

# **1 INTRODUÇÃO**

## **1.1 Apresentação**

## **1.2 Objetivos**

### **1.2.1 Objetivo geral:**

### **1.2.2 Objetivos específicos:**

# **2 JUSTIFICATIVA**

# **3 CONTEXTUALIZAÇÃO**

## **3.1 Processamento da linguagem natural**

## **3.2 Ambiguidades**

## **3.3 Ferramentas para processamento da linguagem natural**

# **4 ANALISE**

## **4.1 Requisitos**

## **4.2 Descrição**

## **4.3 Arquitetura**

## **4.4 Cronograma de execução**

## **4.5 Diagramas**

### **4.5.1 Diagrama de caso de uso**

### **4.5.2 Diagrama de classe**

### **4.5.3 Diagrama de atividade**

### **4.5.4 Diagrama de sequência**

## **4.6 Algoritmo para solução de ambiguidades**

## **4.7 Detalhes da Implementação**

# **5 ANÁLISE DOS RESULTADOS**

## **5.1 Métricas de avaliação**

## **5.2 Testes**

# **6 CONCLUSÕES E TRABALHOS FUTUROS**

# **REFERÊNCIA**

# 1 INTRODUÇÃO

## 1.1 Apresentação

O presente trabalho de conclusão de curso visa desenvolver um algoritmo para solucionar o problema da correferência em aplicações no âmbito do processamento da linguagem natural, que é uma subárea da Inteligência Artificial, na qual é estudada a capacidade dos programas em interpretar a linguagem natural humana.

Para uma máquina analisar a linguagem humana, é necessário que os algoritmos consigam interpretar a morfologia, sintaxe, semântica e pragmática dos textos. Assim, este trabalho está diretamente relacionado com a análise pragmática, mais especificamente com a ambiguidade anafórica dentro de uma relação de correferência, em que são analisadas as relações entre palavras de diversas sentenças, dentro de um contexto.

Quando uma entidade é mencionada mais de uma vez em um texto, poderá ser utilizado um termo anafórico para referenciá-la, podendo ser usado, por exemplo, um pronome (ele, ela, isso) ou um artigo definido (o, a, as). A relação entre a entidade que é dita como antecedente e o termo anafórico é chamada de relação de correferência. Existem casos em que podem haver mais de uma interpretação em relação de qual antecedente está sendo referenciado pelo termo anafórico, provocando a ambiguidade. O algoritmo desenvolvido deverá interpretar os termos anafóricos, considerando o contexto das frases, para conseguir compreender o que está sendo dito.

Para tanto, buscaremos analisar trabalhos anteriores similares ao tema, com o objetivo de conhecer o estado da arte. O resultado que esperamos obter será o de mostrar o uso da solução através de exemplos práticos. A partir desta etapa, será possível escrever a conclusão e pensar em trabalhos futuros.

## 1.2 Objetivos

### 1.2.1 Objetivo geral:

- Desenvolver algoritmo para solução da ambiguidade anafórica pronominal em relações de correferência, voltado para a língua portuguesa, em aplicações de processamento de linguagem natural

### 1.2.2 Objetivos específicos:

- Buscar um *framework* que permita implementar uma solução para a ambiguidade anafórica pronominal em relações de correferência ou uma estrutura que permita testar o algoritmo implementado.
  - Implementar um algoritmo de solução do problema de correferência
  - Para a implementação utilizaremos *frameworks* já disponíveis na Internet e que já possuem programados módulos de morfologia e sintaxe. Implementaremos o algoritmo como um ponto extensível (*hotspot*) para a resolução do problema anafórico.
  - Conectar o algoritmo implementado no framework/sistema.
  - Testar o *framework*/sistema com e sem o algoritmo de solução do problema de correferência, com o objetivo de comparar os resultados e saber a eficiência do algoritmo.

## 2 JUSTIFICATIVA

Em aplicações de processamento de linguagem natural que buscam simular o comportamento humano, o computador deve ser capaz de interpretar perguntas feitas por usuários, no entanto esta tarefa nem sempre é simples. A análise pragmática é uma das etapas da linguística necessária para a compreensão da pergunta, pois é responsável pela análise do significado das palavras da sentença, dentro de um contexto e a relevância deste trabalho está em transpor a lógica pragmática ao algoritmo que queremos desenvolver.

Dentre os diversos problemas relacionados à análise pragmática, um dos principais é o problema da resolução de anáforas, em que o computador precisa identificar corretamente a quem o termo anafórico está se referindo, e é por esse motivo que focaremos nossa pesquisa nesse campo.

Existem casos em que ocorrem a ambiguidade anafórica, assim é necessária a interpretação de qual antecedente está sendo referido pelo termo anafórico, para uma melhor compreensão do sentido da frase, sendo solucionada pela relação de correferência entre essas duas expressões.

Assim, esta monografia justifica-se ao tentar solucionar tais ambiguidades anafóricas dentro de aplicações de processamento de linguagem natural.

Minha abordagem tem como objetivo reproduzir a mesma lógica utilizada pelas pessoas quando, em uma conversa, conseguem resolver as ambiguidades. No caso de ambiguidades anafóricas com pronomes nulos, o antecedente será o sujeito da sentença.

Os casos em que ocorrem ambiguidades com pronomes plenos são mais complexos, será analisada a possibilidade de se utilizar o objeto da frase, o complemento verbal ou nos casos sem solução, se existirem, a aplicação terá que obter mais informações para definir o antecedente.

### **3 CONTEXTUALIZAÇÃO**

#### **3.1 Processamento da linguagem natural**

O processamento da linguagem natural estuda e desenvolve aplicações que reconhecem a linguagem humana, através do estudo da fonologia e análises linguísticas. A recuperação de informação, reconhecimento de fala, tradução automática e sistemas de perguntas e respostas são aplicações desta área.

Para o desenvolvimento deste trabalho está sendo analisado o funcionamento de sistemas de perguntas e respostas. Tais sistemas têm como finalidade reproduzir o comportamento humano, na interpretação de uma pergunta, permitindo formar uma resposta através do seu conhecimento, construído a partir de documentos e texto locais ou pesquisando em outras fontes externas. Um exemplo de uso desse sistema seria, por exemplo, para auxiliar a aprendizagem de estudantes, desenvolvendo tutores virtuais para interagir com o aluno. (SAIAS, 2010)

De uma maneira geral, os sistemas de processamento da linguagem natural funcionam através das seguintes etapas: (i) análise da pergunta; (ii) consulta de fontes de respostas; (iii) extração de respostas e (iv) filtragem dos resultados, descritas a seguir:

- (i) A análise da pergunta inclui a morfologia, sintaxe, semântica e pragmática. Primeiro é verificada a morfologia, em que é analisada a estrutura das palavras e a definição de suas categorias, se é um substantivo, verbo, adjetivo, preposição ou advérbio. Na sequência, é realizada a análise sintática, em que é estudada a estrutura da frase e a função das palavras na mesma, identificando, por exemplo, o sujeito e predicado. Na análise semântica, é identificado o significado das palavras e o sentido que ela representa nas frases, e por último, na análise pragmática, é feita a interpretação da mensagem, analisando-se a relação das frases de acordo com o contexto. Após as análises linguísticas, o sistema obtém uma estrutura da pergunta que pode ser processada computacionalmente. Algumas abordagens adicionam a esta estrutura uma ontologia com o objetivo de classificar a pergunta para identificar o tipo da resposta, se será um valor numérico, data ou definição. (VIEIRA, 2011)
- (ii) O sistema poderá buscar as respostas em documentos locais, como uma coleção de textos no próprio computador ou poderá consultar também conteúdo da internet, usando um motor de pesquisa como o Google ou dicionários online.(SAIAS, 2010)
- (iii) Na etapa de extração das respostas, os sistemas utilizam abordagens que permitem selecionar um grupo de possíveis respostas. Um exemplo de abordagem seria considerar os advérbios e pronomes interrogativos o foco da pergunta como o sujeito e a ação do verbo, se houver.(SAIAS, 2010)
- (iv) Na etapa final, será filtrada a melhor resposta. Para essa escolha, o sistema vai considerar a maior pontuação dentre as respostas selecionadas.(SAIAS, 2010)

### 3.2 Ambiguidades

A ambiguidade em uma frase ocorre quando ela pode ser interpretada de várias maneiras, prejudicando a compreensão da mensagem. Ambiguidades podem ser do tipo lexical, sintática, escopo e semântica. A ambiguidade lexical ocorre quando uma palavra provoca mais de uma forma de interpretação, pode ocorrer como forma de homonímia, em que as palavras são escrita da mesma forma, mas possuem significados diferentes ou a polissemia, em que uma palavra pode ter vários significados. Um exemplo de homonímia: *Qualquer canto está bom para Silvia*, a palavra canto pode representar um lugar ou uma música (SILVA, 2016).

Na ambiguidade sintática as interpretações variam em relações às estruturas sintáticas e não às palavras, como na frase: *eu vi o acidente do carro*, a sentença pode ser interpretada como sendo uma pessoa vendo um acidente envolvendo um carro ou uma pessoa viu um acidente dentro de um carro.

A ambiguidade de escopo, ocorre em frases que mencionam uma distribuição coletiva, a forma de interpretação, pode ser feita em relação ao coletivo ou individual, como no exemplo: *Os alunos comeram cinco biscoitos*, a frase pode ser interpretada como todos os alunos comeram os cinco biscoitos ou cada aluno da sala comeu cinco biscoitos.

A ambiguidade semântica, que é o foco deste trabalho, ocorre nas frases em que existe a relação de correferência, em que são usados pronomes para referenciar um termo, nesse caso o pronome faz referência a mais de uma expressão (CANÇADO, 2008), como no exemplo: *Ana conversava com Júlia, enquanto ela estudava*. O pronome ela pode estar relacionada com Ana ou Júlia.

Na literatura existem soluções distintas para problemas de ambiguidade em anáforas. Em (ISRAEL, 1990) é apresentada uma resolução da anáfora em casos de ambiguidades onde, primeiramente, são extraídas as sentenças que contém o termo anafórico, como um pronome, por exemplo, e depois são selecionados os candidatos a antecedentes que se referem a esse pronome. Nesta abordagem, o pronome é substituído pelos antecedentes selecionados formando pares, cada par contendo o antecedente acrescido do verbo da sentença em questão.

Em seguida é calculado o número de ocorrências dos pares de palavras no texto. Será escolhido o antecedente que está no par (antecedente + verbo) que for mais frequente no texto.

Segundo (COHEN, 2007) alguns casos de ambiguidades em anáforas podem ser resolvidos por inferência. Entretanto, alguns casos não possuem solução e somente podem ser resolvidos de forma arbitrária.

Em (VIEIRA, 2012), a ambiguidade pode ser desfeita através da coesão, que é responsável em fazer a conexão entre os elementos do texto, pode ser do tipo referencial ou sequencial. A coesão referencial é usada para evitar o uso de palavras repetidas através de anáforas ou catáforas, por exemplo, e a coesão sequencial ajuda a compreender e interpretar melhor o texto com o uso de conjunções, evitando frases desconexas e dando ideia de sequencialidade ao texto.

Ambiguidades anafóricas pronominais pode ter pronomes nulos ou plenos. (TEIXEIRA, 2014) concluiu, através de um experimento, que nas frases subordinadas que possuem ambiguidades anafóricas pronominais com pronomes nulos são possíveis resolvê-las, relacionando o antecedente com o sujeito. Nos casos com pronomes plenos nem sempre é possível resolver a ambiguidade.

Na pesquisa de (CARMINATI, 2002), feito para a língua italiana, que também tem em sua gramática o pronome nulo e pleno, observou que nas frases com pronome nulo, o antecedente deve ser o sujeito, e nas frases com pronome pleno o antecedente deve ser o objeto.

Em (FONSECA, 2012) foi feito um experimento a fim de verificar de que forma as pessoas escolhem os antecedentes de frases com ambiguidades anafóricas pronominais. Foram usadas frases com uma oração principal e uma oração subordinada, com pronome nulo e outras com a mesma estrutura sintática, mas com o pronome pleno. Como resultado, foi concluído que as pessoas preferem atribuir como antecedente o sujeito, quando a frase tiver o pronome nulo, e o complemento do verbo, quando o pronome for pleno.

### 3.3 Ferramentas para Processamento da Linguagem Natural

Foram pesquisadas ferramentas, que implementam as análises morfológica e sintáticas para serem usadas em conjunto com o algoritmo que será criado. O Python NLTK foi a primeira a ser estudada, é um livro que disponibiliza uma introdução no campo do processamento da linguagem natural e contém uma biblioteca de código aberto chamada Natural Language Toolkit (NLTK) e possui tutorias utilizando a linguagem python.

Os tutorias incluem demonstrações de como os conceitos da linguagem natural são usados para descrever e analisar a linguagem, como os analisadores analisam as sentenças e criam árvores sintáticas, como marcar automaticamente as palavras de acordo com sua classe gramatical, que são chamadas de etiquetagem gramatical e como são utilizados as estruturas e os algoritmos nesse campo. O NLTK disponibiliza documentação, software e dados gratuitos, incluindo algoritmos para a análise morfológica e sintática de sentenças.

A seguir é mostrado um exemplo de tutorial presentes no livro, sobre o etiquetador gramatical. O método `nltk.pos_tag(text)` atribui uma tag para cada palavra da frase ou texto, essa tag corresponde a uma classe gramatical:

```
>>> text = word_tokenize("And now for something completely different")
>>> nltk.pos_tag(text)
[('And', 'CC'), ('now', 'RB'), ('for', 'IN'), ('something', 'NN'),
 ('completely', 'RB'), ('different', 'JJ')]
```

Na conversão do NLTK o `tagged_token` (etiquetador de token\*) é representado usando uma tupla que consiste do token e tag. As tuplas são criadas a partir da função `str2tuple()`:

```
>>> tagged_token = nltk.tag.str2tuple('fly/NN')
>>> tagged_token
('fly', 'NN')
>>> tagged_token[0]
'fly'
>>> tagged_token[1]
'NN'
```



Existem vários modelos de etiquetagem. O NLTK utiliza o Universal Part-of-Speech Tagset:

Tag	Meaning	English Examples
ADJ	adjective	<i>new, good, high, special, big, local</i>
ADP	adposition	<i>on, of, at, with, by, into, under</i>
ADV	adverb	<i>really, already, still, early, now</i>
CONJ	conjunction	<i>and, or, but, if, while, although</i>
DET	determiner, article	<i>the, a, some, most, every, no, which</i>
NOUN	noun	<i>year, home, costs, time, Africa</i>
NUM	numeral	<i>twenty-four, fourth, 1991, 14:24</i>
PRT	particle	<i>at, on, out, over per, that, up, with</i>
PRON	pronoun	<i>he, their, her, its, my, I, us</i>
VERB	verb	<i>is, say, told, given, playing, would</i>
.	punctuation marks	<i>. , ; !</i>
X	other	<i>ersatz, esprit, dunno, gr8, univeristy</i>

Assim é possível, por exemplo, verificar quais são as tags que mais aparecem na categoria notícias do livro importado:

```
>>> from nltk.corpus import brown
>>> brown_news_tagged = brown.tagged_words(categories='news', tagset='universal')
>>> tag_fd = nltk.FreqDist(tag for (word, tag) in brown_news_tagged)
>>> tag_fd.most_common()
[('NOUN', 30640), ('VERB', 14399), ('ADP', 12355), ('.', 11928), ('DET', 11389),
 ('ADJ', 6706), ('ADV', 3349), ('CONJ', 2717), ('PRON', 2535), ('PRT', 2264),
 ('NUM', 2166), ('X', 106)]
```

No exemplo de código da figura xxx as palavras são classificadas de acordo com sua classe gramatical:

O Lx-Center foi outra ferramenta pesquisada que disponibiliza serviços e ferramentas online, voltadas para o processamento da linguagem natural, contendo interfaces para conjugar verbos, separar sílabas, análises morfológica e sintaxe de textos. Desenvolvida pelo grupo de pesquisa na universidade de Lisboa que realiza pesquisas no campo da inteligência artificial, com foco na linguagem natural.

```
>>> sent = brown.sents(categories='news')[3]
>>> baseline_tagger.tag(sent)
[('', ''), ('Only', None), ('a', 'AT'), ('relative', None),
 ('handful', None), ('of', 'IN'), ('such', None), ('reports', None),
 ('was', 'BEDZ'), ('received', None), ('', ''), ('', ''),
 ('the', 'AT'), ('jury', None), ('said', 'VBD'), ('', ''),
 ('', ''), ('considering', None), ('the', 'AT'), ('widespread', None),
 ('interest', None), ('in', 'IN'), ('the', 'AT'), ('election', None),
 ('', ''), ('the', 'AT'), ('number', None), ('of', 'IN'),
 ('voters', None), ('and', 'CC'), ('the', 'AT'), ('size', None),
 ('of', 'IN'), ('this', 'DT'), ('city', None), ('', ''), ('.', '.')]

```

A tabela de descrição das tags, está disponível no site, é usada para reconhecer quais são as categorias ou classes gramaticais das tags.

Tag	Category	Examples
ADJ	Adjectives	bom, brilhante, eficaz, ...
ADV	Adverbs	hoje, já, sim, felizmente, ...
CARD	Cardinals	zero, dez, cem, mil, ...
CJ	Conjunctions	e, ou, tal como, ...
CL	Clitics	o, lhe, se, ...
CN	Common Nouns	computador, cidade, ideia, ...
DA	Definite Articles	o, os, ...
DEM	Demonstratives	este, esses, aquele, ...
DFR	Denominators of Fractions	meio, terço, décimo, %, ...
DGTR	Roman Numerals	VI, LX, MMIII, MCMXCIX, ...
DGT	Digits	0, 1, 42, 12345, 67890, ...
DM	Discourse Marker	olá, ...
EADR	Electronic Addresses	http://www.di.fc.ul.pt, ...
EOE	End of Enumeration	etc
EXC	Exclamative	ah, ei, etc.
GER	Gerunds	sendo, afirmando, vivendo, ...
GERAUX	Gerund "ter"/"haver" in compound tenses	tendo, havendo ...
IA	Indefinite Articles	uns, umas, ...
IND	Indefinites	tudo, alguém, ninguém, ...
INF	Infinitive	ser, afirmar, viver, ...
INFAUX	Infinitive "ter"/"haver" in compound tenses	ter, haver ...
INT	Interrogatives	quem, como, quando, ...
ITJ	Interjection	bolas, caramba, ...
LTR	Letters	a, b, c, ...
MGT	Magnitude Classes	unidade, dezena, dúzia, resma, ...
MTH	Months	Janeiro, Dezembro, ...
NP	Noun Phrases	idem, ...
ORD	Ordinals	primeiro, centésimo, penúltimo, ...
PADR	Part of Address	Rua, av., rot., ...
PNM	Part of Name	Lisboa, António, João, ...

As ferramentas estão disponíveis no site, como no exemplo do item Pos-Tagger, onde é feita a etiquetagem gramatical, as palavras da frase ou texto escrita pelo usuário serão separadas e classificadas gramaticalmente.



Developed at the [University of Lisbon, Dept. of Informatics](#), by the [NLX-Natural Language and Speech Group](#).

[see an example](#) | [features](#) | [tagset](#) | [+services](#)

**Enter text in Portuguese, separating paragraphs with an empty line:**

Ana comprou um carro

Annotate

Clear

```
<p><s> Ana/PNM comprou/COMPRAR/V#ppi-3s um/U#ms  
carro/CARRO/CN#ms </s></p>
```

\*ainda vou fazer

TreeTagger:

Ferramenta que separa as palavras do texto e classifica elas morfológicamente, foi desenvolvida por Helmut Schmid em um projeto na universidade de Estugarda na Alemanha.

VISL:

Ferramenta escolhida para a pesquisa, é um projeto de pesquisa feito na universidade do sul da Dinamarca, disponibiliza ferramentas gramaticais online, possui analisadores morfológicos, sintáticos, tradução automática.

O projeto começou em 1996, com o objetivo de contribuição de estudo de análises gramáticas na internet, e o desenvolvimento de ferramentas interativas, em 3 anos o projeto cresceu, sendo disponibilizado para sete idiomas, foram construídos banco de árvores para esses idiomas, criados e revisados automaticamente.

## 4 REQUISITOS

### 4.1 Requisitos

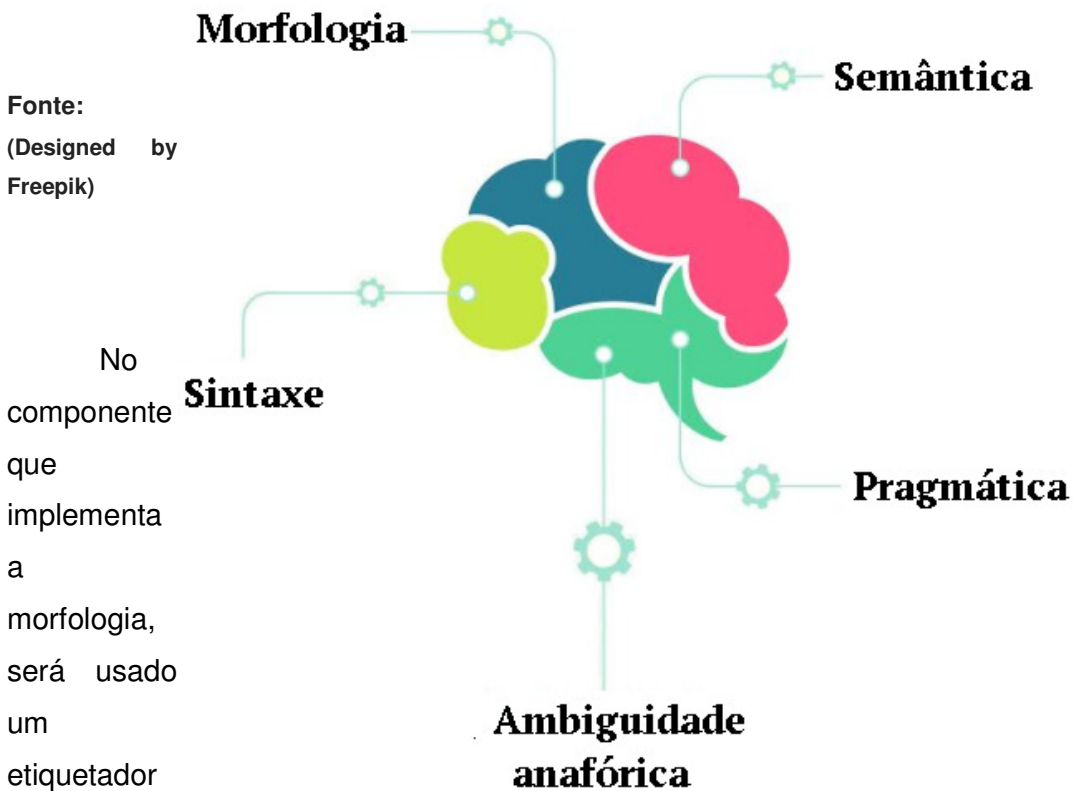
Caso de uso	Requisito funcionais
RF1	UC1 O programa deve ter uma interface para receber as perguntas
RF2	UC2 O programa deve ter uma interface responder as perguntas
RF3	UC3 O programa deve exportar o resultado da análise morfológica
RF4	UC4 O programa deve exportar o resultado da análise sintática
RF5	UC5 O programa deve identificar a ambiguidade anafórica pronominal na frase

### 4.2 Descrição

Uma aplicação que faça o processamento da linguagem natural é complexa, possui componentes para análises morfológicas, sintáticas, semânticas e pragmáticas. Tais elementos são utilizados para interpretação da linguagem, apresentados na figura 1. Este trabalho está preocupado apenas em resolver o problema da ambiguidade anafórica pronominal.

Desta maneira, implementaremos somente o algoritmo para resolver este problema específico e utilizaremos componentes de outros autores que implementam as análises morfológica, sintática, semântica e pragmática, para, em conjunto com o algoritmo, testar a sua eficácia.

FIGURA 1: ELEMENTOS DE INTERPRETAÇÃO



de categorias gramaticais, responsável por classificar as palavras de acordo com a sua morfologia, se é um verbo, substantivo, adjetivo, advérbio, assim como a atribuição do gênero, número das palavras e a flexão do verbo. A identificação da categoria das palavras poderá ajudar a entender o sentido delas na frase, por exemplo, o adjetivo tem o papel de qualificar um substantivo da frase e o artigo de definir o substantivo.

Na sintaxe será usado um analisador sintático, responsável por gerar as estruturas sintáticas das frases, com essas estruturas é possível, por exemplo, identificar quais são as ações do sujeito e os complementos do predicado.

Na análise semântica, será usado um programa que deverá entender os significados das palavras, a partir de inferências, levando em consideração as premissas e conclusões expressadas nas sentenças. O algoritmo de analisador sintático recebe a frase e a árvore de estrutura sintática, feita pelo analisador sintático, a frase será dividida em períodos simples, depois será estabelecido os relacionamentos e combinações entre esses conceitos, consultando uma base de sentença-padrão que contém modelos de estruturas de sentenças, resultando em uma rede inferencial, contendo as premissas e conclusões das sentenças.

Na análise pragmática o algoritmo deve interpretar a frase, levando em consideração o contexto. A figura 2 apresenta o fluxo da solução. Nosso algoritmo deve verificar quando ocorre uma ambiguidade anafórica pronominal. Para desfazer a ambiguidade deverá ser verificado se o pronome da frase é pleno ou nulo. Se for pronome pleno, o programa deve dar como resposta o sujeito da frase, se for nulo, a resposta será o complemento verbal.

### **4.3 Arquitetura**

Será utilizado como cliente um notebook com windows 10. O algoritmo será implementado utilizando a linguagem .....

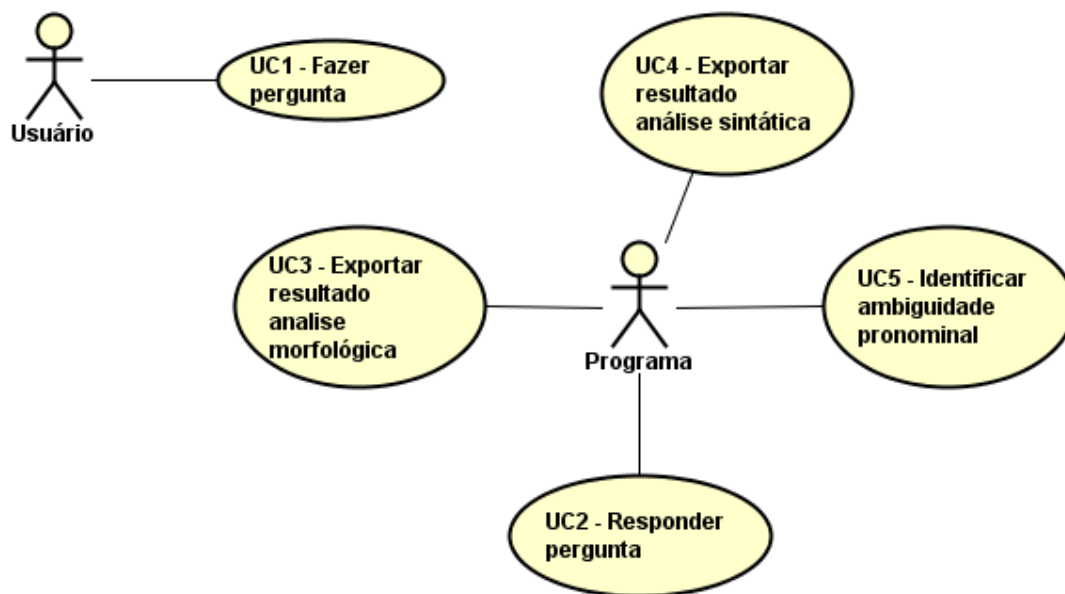
A análise morfológica das palavras nas frases serão feitas pela ferramenta do site VISL, classificando cada palavra de acordo com sua classe gramatical, assim como a análise sintática, identificando função de cada palavra nas frases. O algoritmo será feito para solucionar o problema de ambiguidades anafóricas pronominais. O resultado das análises morfológicas e sintática será exportada pelo algoritmo e de acordo com as regras implementadas, deverá identificar se a frase possui uma ambiguidade anafórica pronominal, para desfazer essa ambiguidade e responder a pergunta.

#### 4.4 Cronograma de execução

Etapa	Abril	Maiο	Junho	Julho	Agosto	Set	Out	Nov
Apresentação	x							
Objetivo	x							
Justificativa		x						
Contextualização		x						
Descrição			x					
Diagramas			x					
Implementação				x	x	x	x	x

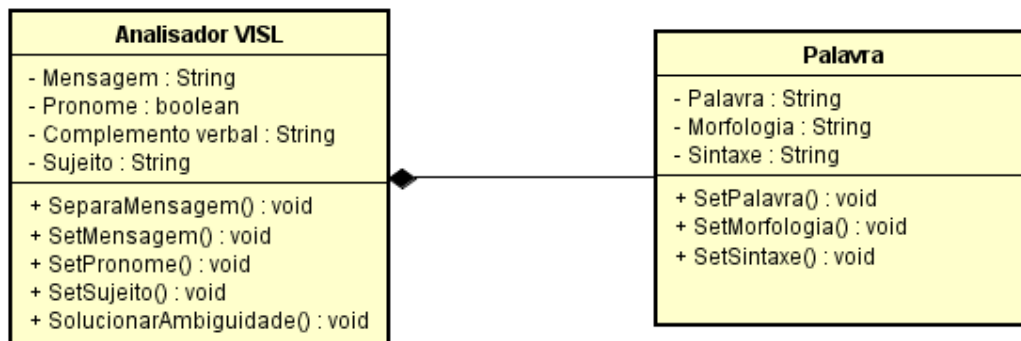
#### 4.5 DIAGRAMAS

##### 4.5.1 Diagrama de caso de uso

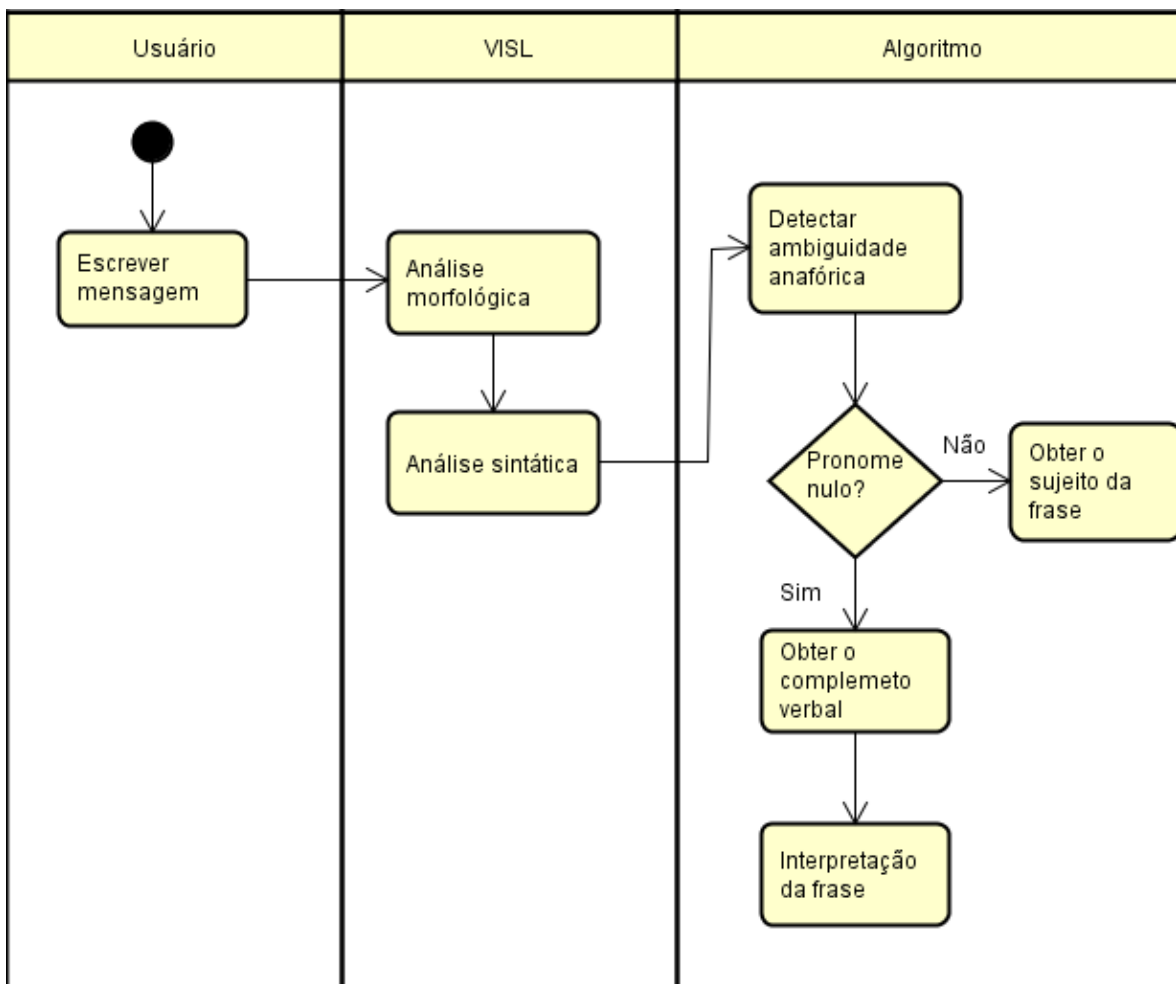




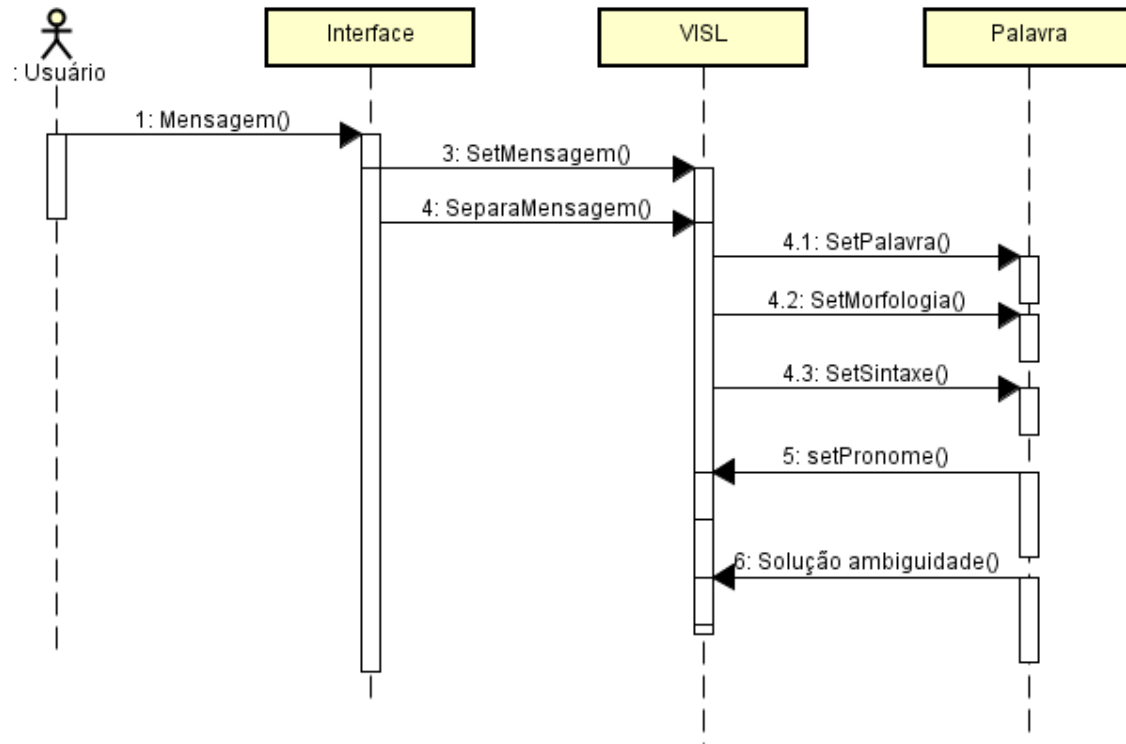
#### 4.5.2 Diagrama de classe



#### 4.5.3 Diagrama de atividade

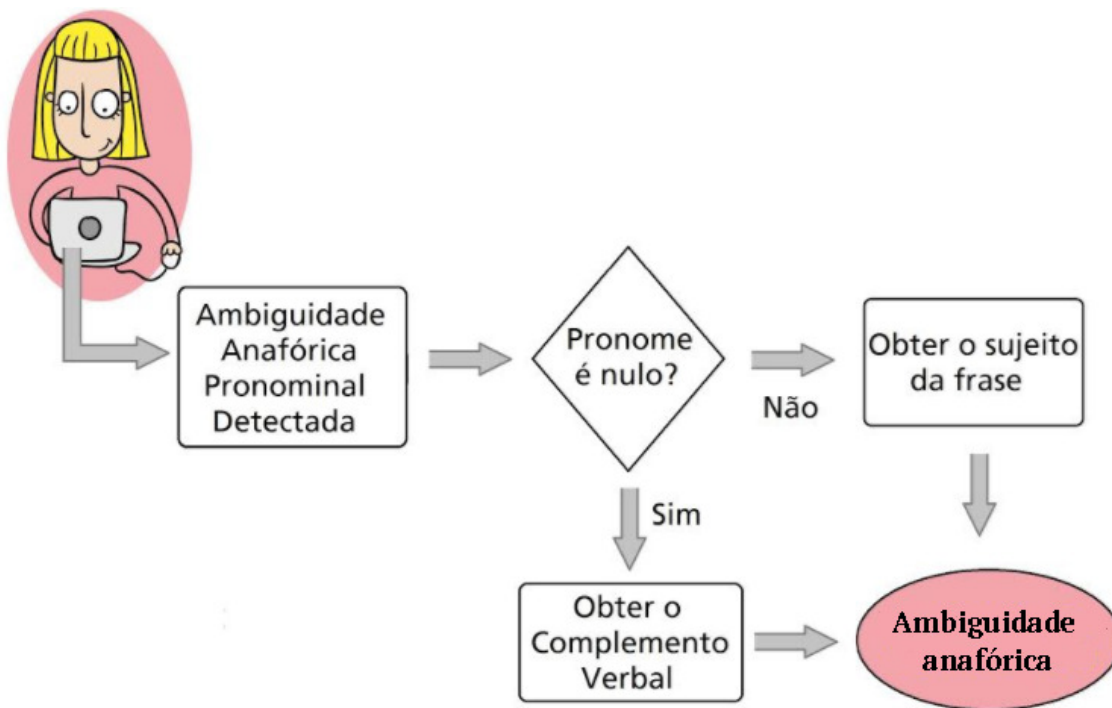


#### 4.5.4 Diagrama de sequência



#### 4.6 Algoritmo para solução de ambiguidades

FIGURA 2: ALGORITMO UTILIZADO



**Fonte:** (Designed by Freepik)

#### 4.7 Detalhes da implementação

## **5 ANÁLISE DOS RESULTADOS**

### **5.1 Métricas de avaliação**

Uma tabela de testes será usada para verificar se o algoritmo está identificando a ambiguidade anafórica pronominal corretamente nas frases usadas como exemplo.

### **5.2 Testes**

#### **Como executar o teste:**

No ítem Sentences Analysis, clicar em Machine Analysis, depois em Tree structure. Em Visualizacion, escolher a opção vertical e em notacion convention, escolher a opção CG-style. Escrever na caixa de texto a pergunta e clicar em Export and Download. Será exportado as informações morfológicas, com a classificação gramatical de cada palavra, e as informações sintáticas também.

**Site:** <http://visl.sdu.dk/visl/pt/>

#### **Tabela de testes**

A tabela de testes contém alguns exemplos de frases que apresentam anáforas pronominais(pronome pleno ou nulo), com ou sem ambiguidade. O objetivo é verificar se de acordo com as regras estabelecidas pelo algoritmo criado, é possível identificar se a frase possui ambiguidade anafórica pronominal.

As regras da identificação estão na primeira linha na vertical da tabela, a frase deve ter no sujeito um nome próprio; um segundo nome próprio e um pronome pessoal para ser identificada uma ambiguidade anafórica pronominal. O site do VISL contém tags para representação dos itens gramaticais, as representações que aparecem na tabela são aquelas que o algoritmo deve reconhecer na frase, que são: SUBJ para o nome próprio que compõe sujeito, prop para o segundo nome próprio e pron\*pers para o pronome pessoal.

No item de resultado esperado é classificado a expressão linguística presente na frase que podem ser: anáfora pronominal, ocorre quando um pronome retoma uma expressão, chamada de antecedente, que já foi dita antes, como na frase “Quando Paulo foi operado, ele se recuperou rapidamente”, o pronome ele se refere a Paulo, o pronome nesse caso é chamado pronome pleno, quando o pronome não aparece na frase é chamado de pronome nulo, como por exemplo: “Quando Paulo foi operado, se recuperou rapidamente”, o antecedente que é referenciado é indicado pela flexão do verbo; e a ambiguidade anafórica pronominal com pronome pleno ou nulo ocorre quando existem mais de uma opção de antecedente que o pronome pode estar fazendo referência na frase.

No item resultado obtido é conferido se o algoritmo conseguiu identificar se a frase possui ambiguidade anafórica pronominal, se o resultado for diferente é marcada de vermelho a coluna.

## Exemplos:

Exemplos do resultado da análise morfológica e sintática de algumas frases a partir do resultado da tree structure (VISL), junto com as observações das regras para identificação de ambiguidade anafórica pronominal.

<http://visl.sdu.dk/visl/pt/parsing/automatic/trees.php>

- Frase com ambiguidade anafórica pronominal(pronome pleno):

Maria conversava com Ana enquanto ela cozinhava

</p>

SOURCE: Running text

1. Maria conversava com Ana enquanto ela cozinhava

A1

STA:fcl

| -SUBJ:prop("Maria" F S) Maria <<<< SUJEITO

| -P:v\*fin("conversar" &fmc; IMPF 3S IND VFIN) conversava

| -PIV:pp

| | -H:prp("com" &right; ) com

| | -P&lt;:prop("Ana" F S) Ana <<<< OUTRO NOME PROPRIO

| -N&lt;:fcl

| -ADVL:adv("enquanto" &rel; &ks; &left;) enquanto

| -SUBJ:pron\*pers("ela" F 3S NOM) ela <<<< PRONOME PESSOAL ENCONTRADO

| -P:v\*fin("cozinhar" IMPF 3S IND VFIN) cozinhava

SOURCE: live

1. running text

A1

</p>

- Frase com ambiguidade anafórica pronominal (pronome nulo):  
Maria conversava com Ana enquanto cozinhava

&lt;/β&gt;

SOURCE: Running text

1. Maria conversava com Ana enquanto cozinhava

A1

STA:fcl

| -SUBJ:prop("Maria" F S) Maria <<<< SUJEITO

| -P:v\*fin("conversar" &lt;fmc&gt; IMPF 3S IND VFIN) conversava

| -PIV:pp

| | -H:prp("com" &lt;right&gt;) com

| | -P&lt;:prop("Ana" F S) Ana <<<< OUTRO NOME PROPRIO

| -N&lt;:fcl

| -ADVL:adv("enquanto" &lt;rel&gt; &lt;ks&gt; &lt;left&gt;) enquanto

| -P:v\*fin("cozinhar" IMPF 3S IND VFIN) cozinhava

<<<<<< AUSENCIA DE PRONOME PESSOAL NO ITEM -N&lt; - PRONOME NULO

SOURCE: live

1. running text

A1

&lt;/β&gt;

- Frase com anáfora pronominal:

Maria fez a prova porque ela estava com nota baixa

&lt;β&gt;

SOURCEX: live

2. running text

A1

SOURCE: Running text

1. Maria fez a prova porque ela estava com nota baixa

A1

STA:fcl

| -S:prop("Maria" F S)      Maria   <<<<< SUJEITO, MAS NÃO TEM OUTRO NOME

PROPRIO

| -P:v\*fin("fazer" &lt;fmc&gt; PS 3S IND VFIN)   fez

| -Od:np

| | -DN:pron("o" &lt;artd&gt; DET F &lt;indf&gt; S)      a

| | -H:n("prova" F S)   prova

| -fA:fcl

| -SUB:conj\*s("porque" &lt;fr:96&gt;)    porque

| -S:pron("ela" &lt;pers&gt; F 3S NOM)   ela

| -P:v\*fin("estar" IMPF 3S IND VFIN)   estava

| -Cs:pp

| -H:prp("com" &lt;right&gt;)      com

| -DP:np

| -H:n("nota" F S)      nota

| -DN:adj("baixo" F S)   baixa



## Referências:

CANÇADO, Márcia. **Manual de semântica noções básicas e exercícios**. 4 ed. Belo Horizonte, 2008.

FONSECA, Maria Cristina Micelli et al. Resolução de correferência pronominal no português do Brasil. **Revista Linguística**/ Revista do Programa de Pós-Graduação em Linguística da Universidade Federal do Rio de Janeiro. Volume 8, número 2, dezembro de 2012.

HAAG, Cassiano Ricardo et al. O processamento anafórico: Um experimento sobre a resolução de ambiguidades em anáforas pronominais. **Revista Linguagem em Discurso, Tubarão**, v. 4, n 1, p. 65-80, jul./dez. 2003.

SAIAS, Jose M. Gomes. **Contextualização e Activação Semântica da Selecção de Resultados em Sistemas de Pergunta-Resposta**. 2010. 192f Tese (Doutorado em Informática) – Universidade de Évora, Portugal.

SILVA, Nísia R Melo. **A ambiguidade Lexical No Gênero Textual Charge**: polissemia e hominímia. Campina Grande. 21 ed, 2016.

SILVEIRA, Gessilene. Anáfora: Algumas Considerações Sintáticas E Semânticas. **Working papers em linguística, UFSC**, n. 3, 1999.

QUARESMA, Paulo. Um sistema de Pergunta-Resposta para uma base de Documentos. **Letras de Hoje. Porto Alegre**. v. 41, n 2, p 43-63, junho, 2016.

TURNEY D. Peter. From Frequency to Machine: Vector Space Models of Semantics. **Journal of Artificial Intelligence Research** 37, p 141-188, 2010.

VIEIRA, Renata. **Linguística computacional**: princípios e aplicações. Faculdade de informática PUCRS, Porto Alegre.

VIEIRA, Renata et al. Processamento computacional de anáfora e correferência. **Revista Est. Ling**, Belo Horizonte, v 16, n. 1, p. 263-284, jan./jun. 2008.

PINHEIRO, Vláudia et al. Um analisador semântico inferencial de sentenças em linguagem natural. **Revista Linguamática**, vol 2, n 1, p 111-131.

Teste: <http://visl.sdu.dk/visl/pt/parsing/automatic/trees.php>

Nomeclatura: <http://visl.sdu.dk/visl/pt/info/symbolset-floresta.html>

<http://visl.sdu.dk/visl/pt/info/portsymbol.html>



Quando tiver ambiguidade anafórica pronominal vai aparecer o sujeito e também o P< feito na análise sintática e um nome próprio (prop)

O algoritmo desenvolvido deverá interpretar os termos anafóricos, considerando o contexto das frases, para conseguir compreender o que está sendo dito.