UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE
DEPARTAMENTO DE COMPUTAÇÃO

THIAGO GANDES FREIRE

GERAÇÃO DE GRAMÁTICAS LOCAIS A PARTIR DE EXEMPLOS

ALEGRE - ES

2021

GERAÇÃO DE GRAMÁTICAS LOCAIS A PARTIR DE EXEMPLOS

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

por

THIAGO GANDES FREIRE

Orientador
Juliana Pinheiro Campos Pirovani

Universidade Federal do Espírito Santo

ALEGRE - ES

2021

THIAGO GANDES FREIRE

GERAÇÃO DE GRAMÁTICAS LOCAIS A PARTIR DE EXEMPLOS

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 12 de maio de 2021.

Profa. Dr. Juliana Pinheiro campos Pirovani
Universidade Federal do Espírito Santo
Orientadora

Prof. Dr. Edmar Hell Kampke
Universidade Federal do Espírito Santo

Profa. Valeria Alves da Silva
Universidade Federal do Espírito Santo

ALEGRE – ES
MAIO 2021

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA

O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
VALERIA ALVES DA SILVA - SIAPE 1207469
Departamento de Computação - DC/CCENS
Em 19/05/2021 às 21:52

Para verificar as assinaturas e visualizar o documento original acesse o link:
https://api.lepisma.ufes.br/arquivos-assinados/193533?tipoArquivo=O

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA

O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
EDMAR HELL KAMPKE - SIAPE 1804453
Departamento de Computação - DC/CCENS
Em 20/05/2021 às 00:40

Para verificar as assinaturas e visualizar o documento original acesse o link:
https://api.lepisma.ufes.br/arquivos-assinados/193598?tipoArquivo=O

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

PROTOCOLO DE ASSINATURA

O documento acima foi assinado digitalmente com senha eletrônica através do Protocolo Web, conforme Portaria UFES nº 1.269 de 30/08/2018, por
JULIANA PINHEIRO CAMPOS PIROVANI - SIAPE 1702411
Departamento de Computação - DC/CCENS
Em 21/05/2021 às 09:21

Para verificar as assinaturas e visualizar o documento original acesse o link:
https://api.lepisma.ufes.br/arquivos-assinados/194518?tipoArquivo=O

SUMÁRIO

LISTA DE FIGURAS

LISTA DE SIGLAS

PLN Processamento de Linguagem Natural

EI Extração de Informações

GL Gramáticas Locais

REN Reconhecimento de Entidades Nomeadas

RESUMO

Este trabalho trata da elaboração e desenvolvimento do “Gerador de GL”, uma ferramenta geradora de regras para Processamento de Linguagem Natural, representadas como gramáticas locais, de modo que o usuário não precise de conhecimento computacional ou linguístico para construí-las. As regras geradas são compatíveis com o ambiente de Processamento de Linguagem Natural Unitex. Para este fim, o texto é segmentado, tokenizado e é gerado um arquivo com informações como lema e as categorias POS dos tokens utilizando o próprio conjunto de ferramentas disponíveis no Unitex. O arquivo é submetido a um algoritmo de descoberta de eventos frequentes dentro dos episódios para generalização e, finalmente, é construída a regra resultante.

Palavras-chave: Geração Automática de Regras; Gramáticas Locais; Exemplos; Unitex.

1 INTRODUÇÃO

Na era da informação, principalmente com o advento da internet que atualmente está amplamente difundida por todo o mundo, tem-se uma quantidade gigantesca de dados disponíveis (LOHR, 2012). Com o intuito de aproveitar o potencial informativo desses dados, áreas de Inteligência Artificial como Aprendizado de Máquina, Processamento de Linguagem Natural (PLN) e Reconhecimento de Padrões tem avançado com o objetivo de entender e extrair conhecimento desses grandes acervos de dados.

A Extração de Informação (EI) pode ser vista como uma aplicação de Processamento de Linguagem Natural (PLN) que visa a extração de informação estruturada a partir de dados não estruturados (comentários em redes sociais, *tweets*, *feedback* em sites de compras, textos em sites de notícias, etc) que possam ser usados por empresas na análise do perfil de seus consumidores ou usuários, por instituições governamentais e políticas para tomar ciência das intenções e necessidades dos eleitores, para estudos linguísticos e outras atividades.

O PLN não se limita a EI, é uma área rica e promissora com diversas aplicações possíveis como: mineração de texto, detectores e corretores de erros para editores de texto, sistemas interativos de aprendizagem com linguagem natural escrita e falada, sistemas que transcrevem a linguagem falada para linguagem escrita, *chatbots*, assistentes virtuais etc. O PLN auxilia a compreensão da língua natural, facilitando a identificação das informações nos textos.

> O Processamento de Linguagem Natural é uma área de pesquisa e aplicação que explora como computadores podem ser usados para entender e manipular linguagem natural escrita ou falada para realizar coisas úteis. Pesquisadores de PLN empenham-se em reunir conhecimentos sobre como os seres humanos entendem e usam linguagem para que ferramentas e técnicas apropriadas possam ser desenvolvidas para tornar os sistemas de computador capazes de entender e manipular linguagens naturais para executar as tarefas desejadas (CHOWDHURY, 2003).

Parte do que as aplicações de PLN precisam realizar é reconhecer alguns padrões

recorrentes na linguagem natural, seja escrita ou falada. Gramática Local (GL) é uma técnica de abordagem linguística usada para representar esses padrões para PLN muito útil para EI.

Segundo (GROSS, 1999), Gramáticas locais são gramáticas de estados finitos ou autômatos de estados finitos que representam conjuntos de expressões de uma língua natural. Estas gramáticas são formadas por regras elaboradas manualmente que representam padrões. Por sua vez, tais padrões representam um grupo de sentenças de sintaxe e semântica semelhantes.

Neste trabalho foi realizado o desenvolvimento de uma ferramenta com interface amigável para facilitar a construção de GL's para uma ferramenta chamada Unitex. Ao final a ferramenta desenvolvida foi capaz de receber exemplos textuais e construir a GL capaz de reconhecer os padrões fornecidos.

1.1 Problema e sua Importância

Segundo Lima, Nunes e Vieira (2007), a área de Processamento de Linguagem Natural evoluiu bastante nas últimas décadas, sendo desenvolvida em importantes universidades e centros de pesquisa pelo mundo. No atual contexto da sociedade da informação, o PLN é fundamental para o desenvolvimento de novas tecnologias. No entanto, os recursos disponíveis ainda não acompanham as demandas existentes.

O Inglês atualmente é a língua com maior disponibilidade de recursos para PLN. A situação do Português é menos favorável, pois possui menos recursos. As aplicações desenvolvidas para PLN não são facilmente adaptáveis a várias línguas. Isso se deve às peculiaridades linguísticas de cada idioma, como classes gramaticais diferentes por exemplo. Assim sendo, adaptar um recurso desenvolvido em outra língua para o português, mesmo se for possível, não é uma tarefa simples.

Existem três principais abordagens usadas nas tarefas de PLN: linguística, probabilística e híbrida. O aspecto importante aqui, é que as abordagens linguística e probabilística são complementares e portanto são frequentemente usadas em conjunto, o que resulta na abordagem híbrida. No entanto, este trabalho se encaixa apenas no espectro de abordagem linguística, mesmo que possa ser utilizado futuramente junto a outras abordagens. Esta abordagem se baseia em aspectos linguísticos e se aproxima mais da forma humana de entender a língua do que a abordagem probabilística

por exemplo, que se baseia em métodos matemáticos. Embora as regras da abordagem linguística se beneficiem da inteligência humana e capturem detalhes que outras abordagens não capturam, essas regras precisam ser construídas manualmente e demandam conhecimento linguístico e conhecimento de ferramentas computacionais que empregam esse tipo de regra.

Picoli et al. (2015), em uma colaboração entre linguistas e cientistas da computação, concluíram que foi possível facilitar o estudo linguístico de propriedades sintático-semânticas de verbos utilizando o Unitex para construção de Gramáticas Locais para recuperar exemplos em corpora. Também notou-se que seria bem acolhido um sistema mais amigável para a construção das GL's, de forma que linguistas e outros possíveis interessados tenham autonomia para construir GL's sem a necessidade de aprender a utilizar ferramentas computacionais.

Pirovani (2019), ao trabalhar com uma abordagem híbrida (linguística e aprendizado de máquina) para a tarefa de Reconhecimento de Entidades Nomeadas(REN), notou melhorias no desempenho ao utilizar uma GL com a técnica de aprendizado de máquina CRF. Além disso, constatou que poucas modificações nas GL's utilizadas promoveram um ganho positivo nas métricas computadas, destacando a importância da adaptabilidade destas quando é necessário processar um corpus diferente do corpus de treino ou em situações em que o corpus de treino não é suficientemente grande.

A abordagem linguística é uma abordagem poderosa e complementar às outras abordagens, entretanto sua construção depende de habilidades e esforço humano. Este trabalho pretende facilitar a construção destas regras minimizando o esforço humano envolvido.

1.2 Objetivo Geral

Desenvolver uma ferramenta com interface amigável para construção semi-automática de GL's para o ambiente de PLN Unitex. Ao final, espera-se que a ferramenta desenvolvida seja capaz de receber exemplos textuais e construir regras capazes de reconhecer os padrões pretendidos.

1.3 Objetivos Específicos

1. Compreender como as gramáticas locais são representadas internamente no Unitex, qual o formato dos arquivos de GL's gerados pelo Unitex para que a ferramenta possa gerá-lo adequadamente posteriormente.

2. Elaborar uma estratégia para identificar e generalizar padrões nos exemplos selecionados.

3. Implementar a ferramenta.

4. Selecionar um conjunto de exemplos que podem ser inseridos na ferramenta pelos usuários para testar e validar a ferramenta.

5. Testar e validar a ferramenta com os exemplos selecionados.

2 REVISÃO DE LITERATURA

Este capítulo apresenta outros trabalhos e fundamentos teóricos que foram utilizados como base para o desenvolvido deste trabalho.

2.1 Gramáticas Locais e o Unitex

Gramáticas locais permitem que representemos uma família de expressões com características comuns. Segundo Gross (1997), os formalismos da abordagem linguística para representação da linguagem focavam em regras super genéricas, de modo que estas regras acabavam representando variações não desejadas e irrelevantes para o objetivo pretendido. Então, ele propôs uma nova técnica em 1993, para a abordagem linguística, com regras menos genéricas capazes de representar uma parte bem específica da linguagem. O Unitex é uma ferramenta que permite a construção e utilização desse tipo de regra.

> De acordo com Paumier (2011), "O Unitex é um conjunto de softwares que permite processar os textos em línguas naturais utilizando recursos linguísticos. Esses recursos se apresentam na forma de dicionários eletrônicos, de gramáticas e tabelas de léxico-gramática. É resultado de trabalhos iniciados no francês por Maurice Gross no “Laboratório de Automação Documental e Linguística” (LADL). Esses trabalhos foram estendidos a outras línguas através da rede de laboratórios RELEX".

Algumas das funcionalidades do Unitex são: tokenização, segmentação em frases, normalização de separadores, buscas através de expressões regulares e reconhecimento de padrão através de gramáticas locais.

As Gramáticas Locais são representadas no Unitex como um conjunto de um ou mais grafos e podem ser construídas através da interface gráfica. Para construir uma gramática local no Unitex (versão 3.1) deve-se selecionar a opção “new” no menu, FS-Graph, conforme a Figura 1.

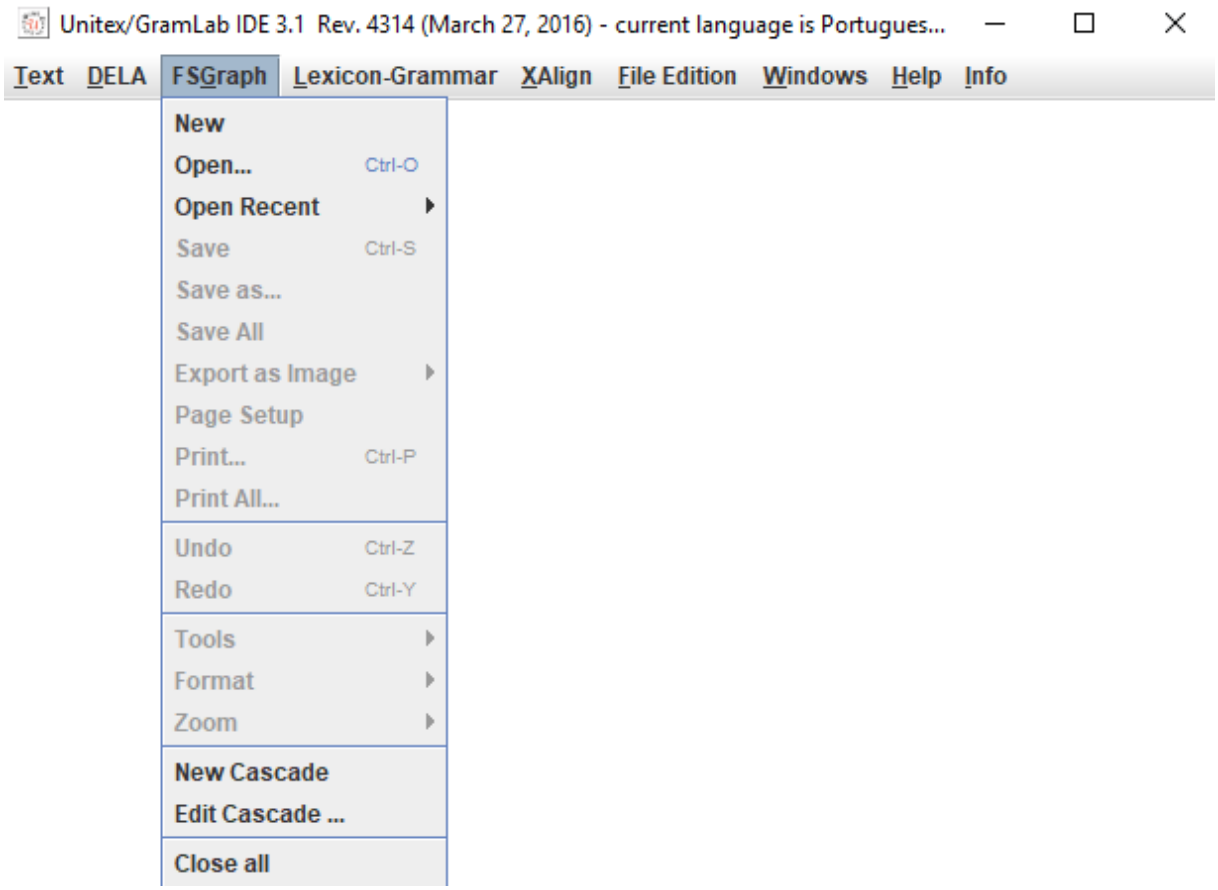


Figura 1: Menu FsGrahp.

Após selecionar a opção “new” será exibido ao usuário uma nova janela que permite a criação ou edição de grafos, conforme mostrado na Figura 2.

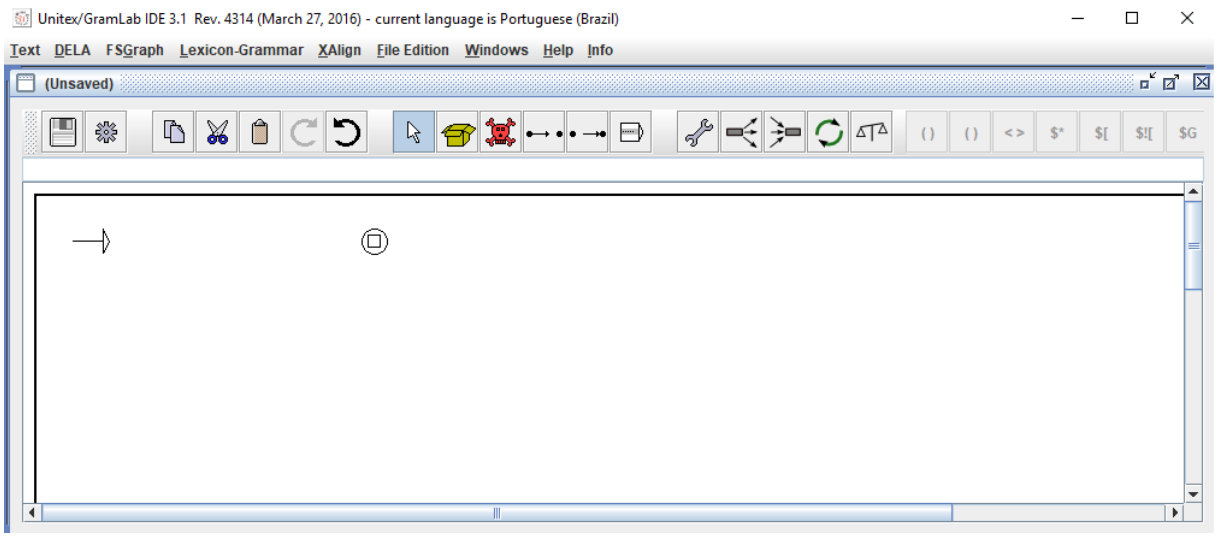


Figura 2: Interface para criação de grafos.

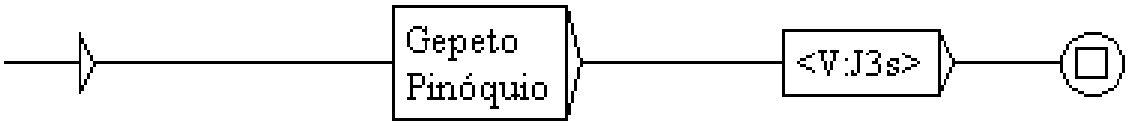
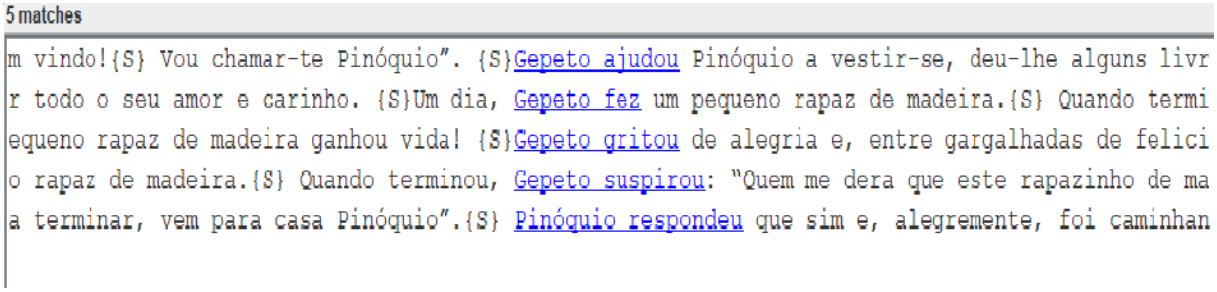


Figura 3: Exemplo de um grafo.

A Figura 3 representa uma gramática que reconhece a palavra “Gepeto” ou “Pinóquio” seguido por algum verbo no pretérito perfeito do indicativo. O código “<V:J3s>” contém meta símbolos do Unitex e representam respectivamente verbo (V), pretérito perfeito do indicativo (J), terceira pessoa (3) e singular (s). O primeiro e último nós são obrigatórios e representam respectivamente o estado inicial e final do autômato.

O Unitex permite que façamos buscas por padrões utilizando as gramáticas construídas, gerando um arquivo de concordância como saída. Este arquivo mostra os trechos reconhecidos destacados em azul entre uma parte do contexto a esquerda e a direita do trecho. No exemplo a seguir, a concordância foi apenas exibida na tela. A concordância resultante é automaticamente armazenada na pasta “_snt” e pode também ser exportada manualmente em um arquivo “.txt”.


5 matches

m vindo!{S} Vou chamar-te Pinóquio". {S}Gepeto ajudou Pinóquio a vestir-se, deu-lhe alguns livr
r todo o seu amor e carinho. {S}Um dia, Gepeto fez um pequeno rapaz de madeira.{S} Quando termi
equeno rapaz de madeira ganhou vida! {S}Gepeto gritou de alegria e, entre gargalhadas de felici
o rapaz de madeira.{S} Quando terminou, Gepeto suspirou: "Quem me dera que este rapazinho de ma
a terminar, vem para casa Pinóquio".{S} Pinóquio respondeu que sim e, alegremente, foi caminhan

Figura 4: Exemplo de concordância.

Os grafos construídos são salvos num arquivo de formato .grf. A ferramenta proposta neste trabalho precisa ser capaz de construir arquivos .grf compatíveis com o Unitex. Portanto, estes arquivos serão apresentados e explorados na seção de metodologia.

2.2 Trabalhos Correlatos

Esta seção tem por objetivo apresentar alguns trabalhos que foram realizados para extração automática de regras de abordagem linguística.

Bartoli et al. (2012) desenvolveram um sistema para geração automática de expressões regulares usando algoritmo genético. O sistema recebe um conjunto de exemplos rotulados pelo usuário e o algoritmo genético retorna a expressão regular mais adequada de acordo com a menor distância possível entre a palavra detectada e o exemplo rotulado e o menor tamanho possível do indivíduo.

Um sistema para geração automática de expressões regulares foi apresentado no trabalho de Feltrim e Souza (2009). O sistema visa gerar expressões regulares para o SciPo, um sistema que auxilia a escrita de textos acadêmicos em língua portuguesa para a área de Ciência da Computação. O sistema desenvolvido visou auxiliar o módulo AZPort, gerando expressões regulares por meio de *clustering*, para que o SciPo possa ser mais facilmente adaptado a outros domínios.

Para reconhecimento de entidades nomeadas em francês, Nouvel et al. (2011) , desenvolveram um que utiliza regras de transdução extraídas automaticamente. O sistema foi inicialmente construído para lidar com texto escrito e depois adaptado para linguagem oral. O módulo de extração de regras processa um corpus anotado com as categorias de Entidades Nomeadas realizando a lematização e o POS-tagging dos tokens de uma sentença que são organizados em uma hierarquia possibilitando a extração dos padrões.

Tatar e Cicekli (2011), desenvolveram um sistema para reconhecimento de entidades nomeadas em Turco utilizando aprendizado automático de regras. Para a aprendizagem automática das regras utiliza a tokenização, segmentação em frases, aplicação de dicionários e por fim rotulação com as características morfológicas dos tokens no texto de treino. Após o processamento inicial do texto, padrões simples são extraídos e depois generalizados para outro padrão que é validado no texto de treino.

A fonte da qual são extraídas as regras deste trabalho se assemelha a do trabalho de Bartoli et al. (2012). No entanto, os exemplos fornecidos pelos usuários não são rotulados. As expressões regulares tem um um grande poder para representação de regras, porém quando as regras crescem e se tornam mais complexas, as expressões regulares tornam-se de difícil leitura para humanos. O trabalho de Tatar e Cicekli é focado na extração de Entidades Nomeadas e para isso utilizaram uma represen-

tação própria de regras que leva em consideração uma categoria alvo de Entidade Nomeada. No trabalho de Nouvel et al. (2011), o propósito final também se concentra na extração de Entidades Nomeadas, mas a representação das regras utilizadas é a mesma utilizada neste trabalho.

3 METODOLOGIA

3.1 Tecnologias

Para desenvolver a ferramenta foi utilizada a linguagem java junto com a *Java Native Interface* (JNI) do Unitex. JNI é um padrão de programação que possibilita que o java utilize bibliotecas desenvolvidas em linguagens nativas como C, C++ e assembly, isto é: linguagens em que o código é compilado especificamente para um determinado Sistema Operacional (SO).

3.2 O formato grf

Para este trabalho construir uma gramática local compatível com o Unitex, foi estudado o formato do arquivo .grf. Para este propósito, foi construído o grafo apresentado na Figura 5, observando as características do arquivo .grf correspondente num editor de texto (Figura 6).

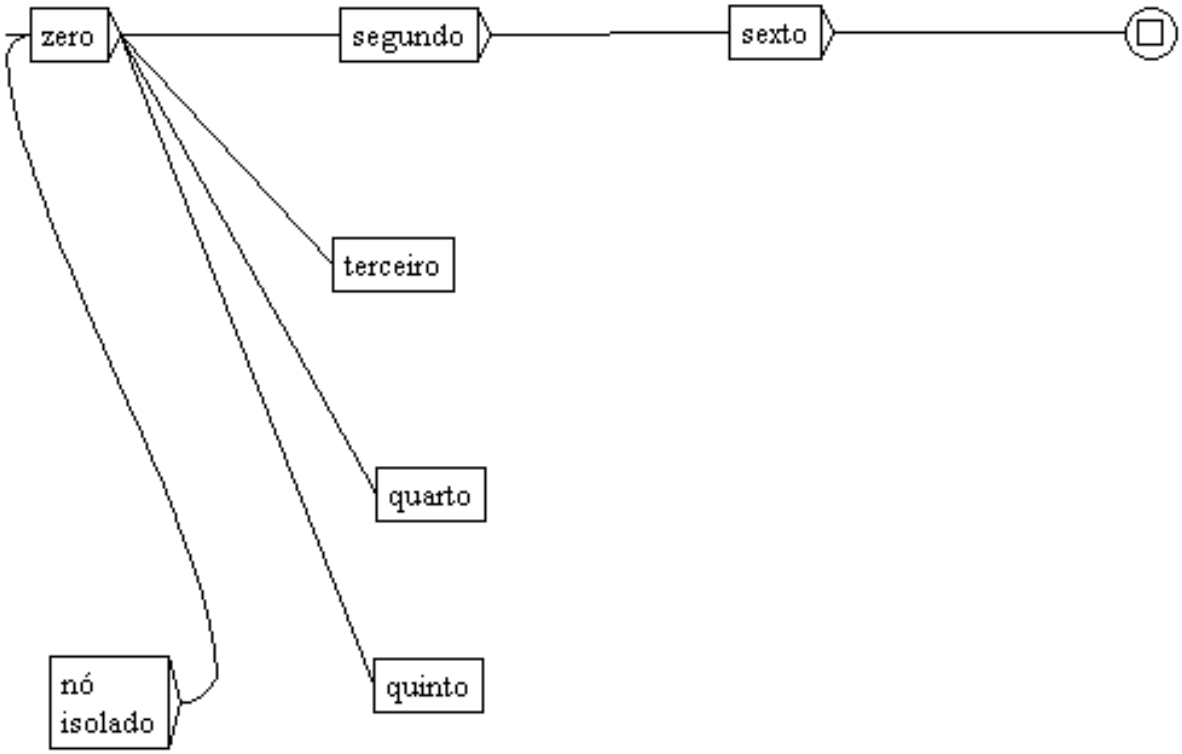


Figura 5: Grafo para compreensão do formato grf.

```
#Unigraph
SIZE 1318 840
FONT Times New Roman:  10
OFONT Arial Unicode MS:B 12
BCOLOR 16777215
FCOLOR 0
ACOLOR 13487565
SCOLOR 16711680
CCOLOR 255
DBOXES y
DFRAME y
DDATE y
DFILE y
DDIR n
DRIG n
DRST n
FITS 100
PORIENT L
#
8
"zero" 50 84 4 2 3 4 5
"" 503 83 0
"segundo" 178 84 1 6
"terceiro" 175 179 0
"quarto" 193 274 0
"quinto" 192 353 0
"sexto" 339 83 1 1
"nó+isolado" 58 360 1 0
```

Figura 6: O formato grf.

O conteúdo entre a cadeia “#Unigraph” e o símbolo “#” é praticamente constante e se refere às configurações de exibição da gramática como o tamanho da janela.

O conteúdo após o símbolo “#” é o responsável pela configuração desejada para a gramática. O algarismo “8” após a “#” representa a quantidade de nós existentes na gramática. Cada linha subsequente ao algarismo “8” representa um nó. Cada nó é representado por um número implícito, dada a ordem em que ele aparece no arquivo. Por exemplo, o nó inicial é sempre o primeiro a aparecer e ele é representado como o nó de número 0 (zero). Já o nó que representa o estado final deve ser sempre o segundo a aparecer e seu número implícito é o 1 (um).

Cada nó possui 5 informações básicas separadas por espaço. A primeira informação entre aspas duplas é o conteúdo que o nó deverá representar. O nó inicial por exemplo, representa uma estrutura que reconhece a palavra “zero” e poderia conter também um código lexical do Unitex como “<N>” que representa os substantivos. A segunda informação diz respeito ao posicionamento horizontal do nó ou também sua posição em relação ao eixo x. Já a terceira informação diz respeito ao posicionamento vertical do nó ou sua posição em relação ao eixo y. A quarta informação se refere a quantas transições de saída o nó possui. E, por fim, a quinta informação são todos os nós alcançáveis a partir do nó atual. A Figura 7 representa todas essas informações.

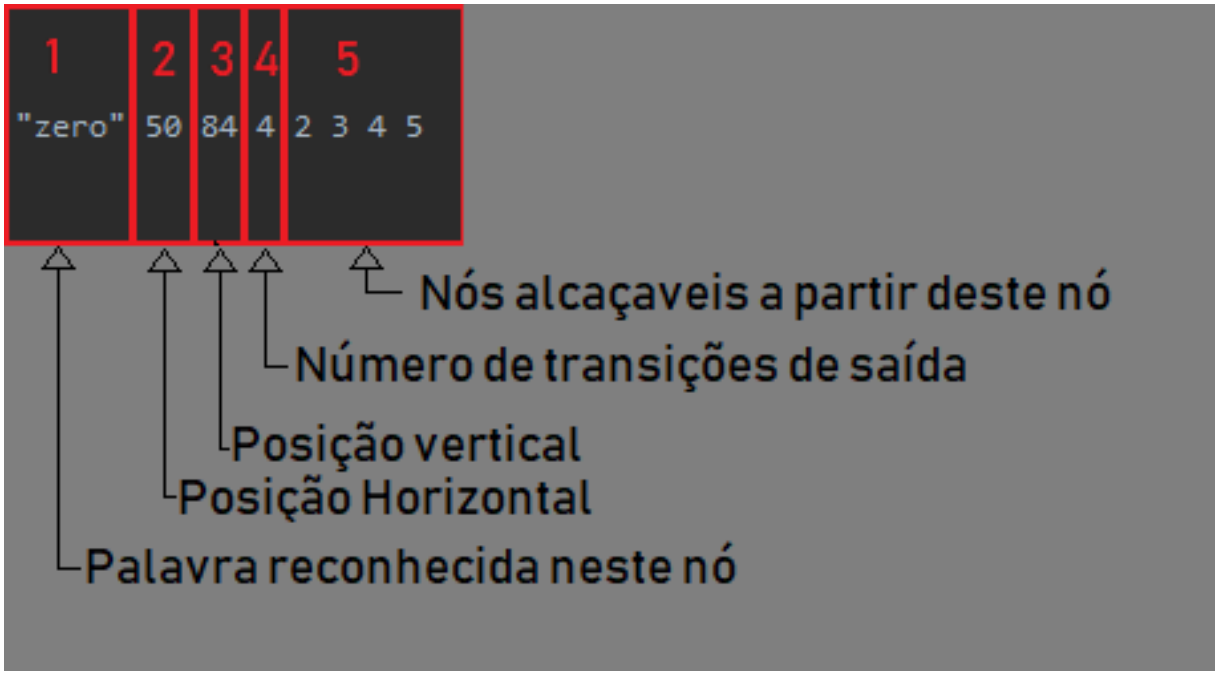


Figura 7: As partes de um nó.

3.3 Construção e Generalização da Gramática

As sentenças marcadas pelo usuário são salvas em apenas um arquivo “.txt”. O conteúdo deste arquivo de texto é pré-processado conforme o processo *default* do Unitex. Em seguida é usada a ferramenta FST-Text disponível no Unitex para gerar um aquivo de exemplos mapeados com informações como o lema, as categorias *Part Of Speech* (POS) possíveis e suas flexões. Cada arquivo gerado irá conter uma única sentença mapeada.

A fim de generalizar estas regras e obter uma única regra que represente os exemplos pretendidos, foi construído um algoritmo para descobrir os padrões que ocorrem mais frequentemente nas sequências. Tal algoritmo foi inspirado no trabalho de (MANNILA; TOIVONEN; VERKAMO, 1997) que foca na descoberta de episódios frequentes em sequências de eventos. Este processo elimina caminhos alternativos irrelevantes no reconhecimento do padrão desejado, isto é, caminhos que ao serem retirados não prejudicam o reconhecimento do padrão desejado e que possivelmente representam características não desejadas.

Para ilustrar como as regras são generalizadas, segue um exemplo: Para a frase “João é bonito.” foi obtido o arquivo da Figura 8.

```
{João,joão.N+Pr:ms} {é,ser.V:P3s} ({bonito,bonito.N:ms}|{bonito,bonito.ADV}|{bonito,bonito.A:ms}) .
```

Figura 8: Arquivo do primeiro exemplo.

E para a frase “Luana é advogada.” foi obtido o arquivo da Figura 9.

```
{Luana,luana.N+Pr:fs} {é,ser.V:P3s} ({advogada,advogar.V:Kfs}|{advogada,advogado.N:fs}|{advogada,advogado.A:fs}).
```

Figura 9: Arquivo do segundo exemplo.

Neste trabalho um evento é uma palavra juntamente com suas características: lema, classe gramatical e flexões. Já um episódio é uma sequência de eventos que contém todos os eventos de um exemplo. O tamanho dos episódios (denotado por “janela”) é a quantidade de eventos que um episódio possui. Extraindo manualmente episódios com um tamanho de janela igual a 3 dos arquivos, tem-se os episódios apresentados na Figura 10.

		evento1	evento2	evento3
Episódio 1	literal	João	é	bonito
	lema	joão	ser	bonito
	classe gramátical	N+Pr	V	N ou ADV ou A
	flexões	ms	P3s	ms
Episódio 2	literal	Luana	é	advogada
	lema	luana	ser	advogar ou advogado
	classe gramátical	N+Pr	V	V ou N ou A
	flexões	fs	P3s	Kfs ou fs

Figura 10: Tabela de características.

Para cada posição de evento calcula-se a frequência (numero de repetições) de cada característica entre os 2 episódios. Cada característica é contada apenas uma vez em cada episódio ou seja, repetições dentro do mesmo episódio não são consideradas. Na tabela apresentada na Figura 10 tem-se o seguinte:

Posição 1 Classe gramatical “N+Pr” e a frequência que ocorre é 2.

Posição 2 Todas as características são iguais, e a frequência que ocorre é 2.

Posição 3 Classe gramatical “N” ou “A” e a frequência de ambas as classes é 2.

A partir do cálculo de frequência realizado, para cada posição escolhe-se os resultados com maior frequência. Em caso de empate na mesma categoria são consideradas todas as possibilidades e em caso de empate entre categorias diferentes a preferência segue a seguinte ordem:

1. literal

2. lema

3. classe gramatical

As flexões por si só não conseguem caracterizar um padrão e portanto são casos especiais que ainda não são tratados na presente versão.

O resultado para o exemplo apresentado é o que se obteve no cálculo de frequência.

4 A FERRAMENTA GERADOR DE GL

Todas as funcionalidades da ferramenta desenvolvida são acessíveis por meio da interface principal. Ao abrir o programa o usuário irá se deparar com essa interface como mostra a Figura 11:

Figura 11: Interface Principal.

Ao selecionar a opção “Carregar Texto” (Figura 12) no menu “Texto” uma janela de seleção de arquivos será exibida (Figura 13) e permitirá que o usuário carregue um texto por vez.

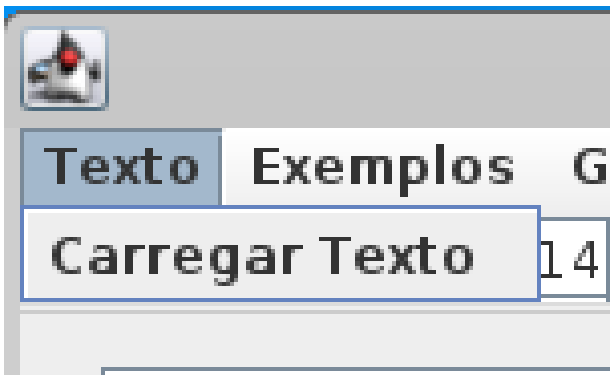


Figura 12: Menu Texto.

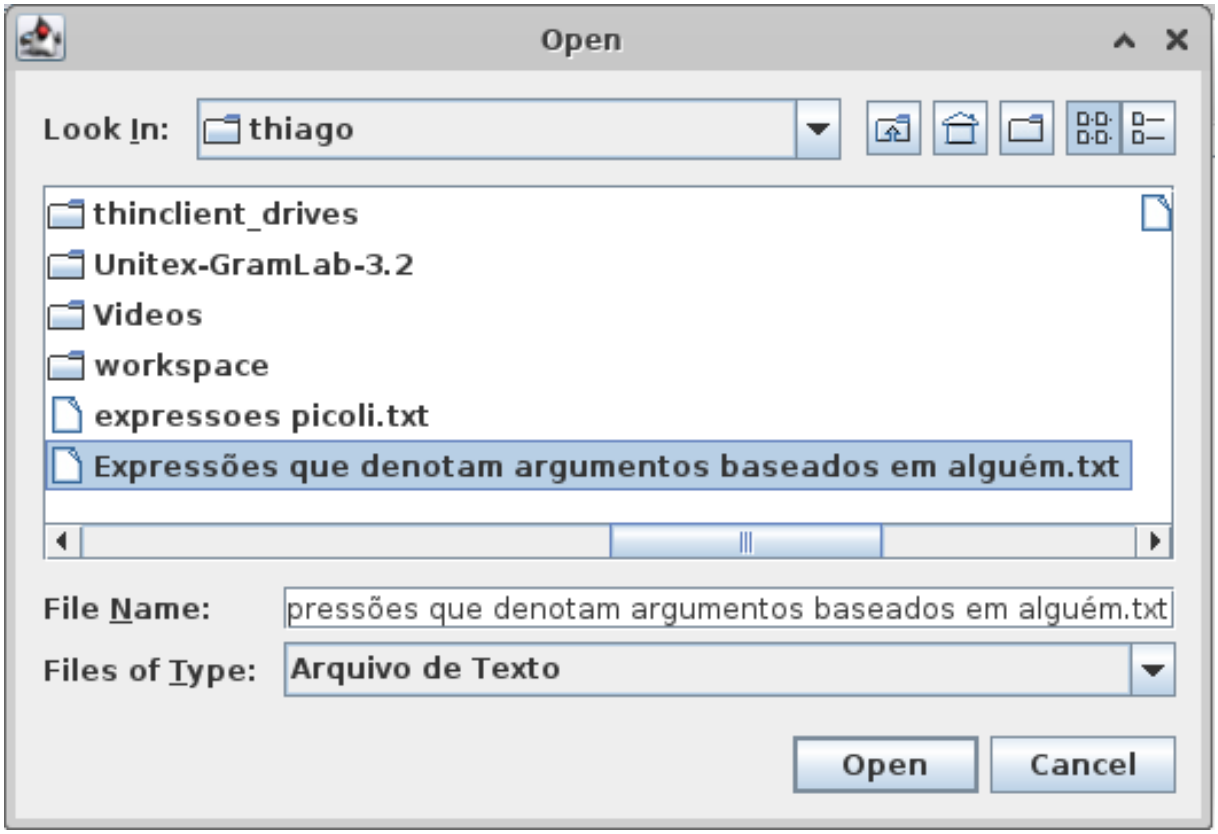


Figura 13: Seleção de arquivo.

O texto selecionado é carregado para o painel de texto na interface principal. O usuário pode aumentar o tamanho do texto conforme desejado e também pode pesquisar por palavras específicas contidas nele. Ao carregar um texto, este estará disponível para que o usuário selecione os exemplos que desejar por meio das teclas de atalho “ctrl+s”, ou na opção “Selecionar Texto Marcado Como Exemplo” no menu “Exemplos” como mostrado na Figura 14.

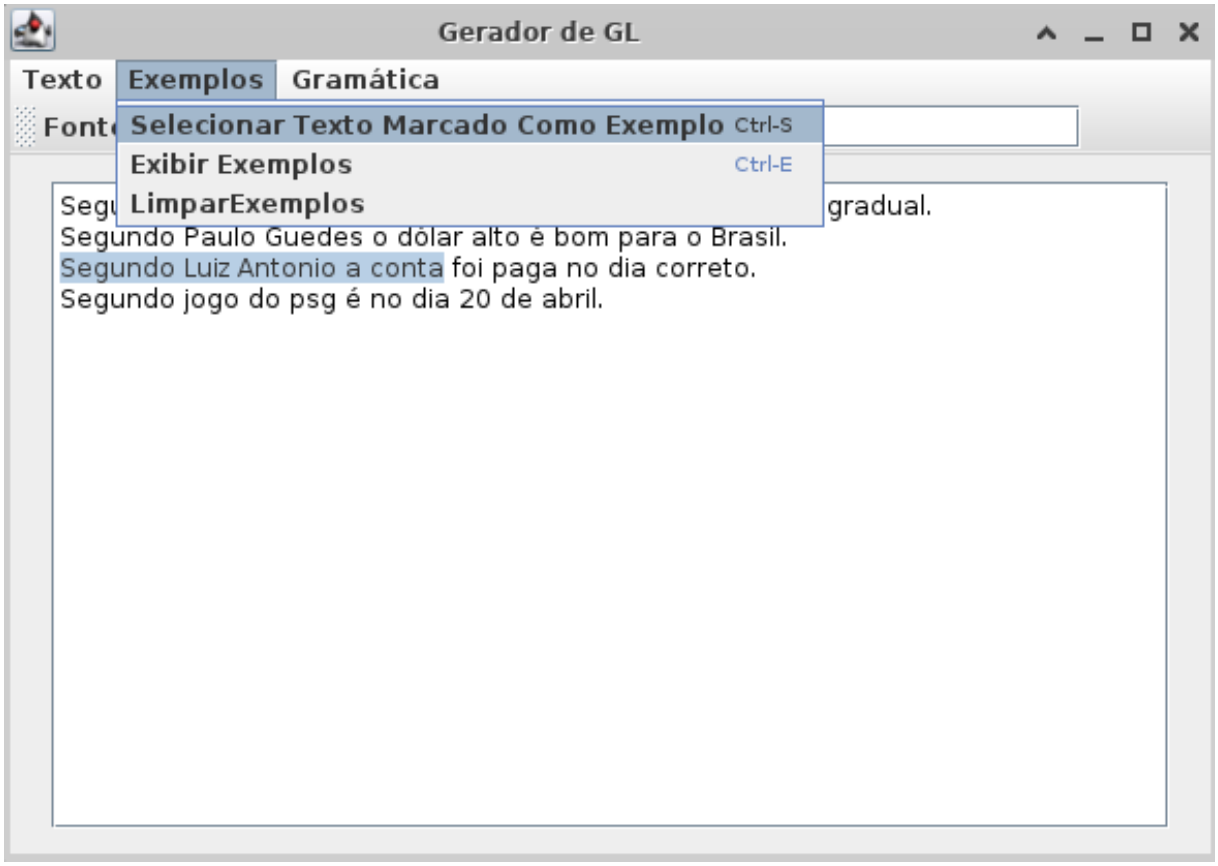


Figura 14: Menu Exemplos.

Os exemplos selecionados são destacados com a cor vermelha no texto e também podem ser visualizados por meio das teclas de atalho “ctrl+e” ou pela opção “Exibir Exemplos” no menu “Exemplos”.

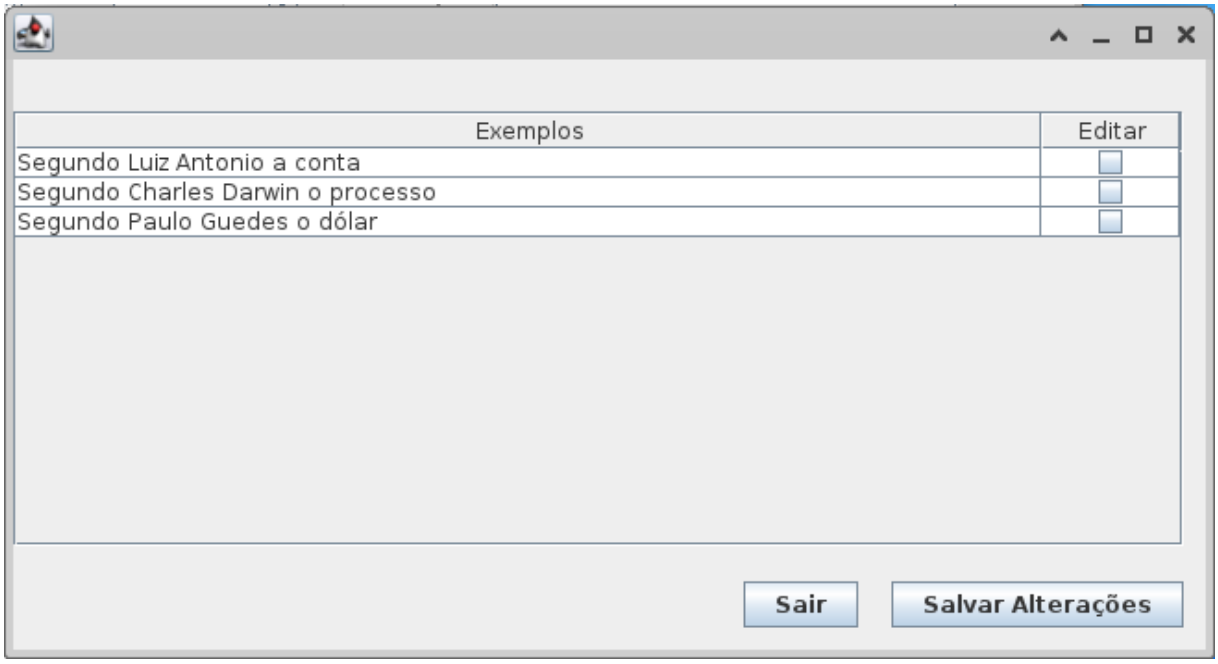


Figura 15: Tela de Visualização de Exemplos.

Após a seleção de todos os exemplos desejados, o usuário poderá gerar a gramática que reconhece os exemplos a partir da opção “Gerar Gramática” no menu “Gramática” ou tecla de atalho “ctrl+g” (Figura 16).

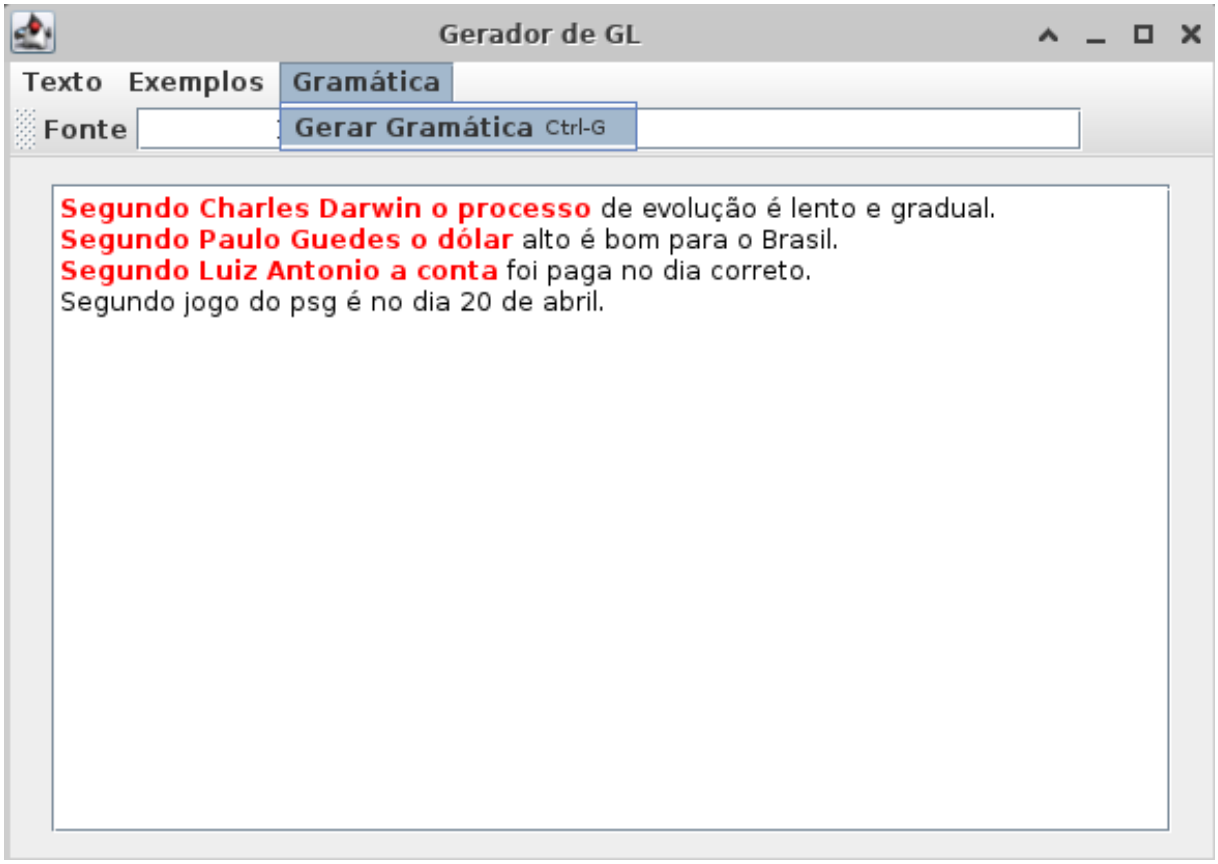


Figura 16: Menu Gramática.

Em seguida será exibida uma janela que possibilita a navegação pelos diretórios do SO e a escolha de um local em que a gramática resultante será salva.

O Gerador de GL está disponível para *download* em https://github.com/geradordegl/geradordegl.

4.1 Possibilidades de Melhorias

Generalização de flexões: Devido ao tratamento diferenciado que esta característica exige, foi decidido que, em primeiro momento, essa característica não é trivial para a generalização e construção de regras.

Generalização de exemplos com tamanhos diferentes: Atualmente o algoritmo de generalização é capaz de lidar apenas com episódios de tamanhos iguais.

4.2 Testes

A seguir são apresentados alguns dos testes realizados e os resultados obtidos. Para o primeiro teste foram selecionados os exemplos “Segundo Charles Darwin o processo”, “Segundo Paulo Guedes o dólar” e “Segundo Luiz Antonio a conta”. A Figura 18 representa a gramática obtida. Este teste, em especial, trata-se de uma regra com intuito de capturar algumas referências a autores e mostra que a GL construída foi genérica o suficiente para reconhecer os exemplos marcados e ao mesmo tempo não capturou a frase não desejada “Segundo jogo confirmado será”.

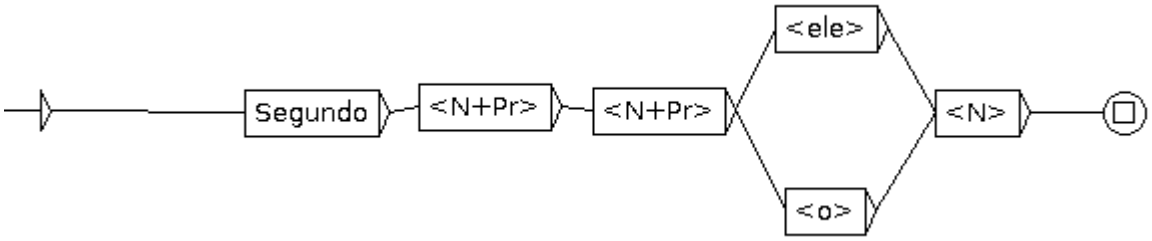


Figura 17: Gramática local do teste 1.

A Figura a seguir representa o texto onde a gramática obtida anteriormente foi aplicada.

Segundo Charles Darwin o processo de evolução é lento e gradual.
{S}Segundo Paulo Guedes o dólar alto é bom para o Brasil.
{S}Segundo Luiz Antonio a conta foi paga no dia correto.
{S}Segundo jogo confirmado será no dia 20 de abril.

Figura 18: Texto do teste 1.

A Figura a seguir representa o resultado da aplicação da GL obtida no texto.

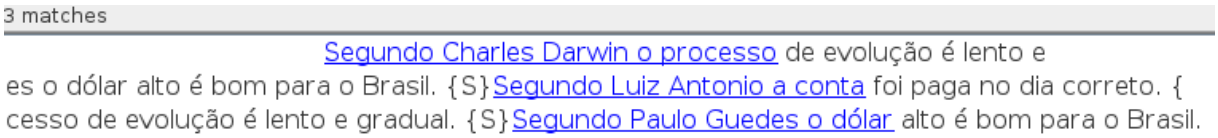


Figura 19: Concordância do teste 1.

Para o segundo teste foram selecionados os exemplos “O sol aqueceu a areia”, “A

água amoleceu o papel” e “O capitão enlouqueceu a tripulação”. A Figura 20 representa a gramática obtida.

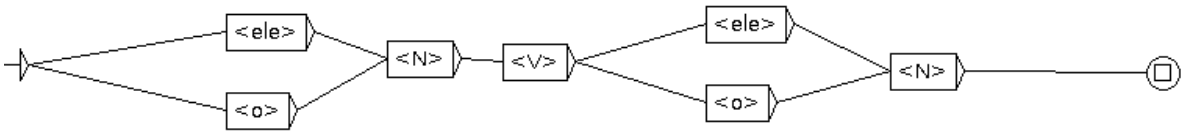


Figura 20: Gramática local do teste 2.

A Figura a seguir representa o corpus onde a gramática obtida anteriormente foi aplicada.

O sol aqueceu a areia
{S}A água amoleceu o papel
{S}O capitão enlouqueceu a tripulação
{S}A guerra empobreceu a população

Figura 21: Texto do teste2.

A Figura a seguir representa o resultado da aplicação da GL obtida no texto..

O sol aqueceu a areia {S}A água amoleceu o papel {S}O capitão enlouqueceu a trip
S}O capitão enlouqueceu a tripulação {S}A guerra empobreceu a população
u a areia {S}A água amoleceu o papel {S}O capitão enlouqueceu a tripulação {S}A guerra empobrec
O sol aqueceu a areia {S}A água amoleceu o papel {S}O c

Figura 22: Concordância do teste 2.

Para o terceiro teste foram selecionados os exemplos “João é bonito.” e “Luana é advogada.”. A Figura 23 representa a gramática obtida.

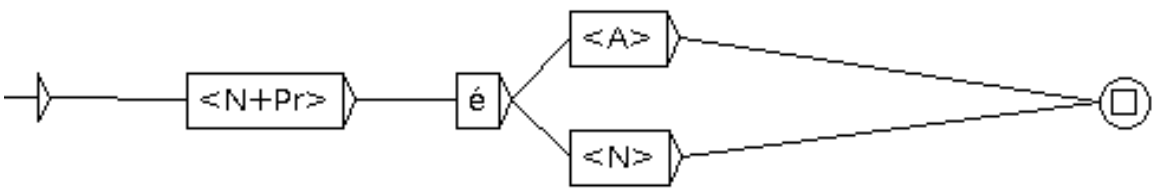


Figura 23: Gramática local do teste 3.

A Figura a seguir representa o corpus onde a gramática obtida anteriormente foi aplicada.

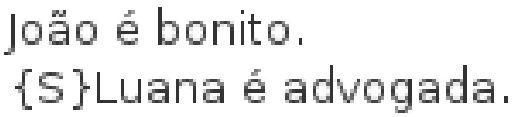


Figura 24: Texto do teste 3.

O resultado obtido apresenta-se na Figura 25.

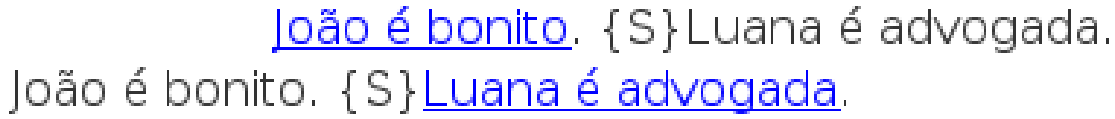


Figura 25: Concordância do teste 3.

Todos os testes realizados foram capazes de reconhecer os exemplos marcados. A gramática do teste 2 também foi utilizada para procurar padrões no Livro “Senhora” de José de Alencar, disponível em domínio público e também disponível nos recursos da língua portuguesa do ambiente Unitex. Ao todo a GL conseguiu localizar 222 ocorrências do padrão. A Figura 26, mostra algumas ocorrências do padrão.

[1]

[1] http://www.dominiopublico.gov.br/pesquisa/PesquisaObraForm.jsp

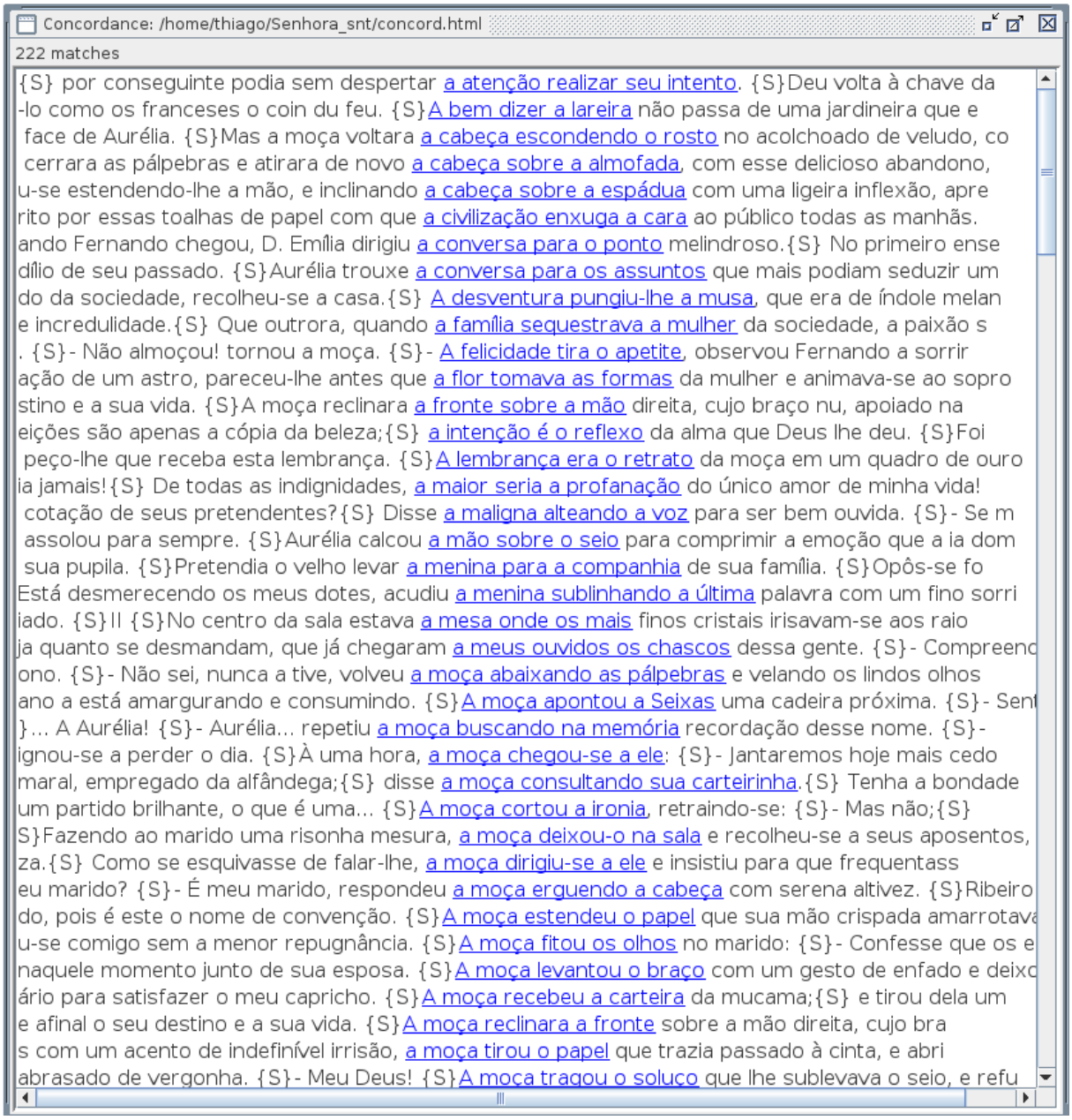


Figura 26: Concordância da GL aplicada ao Livro Senhora.

5 DIFICULDADES E OBSERVAÇÕES

A documentação existente sobre a integração da JNI Unitex é escassa e as existentes são pouco detalhadas e isto impossibilitou a integração da JNI para o sistema operacional Windows.

Houve dificuldades em encontrar trabalhos relacionados à generalização de regras a partir de exemplos, o que também dificultou que a ferramenta se apropriasse de um algoritmo de generalização de regras consolidado na comunidade científica.

Devido a enorme quantidade de comparações exigidas para construir as GLs, diminuir a complexidade assintótica do algoritmo de generalização foi uma tarefa imprescindível. Portanto o desenvolvimento deste algoritmo se mostrou um processo trabalhoso.

6 CONSIDERAÇÕES FINAIS

Neste trabalho, foi implementada uma ferramenta com o objetivo de facilitar a criação de regras para reconhecimento de padrões textuais em linguagem natural. A ferramenta desenvolvida mostrou-se capaz de gerar regras genéricas para reconhecer frases conforme os exemplos fornecidos. Entretanto, já foram identificadas possíveis melhorias no quesito generalização. Para melhorias futuras espera-se que o Gerador de GL conte com a generalização de flexões verbais e seja capaz de lidar com exemplos de tamanhos diferentes.

REFERÊNCIAS

BARTOLI, A. et al. Automatic generation of regular expressions from examples with genetic programming. In: ACM. *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. [S.l.], 2012. p. 1477–1478. Citado na página 16.

CHOWDHURY, G. G. Natural language processing. *Annual review of information science and technology*, Wiley Online Library, v. 37, n. 1, p. 51–89, 2003. Citado na página 9.

FELTRIM, V. D.; SOUZA, V. Pyger: Uma ferramenta geradora de expressões regulares a partir de um conjunto de expressões em linguagem natural. In: *Proceedings of the 1st Student Workshop on Information and Human Language Technology (TILic–STIL)*. [S.l.: s.n.], 2009. p. 1–5. Citado na página 16.

GROSS, M. 1 the construction of local grammars. *Finite-state language processing*, MIT Press, p. 329, 1997. Citado na página 13.

GROSS, M. *A bootstrap method for constructing local grammars*. 1999. Citado na página 10.

LIDDY, E. D. Natural language processing. 2001. Nenhuma citação no texto.

LIMA, V. L. S. de; NUNES, M. d. G. V.; VIEIRA, R. Desafios do processamento de línguas naturais. *SEMISH-Seminário Integrado de Software e Hardware*, v. 34, p. 1, 2007. Citado na página 10.

LOHR, S. The age of big data. *New York Times*, v. 11, n. 2012, 2012. Citado na página 9.

MANNILA, H.; TOIVONEN, H.; VERKAMO, A. I. Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery*, Springer, v. 1, n. 3, p. 259–289, 1997. Citado na página 22.

NOUVEL, D. et al. Recognizing named entities using automatically extracted transduction rules. In: *Language & Technology Conference (LTC'11)*. [S.l.: s.n.], 2011. Citado 2 vezes nas páginas 16 e 17.

PAUMIER, S. Unitex-manuel d'utilisation. 2011. Citado na página 13.

PICOLI, L. et al. Uso de uma ferramenta de processamento de linguagem natural como auxílio à coleta de exemplos para o estudo de propriedades sintático-semânticas de verbos. *Linguamática*, v. 7, n. 2, p. 35–44, 2015. Citado na página 11.

PIROVANI, J. P. C. Crf+ lg: uma abordagem híbrida para o reconhecimento de entidades nomeadas em português. Universidade Federal do Espírito Santo, 2019. Citado na página 11.

TATAR, S.; CICEKLI, I. Automatic rule learning exploiting morphological features for named entity recognition in turkish. *Journal of Information Science*, Sage Publications Sage UK: London, England, v. 37, n. 2, p. 137–151, 2011. Citado na página 16.