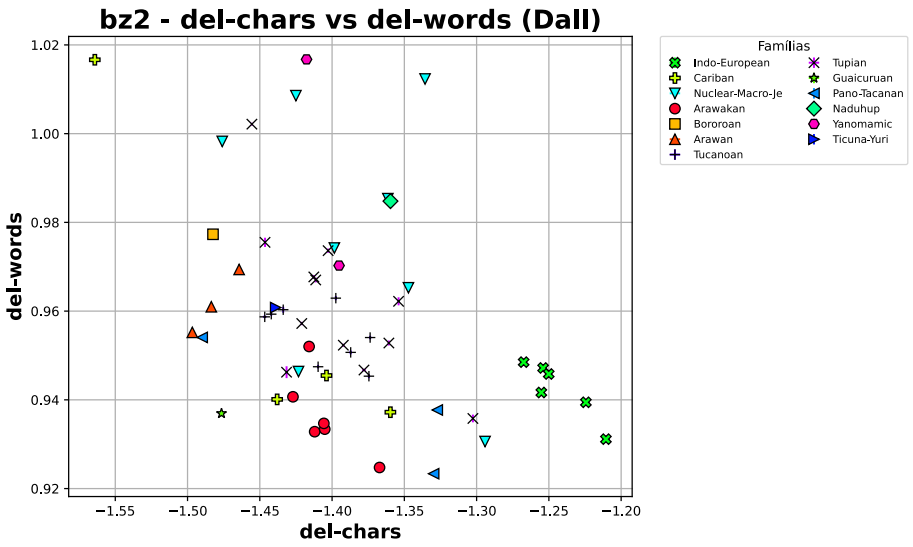

bz2 - del-verses vs del-words (D90)
del-words
del-verses
1.02
1.00
0.98
0.96
0.94
0.92
0.8975
0.9000
0.9025
0.9050
0.9075
0.9100
0.9125
0.9150
0.9175
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

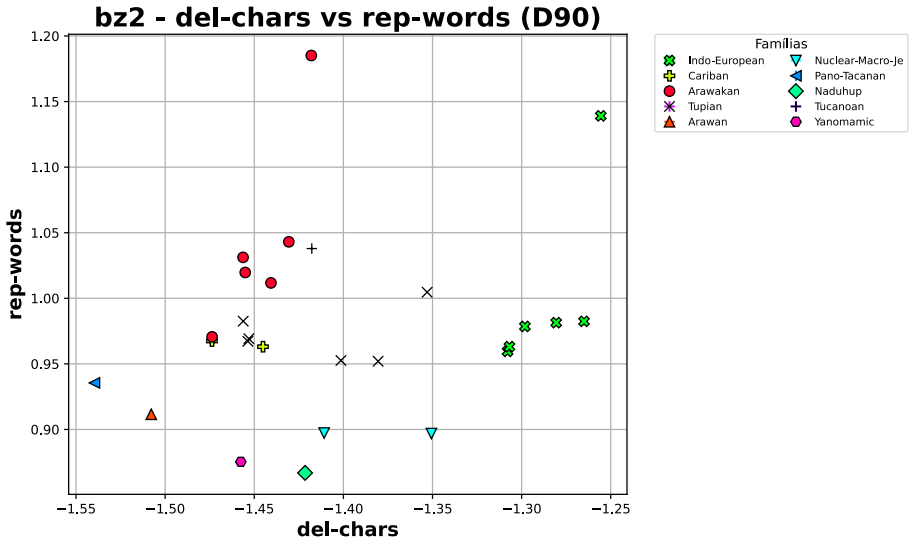

bz2 - del-verses vs del-words (Dall)
del-words
del-verses
1.02
1.00
0.98
0.96
0.94
0.92
0.890
0.895
0.900
0.905
0.910
0.915
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

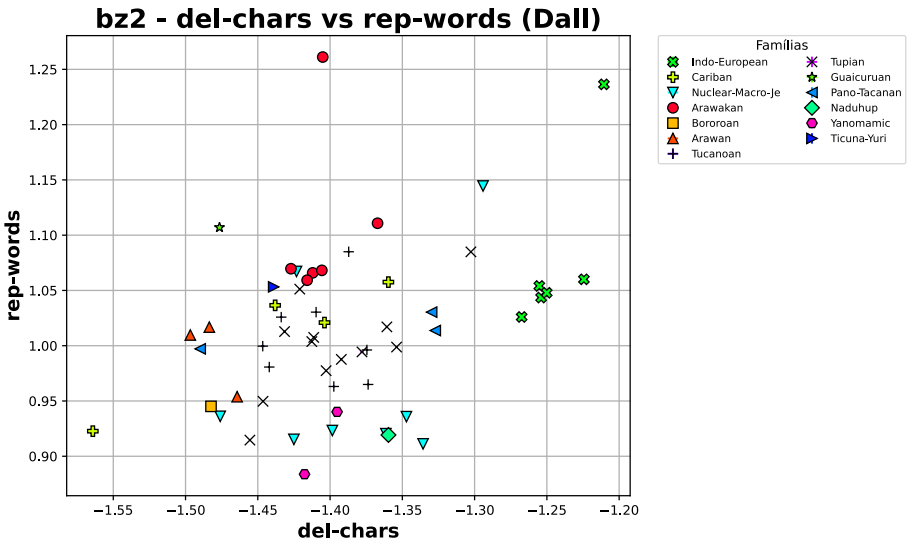

bz2 - del-verses vs rep-words (D90)
rep-words
del-verses
1.20
1.15
1.10
1.05
1.00
0.95
0.90
0.8975
0.9000
0.9025
0.9050
0.9075
0.9100
0.9125
0.9150
0.9175
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

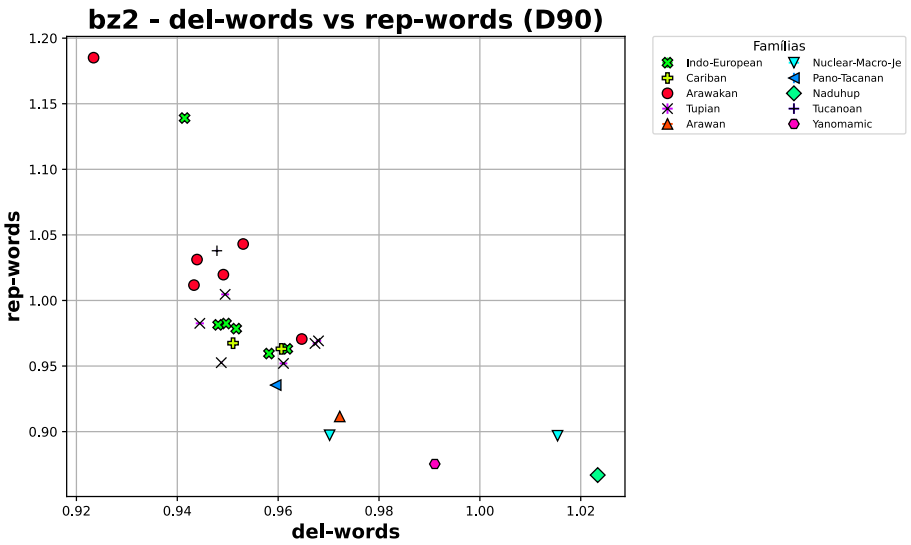

bz2 - del-verses vs rep-words (Dall)
rep-words
del-verses
1.25
1.20
1.15
1.10
1.05
1.00
0.95
0.90
0.890
0.895
0.900
0.905
0.910
0.915
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

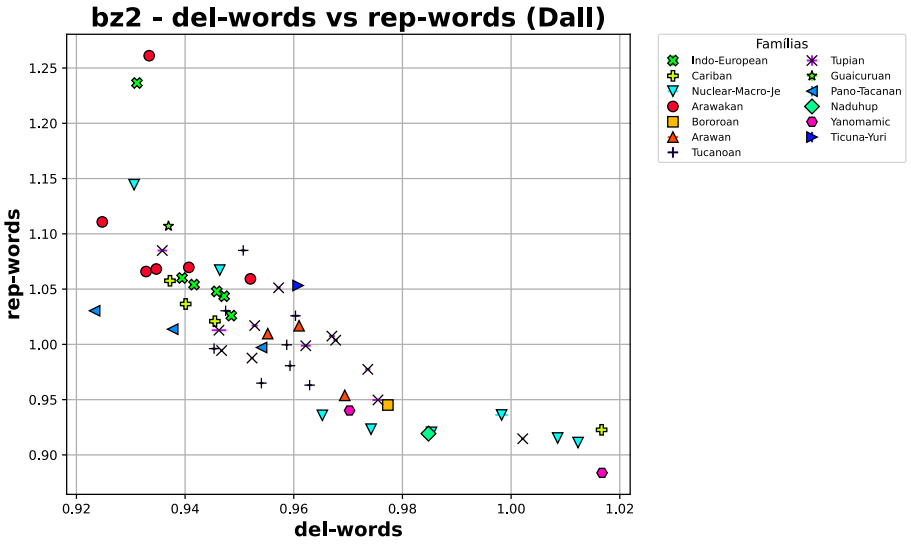

bz2 - del-chars vs del-words (D90)
del-words
del-chars
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

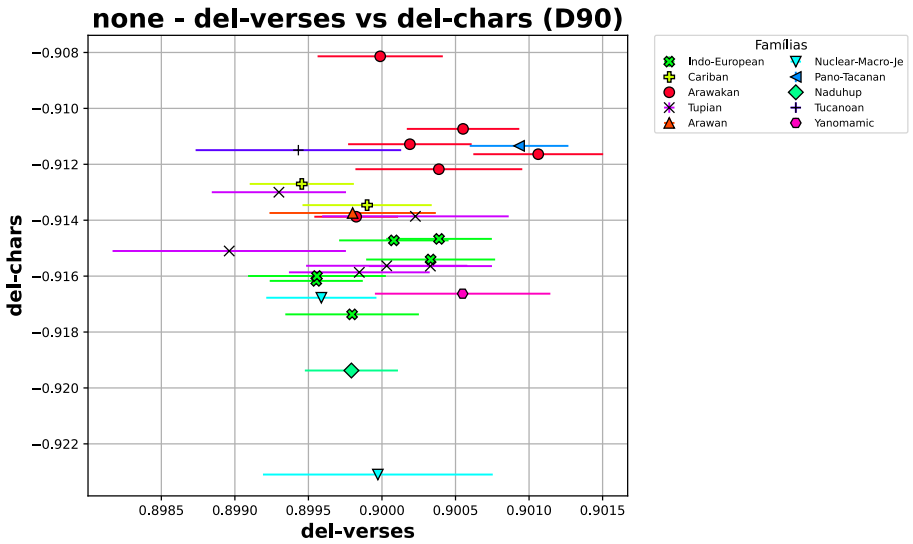

bz2 - del-chars vs del-words (Dall)
del-words
del-chars
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

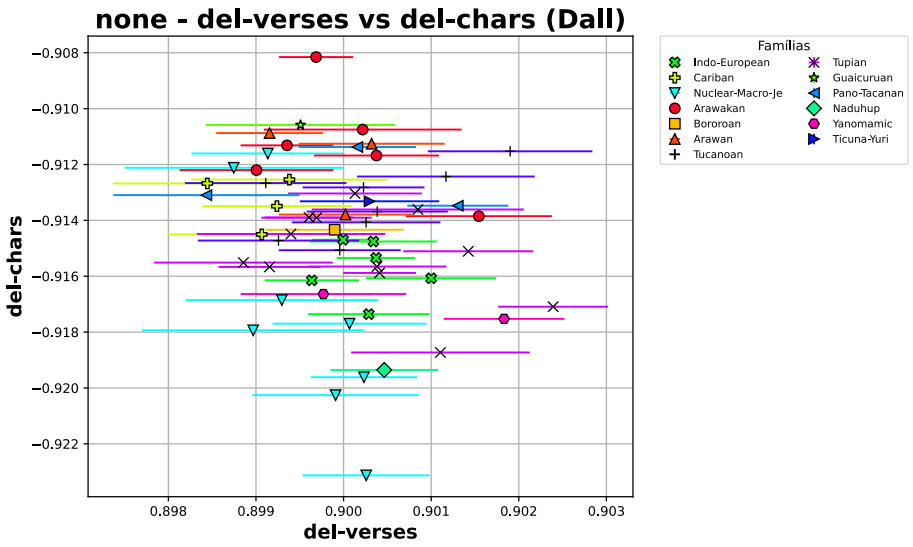

bz2 - del-chars vs rep-words (D90)
rep-words
del-chars
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

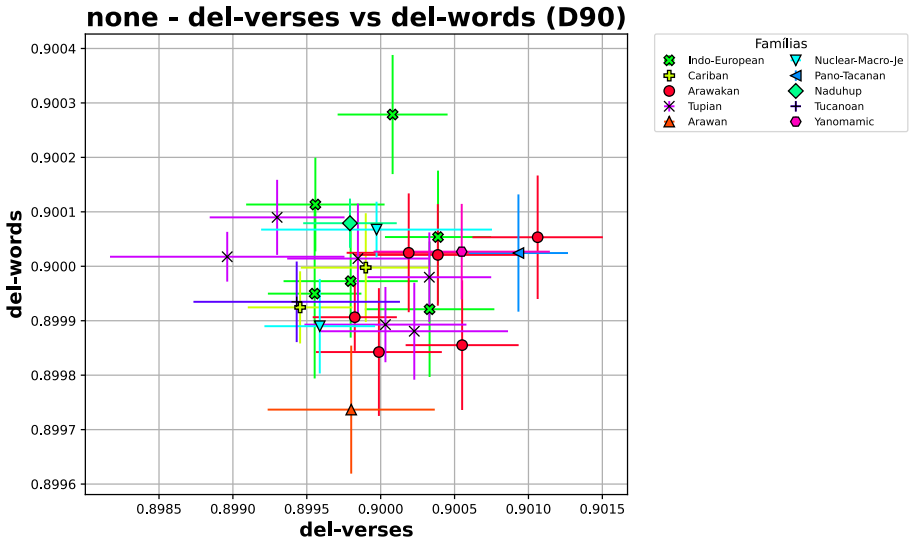

bz2 - del-chars vs rep-words (Dall)
rep-words
del-chars
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

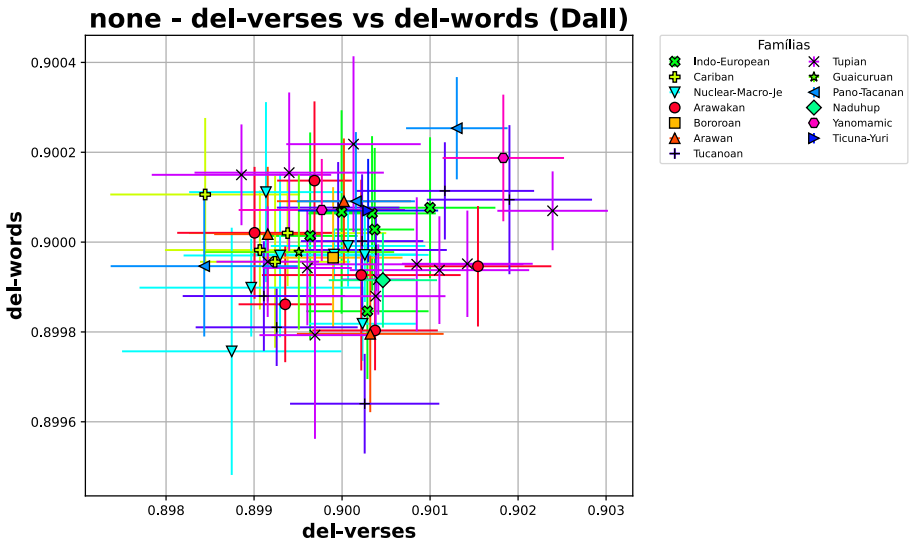

bz2 - del-words vs rep-words (D90)
rep-words
del-words
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

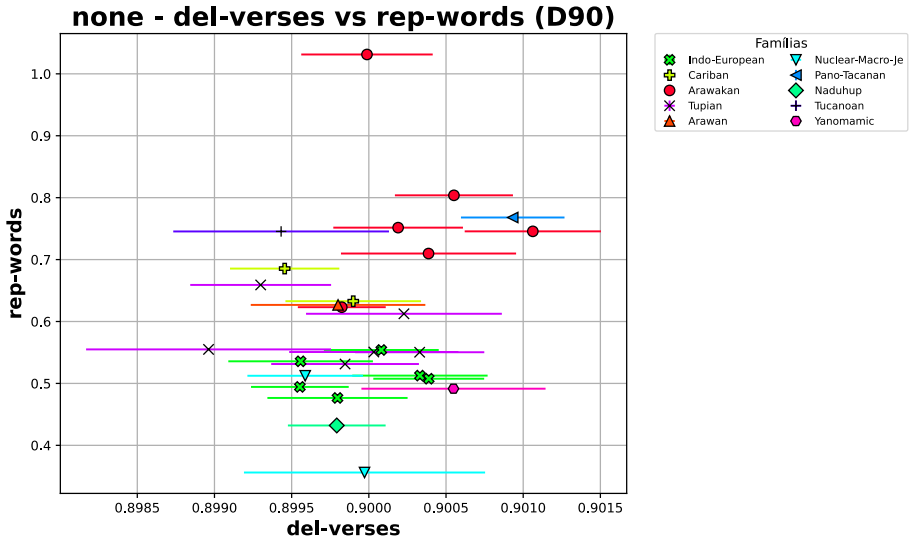

bz2 - del-words vs rep-words (Dall)
rep-words
del-words
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

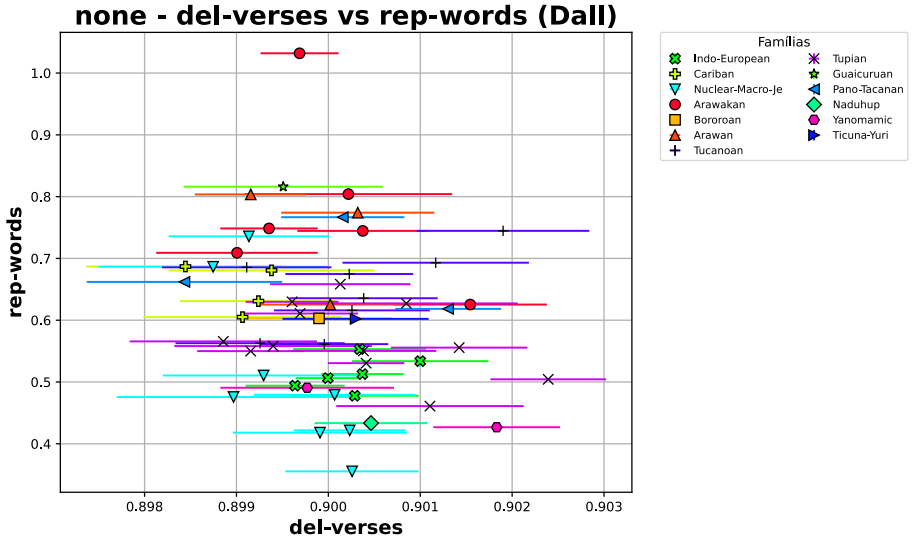

none - del-verses vs del-chars (D90)
del-chars
del-verses
−0.908
−0.910
−0.912
−0.914
−0.916
−0.918
−0.920
−0.922
0.8985
0.8990
0.8995
0.9000
0.9005
0.9010
0.9015
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

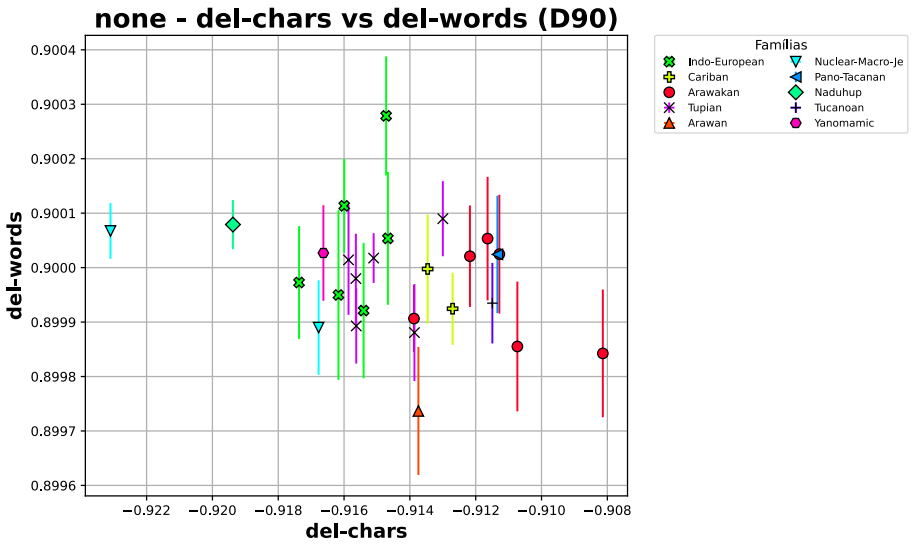

none - del-verses vs del-chars (Dall)
del-chars
del-verses
−0.908
−0.910
−0.912
−0.914
−0.916
−0.918
−0.920
−0.922
0.898
0.899
0.900
0.901
0.902
0.903
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

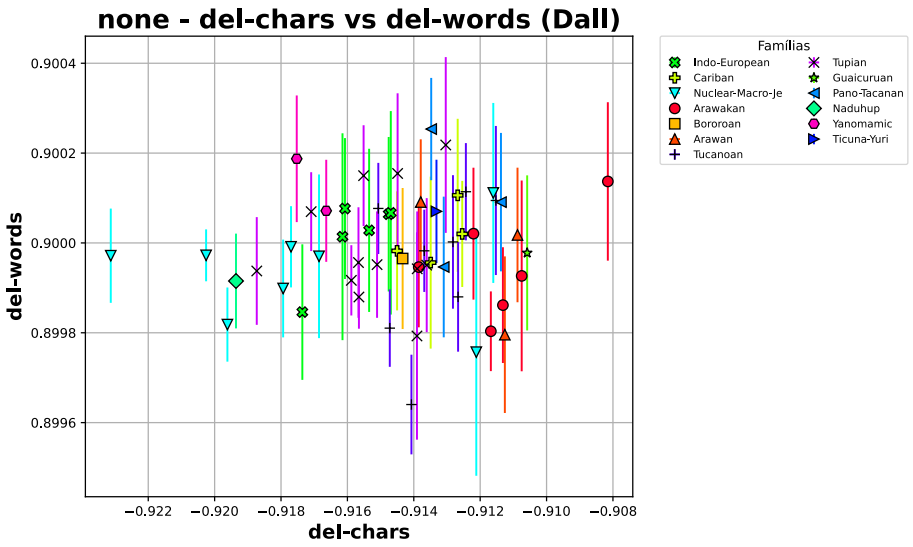

none - del-verses vs del-words (D90)
del-words
del-verses
0.9004
0.9003
0.9002
0.9001
0.9000
0.8999
0.8998
0.8997
0.8996
0.8985
0.8990
0.8995
0.9000
0.9005
0.9010
0.9015
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

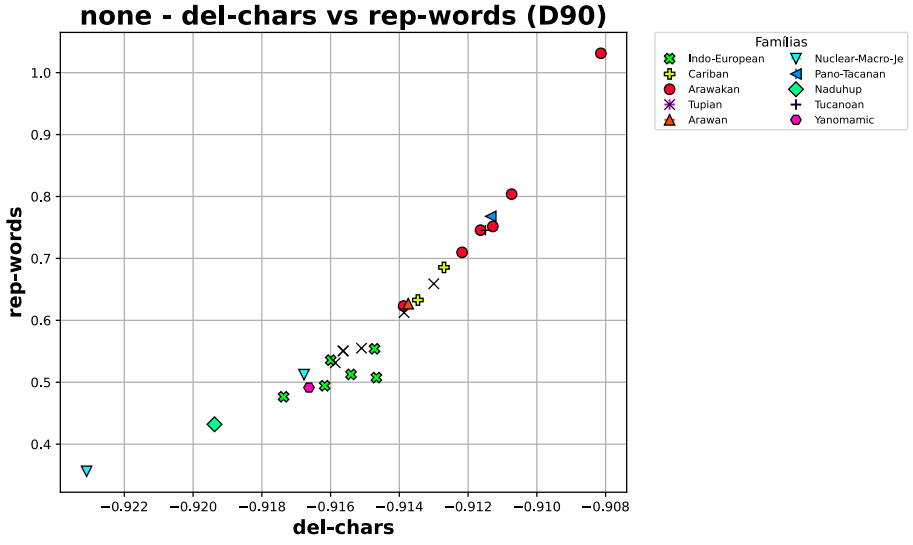

none - del-verses vs del-words (Dall)
del-words
del-verses
0.9004
0.9002
0.9000
0.8998
0.8996
0.898
0.899
0.900
0.901
0.902
0.903
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

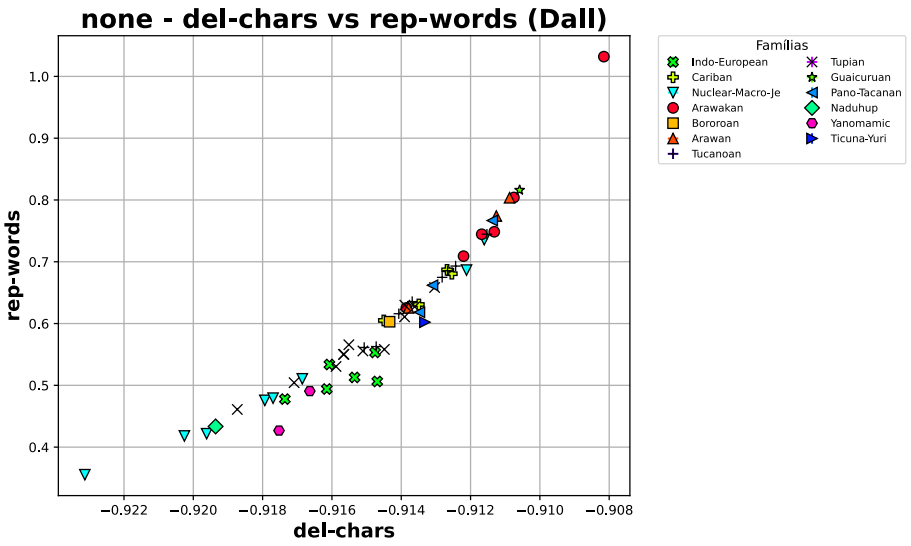

none - del-verses vs rep-words (D90)
rep-words
del-verses
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.8985
0.8990
0.8995
0.9000
0.9005
0.9010
0.9015
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic

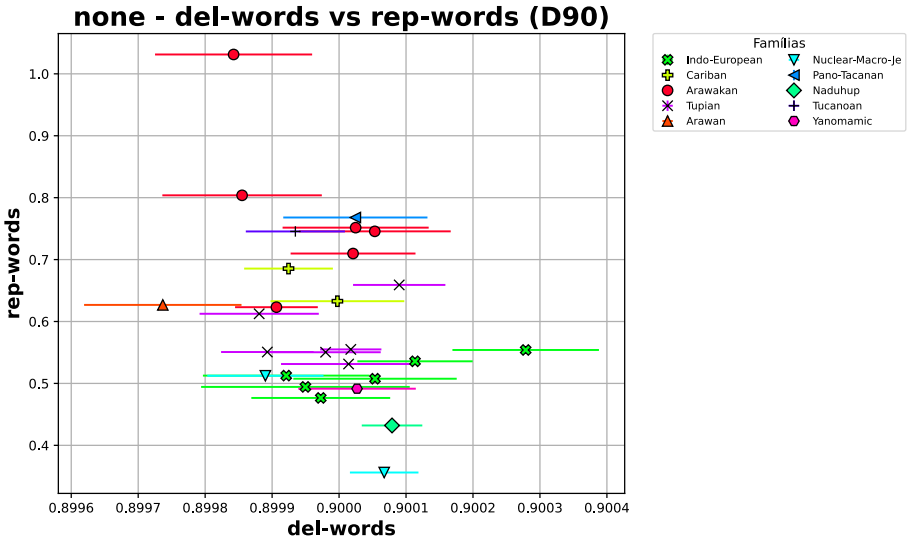

none - del-verses vs rep-words (Dall)
rep-words
del-verses
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.898
0.899
0.900
0.901
0.902
0.903
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

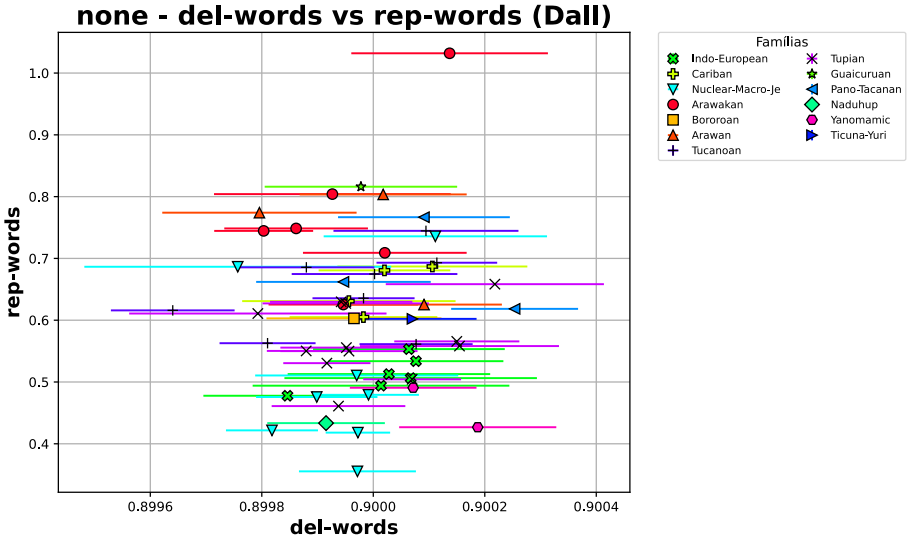

none - del-chars vs del-words (D90)
del-words
del-chars
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic


none - del-chars vs del-words (Dall)
del-words
del-chars
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri


none - del-chars vs rep-words (D90)
rep-words
del-chars
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic


none - del-chars vs rep-words (Dall)
rep-words
del-chars
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri


none - del-words vs rep-words (D90)
rep-words
del-words
Famílias
Indo-European
Cariban
Arawakan
Tupian
Arawan
Nuclear-Macro-Je
Pano-Tacanan
Naduhup
Tucanoan
Yanomamic


none - del-words vs rep-words (Dall)
rep-words
del-words
Famílias
Indo-European
Cariban
Nuclear-Macro-Je
Arawakan
Bororoan
Arawan
Tucanoan
Tupian
Guaicuruan
Pano-Tacanan
Naduhup
Yanomamic
Ticuna-Yuri

Capítulo 11

Conclusão

O presente trabalho explorou a complexidade linguística através de uma abordagem baseada na Teoria da Informação, utilizando a compressão de textos para medir a complexidade em diferentes níveis: morfológico, sintático e pragmático. A implementação da biblioteca *lang-complexity* objetivou proporcionar uma ferramenta robusta para análise de complexidade linguística, permitindo a aplicação das métricas propostas por Juola [8] a uma ampla gama de línguas, incluindo aquelas com características distintas das línguas indo-europeias.

A análise das hipóteses e observações de Juola no contexto de línguas indígenas sul-americanas corroborou a ideia de equicomplexidade entre línguas, ressaltando que, embora as línguas possam diferir em quais níveis a complexidade se manifesta, a complexidade total tende a ser equivalente. Isso reforça a noção de que todas as línguas carregam uma riqueza estrutural única, sem que uma seja inerentemente superior a outra em termos de complexidade.

A hipótese de *Trade-off* sintático-morfológico destacou-se particularmente ao evidenciar que, para maioria das línguas, existe uma relação inversa entre complexidade sintática e morfológica, principalmente ao considerar subgrupos de línguas dentro de famílias linguísticas. Esta abordagem permitiu observar que o agrupamento por famílias linguísticas oferece uma correlação mais forte, com valores de R^2 mais elevados, indicando maior coerência interna. No entanto, algumas exceções, como as línguas da família **Nuclear-Macro-Je**, mostraram desvios que foram abordados através do cálculo do coeficiente de correlação de Pearson, menos sensível a outliers.

Além disso, as modificações feitas a bilioteca trouxeram uma camada adicional de robustez à análise, garantindo a precisão e consistência dos resultados obtidos. A implementação dos testes e do erro padrão da média, demonstraram que a variância entre as amostras era mínima, validando a consistência do processamento de dados pela biblioteca, e sugerindo utilidade em futuras extensões para análises mais complexas.

Portanto, este trabalho não só ampliou a aplicabilidade das métricas de complexidade linguística, como também reforçou a importância de considerar variações internas entre grupos linguísticos, oferecendo uma ferramenta aprimorada para futuras pesquisas em PLN e estudos linguísticos computacionais.

Referências

[1] https://u-channel.ca/indigenous-languages-series-inuktitut-season-2/. Acessado em: 22 de Novembro de 2024 (ver p. 5).

[2] (Ver p. 9).

[3] ACKERMAN, Farrell, MALOUF, Robert. "Morphological organization: The low conditional entropy conjecture". Em: *Language* (2013) (ver p. 3).

[4] EHRET, Katharina. "An information-theroetic view on language complexity and register variation: Compressing naturalistic corpus data." Em: *Corpus Linguistics and Linguistic Theory, v17, n. 2, p.383-410* (2021) (ver p. 3).

[5] EHRET, Katharina, DRAMÉ, Alice Blumenthal, BENTZ, Christian, BERDICEVSKIS, Aleksandrs. "Meaning and Measures: Interpreting and evaluating complexity metrics". Em: *Frontiers in Communication* (2021) (ver p. 3).

[6] FENK, August, FENK-OCZLON, Gertraud. "Complexity trade-offs between the subsystems of language". Em: (2008) (ver p. 3).

[7] HOUSEN, Alex, CLERQ, Bastien De, KUIKEN, Folkert, VERDEN, Ineke. "Multiple approaches to complexity in second language research". Em: *Second Language Research* (2019) (ver p. 3).

[8] JUOLA, Patrick. "Assessing linguistic complexity". Em: *Language Complexity: Typology, Contact, Change. John Benjamins Press, Amsterdam, Netherlands* (2008) (ver pp. 3, 6, 27, 75).

[9] JUOLA, Patrick. "Compression-based analysis of language complexity". Em: *Approaches to Complexity in Language* (2005) (ver p. 3).

[10] JUOLA, Patrick. "Measuring Linguistic complexity: The morphological tier". Em: *Journal of Quantitative Lin-Druckguistics* (1998) (ver p. 3).

[11] JUVONEN, Paivi. "Complexity and simplicity in minimal lexica: The lexicon of Chinook Jargon". Em: (2008) (ver p. 3).

[12] KETTUNEN, Kimmo. "Can type-token ratio be used to show morphological complexity of languages?" Em: *Journal of Quantitative Linguistics* (2014) (ver p. 3).

[13] NICHOLS, Johanna. "Linguistic diversity in space and time". Em: *University of Chicago Press* (1998) (ver p. 3).

[14] PELLEGRINO, François. "Across-language perspective on speech information rate". Em: *Language* (2011) (ver p. 3).

[15] SADENIEMI, Markus, KETTUNEN, Kimmo, LINDHKNUUTILA, Tiina, HONKELA, Timo. "Complexity of european union languages: A comparative approach". Em: *Journal of Quantitative Linguistics* (2008) (ver p. 3).

[16] SERRAS, Felipe Ribas, CARPI, Miguel de Melo, BRANCO, Matheus Castello, FINGER, Marcelo. “Analysing and Validating Language Complexity Metrics Across South American Indigenous Languages”. Em: *Institute of Mathematics and Statistics, University of São Paulo* (2024) (ver pp. 2, 3, 6, 9, 27).