



UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM  
INSTITUTO DE COMPUTAÇÃO- ICOMP  
PROGRAMA PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

Impacto de técnicas de pré-processamento de texto na  
detecção de intenção e extração de parâmetros em  
sistemas de diálogo orientados a tarefa

Erick Rego Ribeiro

Brasil

2020

Erick Rego Ribeiro

Impacto de técnicas de pré-processamento de texto na  
detecção de intenção e extração de parâmetros em  
sistemas de diálogo orientados a tarefa

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, como requisito necessário para obtenção do título de Mestre em Informática.

Orientador(a)

Prof. Dr. Edleno Silva de Moura

Universidade Federal do Amazonas - UFAM

Instituto de Computação- IComp

Brasil

2020

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

R484i      Ribeiro, Erick Rego  
Impacto de técnicas de pré-processamento de texto na detecção de intenção e extração de parâmetros em sistemas de diálogo orientados a tarefa / Erick Rego Ribeiro . 2020  
64 f.: il. color; 31 cm.

Orientador: Edleno Silva de Moura  
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. sistemas de diálogo. 2. chatbot. 3. processamento de linguagem natural. 4. detecção de intenção. 5. extração de parâmetros. I. Moura, Edleno Silva de. II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



UFAM

## FOLHA DE APROVAÇÃO

**"Impacto de técnicas de pré-processamento de texto na detecção de intenção e extração de parâmetros em sistemas de diálogo orientados a tarefa"**

**ERICK REGO RIBEIRO**

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

  
Prof. Edleno Silva de Moura - PRESIDENTE

  
Prof. Aligran Soares da Silva - MEMBRO INTERNO

  
Prof. Thierson Couto Rosa - MEMBRO EXTERNO

Manaus, 27 de Março de 2020

*Este trabalho é dedicado a minha mãe Edinelza Ribeiro, que em vida me educou e criou com muito amor, além de sempre acreditar no potencial de todos os filhos.*

---

# Agradecimentos

Primeiramente agradeço aos meus pais, Edinelza e James, que me proporcionaram o privilégio de poder chegar onde cheguei, graças a eles tive a oportunidade de sanar minhas necessidades mais básicas desde o dia do meu nascimento. Além disso serei eternamente grato pelo empenho deles para me oferecer o maior conforto que poderiam.

Agradeço ao meu orientador, Edleno Silva de Moura pela oportunidade de trabalhar com um dos maiores nomes da ciência mundial em sua área. Sou muito grato por sua paciência e clareza nos ensinamentos compartilhados, pra mim é uma honra.

Sou muito grato também à Larissa Bentes, que contribuiu de forma majestosa durante vários momentos da escrita desse trabalho. Foi ela quem aguentou minhas crises de estresse, ansiedade, soube me motivar em situações difíceis. Essa pessoa é uma referência profissional e minha confidente pra todos os momentos.

Agradeço ao meu amigo Henry Vieira pelas incontáveis vezes onde debatemos o nosso entendimento sobre a construção de sistemas de diálogo. Agradeço também a todas as vezes em que ele me lembrou que é fundamental manter a humildade e estar aberto a admitir equívocos durante a busca por conhecimento. Essas conversas me ajudaram muito na fase de concepção e entendimento do problema, contribuindo diretamente para que este trabalho fosse possível.

Agradeço também à todos os professores do ICOMP que desde minha graduação contribuíram para que eu pudesse chegar a este ponto. Agradeço especialmente também aos que marcaram não apenas meu caminho acadêmico, mas minha vida, professores

Raimundo Barreto, Marco Cristo, Altigran da Silva, César Melo, Moisés Carvalho e Arilo Dias. Obrigado por toda contribuição.

Agradeço à toda equipe do Teewa, com quem eu passei mais tempo do que minha própria família, Xyah, Taigo, Matheus, Caio, Daniel, Raphael, Rúben, Marcos, Samantha, Gercidara, Ivo, Giulia, Rodrigo, Taigo, Daniel Zordan e Henrique.

Agradeço também à todos os funcionários da parte administrativa do ICOMP, que se esforçam para que não seja preciso nos preocupar com burocracias, resolvendo todo e qualquer tipo de problema que temos durante nossa jornada.

À CAPES pelo auxílio financeiro para o desenvolvimento deste trabalho.

*"N3o existe um caminho para a felicidade. A felicidade 3 o caminho."*

Thich Nhat Hanh

# Impacto de técnicas de pré-processamento de texto na detecção de intenção e extração de parâmetros em sistemas de diálogo orientados a tarefa

Autor: Erick Rego Ribeiro

Orientador: Prof. Dr. Edleno Silva de Moura

## Resumo

Devido à popularidade da internet e ao baixo preço de dispositivos móveis, as pessoas mudaram a forma como interagem umas com as outras e com empresas. No passado, a internet impulsionou o crescimento do comércio eletrônico, inicialmente com pessoas fazendo compras por meio de computadores pessoais e mais recentemente utilizando dispositivos móveis, quando o comércio eletrônico passou a ser chamado também como comércio móvel. Nos últimos anos, ao lado do crescimento do comércio móvel, o número de usuários ativos em aplicativos de trocas de mensagens também cresceu. Em resposta a tal fenômeno, empresas de vários setores têm investido no atendimento a seus clientes por meio deste tipo de aplicativos, contudo, manter pessoal qualificado para atender os clientes pode gerar alto custo. Além do custo, o atendimento também pode ficar demorado em momentos de pico, gerando insatisfação por parte dos clientes. Neste cenário o desenvolvimento de *sistemas de diálogo orientados a tarefa* surge como uma alternativa para o atendimento à clientes, graças a sua capacidade de atender a um grande número de clientes de forma ininterrupta, com boa velocidade de resposta e baixo custo. A crescente demanda por esses sistemas e os desafios envolvidos em sua construção, nos motivou a estudar sobre esse tipo de sistema. Neste estudo aprendemos que existe uma fase no desenvolvimento chamada de *compreensão de linguagem natural*,

cujo propósito é identificar a intenção do usuário a cada frase dita por ele, assim como parâmetros relacionados a essa intenção identificada. Este propósito pode ser alcançado por meio de duas tarefas, conhecidas como: detecção de intenção e extração de parâmetros. Por serem tarefas conhecidas na literatura de sistema de diálogos, e possuírem diversos trabalhos já publicados ao longo de anos, propomos nesta dissertação um estudo sobre o impacto do uso de técnicas de pré-processamento de texto aplicado em modelos utilizados nessas duas tarefas. Mais precisamente, escolhemos técnicas como *stemmer*, lematização, remoção de *stopwords* e uso de *Word Embeddings* para serem utilizados em nossos experimentos. Os experimentos foram realizados em 3 bases de dados de referência para o problema estudado, concluímos que nem todas as técnicas de pré-processamento escolhidas impactam de forma positiva quando aplicadas em trabalhos publicados na literatura. Diante das técnicas comparadas, apenas o *stemmer* resultou em ganho, um ganho de até 3% na revocação da tarefa de extração de parâmetros custando uma pequena perda de 0.9% na precisão da mesma tarefa. Já as técnicas de lematização, remoção de *stopwords* e *Word Embeddings* resultaram em perdas na revocação e precisão. Ao analisar os resultados negativos concluímos que a lematização confunde o modelo por apresentar diferentes lemas para uma mesma palavra, enquanto que a remoção de *stopwords* remove as preposições e artigos que são importantes para contextualizar e localizar os parâmetros a serem extraídos, no caso dos *Word Embeddings* a configuração dos trabalhos comparados não favoreceu o uso da técnica.

*Palavras-chave:* comércio eletrônico, sistemas de diálogo, processamento de linguagem natural, detecção de intenção, extração de parâmetros, chatbot.

# Impacto de técnicas de pré-processamento de texto na detecção de intenção e extração de parâmetros em sistemas de diálogo orientados a tarefa

Autor: Erick Rego Ribeiro

Orientador: Prof. Dr. Edleno Silva de Moura

## Abstract

After the popularity of the internet and the low price of mobile devices, the people have changed the way they interact with each other and with companies. In the past, the internet has driven the growth of e-commerce, initially with people shopping through personal computers and more recently using mobile devices, when e-commerce has come to be called also as mobile commerce. In recent years, alongside the growth of mobile commerce, the number of active users in messaging applications has also grown. In response to this phenomenon, companies from various sectors have invested in serving their customers through these types of applications, however, maintaining qualified personnel to serve can generate high costs. In addition to the cost, service can also be time-consuming at peak times, generating customer dissatisfaction. In this scenario, the development of *task-oriented dialogue systems* emerges as an alternative to customer service, thanks to its ability to serve a large number of customers continuously, with good response speed and low cost. The growing demand for these systems and the challenges involved in their construction, motivated us to study about this type of system. In this study, we learned that there is a phase in the development called *natural language understanding*, which purpose is to identify the user's intention for each sentence spoken by him, as well as parameters related to that identified intention.

This purpose can be achieved through two tasks, known as: intention detection and slot filling. As they are tasks known in the dialog system literature, and since there are several works already published over the years, we propose in this dissertation a study on the impact of the use of text pre-processing techniques applied in models used in these two tasks. More precisely, we chose techniques such as stemmer, lemmatization, *stopwords* removal and using *Word Embeddings* to be used in our experiments. Experiments carried out in reference datasets for the problem studied indicate that not all the pre-processing techniques chosen had a positive impact when applied in works published in the literature. In view of the compared techniques, only the *stemmer* results in a gain, a gain of up to 3% in the recall of the parameter extraction task, costing a small loss of 0.9% in the same task. Since stemming techniques, removing *stopwords* and *Word Embeddings* resulted in changes in recall and accuracy. When analyzing the completed results, which shows a confused reader or model for presenting different slogans for the same word, while removing *stopwords* removes prepositions and articles that are important to contextualize and use the items to be extracted, no case *Word Embeddings*, a configuration of the compared works did not favor the use of the technique.

*Keywords:* electronic commerce, dialogue systems, natural language processing, intention detection, slot filling, chatbot.

---

## Lista de ilustrações

Figura 1 – Divisão de um sistema de diálogo orientado a tarefa em etapas. . . . .	18
Figura 2 – Ilustração de uma rede neural artificial com 4 camadas. . . . .	23
Figura 3 – Ilustração da relação entre palavras condensadas em um espaço vetorial. . . . .	28
Figura 4 – Ilustração do funcionamento do <i>CBOW</i> e <i>skip-gram</i> . . . . .	29
Figura 5 – Arquitetura no modelo Triangular-CRF. . . . .	32
Figura 6 – Arquitetura do modelo RNN-NLU. . . . .	34
Figura 7 – Arquitetura do modelo slot-gated. . . . .	34
Figura 8 – Arquitetura do modelo SF-ID Network. . . . .	35

---

## Lista de tabelas

Tabela 1 – Ilustração das tarefas em destaque na etapa de compreensão de linguagem natural. . . . .	20
Tabela 2 – Principais características das bases de dados utilizadas . . . . .	39
Tabela 3 – Ilustração de uma matriz de confusão . . . . .	42
Tabela 4 – Resultado das experimentações com trabalhos comparados sem o uso de técnicas de processamento de texto nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol) . . . . .	49
Tabela 5 – Resultado das experimentações com trabalhos comparados após utilizar a técnica de stemmer nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol) . . . . .	51
Tabela 6 – Resultado das experimentações com trabalhos comparados após utilizar a técnica de lematização nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol) . . . . .	53
Tabela 7 – Resultado das experimentações com trabalhos comparados após utilizar a técnica remoção de stopwords nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol) . . . . .	54
Tabela 8 – Resultado das experimentações com trabalhos comparados após utilizar a técnica de <i>Word Embeddings</i> nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol) . . . . .	56

---

# Lista de abreviaturas e siglas

**AM** *Aprendizado de Máquina*

**DI** *Detecção de Intenção*

**EP** *Extração de Parâmetros*

**IA** *Inteligência Artificial*

**PLN** *Processamento de Linguagem Natural*

**RI** *Recuperação de Informação*

**RNA** *Rede Neural Artificial*

**SDOT** *Sistemas de diálogo orientados a tarefa*

---

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>1.1</b>	<b>Introdução</b>	<b>17</b>
<b>1.2</b>	<b>Objetivos</b>	<b>21</b>
1.2.1	Objetivo geral	21
<b>1.3</b>	<b>Organização do Trabalho</b>	<b>21</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>22</b>
<b>2.1</b>	<b>Aprendizagem de Máquina</b>	<b>22</b>
2.1.1	Rede Neural Artificial	23
<b>2.2</b>	<b>Sistemas de diálogo e compreensão de linguagem natural</b>	<b>24</b>
2.2.1	Detecção de Intenção e Extração de Parâmetros	24
<b>2.3</b>	<b>Técnicas de pré-processamento de texto</b>	<b>26</b>
2.3.1	Stemmer	26
2.3.2	Lematização	26
2.3.3	Remoção de stopwords	27
2.3.4	Word Embeddings	27
2.3.5	GloVe	29
2.3.6	FastText	29
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>31</b>
<b>3.1</b>	<b>Modelagem da tarefa de detecção de intenção e extração de parâmetros</b>	<b>31</b>
3.1.1	Modelagem Independente	32
<b>3.2</b>	<b>Impacto do uso de técnicas de pré-processamento de texto</b>	<b>35</b>

<b>4</b>	<b>EXPERIMENTOS</b>	<b>38</b>
<b>4.1</b>	<b>Configuração dos experimentos</b>	<b>39</b>
4.1.1	Bases de dados utilizada	39
4.1.2	Trabalhos comparados	40
4.1.3	Ambiente de execução	41
4.1.4	Métricas de avaliação	41
4.1.4.1	Acurácia	43
4.1.4.2	Revocação	43
4.1.4.3	Precisão	43
4.1.4.4	F-score	44
4.1.5	Metodologia	44
4.1.5.1	Processo de escolha da melhor implementação para cada uma das técnicas de pré-processamento utilizadas	44
4.1.5.2	Processo de adaptação dos trabalhos comparados para suportar as técnicas de pré-processamento	45
4.1.5.3	Processo de adaptação dos trabalhos comparados para suportar novas métricas de avaliação	47
4.1.6	Resultados	47
4.1.6.1	Impacto do uso de stemmer nas tarefas de detecção de intenção e extração de parâmetros	49
4.1.6.2	Impacto do uso de lematização nas tarefas de detecção de intenção e extração de parâmetros	51
4.1.6.3	Impacto do uso de remoção de stopwords nas tarefas de detecção de intenção e extração de parâmetros	53
4.1.6.4	Impacto do uso de Word Embeddings nas tarefas de detecção de intenção e extração de parâmetros	54
<b>5</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS</b>	<b>57</b>
	<b>Referências</b>	<b>61</b>

---

# Introdução

**E**ste capítulo apresenta e contextualiza o campo de estudo denominado como Sistemas de diálogo Orientados a Tarefas (SDOT). O capítulo também apresenta a relação de fatos históricos, aplicações comerciais e produções científicas que culminaram na motivação deste trabalho.

## 1.1 Introdução

Devido à grande popularidade da internet e ao baixo preço de dispositivos móveis, as pessoas mudaram a forma como interagem umas com as outras e também com empresas. No passado, a internet impulsionou o crescimento do comércio eletrônico, inicialmente com pessoas fazendo compras por meio de computadores pessoais e mais recentemente com acessos por meio de dispositivos móveis, quando o comércio eletrônico passou a ser chamado também de comércio móvel ([FELDMAN, 2000](#); [STAFFORD; GILLENSON, 2003](#)).

Nos últimos anos, o número de usuários ativos mensais em aplicativos de troca de mensagens instantâneas ultrapassou os quatro maiores aplicativos de redes sociais em todo mundo ([INTELLIGENCE, 2016](#)). Em resposta a tal fenômeno, empresas de vários setores, tais como companhias aéreas, marcas de moda e serviços de entrega, têm investido no atendimento a seus clientes por meio de aplicativos de troca de mensagens. Algumas companhias aéreas já utilizam esses aplicativos para oferecer a seus clientes a possibilidade de fazer *check-in*, de fazer perguntas ou ainda de fornecer atualizações de

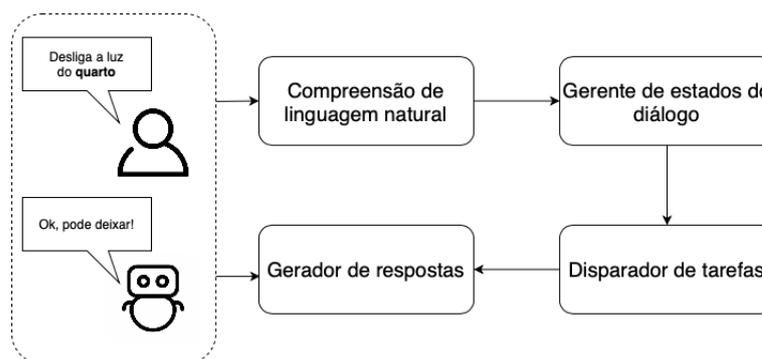


Figura 1 – Divisão de um sistema de diálogo orientado a tarefa em etapas.

informações sobre os voos.

Apesar do uso crescente, um problema com o atendimento por meio de aplicativos de troca de mensagens é a necessidade de se manter pessoal qualificado para interagir com os clientes, o que pode gerar alto custo. Além do custo, o atendimento também pode demorar em momentos de pico de demanda, gerando insatisfação por parte dos clientes. O desenvolvimento de sistemas de conversação automática, também conhecidos como Sistemas de Diálogo Orientados a Tarefa (SDOT), surge nesse cenário como uma alternativa interessante para o atendimento a clientes. Tais sistemas são capazes de atender a um grande número de clientes de forma ininterrupta, com velocidade de resposta e baixo custo (YOUNG et al., 2013; SHAWAR; ATWELL, 2007).

A crescente demanda por sistemas de diálogo orientados a tarefa e os desafios envolvidos em sua construção motiva o estudo sobre esse tipo de sistema. Os sistemas de diálogo orientados a tarefa são uma classe específica de sistemas de diálogo cujo propósito é realizar uma tarefa específica. Por exemplo, realizar reserva de voos (HEMPHILL; GODFREY; DODDINGTON, 1990), buscar por restaurantes (WILLIAMS, 2012) ou ainda buscar por informações turísticas (HENDERSON; THOMSON; WILLIAMS, 2014). Os diálogos são caracterizados por conversas curtas, normalmente de apenas uma interação simples, também conhecida como turno, ou talvez meia dúzia de interações, com o objetivo do sistema obter as informações necessárias para concluir a tarefa desejada pelo usuário. Considera-se que um diálogo acaba sempre que uma determinada tarefa alvo do usuário é concluída.

Lembrando que um diálogo pode ser composto por uma ou mais interações entre o usuário e o sistema, pode-se dizer que um sistema orientado a tarefa tem como

missão identificar a tarefa que o usuário deseja dentro de um diálogo e ser capaz de executá-la. Atualmente trabalhos relacionados à área dividem esses sistemas em quatro componentes principais (CHEN et al., 2017), como ilustrado na Figura 1. O primeiro é o componente de compreensão de linguagem natural. Após uma mensagem escrita pelo usuário, esse componente a classifica de acordo com uma lista de intenções previamente determinada. Além de determinar a intenção, o componente também extrai do texto parâmetros associados à intenção, associando cada palavra do texto a no máximo 1 parâmetro.

O segundo componente é o gerente de estados do diálogo. Ele verifica se a lista de parâmetros fornecidos até o momento é suficiente para executar a tarefa associada à intenção do usuário. Esse componente controla os diálogos entre o usuário e o sistema, sendo capaz, por exemplo, de identificar que uma lista de parâmetros fornecida na última mensagem escrita complementa informação já passada em mensagens anteriores no diálogo. Dessa forma, pode-se dizer que ele gerencia o estado atual do diálogo tratado, sendo o único componente que apresenta informação sobre o diálogo como um todo, e não somente informação sobre a mensagem do turno atual.

O terceiro componente é o disparador de tarefas. Ele é responsável por executar a tarefa em si caso o gerente de estados do diálogo conclua que uma tarefa que está pronta para ser executada. O quarto componente é o gerador de respostas. Ele tem como propósito transformar os resultados obtidos pelos componentes anteriores em uma resposta em linguagem natural. Essa resposta pode ser um pedido de informação complementar de parâmetros para uma intenção, de modo a sinalizar que uma mensagem ou intenção não foi entendida, pode ser uma mensagem de erro ou ainda a resposta com o resultado da tarefa executada.

Nesta dissertação iremos estudar apenas o problema de compreensão de texto em linguagem natural (primeiro componente) em sistemas de diálogo orientados a tarefa, ou seja, estudaremos apenas melhorias no módulo de compreensão de texto em linguagem natural. O cenário para este problema pode ser melhor descrito como uma sessão de conversação entre o usuário e o sistema, onde o usuário deve descrever de forma textual o seu interesse em executar uma ação, especificando no mesmo texto todos

os parâmetros necessários para fazer que o sistema possa executar a tarefa especificada. Assumimos aqui portanto que as conversas terão apenas um turno.

<b>Sentença</b>	Apagar	a	lâmpada	da	sala	de	estar
<b>Parâmetros</b>	O	O	B-dispositivo	O	B-local	I-local	I-local
<b>Intenção</b>	Desligar um dispositivo						

Tabela 1 – Ilustração das tarefas em destaque na etapa de compreensão de linguagem natural.

Como vimos, o componente de compreensão de texto em linguagem natural é composto pelas tarefas de detecção de intenção e a extração de parâmetros associados a tal intenção. A Tabela 1 exemplifica a combinação dessas duas tarefas. No exemplo da Tabela 1 é apresentado a sentença "Apagar a lâmpada da sala de estar". Um ser humano ao ler essa sentença intuitivamente entende o seu propósito e percebe minúcias no texto que o ajudará a executar tal ação, compreende que tal sentença corresponde ao ato de "Desligar um dispositivo" que está localizado na sala de estar. No entanto, uma máquina não é tão inteligente assim, é preciso ensiná-la. Primeiramente, por meio da detecção de intenção o texto por completo é analisado, e de forma algorítmica é detectada a intenção de "Desligar um dispositivo". Depois a extração de parâmetros extrai uma ou mais partes importantes do texto por meio de algoritmos que analisam palavra por palavra e extraem partes importantes. Um formato bastante utilizado para extrair pedaços importantes do texto é o IOB (RAMSHAW; MARCUS, 1999), onde I (*Inside*) significa que a palavra atual está contida no trecho importante que será extraído, O (*Outside*) significa que a palavra atual não será extraída e B (*Begin*) significa que a palavra atual é o início de um trecho que será extraído.

O campo de estudo de sistemas de diálogo orientados a tarefa é repleto de modelo computacionais capazes de solucionar essas duas tarefas, assumindo uma margem de erro esperada. No entanto uma parte muitas vezes deixada em segundo plano no estudo desses modelos computacionais é o uso de técnicas de pré-processamento de textos que podem ser empregadas antes da aplicação do modelo. De modo a exemplificar tais técnicas, pode-se citar operações como a remoção de *stopwords*, extração de radicais de palavras, dentre outras.

Diante do que foi apresentado até o momento sobre o problema de compreensão

de texto em linguagem natural, vimos a oportunidade de investigar a seguinte hipótese de pesquisa. É possível obter ganhos significativos com a aplicação de técnicas de pré-processamento de texto, ao se desenvolver um módulo de compreensão de texto em linguagem natural para um sistema de diálogo orientado a tarefa?

Para responder a tal pergunta, apresenta-se nesta dissertação um estudo onde aplicou-se técnicas de pré-processamento de texto em trabalhos de referências nas tarefas de detecção de intenção e extração de parâmetros, comparando-se sempre o resultado da tarefa com e sem o uso do pré-processamento de texto. Ao experimentar essa hipótese, buscou-se fazer um estudo com diferentes métodos e bases de dados publicas.

## 1.2 Objetivos

### 1.2.1 Objetivo geral

O objetivo geral desta dissertação consiste em estudar, implementar e avaliar o impacto do uso de técnicas de pré-processamento textuais aplicadas ao problema de compreensão de texto em linguagem natural no contexto de sistemas de diálogo orientados a tarefa. Pretende-se trabalhar mais especificamente com as tarefas de detecção de intenção e extração de parâmetros. Dentre as possíveis técnicas de pré-processamento de texto escolhemos as técnicas de extração de radicais (*stemmer*), lematização, remoção de *stopwords* e representações vetoriais de palavras com embeddings pré-treinados.

## 1.3 Organização do Trabalho

O texto desta dissertação está organizado da seguinte forma. O capítulo 2 apresenta a referencial teórico, de modo que sirva como base para o entendimento do trabalho. O capítulo 3 apresenta os trabalhos da literatura relacionado ao problema apresentado. No capítulo 4 são descritos os experimentos e os resultados obtidos. Por fim, no capítulo 5 são apresentadas as conclusões da dissertação e trabalhos futuros.

---

## Referencial Teórico

**E**ste capítulo tem como objetivo fundamentar o escopo teórico no qual esta dissertação está inserida, apresentar os termos da área e introduzir conceitos que auxiliem a leitura do trabalho.

### 2.1 Aprendizagem de Máquina

O Aprendizado de Máquina (AM) é uma subárea da Inteligência Artificial (IA). Essa subárea tem como objetivo desenvolver técnicas capazes de ensinar ao computador como aprender determinada tarefa. Segundo ([BISHOP, 2006](#)) o aprendizado de máquina é um campo de pesquisa fundamentado na Inteligência Artificial e na Estatística. Já ([MITCHELL, 1997](#)), acrescenta que o aprendizado de máquina é definido como a capacidade de melhorar o desempenho na realização de alguma tarefa por meio da experiência. Atualmente, algoritmos de aprendizagem são aplicados em diversas áreas como, por exemplo, Processamento de Linguagem Natural (PLN) e Recuperação de Informação (RI).

De modo geral, os algoritmos de aprendizado de máquina podem ser divididos em três tipos de aprendizado: supervisionado, não supervisionado e por reforço. Nessa dissertação iremos focar em classificação, um tipo de algoritmo supervisionado. Isso porque as tarefas envolvidas no problema estudo são tratadas como classificação, a Seção [2.2.1](#) abordará esse assunto com mais detalhes. A metodologia da classificação consiste em receber um conjunto de dados de entrada e um valor esperado para a

variável de saída, conhecida como rótulo ou classe. O algoritmo de classificação deve convergir para o rótulo esperado por meio de uma fase chamada de treinamento. No treinamento um conjunto de dados chamado, dados de treino, é passado para o algoritmo de modo que ele aprenda a prever corretamente o rótulo para novos dados. Ao término do treinamento uma nova fase é aplicada, a fase de teste. No teste o desempenho do algoritmo de classificação é avaliado por meio de métricas que aferem quantos rótulos ele acertou para dados ainda não vistos. Essa fase serve para verificar se o algoritmo aprendeu a resolver o problema modelado.

### 2.1.1 Rede Neural Artificial

Uma rede neural artificial (RNA) é um modelo computacional inspirado no comportamento do sistema nervoso central de animais. As RNAs são compostas por neurônios artificiais, onde cada neurônio possui um vetor de entrada  $x_i$ , pesos  $w$  que conectam os neurônios e uma função ativação que determina o valor de saída do neurônio. Os neurônios estão organizados em camadas, de modo que os neurônios de uma camada enviem seus valores de saída para a camada posterior. A Figura 2 ilustra a arquitetura padrão de uma rede neural artificial com 4 camadas. A camada mais à esquerda é conhecida como camada de entrada, enquanto a camada mais à direita é a camada de saída, e as duas camadas intermediárias são as camadas escondidas.

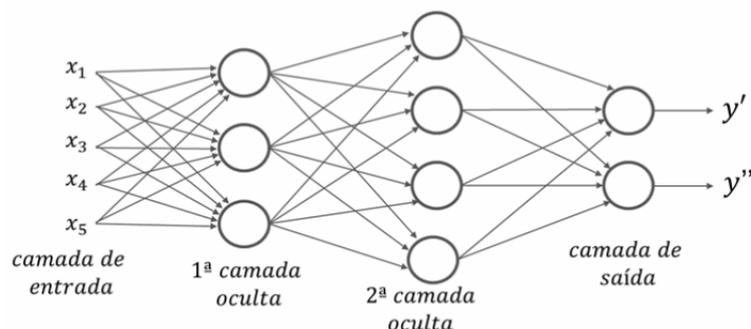


Figura 2 – Ilustração de uma rede neural artificial com 4 camadas.

## 2.2 Sistemas de diálogo e compreensão de linguagem natural

A construção de sistemas de diálogo orientados a tarefa tem sido alvo de estudo ao longo dos anos. Recentemente esse tipo de sistema ganhou maior notoriedade e se popularizou junto com aplicações como *chatbot* e assistente virtual. Exemplos icônicos podem ser dados, tais como as soluções comerciais *Google Assistant*<sup>1</sup> e *Amazon Alexa*<sup>2</sup>, sistemas que auxiliam os usuários a realizarem pequenas tarefas cotidianas, incluindo a compra de roupas ou comida, desligar uma lâmpada, criar lembretes ou tocar uma música. O componente principal por trás da construção de bons sistemas de diálogo orientados a tarefa é conhecido como compreensão de linguagem natural.

O componente de compreensão de linguagem natural é o responsável por resolver o principal problema enfrentado na construção desses sistemas. O problema consiste em entender quais ações o usuário deseja realizar sobre o sistema, além de identificar argumentos relacionados à intenção principal. Por exemplo, supondo que um usuário informe a seguinte frase, "mostre-me todos os voos que vão de Manaus para Belém", o sistema deve ser capaz de entender que o usuário tem como intenção principal a listagem de voos, mas também adiciona em sua frase duas informações que complementam a intenção principal. Chamamos essas informações complementares de parâmetros, que no exemplo, são as palavras "Manaus" e "Belém". Esse problema é resolvido por meio de duas tarefas conhecidas como Detecção de Intenção (DI) e Extração de Parâmetros (EP).

### 2.2.1 Detecção de Intenção e Extração de Parâmetros

A tarefa de detecção de intenção consiste em detectar automaticamente no texto a intenção do usuário dentro de um conjunto finito de possibilidades. As intenções são propriedades globais do enunciado, e entendê-las significa conhecer o propósito do usuário a cada frase dita por ele. Em outras palavras, é a captura do objetivo ou a ação

---

<sup>1</sup> <https://assistant.google.com/>

<sup>2</sup> <https://www.amazon.com>

principal que o usuário deseja executar. Esta tarefa é uma instância de um problema de classificação, em que dada uma frase em formato de texto para um algoritmo especializado, o algoritmo deve inferir um único rótulo  $y^{\text{intenção}}$  para a frase toda. Essa tarefa originou-se em meados da década 90 com o trabalho de (GORIN; RICCARDI; WRIGHT, 1997). O propósito desse trabalho era desenvolver um sistema capaz de realizar o redirecionamento de chamadas para o teleatendimento, também conhecido como *call-router*. Neste cenário o sistema não precisava resolver o problema do usuário, mas apenas transferir a chamada para alguém ou algo que pudesse resolver. Para isso, o sistema deveria reconhecer aspectos da fala do usuário que pudessem ser utilizados na identificação do tipo da chamada, e posteriormente mapear para o setor responsável por continuar o atendimento. Assumindo que cada setor pode ser representado como um rótulo, esta tarefa foi tratada como um problema de classificação.

A tarefa de extração de parâmetros é outra tarefa importante no problema de compreensão de linguagem natural. Os parâmetros são propriedades locais do enunciado, e entendê-los significa conhecer especificidades do que o usuário deseja. Em outras palavras, a extração de parâmetros tem como objetivo extrair informações obrigatórias para a execução do objetivo do usuário. Por exemplo, quando o usuário utiliza o comando "ligue a lâmpada do quarto", além da identificação da intenção "ligar a lâmpada" é necessário também extrair do texto qual a localização do dispositivo a ser ligado, neste caso a lâmpada localizada no quarto. Note que a tarefa de extração de parâmetros está diretamente relacionada à tarefa de detecção de intenção. A tarefa de extração de parâmetros é uma instância de um problema conhecido como rotulagem de sequência. A rotulagem de sequência é um tipo de tarefa que envolve a atribuição algorítmica de um rótulo para cada membro de uma sequência de valores observados. Quando os valores observados são textos, a representação é feita por uma sequência de palavras  $x = (x_1, x_2, \dots, x_n)$  que deve ser mapeada para um conjunto de rótulos  $y^{\text{parâmetros}} = (y_1, y_2, \dots, y_n)$  com o mesmo tamanho da entrada.

## 2.3 Técnicas de pré-processamento de texto

Ao trabalhar com um problema cuja entrada é texto, utilizar técnicas de pré-processamento de texto é fundamental. No pré-processamento os dados são processados e normalizados utilizando-se técnicas que buscam transformar os dados para formatos em que os classificadores possam alcançar bons resultados. As técnicas de pré-processamento podem afetar significativamente o desempenho individual dos algoritmos utilizados, de forma positiva ou negativa, e por consequência o desempenho final do sistema. Dentre as diversas técnicas de pré-processamento existentes na literatura, escolhemos para esta dissertação estudarmos o impacto do uso das seguintes técnicas: *stemmer*, lematização, remoção de *stopwords* e representação de palavras por meio de *Word embeddings*.

### 2.3.1 Stemmer

Uma técnica bastante utilizada na literatura é o processo de radicalização ou *stemmer*. A técnica de *stemmer* consiste em reduzir as palavras aos seus radicais por meio da remoção de seus afixos, prefixos e sufixos, de forma que palavras morfológica-mente relacionadas sejam representadas em uma única forma comum. Por exemplo, as palavras “*quero*”, “*queria*”, “*queremos*” (todas derivadas do verbo “*querer*”), após o processo de radicalização devem ser reduzidas ao radical “*quer*”. Um possível efeito colateral negativo dessa técnica é a possibilidade de produzir palavras inexistentes no idioma (KRAAIJ; POHLMANN, 1994; ORENGO; HUYCK, 2001).

### 2.3.2 Lematização

Semelhante ao *stemmer*, a técnica de lematização também busca reduzir palavras a uma forma comum. No entanto, a lematização difere do *stemmer* por garantir que a forma comum exista no idioma. Essa forma comum na lematização, chama-se *lema*. Um *lema* é a forma canônica de uma palavra que leva em consideração a classe gramatical da palavra, ou seja, o contexto em que ela foi utilizada. Por exemplo, no inglês temos a

palavra "*better*", cuja tradução significa "*melhor*" quando traduzida para o português, após aplicar a lematização o lema correspondente é "*good*", cuja tradução é "bom". Outro exemplo ainda mais peculiar no uso da lematização, é o da palavra "*meeting*", que no inglês pode significar o verbo "*conhecer*" ou o substantivo "*reunião*".

### 2.3.3 Remoção de stopwords

A técnica de remoção de *stopwords* é o processo de remover palavras irrelevantes do texto. Partindo da premissa de que a maioria dos conjuntos de dados de texto contém palavras desnecessárias, como palavras com alta ocorrência, com erros ortográficos, artigos, pronome, gírias e etc. Esta técnica consiste em criar uma lista de palavras irrelevantes e garantir que essas palavras não sejam consideradas no texto. Por exemplo, uma lista padrão de *stopwords* no português contém palavras como "*a*", "*um*", "*of*", "*since*", ou seja, palavras que são usadas no respectivo idioma com muita frequência e que comunica pouca informação sobre o conteúdo do documento. Em muitos algoritmos, especialmente algoritmos de aprendizado estatístico e probabilístico, esse tipo de ruído pode resultar em efeitos adversos no desempenho do sistema. Logo, a suposição por trás da remoção de *stopwords* é de que as palavras que permanecem são palavras com maior relevância, o que resulta em modelos de maior qualidade.

### 2.3.4 Word Embeddings

*Word embeddings* é uma técnica de representação de palavras, onde cada palavra é codificada como um vetor numérico de  $D$  dimensões em um espaço vetorial. Essa representação é capaz de capturar informações semânticas e sintáticas das palavras, de tal forma, que palavras semanticamente similares sejam mapeadas para regiões próximas no espaço vetorial. Por exemplo, as palavras "avião", "aeronave", "helicóptero" e "dirigível", provavelmente estarão mais próximas umas das outras. Por outro lado, estas mesmas palavras devem estar mais distantes de palavras como "homem", "mulher", "criança" e "menina" que representam outro conjunto de palavras similares. A represen-

tação de palavras com *Word embeddings* é utilizada como codificação de palavras em diferentes áreas que envolvem processamento de texto, tais como, processamento de linguagem natural, recuperação de informação e análise de sentimentos.

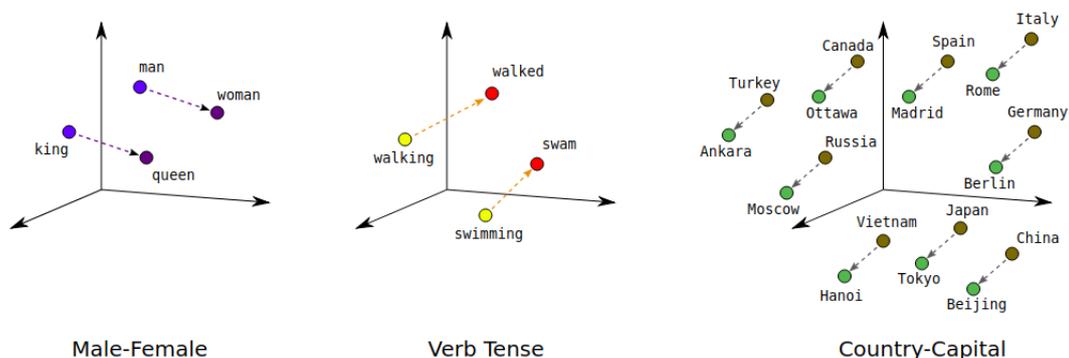


Figura 3 – Ilustração da relação entre palavras condensadas em um espaço vetorial.

Além desse agrupamento de palavras utilizadas em contextos similares, também é possível extrair relações entre duas palavras, tais como, o gênero masculino e o feminino, localidades como países e cidades, além de variações gramaticais como tempos verbais e outras relações. A Figura 3 ilustra o que foi dito até o momento sobre *Word Embeddings*. Diversos métodos de geração de *Word Embeddings* foram propostos com o objetivo de aprender a melhor representação de palavras no espaço vetorial. Um dos mais conhecidos é o trabalho de (MIKOLOV et al., 2013), onde os autores apresentaram um modelo baseado em redes neurais para o processo de representação de palavras no espaço vetorial. Esse modelo foi chamado de *Word2vec* e possibilita dois processos de representação de palavras: *continuous bag-of-words (CBOW)* e *skip-gram*.<sup>4</sup> No CBOW, o modelo é treinado para prever a palavra atual baseada em uma janela de palavras de contexto ao seu redor. No processo de treinamento a ordem das palavras utilizadas como contexto não influencia a inferência. Já o *skip-gram*, faz esse processo de maneira inversa, ele utiliza a palavra atual para inferir o contexto no qual ela foi utilizada. Os dois processos são ilustrados pela Figura 4.

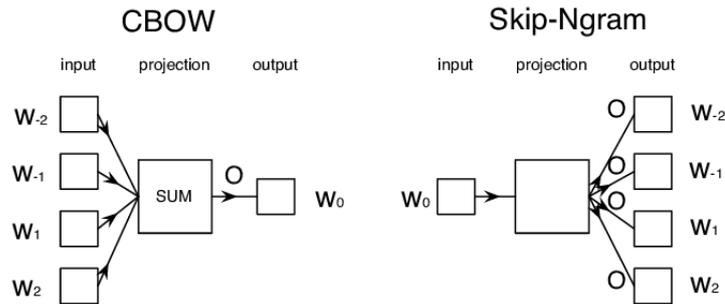


Figura 4 – Ilustração do funcionamento do CBOW e skip-gram.

### 2.3.5 GloVe

Após a publicação do modelo *Word2vec* em (MIKOLOV et al., 2013), houve um aumento de trabalhos sobre *Word Embeddings*. Um dos trabalhos mais bem sucedidos nessa área foi o *GloVe* (PENNINGTON; SOCHER; MANNING, 2014). Nesse trabalho, o autor explica o processo de otimização obtido pelo *Word2vec* como um tipo especial de fatoração para matrizes de ocorrência conjunta de palavras. No *GloVe* uma nova forma de aprender os vetores de palavras é apresentada. Esta nova forma consiste em examinar as palavras como uma matriz global de co-ocorrências. A vantagem na utilização desse método é que diferentemente do *Word2vec*, o *GloVe* não dependerá apenas de estatísticas locais, mas incorpora estatísticas globais da co-ocorrência de palavras para obter vetores.

### 2.3.6 FastText

O *FastText* é uma extensão do modelo *Word2vec* de (MIKOLOV et al., 2013), também publicado por Tomas Mikolov. No entanto, o *FastText* inova ao propor o aprendizado de representações para sub-partes da palavra e chegar na representação da palavra pela soma dos vetores de suas sub-partes. Esta nova abordagem permite que o *FastText* aprenda representações vetoriais semelhantes para palavras como "amor", "amado" e "amando", mesmo que elas possam aparecer em contextos diferentes. Este recurso melhora a aprendizagem em linguagens fortemente flexionadas (BOJANOWSKI et al., 2017), fazendo com que os vetores aprendidos pelo *FastText* sejam mais eficientes na representação de palavras, porém o processo de treinamento é mais demorado

comparado a outros modelos.

---

## Trabalhos Relacionados

**E**ste capítulo é dividido em duas partes. A primeira parte apresenta os principais trabalhos relacionados ao problema de compreensão de linguagem natural e duas possíveis modelagens nas quais os algoritmos para as tarefas de detecção de intenção e extração de parâmetros podem ser aplicados.

Já a segunda parte apresenta trabalhos semelhantes a hipótese de pesquisa dessa dissertação. Vale lembrar que o foco dessa dissertação é analisar o impacto do uso de técnicas de pré-processamento de texto aplicadas ao problema de compreensão de linguagem natural. De modo análogo, a segunda parte apresenta os trabalhos que também analisaram o impacto de técnicas de processamento de texto, porém em outros cenários.

### 3.1 Modelagem da tarefa de detecção de intenção e extração de parâmetros

A literatura que envolve as tarefas detecção de intenção e extração de parâmetros é dividida em duas modelagens: métodos que utilizam a modelagem independente e métodos que utilizam a modelagem conjunta.

### 3.1.1 Modelagem Independente

A modelagem independente foi a primeira a surgir. Essa modelagem parte da hipótese de que as tarefas a serem resolvidas são instâncias de problemas isolados. Assumindo que são problemas isolados, os pesquisadores que defendem essa abordagem utilizam o estado-da-arte em classificação de textos para resolver a tarefa de detecção de intenção e o estado-da-arte em rotulagem de sequências para resolver a tarefa de extração de parâmetros.

No geral, ao contribuir para modelagem independente os autores devem propor a aplicação de um novo algoritmo ou melhoria de um existente para cada uma das tarefas citadas. Por fim, na construção de um sistema de diálogo por completo, a etapa de compreensão de linguagem natural é a mescla dos resultados isolados de ambas as tarefas. Nesta dissertação não iremos nos aprofundar em trabalhos que utilizam a modelagem independente.

Em (JEONG; LEE, 2008) foi proposto o trabalho pioneiro na modelagem conjunta. Nesse trabalho os autores propuseram um modelo chamado *Triangular-CRF*, o qual consiste em acoplar ao algoritmo Conditional Random Field (CRF) um campo aleatório adicional, como ilustrado na Figura 5. O novo campo permite ao modelo aprender as intenções do usuário e, para cada intenção, aprender um conjunto único de padrões de parâmetros a serem extraídos. A principal desvantagem desse modelo é a falta de escalabilidade, pois a medida que a quantidade de intenções cresce o modelo torna-se impreciso.

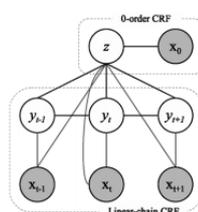


Figura 5 – Arquitetura no modelo Triangular-CRF.

Posteriormente com a disseminação de métodos baseados em redes neurais artificiais, em particular arquiteturas neurais profundas, novas propostas de trabalhos envolvendo a modelagem conjunta e arquiteturas neurais profundas foram apresentadas.

Em (HAKKANI-TÜR et al., 2016) foi proposta uma arquitetura baseada em Recurrent Neural Network (RNN) para múltiplos domínios, e que aprende as tarefas de detecção de intenção e extração de parâmetros de forma unificada. A RNN é um tipo de rede neural artificial projetada para reconhecer padrões de dados sequências. Elas são normalmente aplicadas em problemas que envolvem texto ou áudio. Nesse trabalho os autores demonstraram a eficiência de métodos baseados em arquiteturas neurais profundas. Os autores destacaram três contribuições em seu trabalho: 1) uma arquitetura conjunta baseada em RNNs para as tarefas de detecção de intenção e extração de parâmetros; 2) um modelo para múltiplos domínios que permite que os dados de cada domínio se reforcem mutuamente; e 3) a investigação de arquiteturas alternativas para modelar o problema compreensão da linguagem natural.

No mesmo ano, em (LIU; LANE, 2016) foi proposta uma arquitetura neural também baseada em RNNs, porém com a acréscimo de mecanismos de atenção para a resolução conjunta das tarefas de detecção de intenção e extração de parâmetros. O mecanismo de atenção é um recurso que permite a rede concentrar o aprendizado em partes diferentes da sequência de entrada, e correlacionar a cada etapa de geração da sequência de saída, o que melhora o desempenho das RNNs. Os autores propuseram ainda o uso de três camadas de RNNs, como ilustrado na Figura 6. Um das camadas é responsável por codificar cada uma das palavras da sequência de entrada, enquanto que segunda camada realiza o mapeamento de cada palavra para um rótulo correspondente, o que corresponde à tarefa de extração de parâmetros. Simultaneamente, a terceira camada gera o rótulo da intenção detectada. O aprendizado simultâneo entre cada uma das três camadas de RNNs ocorre mediante ao compartilhamento dos pesos de saída de cada camada e uma função de custo que durante o treinamento ajusta os pesos da rede como todo. Nos experimentos os autores mostram que sua proposta é superior aos resultados obtidos em trabalhos anteriores que utilizavam a abordagem independente.

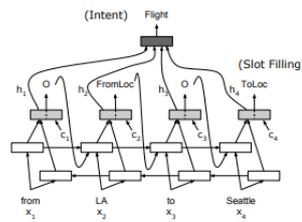


Figura 6 – Arquitetura do modelo RNN-NLU.

Em (GOO et al., 2018) foi proposto um modelo chamado Slot-Gated, uma abordagem que também utiliza RNNs e mecanismos de atenção. No entanto, nesse trabalho os autores propuseram um modelo mais flexível, no qual existe a possibilidade de escolher a que tarefa o modelo deve direcionar a sua atenção, por exemplo, é possível escolher se durante o treinamento o modelo deve focar nos parâmetros a serem extraídos em conjunto com a intenção principal do enunciado ou apenas na intenção. As duas variantes do modelo Slot-Gated são ilustradas na Figura 7.

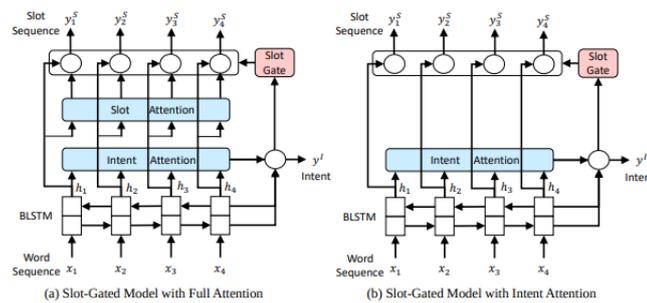


Figura 7 – Arquitetura do modelo slot-gated.

A justificativa para esse novo modelo proposto é fundamentada no argumento de que o trabalho anterior não modela de forma explícita a relação entre a intenção detectada e cada parâmetro extraído. No geral, ao comparar o modelo Slot-Gated como trabalhos anteriores os autores destacam três contribuições: 1) um novo modelo de atenção com melhores resultados que os trabalhos anteriores; 2) a experimentação em mais de uma base de dados, o que reforça a superioridade do modelo proposto; e 3) o modelo que segundo eles relaciona as tarefas de detecção de intenção e extração de parâmetros de forma explícita.

Diante dos avanços que os trabalhos anteriores apresentaram ao utilizar o mecanismo de atenção em conjunto com RNNs. Em (HAIHONG et al., 2019) propuseram o modelo chamado SF-ID Network, que possui dois mecanismos de atenção em

sua composição e duas sub-redes, a rede SF e rede ID. A rede ID é utilizada para a tarefa de detecção de intenção e a outra para extração de parâmetros. Ideia por trás da utilização de duas sub-redes e dos dois mecanismos de atenção é semelhante ao conceito de compartilhamento de pesos apresentado no trabalho de (LIU; LANE, 2016). No entanto, em vez de compartilhar os pesos de cada camada, Haihong compartilha os vetores de contexto aprendidos por cada um dos mecanismos de atenção. A sub-rede SF, além de utilizar cada palavra da sequência de entrada, também utiliza as informações do vetor de contexto que a sub-rede ID aprendeu. Em contrapartida, a sub-rede ID também utiliza o vetor de contexto aprendido pela sub-rede SF, o modelo por completo é ilustrado na Figura 8.

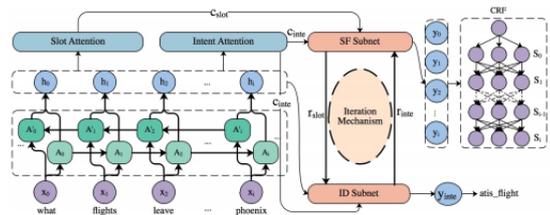


Figura 8 – Arquitetura do modelo SF-ID Network.

## 3.2 Impacto do uso de técnicas de pré-processamento de texto

As técnicas de *stemmer*, lematização e remoção de *stopwords* são técnicas clássicas amplamente utilizadas quando a natureza do problema envolve texto. Recentemente com a disseminação de métodos baseados em redes neurais artificiais o uso de *Word Embeddings* também ganhou destaque quando o natureza do problema envolve texto.

Para um desavisado as técnicas de *stemmer* e lematização podem ser consideradas equivalentes. Assim como o ato de remover ou não *stopwords* pode ser considerado irrelevante. Quem sabe até mesmo desconsiderar ou utilizar cegamente *Word embeddings* para todo problema envolve o uso rede neurais artificiais. No entanto, trabalhos que estudam o impacto do uso dessas técnicas relatam que não existe uma conclusão universal sobre o impacto de seu uso. Na verdade a conclusão é de que o resultado

pode variar dependendo do problema a ser resolvido.

Em (HARMAN, 1991) foi realizado um estudo comparativo baseado em diferentes algoritmos de *stemmer* para o problema de recuperação de informação com documentos em inglês, e verificou-se a ineficácia do uso de *stemmer*. Nos resultados dos experimentos os autores concluíram que o uso da técnica piora a precisão dos mecanismos de busca. No entanto, em (POPOVIČ; WILLETT, 1992) os autores também avaliaram o impacto do uso de *stemmer* no problema de recuperação de informação e observaram que esta técnica pode ser eficaz na precisão em especial para textos curtos.

Em (KORENIUS et al., 2004) foi conduzido um experimento para avaliar se a lematização pode melhorar o desempenho de algoritmos de agrupamento de documentos em finlandês. Diante do desempenho positivo da técnica de *stemmer* em outros trabalhos, os autores decidiram também acrescentar o uso do *stemmer* no experimentos. Nesse experimento, 5000 textos em finlandês foram selecionados e 4 algoritmos hierárquicos de agrupamento foram empregados. Nos resultados observou-se a superioridade da lematização em comparação ao *stemmer*, quando aplicado ao problema de agrupamento de documentos em finlandês.

De forma semelhante, em (TOMAN; TESAR; JEZEK, 2006) também foram conduzidos experimentos com *stemmer* e lematização, nesse caso para o problema de classificação de documentos nos idiomas inglês e tcheco. Nos experimentos os autores chegaram a conclusão de que as técnicas de *stemmer* e lematização impactam de forma insignificante para o problema abordado. Um resultado diferente foi obtido em (HAN et al., 2012), nesse trabalho os autores novamente conduziram experimentos com as técnicas de *stemmer* e lematização, dessa vez para o problema de agrupamento de documentos. A conclusão foi de que ambas as técnicas geram melhorias significativas para o problema de agrupamento de documentos. Além disso, acrescentaram que deve-se adotar uma das técnicas de acordo com o campo de aplicação. Quando há uma necessidade por eficiência na compressão da quantidade de termos utilizados no agrupamento de documentos, o *stemmer* é uma opção. Porém quando houver a necessidade de precisão a lematização pode ser uma escolha ainda melhor.

Assim como o *stemmer* e a lematização, a remoção de *stopwords* é outra técnica

de pré-processamento de textos que também apresenta diferentes resultados de acordo com o tipo de problema aplicado. Em (ZOU et al., 2006) os autores avaliaram o impacto de usar ou não *stopwords* em mecanismos de busca com documentos em chinês, e a conclusão foi de que a remoção de *stopwords* tem impacto positivo e melhora o desempenho dos mecanismos de busca. Em trabalhos como (EL-KHAIR, 2017; AZMI; AL-THANYAN, 2009) os autores também evidenciam impacto positivo do uso de *stopwords* para problemas como sumarização de texto e recuperação de informação, dessa vez para o idioma árabe. Ainda sobre a recuperação de informação, a remoção de *stopwords* também pode melhorar o desempenho de mecanismos de busca em domínios com múltiplos idiomas, como bengali-hindi, inglês-turco, japonês-inglês, segundo os trabalhos de (MANDAL et al., 2007; CELEBI; SEN; GUNEL, 2009; LI; SHAWE-TAYLOR, 2006).

No entanto, também existem trabalhos que apresentam conclusões negativas sobre o uso da técnica de remoção de *stopwords*. Em (BAKLIWAL et al., 2012) foi proposto um sistema para análise de sentimentos no Twitter por meio de uma função de pontuação. O estudo utilizou o conteúdo de *tweets* antigos para classificar e ponderar os sentimentos. Nos experimentos foi relatado que a remoção *stopwords* contribuiu negativamente na classificação de sentimentos e que também adicionou ambiguidade na formação do modelo. Em (CHIN; ZAPPONE; ZHAO, 2016) também foi realizado um estudo envolvendo a análise de sentimentos no Twitter. Desse vez, baseado em dados da eleição presidencial dos Estados Unidos referente ao ano 2016. Uma parte da conclusão foi destinada a relatar que a remoção de *stopwords* não obteve impacto significativo na precisão do modelo de classificação utilizado.

Os resultados anteriores mostram a importância de se estudar o impacto de operações sobre texto em cada novo contexto onde as mesmas possam ser empregadas. Tal observação é a principal motivação para o estudo realizado nesta dissertação.

---

## Experimentos

Neste capítulo é apresentada uma avaliação experimental do impacto da aplicação de técnicas de pré-processamento de textos a tarefas de detecção de intenção e extração de parâmetros. As técnicas de pré-processamento escolhidas foram radicalização (stemming), remoção de palavras irrelevantes (stopwords), lematização e representação de palavras por meio de *Word Embeddings* pré-treinados com os métodos Glove e FastText. Para fins de análise e comparação do impacto causado pelas técnicas de pré-processamento, foram escolhidos alguns dos principais trabalhos publicados nos últimos 5 anos, adotando-se como critérios de escolha o fato de terem resultados próximos ao estado-da-arte e de terem seu código fonte disponibilizado pelo autor, o que evita questionamentos quanto a possíveis ruídos introduzidos pela implementação dos métodos. Ao todo foram selecionados 4 trabalhos de referência para a realização dos experimentos. Os métodos foram testados em diferentes bases de dados e foram avaliados com e sem o uso das técnicas de pré-processamento de texto aqui estudadas.

O restante do capítulo é estruturado da seguinte forma. Na Seção 4.1 são descritas as configurações dos experimentos, incluindo a descrição de cada base de dados utilizada, o ambiente onde foram executados os testes, as métricas utilizadas nos experimentos e metodologia utilizada na experimentação.

Na Seção 4.1.6 são apresentados os resultados obtidos nos experimentos. Esses resultados são apresentados na sequência em que foram descritos na metodologia, apresentada na Seção 4.1.5. Cada subseção da seção de resultados retrata os resultados obtidos para cada técnica de pré-processamento de texto avaliada.

## 4.1 Configuração dos experimentos

### 4.1.1 Bases de dados utilizada

Os experimentos foram conduzidos em três bases de dados diferentes, escritas nos idiomas Inglês e Espanhol. Cada base de dados é composta por exemplos de entrada separados em treino, teste e validação. Cada exemplo de entrada é representado por frases com palavras separadas por espaços. Cada palavra contém um rótulo correspondente que indica se uma ou mais palavras representam parâmetros a serem extraídos. Além disso cada frase contém um rótulo adicional que corresponde à intenção ou objetivo da frase. A Tabela 2 sumariza as principais características de cada uma das bases de dados. Nas bases de dados ATIS e SNIPS utilizamos a mesma subdivisão de treino, teste e validação utilizada no trabalho de (GOO et al., 2018). Já no FMD foi utilizada a subdivisão descrita no artigo original de apresentação da coleção (SCHUSTER et al., 2018).

Base	Treino	Teste	Validação	Intenções	Parâmetros	Vocabulário
ATIS	4,778	500	893	21	120	722
SNIPS	13,084	700	700	7	72	11,241
FMD	30,521	4,181	8,621	12	11	3983
FMD (Espanhol)	3,617	1,983	3,043	12	11	1849

Tabela 2 – Principais características das bases de dados utilizadas

A primeira base de dados utilizada nos experimentos foi a *Airline Travel Information System* (ATIS) (HEMPHILL; GODFREY; DODDINGTON, 1990), criada na década de 90 por meio de gravações de áudio das vozes de pessoas que trabalhavam realizando reservas de voos. É uma base de dados pequena, com apenas 4778 mil exemplos de treino, 500 exemplos teste e 893 exemplos de validação. Ao todo são apenas 21 intenções a serem detectadas e 120 parâmetros a serem extraídos. O vocabulário dessa base de dados é composto por apenas 722 palavras. Apesar de ser pequena, é uma base de dados de referência constantemente utilizada em trabalhos que abordam as tarefas de detecção de intenção e extração de parâmetros (JEONG; LEE, 2008; HAKKANI-TÜR et al., 2016; LIU; LANE, 2016; GOO et al., 2018).

A segunda base de dados utilizada foi a SNIPS (COUCKE et al., 2018), criada a

partir de uma coleção de frases cadastradas na plataforma *Snips Voice Platform*<sup>1</sup>, uma plataforma voltada para desenvolvimento de assistentes de voz. Foi publicada no ano de 2018 e apresenta 13,084 frases de treino, 700 de teste e 700 de validação, totalizando um número 3 vezes maior que a ATIS. Essa base de dados foi projetada para ter poucas intenções a serem detectadas e parâmetros a serem extraídos. Ao todo possui são 7 intenções a serem detectadas e 12 parâmetros a serem extraídos.

A terceira base de dados utilizada foi a Facebook Multilingual Dataset (FMD) (SCHUSTER et al., 2018), a qual dividimos em duas bases de dados, uma em Inglês (FMD) e outra em Espanhol (FMD Espanhol). Publicada no ano de 2018, essa base de dados é muito maior que os seus antecessores ATIS e SNIPS. Além de conter menos intenções que o ATIS e menos parâmetros a serem extraídos que o SNIPS. O que garante mais exemplos de treino, teste e validação por classes a ser aprendida.

#### 4.1.2 Trabalhos comparados

Além das três bases de dados apresentadas anteriormente, para maximizar a confiança nos resultados obtidos ao avaliar o impacto do uso de técnicas de pré-processamento de texto, foram escolhidos os principais trabalhos publicados nos últimos 5 anos que resolvem as tarefas de detecção de intenção e extração de parâmetros de forma conjunta. Esses trabalhos foram descritos com mais detalhes no capítulo anterior, são trabalhos com resultados próximos ao estado-da-arte e o código foi disponibilizado por um dos autores:

- **JointSLU:** (HAKKANI-TÜR et al., 2016) uma das primeiras arquiteturas neurais baseadas em RNNs capaz de resolver as tarefas de detecção de intenção e extração de parâmetros<sup>2</sup>.
- **RNN-NLU:** (LIU; LANE, 2016) uma arquitetura neural baseada em duas camadas de RNNs no formato codificador-decodificador. A primeira camada recebe o texto do usuário, enquanto a segunda camada atribui um rótulo para cada palavra da

<sup>1</sup> <https://snips.ai/>

<sup>2</sup> <https://github.com/yvchen/JointSLU>

camada de entrada. Existem pesos que ligam as duas camadas, e estes pesos são utilizados para classificar a intenção do usuário. Para melhorar os resultados, foi utilizada uma técnica conhecida como mecanismos de atenção<sup>3</sup>.

- **Slot-Gated:** (GOO et al., 2018) uma arquitetura neural também baseada em RNNs e mecanismos de atenção. O diferencial desse trabalho é a introdução do conceito de mecanismos *slot-gated*, uma função que torna o modelo mais flexível e possibilita escolher a tarefa para qual a rede neural deve direcionar a sua atenção<sup>4</sup>.
- **SF-ID Network:** (HAIHONG et al., 2019) uma arquitetura neural composta de duas sub-redes baseadas em RNNs e mecanismos de atenção que estabelecem uma conexão entre as tarefas de extração de parâmetros e detecção de intenção por meio do vetor do contexto aprendido por cada uma das sub-redes<sup>5</sup>.

### 4.1.3 Ambiente de execução

Todos os experimentos foram executados no Google Colab<sup>6</sup>. O Google Colab é um serviço de nuvem gratuito oferecido pelo Google, como uma forma de incentivo à pesquisa de Aprendizagem de Máquina e Inteligência Artificial. Todos os experimentos foram codificados na linguagem de programação Python nas versões 2.7 e 3.6.

### 4.1.4 Métricas de avaliação

O impacto das técnicas de pré-processamento de texto utilizado nas tarefas de detecção de intenção e extração de parâmetros foi avaliado por meio das métricas acurácia, precisão, revocação e *f-score*. Normalmente a tarefa de detecção de intenção é avaliada apenas com a métrica acurácia, como feito em trabalhos como (HAKKANI-TÜR et al., 2016; LIU; LANE, 2016). Já para a tarefa de extração de parâmetros é normalmente avaliada apenas com a métrica *f-score* (GOO et al., 2018; HAIHONG et al., 2019).

<sup>3</sup> <https://github.com/HadoopIt/rnn-nlu>

<sup>4</sup> <https://github.com/MiuLab/SlotGated-SLU>

<sup>5</sup> <https://github.com/ZephyrChenzf/SF-ID-Network-For-NLU>

<sup>6</sup> <https://colab.research.google.com>

Neste trabalho optamos por adicionar as métricas precisão e revocação ao conjunto de métricas utilizadas. Essa decisão foi tomada devido ao estudo das técnicas de pré-processamento stemmer e lematização, que geralmente apresentam impactos opostos sobre as métricas precisão e revocação. A técnica de stemmer é conhecida por melhorar a revocação em sacrifício da precisão. Já a técnica de lematização normalmente resulta no ganho de precisão, mas às custas de uma pequena perda na revocação.

O cálculo de todas as métricas utilizadas é definido pelo resultado da matriz de confusão. Uma matriz de confusão contém informações sobre classificações reais e as previstas feitas por um modelo de classificação. A Tabela 3 exemplifica uma matriz de confusão.

		Predição	
		Classe A	Classe B
Real	Classe A	VP	VN
	Classe B	FP	FN

Tabela 3 – Ilustração de uma matriz de confusão

Conceitos relacionados à matriz de confusão são:

- **Verdadeiro Positivo (VP):** é a quantidade de exemplos positivos preditos corretamente. Um exemplo é dito como VP quando a predição do modelo é exatamente a classe que estamos buscando. Por exemplo, quando a intenção do usuário é ligar uma lâmpada e o modelo previu corretamente que ele queria ligar tal lâmpada;
- **Falso Positivo (FP):** é a quantidade de exemplos positivos preditos erroneamente. Um exemplo é dito como FP quando a predição do modelo prevê de forma incorreta a classe que estamos buscando. Por exemplo, quando a intenção do usuário é ligar uma lâmpada e o modelo previu erroneamente que ele queria desligar a tal lâmpada;
- **Verdadeiro Negativo (VN):** é a quantidade de exemplos negativos preditos corretamente. Um exemplo é dito como VN quando a predição de uma classe que não estamos buscando prever foi prevista corretamente. Por exemplo, a intenção é de desligar a lâmpada e o modelo previu corretamente que a intenção não é de desligar a lâmpada;

- **Falso Negativo (FN):** é a quantidade de exemplos negativos preditos erroneamente. Um exemplo é dito como FN quando a predição de uma classe que não estamos buscando prever foi prevista de forma incorreta. Por exemplo, a intenção é de ligar a lâmpada e o modelo previu incorretamente que a intenção não é de desligar a lâmpada.

#### 4.1.4.1 Acurácia

A métrica acurácia é a porcentagem de acertos para todas as classes. Representada pela razão entre o somatório das previsões corretas (verdadeiros positivos e verdadeiros negativos) sobre o somatório das previsões. Essa métrica diz quanto o modelo analisado acertou das previsões possíveis. O cálculo é feito da seguinte forma:

$$\text{acurácia} = \frac{VP + VN}{VP + FN + VN + FP} \quad (4.1)$$

#### 4.1.4.2 Revocação

A métrica revocação indica a frequência de relevância dos resultados preditos. Representada pela razão entre os verdadeiros positivos sobre a soma de verdadeiros positivos com negativos falsos. Essa métrica diz a proporção de positivos que foi identificada corretamente, ou seja, indica o quão bom o modelo é para prever casos positivos de uma classe alvo. O cálculo é feito da seguinte forma:

$$\text{revocação} = \frac{VP}{VP + FN} \quad (4.2)$$

#### 4.1.4.3 Precisão

A métrica precisão indica a corretude por classe dos resultados preditos. Representada pela razão entre os verdadeiros positivos sobre a soma de verdadeiros positivos com falsos positivos. Essa métrica diz o quão bem o modelo trabalhou. O cálculo é feito

da seguinte forma:

$$\text{precisão} = \frac{VP}{VP + FP} \quad (4.3)$$

#### 4.1.4.4 F-score

A métrica f-score é a média harmônica ponderada entre precisão e revocação. Essa métrica diz o balanço entre a precisão e a revocação do modelo analisado. O cálculo é feito da seguinte forma:

$$\text{f-score} = 2 \times \frac{\text{precisão} \times \text{revocação}}{\text{precisão} + \text{revocação}} \quad (4.4)$$

### 4.1.5 Metodologia

O processo de experimentação utilizado nessa dissertação é dividido em duas etapas. A primeira etapa apresenta o processo de escolha das implementações utilizadas por cada uma das técnicas de pré-processamento abordadas nesse trabalho. Já na segunda etapa é descrito o processo de ajustes nos códigos fonte dos trabalhos comparados, dentre os ajustes realizados então inclusos o suporte a *Word Embeddings*, *stemming*, *lematização*, remoção de *stopwords* e a implementação das métricas precisão, revocação, acurácia e f-score.

#### 4.1.5.1 Processo de escolha da melhor implementação para cada uma das técnicas de pré-processamento utilizadas

Neste trabalho optamos por avaliar o impacto do uso das técnicas de stemming, remoção de stopwords, lematização e word embeddings pré-treinados no formato Glove e FastText. No entanto, cada uma dessas técnicas possui variações e diferentes formas de implementação. A avaliação de todas as possíveis variações de implementação para

cada técnica resultaria em uma quantidade enorme de possíveis combinações. Diante desse problema, antes de aplicar as técnicas de pré-processamento de texto diretamente nos trabalhos comparados, optamos por selecionar as melhores implementações para cada uma das técnicas utilizadas.

Para a técnica de radicalização (*stemmer*) foram comparadas as implementações conhecidas como *Porter stemmer*, *Snowball Stemmer* e *Lancaster Stemmer*. Todas as implementações estão disponíveis na biblioteca de ferramentas de linguagem natural conhecida como *Natural Language Toolkit* (NLTK). O critério de escolha utilizado para selecionar o melhor algoritmo de stemmer foi semelhante ao critério de escolha apresentado em (TOMAN; TESAR; JEZEK, 2006). Nesse trabalho, o autor avalia a qualidade do algoritmo de stemmer baseado no quanto ele consegue comprimir o vocabulário de uma base de dados. Na seleção da melhor implementação o algoritmo, *Porter Stemmer* foi o que apresentou a maior taxa de compressão de palavras.

Para a técnica de lematização foram comparadas as implementações disponíveis nas bibliotecas *StanfordNLP*, *spaCy* e *NLTK*. O critério de escolha utilizado para selecionar o melhor algoritmo de lematização foi o mesmo utilizado na seleção do stemming. Em outras palavras, a implementação selecionada foi a biblioteca que apresentou a maior taxa de compressão após ser aplicado nas bases de dados ATIS, SNIPS e FMD. Sendo a nossa avaliação a implementação da técnica de lematização disponibilizada pela biblioteca *spaCy* e a melhor quando comparada aos demais.

Para a técnica de remoção de stopwords, não houve a escolha do melhor algoritmo. Escolhemos a lista de stopwords disponível na biblioteca *NLTK*. Já na técnica de *Word Embeddings* com vetores pré-treinados, foram escolhidos vetores no formato *Glove* e *FastText* nas dimensões 50, 100 e 200.

#### 4.1.5.2 Processo de adaptação dos trabalhos comparados para suportar as técnicas de pré-processamento

Para avaliar o impacto do uso das técnicas de pré-processamento foram escolhidos 4 trabalhos já publicados e com resultados próximos ao estado-da-arte. Cada um

dos trabalhos comparados foi submetido a uma série de experimentos no qual os dados de entrada passaram pelo processo de radicalização (stemmer), lematização, remoção de stopwords e tiveram suas palavras codificadas em vetores densos de tamanho fixo por meio da técnica de *Word Embeddings*. Os dados de entrada para os trabalhos comparados foram retirados de três bases de dados comumente usadas em trabalhos de detecção de intenção e extração de parâmetros.

No entanto, nenhum dos trabalhos comparados estava preparado para receber as técnicas de pré-processamento. Todos eles foram projetados para receber como entrada apenas palavras separadas por espaços. Para os experimentos envolvendo as técnicas de stemmer e lematização, o processo de adaptação dos trabalhos comparados consistiu em previamente aplicar a cada uma das palavras da base de dados as técnicas de stemmer e lematização. O processo resultou em duas novas palavras: a primeira reduzida ao seu radical (*stem*) e a segunda reduzida a sua forma canônica (*lema*). Então, os *stems* e os *lemas* foram submetidos como dados de entrada para cada um dos trabalhos comparados em seus respectivos experimentos.

Nos experimentos envolvendo a técnica de remoção de *stopwords* também foi realizado um pré-processamento na base de dados. Nesse pré-processamento foram removidas todas as ocorrências de palavras pertencentes à lista de stopwords. Isto é, caso a palavra não fosse encontrada na lista de *stopwords* ela permaneceria na base de dados. Esse processo foi aplicado ao conjunto de treino, teste e validação. Por fim, o conjunto de palavras que não foi removido após a técnica de remoção de stopwords foi submetido como dado de entrada para cada um dos trabalhos comparados em seus respectivos experimentos.

Nos experimentos envolvendo o uso da técnica de *Word Embeddings* o processo de adaptação dos trabalhos comparados foi realizado diretamente na implementação dos trabalhos comparados. Para cada trabalho comparado foi necessário entender o processo de codificação das palavras. A codificação utilizada nas implementações originais de cada um dos trabalhos comparados é conhecida como *one-hot-encoding*. Nessa codificação cada palavra é mapeada para um único vetor composto de zeros e apenas um número 1, cada posição assinalada com o número 1 representa uma palavra do voca-

bulário. Para garantir que cada palavra tenha apenas uma única posição assinalada com o valor 1 no vetor, o comprimento do vetor de *one-hot-encoding* deve ser a quantidade de palavras do vocabulário. Ao aplicar a técnica de *Word Embeddings* na implementação dos trabalhos comparados, os vetores de zeros e uns foram substituídos por vetores densos de tamanho fixo repletos de números em ponto flutuante. A dificuldade em realizar esse processo foi entender a implementação de cada um dos trabalhos comparados.

#### 4.1.5.3 Processo de adaptação dos trabalhos comparados para suportar novas métricas de avaliação

No código fonte das implementações dos trabalhos comparados foram implementadas apenas as métricas acurácia e f-score. Normalmente a tarefa de detecção de intenção é avaliada apenas com a métrica acurácia, enquanto que a tarefa de extração de parâmetros é avaliada com a métrica f-score (HAKKANI-TÜR et al., 2016; LIU; LANE, 2016; GOO et al., 2018). Para enriquecer a avaliação do impacto do uso de técnicas de pré-processamento foram incluídas as métricas de precisão e revocação.

Em capítulos anteriores, a tarefa de detecção de intenção foi apresentada como um problema de classificação de texto. Para avaliar esta tarefa foi utilizada a biblioteca *Sklearn*<sup>7</sup>, por conter a implementação das principais métricas necessárias para avaliar um classificador de texto. Já na tarefa de extração de parâmetros foi utilizado o *script* conhecido como *conlleva*<sup>8</sup> um trecho de código bastante utilizado em problemas de rotulagem de sequências (TUR; MORI, 2011).

#### 4.1.6 Resultados

Nesta seção são apresentados os resultados obtidos após os experimentos realizados nas três bases de dados apresentadas na Seção 4.1.1. São reportados os resultados obtidos após o uso de cada uma das técnicas de pré-processamento de texto analisadas ao decorrer desta dissertação. Para garantir uma confiança estatística nos resultados

<sup>7</sup> <https://scikit-learn.org/>

<sup>8</sup> <https://www.clips.uantwerpen.be/conll2000/chunking/output.html>

apresentados, unimos os grupos de treino e validação de cada base de dados utilizada, e aplicamos a técnica de validação cruzada K-Fold com o  $k$  igual a 10 para gerar novos grupos treino e validação. O K-Fold é uma técnica que divide os dados em  $k$  grupos, e um dos grupos é escolhido de maneira aleatória para ser o grupo de validação. O restante dos grupos se juntam, formam o grupo de treino e são utilizados na etapa de treinamento do modelo. Após o treinamento, o modelo é gerado e avaliado nos dados do grupo de validação. Esse método se repete até que todos os grupos tenham passado no grupo de validação e todos tenham suas métricas calculadas. Por esse motivo, todos os resultados reportados neste capítulo correspondem à média aritmética das  $k$  validações experimentadas.

Para fins de comparação a Tabela 4 apresenta os resultados obtidos por cada um dos trabalhos comparados sem o uso de técnicas de pré-processamento. O objetivo dessa tabela é dar uma visão geral e servir como referência para o desempenho dos trabalhos comparados antes do uso das técnicas de pré-processamento. É importante ressaltar que ao observar as métricas de acurácia e  $f$ -score é possível observar que a tarefa de detecção de intenção apresenta valores superiores aos obtidos na tarefa de extração de parâmetros. O mesmo acontece em trabalhos como (LIU; LANE, 2016; HAKKANI-TÜR et al., 2016).

Tabela 4 – Resultado das experimentações com trabalhos comparados sem o uso de técnicas de processamento de texto nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol)

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	97.42	93.50	93.73	93.61	92.61	90.18	92.61	90.87
SF-ID-Network	97.81	95.67	95.06	95.36	96.98	96.72	96.98	96.72
Slot-Gated	97.92	95.21	95.45	95.33	95.74	95.46	95.74	95.33
RNN-NLU	97.45	92.60	92.25	92.42	95.74	95.27	95.74	95.27

(a) Experimentações na base de dados ATIS

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	94.21	84.22	88.88	86.49	97.86	97.92	97.86	97.86
SF-ID-Network	95.12	91.49	92.46	91.97	97.00	96.99	97.00	96.98
Slot-Gated	94.22	86.35	90.45	88.35	96.71	96.74	96.71	96.71
RNN-NLU	93.51	77.70	85.64	81.48	97.14	97.19	97.14	97.13

(b) Experimentações na base de dados SNIPS

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	94.41	93.24	93.76	93.50	98.77	98.77	98.77	98.77
SF-ID-Network	97.58	95.74	95.54	95.64	99.14	99.14	99.14	99.14
Slot-Gated	97.68	94.77	95.30	95.03	98.99	98.98	98.99	98.98
RNN-NLU	97.26	91.14	93.08	92.10	98.57	98.59	98.57	98.50

(c) Experimentações na base de dados FMD

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	91.64	75.77	81.84	78.68	97.21	97.19	97.21	97.14
SF-ID-Network	93.16	86.01	85.08	85.54	97.54	97.62	97.54	97.57
Slot-Gated	93.20	83.15	85.33	84.23	97.01	97.06	97.01	97.02
RNN-NLU	94.74	82.92	85.33	84.11	97.77	97.74	97.77	97.74

(d) Experimentações na base de dados FMD(Espanhol)

#### 4.1.6.1 Impacto do uso de stemmer nas tarefas de detecção de intenção e extração de parâmetros

Na Tabela 5, são apresentados os resultados obtidos após aplicar a técnica de stemmer nos trabalhos comparados. Esta tabela é composta por quatro partes, onde cada parte exibe os resultados obtidos nas tarefas de detecção de intenção e extração de parâmetros separadas por base de dados utilizada. De modo geral, ao analisar os resultados obtidos pela tarefa de extração de parâmetros tanto na bases de dados ATIS, FMD e FMD(espanhol) é possível observar um ganho de até 3% na métrica de revocação. Em contrapartida ao ganho obtido na revocação foi observado uma perda de até 0.9%

na métrica de precisão, para a mesma tarefa. Já na tarefa de detecção de intenção foi observado que em todos os casos onde ocorre uma melhora, ainda que pequena na revocação também existe ganho na tarefa de detecção de intenção. No entanto, este ganho foi obtido em todas as métricas utilizadas. Para tentar entender o impacto positivo obtido na métrica de revocação, para cada um dos trabalhos comparados foram analisados as intenções detectadas e os parâmetros extraídos por base de dados utilizada. Os exemplos observados resultam de casos onde uma ou mais palavras foram modificadas pelo processo de radicalização (*stemmer*). Nesses casos foram observadas situações onde uma ou mais palavras que haviam sido inicialmente preditas de forma incorreta, até o uso do *stemmer*, passaram a ser preditas de forma correta. Em nosso entendimento, isso ocorre porque ao serem reduzidas a radical comum, a ambiguidade introduzida pela flexão das palavras foi removida e a incerteza do modelo para esses casos deixou de existir. Até o momento o ganho obtido na revocação é justificado pelo processo de redução de palavras e suas flexões a um radical comum. Por outro lado, essa mesma justifica não explica o ganho obtido em todas as métricas da tarefa de detecção de intenção. Deste modo, o nosso entendimento atual sobre o problema estudado nos leva a crer que a tarefa de extração de parâmetros tem influência direta sobre o ganho obtido na tarefa de detecção de intenção.

Tabela 5 – Resultado das experimentações com trabalhos comparados após utilizar a técnica de stemmer nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol)

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	-0.01	-0.89	+2.23	+1.89	+3.78	+2.91	+3.78	+3.78
SF-ID-Network	+0.15	-0.01	+0.75	+0.43	+0.44	+0.38	+0.44	+0.45
Slot-Gated	-0.12	-0.63	+0.93	-0.25	+0.22	-1.14	+0.22	+0.06
RNN-NLU	-0.03	-0.34	+0.10	-0.23	+0.23	+0.72	+0.23	+0.15

(a) Experimentações na base de dados ATIS

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	+0.41	-0.12	+3.01	+2.41	+1.12	+0.48	+0.43	+0.42
SF-ID-Network	+0.23	-0.25	+0.34	+0.10	0.00	+0.03	0.00	+0.01
Slot-Gated	-0.09	-0.01	+0.61	-0.19	+1.15	+1.13	+1.15	+1.14
RNN-NLU	-0.47	-0.62	+0.72	-0.93	+0.15	+0.12	+0.15	+0.16

(b) Experimentações na base de dados SNIPS

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	+2.76	+0.01	+2.13	+1.07	+0.10	+0.13	+0.10	+0.10
SF-ID-Network	-0.04	-0.08	+0.30	-0.14	+0.04	+0.03	+0.04	+0.03
Slot-Gated	-0.03	-0.12	+0.35	+0.73	-0.17	-0.17	-0.17	-0.17
RNN-NLU	+0.08	+0.25	+0.01	+0.13	+0.08	+0.07	+0.08	+0.14

(c) Experimentações na base de dados FMD

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	+0.74	+1.81	+0.80	+1.35	+1.46	+1.44	+1.46	+1.49
SF-ID-Network	+0.58	+0.99	+1.06	+1.03	+0.49	+0.44	+0.49	+0.47
Slot-Gated	+0.08	-0.02	+0.24	-0.08	+0.26	+0.17	+0.26	+0.22
RNN-NLU	-0.36	0.00	+0.10	-0.05	+0.42	+0.39	+0.42	+0.42

(d) Experimentações na base de dados FMD(Espanhol)

#### 4.1.6.2 Impacto do uso de lematização nas tarefas de detecção de intenção e extração de parâmetros

Na Tabela 6 são apresentados os resultados obtidos após aplicar a técnica de lematização nos trabalhos comparados. Essa tabela também é composta por quatro partes, onde cada parte exibe os resultados obtidos nas tarefas estudadas separadas por base de dados utilizada. De modo geral, os resultados obtidos por meio do uso da técnica de lematização não foram satisfatórios. Ao analisar os resultados obtidos na tarefa de extração de parâmetros em todas as bases de dados, é possível observar uma perda significativa de até 2% na métrica de f-score. A métrica de f-score reflete a perda acumulada nas métricas de precisão e revocação. Os resultados obtidos na tarefa de

detecção de intenção também foram ruins, reforçando o nosso entendimento sobre a importância da tarefa de extração de parâmetros; por consequência dos resultados ruins na primeira tarefa de extração de parâmetros a tarefa de detecção de intenção também obteve resultados ruins. Para tentar entender o impacto negativo obtido após o uso da técnica de lematização também analisamos as intenções preditas e os parâmetros extraídos para cada trabalho comparado. Os exemplos observados derivam de casos onde uma ou mais palavras foram modificadas pelo processo de lematização. Nesses casos, percebemos que o impacto negativo da técnica advém de situações onde uma ou mais palavras que haviam sido inicialmente preditas de forma correta, deixaram de ser após o uso da lematização. Isso ocorre porque o processo de lematização nem sempre reduz uma palavra para a mesma forma comum (*lemma*). Em suma, os exemplos observados mostram que esse comportamento insere ambiguidade e incerteza no modelo, e essa incerteza resulta em erros.

Tabela 6 – Resultado das experimentações com trabalhos comparados após utilizar a técnica de lematização nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol)

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	-1.53	0.00	-2.10	-2.54	-1.79	-2.87	-1.79	-2.28
SF-ID-Network	-0.34	-0.24	-0.74	-0.49	-0.45	-0.60	-0.45	-0.63
Slot-Gated	-0.50	-0.16	-1.19	-0.68	-1.00	-1.46	-1.00	-1.29
RNN-NLU	-0.57	+0.09	-1.44	-0.68	0.00	-0.05	0.00	-0.07

(a) Experimentações na base de dados ATIS

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	-0.98	-2.07	-5.56	-1.83	-0.86	-0.86	-0.86	-0.86
SF-ID-Network	-1.04	-2.70	-3.10	-2.90	+0.14	+0.16	+0.14	+0.16
Slot-Gated	-0.55	-0.67	-1.68	-1.15	0.00	+0.09	0.00	+0.02
RNN-NLU	-2.24	-5.76	-8.21	-6.89	-0.28	-0.25	-0.28	-0.26

(b) Experimentações na base de dados SNIPS

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	+2.53	-0.32	-0.61	-0.47	+0.41	+0.41	+0.41	+0.39
SF-ID-Network	-0.20	-0.42	-1.02	-0.72	0.00	0.00	0.00	0.00
Slot-Gated	-0.14	-0.25	-0.73	-0.48	-0.08	-0.07	-0.08	-0.07
RNN-NLU	-0.49	-1.29	-2.77	-2.02	+0.20	-1.83	-1.80	-0.93

(c) Experimentações na base de dados FMD

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	+0.74	+0.58	+0.24	+0.52	+0.52	+0.52	+0.52	+0.56
SF-ID-Network	+0.34	-1.51	-2.54	-2.03	+0.42	+0.41	+0.42	+0.42
Slot-Gated	-0.94	-5.04	-4.86	-4.96	+0.16	+0.23	+0.16	+0.18
RNN-NLU	-1.05	-5.57	-5.95	-5.76	-1.71	-1.65	-1.71	-1.68

(d) Experimentações na base de dados FMD(Espanhol)

#### 4.1.6.3 Impacto do uso de remoção de stopwords nas tarefas de detecção de intenção e extração de parâmetros

Na Tabela 7 são apresentados os resultados obtidos após aplicar a técnica de remoção de stopwords nos trabalhos comparados. Essa tabela também é composta por quatro partes, onde cada parte exibe os resultados obtidos nas tarefas de detecção de intenção e extração de parâmetros separadas por base de dados utilizada. De modo geral, essa técnica de pré-processamento também resultou em impactos negativos nas tarefas estudadas. Em alguns casos é possível observar uma perda de até 6% na tarefa de extração de parâmetros por meio da métrica de acurácia. Em suma, acreditamos que as preposições e artigos que foram removidos eram importantes para contextualiza-

ção e localização dos parâmetros a serem extraídos. No entanto por ser uma técnica extremamente simples não encontramos uma forma de analisar com mais exatidão o impacto negativo obtidos após o uso da técnica de remoção de *stopwords*.

Tabela 7 – Resultado das experimentações com trabalhos comparados após utilizar a técnica remoção de stopwords nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol)

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	-2.36	-1.18	-1.64	-1.40	-1.57	-0.47	-1.57	-1.64
SF-ID-Network	-1.32	-0.68	-0.42	-0.55	-2.02	-2.91	-2.02	-2.51
Slot-Gated	-1.72	-1.35	-1.25	-1.30	-0.67	-1.87	-0.67	-1.27
RNN-NLU	-0.91	-0.08	+0.60	+0.27	-1.90	-2.69	-1.90	-2.48

(a) Experimentações na base de dados ATIS

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	-6.03	-2.30	-3.08	-2.67	-1.00	-0.98	-1.00	-1.03
SF-ID-Network	-5.84	-4.49	-4.18	-4.33	-0.43	-0.42	-0.43	-0.42
Slot-Gated	-4.63	-2.32	-2.34	-2.33	-1.14	-1.17	-1.14	-1.16
RNN-NLU	-3.27	+0.83	-1.44	-0.21	-0.57	-0.56	-0.57	-0.56

(b) Experimentações na base de dados SNIPS

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	+0.85	-0.91	-0.16	-0.04	-0.12	-0.08	-0.12	-0.11
SF-ID-Network	-2.32	-0.38	-1.00	-0.69	-0.44	-0.44	-0.44	-0.44
Slot-Gated	-2.17	+0.25	-0.13	+0.06	-0.53	-0.52	-0.53	-0.52
RNN-NLU	-1.22	+1.30	+0.72	+1.01	-0.48	-0.51	-0.48	-0.44

(c) Experimentações na base de dados FMD

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU	+0.39	+3.01	+0.25	+1.72	+0.77	+0.75	+0.77	-87.21
SF-ID-Network	-1.62	-1.30	-1.26	-2.28	-4.00	-10.91	-11.72	-11.31
Slot-Gated	-0.20	-2.45	-4.00	-3.22	+0.69	+0.70	+0.69	+0.69
RNN-NLU	-0.19	-1.46	-0.57	-1.03	-0.33	-0.34	-0.33	-0.33

(d) Experimentações na base de dados FMD(Espanhol)

#### 4.1.6.4 Impacto do uso de Word Embeddings nas tarefas de detecção de intenção e extração de parâmetros

Assim como as tabelas anteriores, na Tabela 8 também são apresentados os resultados obtidos após aplicar uma técnica pré-processamento de texto nos trabalhos comparados. A técnica utilizada nessa parte dos experimentos é a codificação de palavras por meio de *Word Embeddings* com vetores pré-treinados de tamanho fixo. Os

vetores utilizados são vetores pré-treinados no formato *GloVe* e *FastText*, nas dimensões 50, 100 e 200. Ao utilizar dois tipos de *Word Embeddings* tínhamos em mente a ideia de capturar características latentes das implementação de *GloVe* e *FastText*. Enquanto que o ato de variar as dimensões dos vetores de tamanho fixo, tinha como objetivo reduzir a complexidade do modelo e consequentemente reduzir o tempo de treinamento. Para simplificar a visualização dos resultados a Tabela 8 apresenta somente os resultados obtidos da experimentação com 100 dimensões. Em suma, todas as dimensões apresentaram resultados foram insatisfatórios e longe do esperado. A começar pelo *GloVe* que em um dos trabalhos comparados é possível observar uma perda de até 15% na métrica *f-score*, utilizando a extração de parâmetros como referencial. O *FastText* por suas vez apresenta o impacto negativo com valores ainda maiores.

Existem casos onde a perda de acurácia chega a mais de 20%. Esse resultado obtido no *FastText* o torna pior que o *GloVe*, isso porque, a métrica acurácia é foi a métrica com os maiores valores obtidos em nossos experimentos. Para tentar entender o impacto negativo obtido após o uso dessa técnica, comparamos a taxa de erro no conjunto de treino e validação, para ter certeza que os trabalhos comparados estavam aprendendo as tarefas estudadas. Observamos que as redes neurais utilizadas nos trabalhos comparados não foram capazes de aprender o conjunto de treino, mantendo uma taxa de erro acima dos 25%. A nossa conclusão para este caso é de que as configurações utilizadas nas redes neurais implementadas nos trabalhos comparados não favorecem o uso de *Word Embeddings*, e portanto após o uso da técnica, os modelo neurais não conseguiram aprender o problema estudado.

Tabela 8 – Resultado das experimentações com trabalhos comparados após utilizar a técnica de *Word Embeddings* nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol)

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU(GloVe)	-6.53	-8.20	-5.10	-5.54	-1.88	-3.86	-2.89	-3.38
JointSLU(FastText)	-10.90	-9.75	-10.41	-10.60	-10.20	-10.37	-10.33	-11.09
SF-ID-Network(GloVe)	-2.72	-3.37	-5.36	-4.38	-3.03	-4.64	-3.03	-4.19
SF-ID-Network(FastText)	-9.43	-10.77	-20.28	-15.84	-6.27	-8.94	-6.27	-8.45
Slot-Gated(GloVe)	-6.00	-6.20	-5.30	-5.13	-3.21	-4.14	-3.33	-4.06
Slot-Gated(FastText)	-8.24	-6.46	-8.32	-6.31	-14.75	-7.18	-8.55	-9.25
RNN-NLU(GloVe)	-4.55	-3.05	-1.08	-1.20	-2.79	-3.35	-2.83	-3.30
RNN-NLU(FastText)	-21.74	-20.19	-18.31	-17.13	-7.02	-6.82	-6.98	-5.95

(a) Experimentações na base de dados ATIS

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU(GloVe)	-8.97	-5.07	-10.56	-11.80	-0.86	-0.86	-0.86	-0.86
JointSLU(FastText)	-10.20	-6.00	-9.56	-9.00	-7.06	-8.00	-5.87	-7.80
SF-ID-Network(GloVe)	-13.58	-20.78	-29.33	-25.26	-3.71	-2.64	-3.71	-3.57
SF-ID-Network(FastText)	-25.90	-40.97	-57.26	-50.48	-4.29	-3.95	-4.29	-4.30
Slot-Gated(GloVe)	-4.70	-5.16	-2.94	-5.23	-3.54	-4.50	-5.36	-3.93
Slot-Gated(FastText)	-15.31	-16.41	-14.56	-15.36	-6.40	-6.98	-6.89	-6.98
RNN-NLU(GloVe)	-3.59	+4.49	+2.26	+3.65	-2.07	-5.15	-5.19	-4.34
RNN-NLU(FastText)	-14.60	-7.76	-9.75	-9.39	-7.13	-7.29	-9.18	-8.40

(b) Experimentações na base de dados SNIPS

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU(GloVe)	-4.27	-4.32	-6.61	-4.47	-1.59	-2.59	-1.59	-2.61
JointSLU(FastText)	-8.19	-9.84	-10.00	-10.38	-9.44	-10.77	-12.76	-11.00
SF-ID-Network(GloVe)	-6.57	-6.60	-20.87	-14.37	-2.18	-2.19	-2.18	-2.47
SF-ID-Network(FastText)	-16.09	-21.75	-49.91	-39.19	-7.68	-7.83	-7.68	-9.32
Slot-Gated(GloVe)	-14.29	-11.95	-12.83	-13.99	-9.08	-11.30	-9.11	-9.07
Slot-Gated(FastText)	-8.00	-6.02	-8.17	-6.01	-18.00	-10.70	-11.80	-12.90
RNN-NLU(GloVe)	-7.77	-8.22	-10.51	-11.05	-11.56	-11.58	-9.49	-9.48
RNN-NLU(FastText)	-7.58	-2.39	-5.95	-3.01	-17.58	-10.50	-11.56	-12.42

(c) Experimentações na base de dados FMD

Trabalhos	Extração de Parâmetros				Detecção de Intenção			
	Acurácia	Precisão	Revocação	F-score	Acurácia	Precisão	Revocação	F-score
JointSLU(GloVe)	-9.26	-6.42	-2.76	-6.48	+0.52	+0.52	+0.52	+0.56
JointSLU(FastText)	-11.00	-7.75	-9.80	-8.00	-4.88	-5.29	-6.65	-4.16
SF-ID-Network(GloVe)	-10.01	-10.03	-30.65	-22.12	-17.91	-17.85	-17.91	-20.90
SF-ID-Network(FastText)	-20.22	-29.58	-56.44	-47.55	-23.07	-23.28	-23.07	-28.54
Slot-Gated(GloVe)	-7.70	-6.80	-5.40	-6.01	-5.99	-4.98	-3.44	-6.63
Slot-Gated(FastText)	-13.00	-3.60	-5.90	-4.99	-9.32	-9.78	-9.02	-9.03
RNN-NLU(GloVe)	-5.24	-6.57	-5.40	-5.89	-7.75	-5.66	-4.20	-7.35
RNN-NLU(FastText)	-14.54	-3.37	-5.90	-4.86	-9.10	-12.36	-10.78	-7.76

(d) Experimentações na base de dados FMD(Espanhol)

---

## Conclusão e Trabalhos futuros

Neste trabalho foi estudado o problema de compreensão de linguagem natural aplicado a sistemas de diálogo orientados a tarefa. Foi proposta uma avaliação experimental com o objetivo de avaliar o impacto do uso de técnicas de pré-processamento de texto aplicado nas tarefas de detecção de intenção e extração de parâmetros, que compõem o problema estudado. Em suma, toda a pesquisa desenvolvida ao longo desta dissertação visava responder a seguinte hipótese de pesquisa. É possível obter ganhos significativos com a aplicação de técnicas de pré-processamento de texto no problema estudado? A resposta é sim, porém nem todas as técnicas de pré-processamento geram bons resultados.

Para chegar a essa conclusão foram selecionadas dentre as possíveis técnicas de pré-processamento de texto existentes na literatura, as técnicas de *stemmer*, lematização, remoção de *stopwords* e uso de *Word Embeddings* com vetores pré-treinados. Para avaliar e comparar o impacto do uso de cada técnica escolhida foram realizados experimentos com trabalhos comparados. Os trabalhos comparados são trabalhos já publicados na literatura e com o código fonte disponibilizado pelo autor. Cada trabalho comparado foi treinado e avaliado nas bases de dados de referência ATIS, SNIPS, FMD e FMD(espanhol). Os trabalhos comparados seguem uma abordagem chamada de modelagem conjunta, que utiliza um único modelo para aprender as tarefas de detecção de intenção e de extração de parâmetros de modo unificado. Cada trabalho comparado foi avaliado com e sem o uso de técnicas de pré-processamento de texto.

De modo geral, os resultados evidenciam que nem todas as técnicas utilizadas

geram bons resultados para o problema estudado. Detalhadamente, os resultados evidenciam que o uso da técnica de *stemmer* melhora as duas tarefas estudadas. Isso porque nas bases de dados ATIS, SNIPS, FMD e FMD(Espanhol) foram observados ganhos de até 3% na revocação da tarefa de extração de parâmetros custando uma pequena perda de 0.9% na precisão da mesma tarefa. Outro ponto de destaque sobre o *stemmer* foi a melhora significativa de todas as métricas na avaliação da tarefa detecção de intenção sempre que um ganho fora obtido também na tarefa de extração de parâmetros, ainda que pequeno. O ganho obtido na métrica de revocação já era esperado, pois condiz com trabalhos como (POPOVIČ; WILLETT, 1992; TOMAN; TESAR; JEZEK, 2006; HAN et al., 2012) que também relatam ganhos na revocação ao utilizar o *stemmer*. No entanto, obter ganhos em todas as métricas da forma como aconteceu na tarefa de detecção de intenção nos surpreendeu bastante. Esse comportamento nos leva a hipótese de que em trabalhos que utilizam redes neurais para aprender as duas tarefas simultaneamente, a tarefa de extração de parâmetros é a tarefa mais importante dentre as duas tarefas estudadas. Diante dessa hipótese, é válido pensar que direcionar os esforços em propor melhorias para aprendizado da tarefa de extração de parâmetros pode resultar em ganhos significativos também para a tarefa de detecção de intenção. Em trabalhos futuros podemos verificar se esta hipótese é válida.

Os resultados obtidos por meio do uso da técnica de lematização não foram satisfatórios. Ao analisar a tarefa de extração de parâmetros é notório o percentual de perda em todas as métricas utilizadas. Em certos casos houveram perdas significativas de até 2% na métrica de *f-score* em consequência de perdas acumuladas nas métricas de precisão e revocação. Também foram observadas perdas na tarefa de detecção de intenção ao utilizar a técnica de lematização. Para tentar entender o resultado obtido, analisamos as palavras reduzidas ao lema. Concluímos que os casos de erros nas tarefas estudadas ocorrem devido a uma mesma palavra possuir diferentes lemas. Acreditamos que a tarefa de extração de parâmetros não conseguiu capturar de forma precisa essa mudança de padrão, o que resultou na rotulagem incorreta e conseqüentemente valores baixos para as métricas utilizadas.

A remoção de stopwords também não gerou bons resultados. Tanto a tarefa de

extração de parâmetros, quanto a tarefa de detecção de intenção foram prejudicadas. As métricas acurácia e *f-score* da tarefa de extração de parâmetros chegaram a cair até 6% quando comparado com o mesmo trabalho sem a remoção de stopwords. Em suma, acreditamos que as preposições e artigos que foram removidos eram importantes para contextualizar e localizar os parâmetros a serem extraídos. Os resultados obtidos após o uso de *Word Embeddings* com vetores pré-treinados no formato *GloVe* e *FastText* também resultaram em impactos negativos. Nós observamos resultados insatisfatórios mesmo com a variação das dimensões do vetores pré-treinados em 50, 100 e 200 dimensões, pois ao longo das variações os resultados só decaíram. O nosso entendimento é de que os hiper-parâmetros da redes utilizadas não favorecem o uso dessa técnica. Diante disso, para trabalhos futuros podemos estudar a otimização da configuração de hiper-parâmetros que favoreça as redes utilizadas.

A nossa conclusão após todos experimentos é de que: a) diante das técnicas comparadas, apenas o stemmer gera bons resultados para as tarefas estudadas. b) a lematização confunde o modelo por apresentar diferentes lemas para uma mesma palavra c) a remoção de *stopwords* remove preposições e artigos que são importantes para contextualizar e localizar os parâmetros a serem extraídos o que também confunde o modelo; d) A configuração dos trabalhos comparados não favorece o uso da técnica de *Word Embeddings*; e) Existe uma correlação entre as tarefas estudadas, onde a tarefa de extração de parâmetros é a tarefa de maior importância.

Ao longo de todo o processo de pesquisa científica desenvolvido nessa dissertação, aprendemos que a todo momento surgem novas arquiteturas e combinações de modelos neurais para as tarefas estudadas. Muitas vezes, a inclusão de pequenas modificações em arquiteturas neurais existentes é suficiente para aumentar a acurácia ou *f-score* dessas tarefas. No entanto, são poucos os trabalhos que exploram o impacto e a importância das decisões tomadas na etapa de pré-processamento dos dados como feito neste trabalho. Por este motivo, acreditamos que essa linha de pesquisa ainda possui vários pontos inexplorados. Em um trabalho futuro é possível utilizar outras técnicas de pré-processamento, por exemplo, N-gramas de palavras, técnicas de *Tokenização* que não separem as palavras apenas por espaços e até considerar os caracteres

como menor unidade em vez de uma palavra. Diante da exploração de outras técnicas de pré-processamento de texto para problema de compreensão de linguagem natural voltado a sistemas de diálogo orientados a tarefa, podemos contribuir muito para o entendimento do problema e o entendimento de como as tarefas de detecção de intenção e extração de parâmetros se relacionam.

---

## Referências

- AZMI, A.; AL-THANYAN, S. Ikhtasir—a user selected compression ratio arabic text summarization system. In: IEEE. *2009 International Conference on Natural Language Processing and Knowledge Engineering*. [S.l.], 2009. p. 1–7. [37](#)
- BAKLIWAL, A. et al. Mining sentiments from tweets. In: *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*. [S.l.: s.n.], 2012. p. 11–18. [37](#)
- BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: springer, 2006. [22](#)
- BOJANOWSKI, P. et al. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, MIT Press, v. 5, p. 135–146, 2017. [29](#)
- CELEBI, E.; SEN, B.; GUNEL, B. Turkish—english cross language information retrieval using lsi. In: IEEE. *2009 24th International Symposium on Computer and Information Sciences*. [S.l.], 2009. p. 634–638. [37](#)
- CHEN, H. et al. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, ACM, v. 19, n. 2, p. 25–35, 2017. [19](#)
- CHIN, D.; ZAPPONE, A.; ZHAO, J. Analyzing twitter sentiment of the 2016 presidential candidates. *American Journal Of Science and Research*, 2016. [37](#)
- COUCKE, A. et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, 2018. [39](#)
- EL-KHAIR, I. A. Effects of stop words elimination for arabic information retrieval: a comparative study. *arXiv preprint arXiv:1702.01925*, 2017. [37](#)
- FELDMAN, S. I. Mobile commerce for the masses. *IEEE Internet Computing*, v. 4, n. 6, p. 74–75, 2000. [17](#)
- GOO, C.-W. et al. Slot-gated modeling for joint slot filling and intent prediction. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. [S.l.: s.n.], 2018. p. 753–757. [34](#), [39](#), [41](#), [47](#)
- GORIN, A. L.; RICCARDI, G.; WRIGHT, J. H. How may i help you? *Speech communication*, Elsevier, v. 23, n. 1-2, p. 113–127, 1997. [25](#)

- HAIHONG, E. et al. A novel bi-directional interrelated model for joint intent detection and slot filling. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. [S.l.: s.n.], 2019. p. 5467–5471. [34](#), [41](#)
- HAKKANI-TÜR, D. et al. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In: *Interspeech*. [S.l.: s.n.], 2016. p. 715–719. [33](#), [39](#), [40](#), [41](#), [47](#), [48](#)
- HAN, P. et al. The influence of word normalization in english document clustering. In: IEEE. *2012 IEEE international conference on computer science and automation engineering (csae)*. [S.l.], 2012. v. 2, p. 116–120. [36](#), [58](#)
- HARMAN, D. How effective is suffixing? *Journal of the american society for information science*, Wiley Online Library, v. 42, n. 1, p. 7–15, 1991. [36](#)
- HEMPHILL, C. T.; GODFREY, J. J.; DODDINGTON, G. R. The atis spoken language systems pilot corpus. In: *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*. [S.l.: s.n.], 1990. [18](#), [39](#)
- HENDERSON, M.; THOMSON, B.; WILLIAMS, J. D. The third dialog state tracking challenge. In: IEEE. *2014 IEEE Spoken Language Technology Workshop (SLT)*. [S.l.], 2014. p. 324–329. [18](#)
- INTELLIGENCE, B. I. Messaging apps are now bigger than social networks. *Business Insider*, v. 20, 2016. [17](#)
- JEONG, M.; LEE, G. G. Triangular-chain conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*, IEEE, v. 16, n. 7, p. 1287–1302, 2008. [32](#), [39](#)
- KORENIUS, T. et al. Stemming and lemmatization in the clustering of finnish text documents. In: *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. [S.l.: s.n.], 2004. p. 625–633. [36](#)
- KRAAIJ, W.; POHLMANN, R. Porter’s stemming algorithm for dutch. *Informatiewetenschap*, p. 167–180, 1994. [26](#)
- LI, Y.; SHAW-TAYLOR, J. Using kcca for japanese–english cross-language information retrieval and document classification. *Journal of intelligent information systems*, Springer, v. 27, n. 2, p. 117–133, 2006. [37](#)
- LIU, B.; LANE, I. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*, 2016. [33](#), [35](#), [39](#), [40](#), [41](#), [47](#), [48](#)
- MANDAL, D. et al. Bengali and hindi to english clir evaluation. In: SPRINGER. *Workshop of the Cross-Language Evaluation Forum for European Languages*. [S.l.], 2007. p. 95–102. [37](#)
- MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2013. p. 3111–3119. [28](#), [29](#)
- MITCHELL, T. M. *Machine Learning*. 1. ed. USA: McGraw-Hill, Inc., 1997. ISBN 0070428077. [22](#)

- ORENGO, V.; HUYCK, C. Rslp stemmer (removedor de sufixos da lingua portuguesa). *Acesso em*, v. 2, 2001. [26](#)
- PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. [S.l.: s.n.], 2014. p. 1532–1543. [29](#)
- POPOVIČ, M.; WILLETT, P. The effectiveness of stemming for natural-language access to slovene textual data. *Journal of the American Society for Information Science, Wiley Online Library*, v. 43, n. 5, p. 384–390, 1992. [36](#), [58](#)
- RAMSHAW, L. A.; MARCUS, M. P. Text chunking using transformation-based learning. In: *Natural language processing using very large corpora*. [S.l.]: Springer, 1999. p. 157–176. [20](#)
- SCHUSTER, S. et al. Cross-lingual transfer learning for multilingual task oriented dialog. *arXiv preprint arXiv:1810.13327*, 2018. [39](#), [40](#)
- SHAWAR, B. A.; ATWELL, E. Chatbots: are they really useful? In: *Ldv Forum*. [S.l.: s.n.], 2007. v. 22, n. 1, p. 29–49. [18](#)
- STAFFORD, T. F.; GILLENSON, M. L. Mobile commerce: what it is and what it could be. *Communications of the ACM, ACM*, v. 46, n. 12, p. 33–34, 2003. [17](#)
- TOMAN, M.; TESAR, R.; JEZEK, K. Influence of word normalization on text classification. *Proceedings of InSciT*, v. 4, p. 354–358, 2006. [36](#), [45](#), [58](#)
- TUR, G.; MORI, R. D. *Spoken language understanding: Systems for extracting semantic information from speech*. [S.l.]: John Wiley & Sons, 2011. [47](#)
- WILLIAMS, J. A belief tracking challenge task for spoken dialog systems. In: *NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012)*. [S.l.: s.n.], 2012. p. 23–24. [18](#)
- YOUNG, S. et al. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE, IEEE*, v. 101, n. 5, p. 1160–1179, 2013. [18](#)
- ZOU, F. et al. Evaluation of stop word lists in chinese language. In: *LREC*. [S.l.: s.n.], 2006. p. 2497–2500. [37](#)