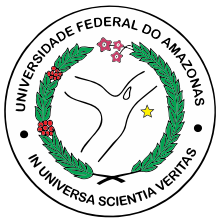


UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Detecção de intenção do usuário utilizando modelos de aprendizagem profunda com uso de hashing semântico

Rodrigo Azevedo da Costa

MANAUS-AM
2019

Rodrigo Azevedo da Costa

Detecção de intenção do usuário utilizando modelos de aprendizagem profunda com uso de hashing semântico

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

Orientador: Prof. Dr. Eduardo James Pereira Souto
Coorientador: Prof. Dr. André Luiz da Costa Carvalho

MANAUS-AM
2019

Ficha Catalográfica

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

Costa, Rodrigo Azevedo da
C837d Detecção de intenção do usuário utilizando modelos de aprendizagem profunda com uso de hashing semântico / Rodrigo Azevedo da Costa. 2019
78 f.: il. color; 31 cm.

Orientador: Eduardo James Pereira Souto
Coorientador: André Luiz da Costa Carvalho
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. detecção de intenção do usuário. 2. aprendizagem de máquina. 3. redes neurais. 4. hashing semântico. I. Souto, Eduardo James Pereira II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

"Detecção de Intenção do Usuário através de modelos de aprendizagem profundo híbrido"

RODRIGO AZEVEDO DA COSTA

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

Prof. Eduardo James Pereira Souto - PRESIDENTE

Prof. André Luiz da Costa Carvalho - MEMBRO EXTERNO

Prof. David Braga Fernandes de Oliveira - MEMBRO EXTERNO

Manaus, 20 de Setembro de 2019

"Há três caminhos para o fracasso: não ensinar o que se sabe, não praticar o que se ensina, e não perguntar o que se ignora."

(São Beda)

Agradecimentos

Agradeço a Deus, por guiar meus passos e renovar minhas forças a cada dia.

Agradeço aos meus pais, Marildo Lopes e Valdeíze Azevedo, pelo amor, apoio incondicional e por terem me dado a maior herança que os pais podem deixar para os filhos, a educação. Obrigado pelas lições de vida que me tornaram um bom filho, um bom marido para minha esposa Poly e um bom pai para a Cecile.

Agradeço aos meus orientadores, Professor André Carvalho e Professor Eduardo Souto, pelo conhecimento compartilhado, pela atenção, por confiar em mim e pelas várias oportunidades que me fizeram crescer na vida acadêmica.

Aos membros da banca examinadora pela participação na defesa deste trabalho com valiosas contribuições.

A todos os meus professores do Instituto de Computação (ICOMP) da Universidade Federal do Amazonas (UFAM), que sempre estiveram dispostos a ajudar, ensinar e mais que isso, servirem de exemplo pessoal e profissional.

A todos que contribuíram, direta ou indiretamente, para conclusão dessa etapa da minha vida. Cheguei até aqui porque tive a oportunidade de conhecer, conviver e aprender com pessoas muitos especiais. Muito obrigado!

Resumo

Um módulo de reconhecimento de intenção do usuário pode ser considerado o componente principal de qualquer sistema de conversação. Intenções são propósitos ou objetivos expressos em uma entrada do usuário por meio de um aplicativo, sistema de busca, chats de conversação, etc. Diversas técnicas são utilizadas e funcionam de maneira satisfatória de acordo com o cenário de aplicação. Atualmente, as técnicas de aprendizagem de máquina são consideradas o estado da arte para este tipo de tarefa e têm sido aplicadas em diversos trabalhos. Tais modelos possuem alta capacidade de representação e podem facilmente aprender as relações existentes entre os termos de um conjunto de treinamento. Entretanto, para alguns cenários não há uma grande quantidade de amostras e, consequentemente, realizar um aprendizado adequado por parte do método fica comprometido. Outro ponto importante é com relação à disposição dos termos dentro de uma sentença, dificuldade à qual muitas pesquisas não levam em consideração. Este trabalho implementa um módulo para reconhecimento de intenção do usuário baseado em modelos de aprendizagem profunda híbrido, levando em consideração a disposição dos termos de uma sentença gerando um valor adicional (hash semântico) e algoritmos de incorporação de texto. Propomos a utilização desse valor extra apenas para as palavras consideradas mais importantes dentro de uma sentença criando uma representação mais direcionada. Como resultado, o modelo desenvolvido atingiu uma precisão média de 93,95%, superando em mais de 2 pontos percentuais os demais trabalhos avaliados mostrando a possibilidade de ganho com a utilização valores adicionais referentes a alguns termos da sentença.

Palavras-chave: detecção de intenção do usuário, aprendizagem de máquina, redes neurais.

Abstract

A user intent recognition module can be considered the main compoenent of any conversation system. Intentions are purposes or goals expressed in a user input through an application, search engine, chat, etc. Several techniques are used and work satisfactorily according to the application scenario. Currently, machine learning techniques are considered state of the art for this type of task and have been applied in many works. Such models have high representational capacity and can easily learn the relationships between the terms of a training set. However, for some scenarios there are not a large number of samples and, as a result, proper learning by the method is compromised. Another important point concerns the disposition of terms within a sentence, a difficulty that many researches do not take into account. This paper implements a user intent recognition module based on hybrid deep learning models, taking into account the disposition of the terms of a sentence generating an additional value (semantic hash) and text embedding algorithms. We propose using this extra value only for the words considered most important within a sentence creating a more targeted representation. As a result, the developed model reached an average accuracy of 93.95%, exceeding by more than 2 percentage points the other works evaluated showing the possibility of gain with the use of additional values referring to some og the sentence.

Keywords: user intent detection, machine learning, neural networks.

Lista de Figuras

2.1 Visão parcial da atividade de detecção de intenção. 7
2.2 Técnicas existentes para a tarefa de PLN. 8
2.3 Processo de classificação usando técnicas baseadas em regras. 9
2.4 Processo de classificação usando técnicas de aprendizagem de máquina. . . 10
2.5 Células cinzas marcam as etapas automatizadas do processo de análise. . . 11

3.1 Processo de atualização do inventário de intenções. 17
3.2 Arquitetura do modelo de treino e teste. Os nós sombreados significam que eles são treinados. 18
3.3 Matriz de dependências para detecção de intenção. 19
3.4 Diferenças entre detecção de intenção e de contexto apresentada pelo autor. 20
3.5 Framework proposto. 21
3.6 Arquitetura da LSTM bidirecional utilizada. 22
3.7 Arquitetura da LSTM bidirecional utilizada. 23

4.1 Estrutura da rede neural LSTM e GRU 32
4.2 Exemplo de classificador CRF. 34
4.3 Arquitetura do método proposto. 34

5.1 Protocolo experimental para definição e execução dos experimentos. 37
5.2 Ilustração do método k-fold. 42
5.3 Estrutura comum para as redes neurais avaliadas. 44
5.4 Desempenho da rede LSTM simples (camada única). 45
5.5 Desempenho da rede LSTM bidirecional. 46
5.6 Desempenho da rede LSTM-GRU. 47
5.7 Desempenho da rede GRU invertida. 48

5.8 Desempenho da rede GRU Bidirecional. 49
5.9 Gráfico da média dos resultados obtidos por cada trabalho. 52
5.10 Matriz de confusão obtida a partir do método proposto. 53
5.11 Plotagem das Curvas ROC para cada um dos trabalhos avaliados. 55

Lista de Tabelas

2.1 Áreas de aplicação para a tarefa de detecção de intenção. 12
2.2 Maneiras de capturar uma intenção do usuário. 13

3.1 Listagem dos trabalhos relacionados e suas características. 26
3.2 Comparativo dos trabalhos relacionados com a proposta apresentada. . . . 27

5.1 Quantidade de intenções por cada base de dados apresentada. 38
5.2 Quantidade de intenções por cada tipo da base ATIS. 39
5.3 Quantidade de intenções por cada tipo da base ATIS. 40
5.4 Dados quantitativos das bases após o processo de aumento de dados. . . . 41
5.5 Resultados obtidos a partir das avaliações. 50
5.6 Avaliação geral dos métodos (10-fold cross validation). 51
5.7 Média geral obtida por cada trabalho. 52
5.8 Avaliação geral da perfomance com valores M e μ. 54

Lista de Abreviaturas e Siglas

ATIS	Air Travel Information System
CTIC	Centro de Tecnologia da Informação e Comunicação
CNN	Convolutional Neural Network
CRF	Conditional Random Fields
GRU	Gated Recorrent Unit
LSTM	Long Short-Term Memory
ML	Machine Learning
PLN	Processamento de Linguagem Natural
ROC	Receiver Operating Characteristic
RNN	Recurrent Neural Network
SVM	Support Vector Machine
UFAM	Universidade Federal do Amazonas

Sumário

1 Introdução **1**

1.1 Motivação e Problema 2

1.2 Objetivos 3

1.2.1 Objetivo Geral 3

1.2.2 Objetivos Específicos 4

1.3 Contribuições do Trabalho 4

1.4 Organização do Documento 5

2 Detecção de Intenção do Usuário **6**

2.1 Detecção de Intenção 6

2.2 Abordagens para Detecção de Intenção 8

2.2.1 Detecção de Intenção do Usuário baseada em regras 9

2.2.2 Detecção de Intenção do Usuário baseada em técnicas de Aprendizagem de Máquina 10

2.2.3 Aprendizagem Profunda 11

2.3 Áreas de Aplicação 11

2.4 Problemas e Desafios 13

2.5 Considerações Finais do Capítulo 14

3 Trabalhos Relacionados **16**

3.1 Detecção de Intenção sem auxílio de processamento adicional 16

3.1.1 Weakly Supervised User Intent Detection for Multi-Domain Dialogues 16

3.1.2 Query Intent Detection Using Convolutional Neural Networks . . . 17

3.1.3 Intent Understanding in a Virtual Agent 18

3.1.4 Intent Detection and Semantic Parsing for Navigation Dialogue Language Processing . . . 19
3.1.5 A Politically-Sensitive Dialog System based on Twitter Data . . . 20
3.2 Detecção de Intenção com auxílio de processamento adicional . . . 21
3.2.1 Intent Detection Using Semantically Enriched Word Embeddings . 21
3.2.2 Multi-Domain Joint Semantic Frame Parsing using Bi-directional RNN-LSTM . . . 22
3.2.3 Subword Semantic Hashing for Intent Classificationon Small Datasets 23
3.2.4 Large-Scale Word Representation Features for Improved Spoken Language Understanding . . . 24
3.3 Discussões . . . 25
3.4 Considerações Finais do Capítulo . . . 28

4 Método Proposto **29**
4.1 Arquitetura da Rede Profunda . . . 29
4.2 Representação da Entrada de Dados . . . 30
4.3 Hashing Semântico . . . 30
4.4 Gated Recorrent Unit . . . 32
4.5 Conditional Random Fields (CRF) . . . 33
4.6 Modelo Inverted GRU + CRF . . . 34
4.7 Considerações Finais do Capítulo . . . 35

5 Experimentos e Resultados **36**
5.1 Protocolo Experimental . . . 36
5.2 Base de Dados . . . 37
5.2.1 Ask Ubuntu, Chatbot e Web Application Corpus . . . 38
5.2.2 ATIS . . . 39
5.2.3 Emails de suporte ao usuário . . . 40
5.2.4 Baselines de Comparação . . . 41
5.3 Métricas de Avaliação . . . 41
5.4 Aumento de dados . . . 43
5.5 Resultados . . . 43
5.5.1 Avaliação dos tipos de redes neurais . . . 44

5.5.2 LSTM 45
5.5.3 LSTM Bidirecional 46
5.5.4 LSTM GRU 47
5.5.5 Inverted GRU 48
5.5.6 GRU Bidirecional 49
5.5.7 Estrutura Final da Rede Neural 50
5.6 Avaliação geral do método 51
5.7 Considerações Finais do Capítulo 55

6 Conclusão **57**
6.1 Considerações Finais 57
6.2 Trabalhos Futuros 58

Referências Bibliográficas **59**

Capítulo 1

Introdução

A detecção de intenção do usuário é uma das principais tarefas de qualquer sistema de conversação. A intenção corresponde ao desejo que o sistema de conversação perceberá após o usuário enviar uma mensagem específica. Por exemplo, ao enviar um "obrigado" a intenção do usuário é agradecer, sendo assim a intenção para a frase poderia ser "agradecimento". Por essa razão é preciso determinar a intenção do usuário com clareza para que o sistema possa realizar a tarefa correta.

A tarefa de detecção de intenção está presente em uma ampla variedade de serviços e aplicações como os mecanismos de pesquisa na Internet e nos chats de conversação (chatbots) de muitos sistemas web (Frey and Osborne, 2017), (Zheng et al., 2017). Por exemplo, nos mecanismos de pesquisa é preciso decidir corretamente a intenção do usuário antes de tentar corresponder entidades ou respostas específicas para a consulta.

No caso de sistemas de conversação mais complexos, nos quais o computador deve manter uma conversa com o usuário, o reconhecimento de intenções pode ser usado para direcionar a interação buscando a resolução de algum problema ou uma necessidade pontual. Essa interação mútua pode ajudar o sistema a aprender cada vez mais com base nas entradas fornecidas pelo usuário.

Detectar essas intenções para os mais diversos tipos de sistemas não é uma tarefa trivial, visto que a maneira de interagir dos usuários é algo muito particular e muda conforme o serviço escolhido, objetivo pretendido, dispositivo utilizado para a realização de uma atividade, existência de mecanismo de busca ou, padrões existentes para detecção de texto (Dale, 2016), etc.

Em geral, as abordagens utilizadas para detectar intenções são concebidas a partir de

regras ou por meio de técnicas de aprendizagem de máquina. As abordagens baseadas em regras empregam templates obtidos a partir de dicionários para fazer a verificação de cada possível intenção. Porém, tais abordagens são limitadas e específicas para um domínio de interesse dificultando a sua utilização quando expostas a grandes quantidades de dados.

As soluções atuais de detecção de intenção empregam técnicas de aprendizagem de máquina. O objetivo é fazer uso de classificadores treinados capazes de detectar e, até mesmo, gerar intenções baseadas na entrada do usuário (Graves et al., 2013). Nessa categoria destacam-se o uso de diferentes tipos de redes neurais bem como a possibilidade do uso de técnicas de aprendizagem profunda.

Outra técnica que proporciona melhoria na classificação está relacionada ao enriquecimento de palavras que facilita na aprendizagem de uma rede neural e direciona para uma predição mais efetiva das classes. Uma maneira de se fazer isso é por meio do uso de hashing semântico que produz uma lista de valores semelhantes em um tempo menor e independente do tamanho da coleção de dados disponíveis. Essa disponibilidade permite o armazenamento de informações adicionais sobre um item específico.

1.1 Motivação e Problema

A tarefa de detectar intenções está ligada diretamente à área de processamento de linguagem natural (PLN). Embora existam diversos algoritmos que realizem a coleta, treino, classificação e, até mesmo, geração de respostas automáticas para as intenções proferidas pelos usuários, um problema recorrente é com relação à representação das palavras para definição de intenções e realização de buscas (Bojanowski et al., 2017).

Para Abdul-Kader and Woods (2015) e Liu and Lane (2016) as principais dificuldades sobre a atividade de detectar intenções estão relacionadas à fase de treino. Os três principais motivos são: 1) quantidade insuficiente de exemplos na base de dados, 2) o conteúdo das bases utilizadas no treino é diferente das interações finais e, 3) desconhecimento sobre a quantidade ideal de treino a ponto de fazer com que o algoritmo se torne específico demais para um determinado cenário gerando o problema conhecido como *overfitting*.

Os modelos de detecção de intenções recentes têm adotado soluções baseadas em redes neurais devido a sua capacidade natural de aprendizagem (Graves et al., 2013), (Goodfellow et al., 2016). Em muitos casos são utilizadas as chamadas redes neurais rasas com uma

ou duas camadas. Porém, esse tipo de rede não tem autonomia suficiente para abranger todos os dados fornecidos na entrada. Por essa razão, muitas soluções atuais têm adotado redes neurais profundas, devido à possibilidade de construir um espaço de características melhorado, além de obter uma representatividade maior dos dados (LeCun et al., 2015).

Outra característica observada é que muitos trabalhos não levam em consideração a distribuição, ordem e peso na qual uma palavra aparece dentro de uma sentença (Liu and Lane, 2016), (Shen and Zhang, 2016). Para superar esse problema, este trabalho propõe um método utilizando a arquitetura de redes neurais profundas do tipo *Inverted Gated Recorrent Unit*, introduzida por (Cho et al., 2014), para análise de sentença e contexto em junção com o classificador *Conditional Random Fields* (CRF) para reconhecimento, cálculo de probabilidade das entidades e transferência de pesos.

Diferentes das abordagens existentes, nós introduzimos a utilização de uma função de hashing semântico, proposto inicialmente por Shen et al. (2014), que acrescenta uma representação junto ao vetor de características das sentenças baseada nos subtokens obtidos a partir dos n-gramas de cada termo. Um n-grama de letras é uma sequência de n letras de uma dada palavra.

A utilização desse valor junto ao vetor de características tem como base o trabalho desenvolvido por Shridhar and Sahu (2018) no qual propõe a inserção de um valor junto ao vetor de entrada representando toda a sentença informada pelo usuário. O diferencial do nosso trabalho está na representação adotada para cálculo do novo valor, baseado unicamente nas palavras com maior peso.

Nem todos os termos são essenciais para compreensão e classificação de uma sentença conforme demonstrado em Mikolov et al. (2013) onde utilizam a ideia de hierarquia de termos em uma rede neural superficial para determinar os termos mais importantes dentro de grandes coleções. Essa modificação pode favorecer a representação de uma intenção por meio das palavras-chaves e da estrutura da sentença utilizada.

1.2 Objetivos

1.2.1 Objetivo Geral

O desenvolvimento deste trabalho tem por objetivo aprimorar a tarefa de detecção de intenções do usuário, por meio do uso de redes neurais profundas do tipo *Inverted*

GRU, juntamente com o uso da técnica de hashing semântico, adicionando à estrutura de representação das palavras um valor que fortalece o peso dos termos mais importantes.

1.2.2 Objetivos Específicos

Para atingir o objetivo geral desta pesquisa, um conjunto de resultados intermediários deve ser alcançado:

1. Adaptar e combinar técnicas de PLN com algoritmos de redes neurais profundas, a fim de criar um modelo que possa ser utilizado na tarefa de detecção de intenções voltado para bases textuais pequenas.

2. Estabelecer procedimentos que permitam a definição de um arcabouço mínimo a ser utilizado para a construção de um modelo de detecção de intenção.

3. Demonstrar a capacidade de classificação geral do modelo em sentenças não incluídas no treinamento.

1.3 Contribuições do Trabalho

Embora existam pesquisas abordando a problemática de detecção de intenção utilizando redes neurais, o método proposto neste trabalho apresenta algumas contribuições importantes, a saber:

1. Utilização de um valor adicional para intensificar a probabilidade de uma sentença a ser classificada em uma intenção válida considerando apenas as palavras classificadas com maior importância na sentença.

2. Avaliação do desempenho de diferentes estruturas de redes neurais nas fases de treino e validação para o cenário de aplicação definindo a mais adequada com base nos valores de precisão, revocação e F1.

3. Investigação do impacto de algumas decisões de pré-processamento de dados no comportamento do algoritmo, verificando que essas decisões podem afetar no desempenho final do método.

1.4 Organização do Documento

Este trabalho está dividido da seguinte forma: no Capitulo 2 é apresentada a fundamentação teórica baseada na revisão da literatura, previamente definida e executada. Esse capítulo fornece a base necessária para que o leitor tenha uma visão geral sobre a área de Detecção de Intenções, conceitos principais, problemas existentes, técnicas utilizadas e o cenário de aplicação envolvido.

No Capítulo 3 são listados os trabalhos relacionados que serviram de base teórica para desenvolvimento desta pesquisa. Os trabalhos estão divididos em duas categorias: 1) pesquisas que não utilizam qualquer processamento adicional dependendendo unicamente da geração obtida a partir do modelo de aprendizagem de máquina e, 2) pesquisas que envolvem o uso de técnicas de aprendizagem de máquina juntamente com a utilização de um processamento adicional para auxiliar na atividade de classificação (semelhante ao adotado neste trabalho). Por fim, a última seção do capítulo mostra um comparativo entre cada um dos trabalhos.

No Capítulo 4 é apresentada a descrição da arquitetura do modelo proposto para a tarefa de detecção de intenções. Esse modelo leva em consideração o uso de técnicas de PLN e uma rede neural profunda do tipo *Inverted* GRU juntamente com o uso do método de hashing semântico.

No Capitulo 5 são apresentados os experimentos realizados para avaliação de diferentes tipos de redes neurais e validação do método proposto bem como os resultados alcançados para cada um dos cenários definidos.

Capítulo 2

Detecção de Intenção do Usuário

Este capítulo introduz os conceitos e as definições necessárias para o entendimento deste trabalho. A primeira seção (2.1) faz uma breve descrição sobre a tarefa de Detecção de Intenção. A segunda seção (2.2) apresenta uma descrição sobre as abordagens utilizadas, agrupando-as de acordo com a categoria correspondente. A terceira seção (2.3) mostra as diferentes aplicações em que essa tarefa é utilizada e, por fim, a seção 2.4 relata sobre os problemas e desafios existentes para essa atividade.

2.1 Detecção de Intenção

Intenções são propósitos ou objetivos expressos em uma entrada do usuário. A partir desse conceito é possível resumir intenção em: uma ação que alguém deseja realizar. Para que essa vontade seja compreendida é preciso expressá-la usando algum canal de comunicação. Ao reconhecer a intenção na entrada fornecida é possível determinar o fluxo de diálogo correto que corresponda às necessidades apresentadas.

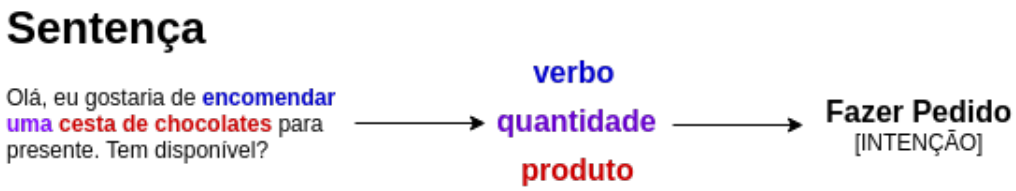
Deoras et al. (2012) definem intenção como uma "solicitação do usuário". Essa definição pode ser utilizada para diversos cenários, pois sempre que há interação com um sistema ou dispositivo, o usuário pode solicitar algum auxílio ou requerer alguma funcionalidade. As informações de contexto são particularmente importantes para o entendimento das intenções proferidas pelos usuários e não considerar tais informações pode resultar em interpretações incorretas (Liu et al., 2016).

Por exemplo, se alguém diz a seguinte frase: "quero comprar um tênis" e, considerando que exista uma base de dados de intenções, a frase dita poderia ser classificada como uma

intenção do tipo Buscar_Produto, ou ainda, Comprar_Produto. De qualquer modo, é preciso que haja um mapeamento correto entre a sentença expressa por um usuário e a intenção detectada pelo sistema utilizado na interação, para então retornar ou realizar alguma ação.

Na Figura 2.1, é possível ter uma visão resumida do processo de detecção de intenção do usuário. A partir de uma sentença fornecida, por meio da fala ou texto, é preciso identificar as palavras mais importantes da sentença. No caso da figura, os termos mais importantes são: 'encomendar', 'uma', 'cesta'. Assim pode-se realizar o processamento utilizando algum algoritmo para obter a intenção desejada pelo usuário.

Figura 2.1: Visão parcial da atividade de detecção de intenção.



Devido às diferentes maneiras de interação e a possibilidade de influenciar no comportamento dos usuários, é preciso que a tarefa de detecção de intenções seja feita de maneira cautelosa e precisa (Khatua et al., 2017). Dessa maneira, será possível extrair corretamente o significado da sentença e dar fluidez à comunicação estabelecida a fim de reconhecer com eficácia o que realmente o usuário está querendo dizer (Venkataraman and Anantha, 2017).

Em alguns casos, é preciso ainda realizar a tarefa de reconhecimento de entidades. Por exemplo, na sentença: "eu quero voar de Manaus para Belém amanhã de manhã". A intenção dessa entrada do usuário pode ser "procurar voo" e há vários atributos específicos a serem identificados para definir a intenção correta como: "local de destino", "local de partida" e "data de partida".

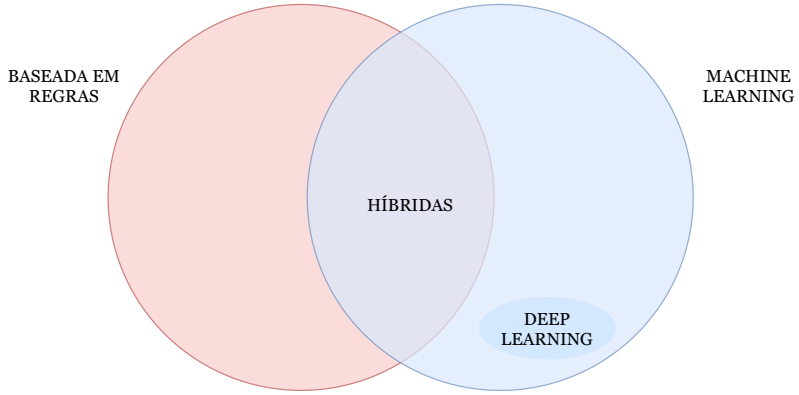
É preciso ressaltar que as intenções a serem detectadas variam de acordo com o cenário de aplicação e meio utilizado na interação. A maneira de utilizar um smartphone, por exemplo, para realizar uma busca não é a mesma quando se utiliza um computador. Essa diferença entre padrões de uso de cada dispositivo tem que ser levada em consideração na hora de analisar e, consequentemente, classificar uma intenção.

2.2 Abordagens para Detecção de Intenção

As soluções para a atividade de detecção de intenções do usuário podem ser agrupadas em abordagens baseadas em regras e abordagens de aprendizagem de máquina. Cada uma possui vantagens e desvantagens que precisam ser analisadas e utilizadas de acordo com as características do cenário de aplicação e do conjunto de dados.

Abordagens baseadas em regras são limitadas e geralmente voltadas para um cenário específico de aplicação, usam expressões regulares com algumas características das entidades de interesse. Por outro lado, as abordagens de aprendizagem de máquina utilizam algoritmos capazes de aprender relações e operam construindo um modelo a partir de entradas padrão de conjuntos de dados. A divisão entre essas duas abordagens principais pode ser vista na Figura 2.2.

Figura 2.2: Técnicas existentes para a tarefa de PLN.



A partir da utilização de duas ou mais abordagens em paralelo é possível obter as chamadas abordagens híbridas. Essas abordagens combinam diferentes características possibilitando uma expansão na utilização de alguma técnica individual. Dentro da abordagem de aprendizagem de máquina temos a chamada aprendizagem profunda (Goodfellow et al., 2016) que tem por objetivo modelar abstrações de alto nível de dados usando várias camadas de processamento.

Definir a arquitetura da rede neural a ser utilizada é muito importante devido ao fato de que seu arranjo depende diretamente do problema a ser tratado. Além disso, a arquitetura da rede está intimamente relacionada ao algoritmo de aprendizagem usado para treinamento. Na escolha da estrutura a ser utilizada são analisados o número de camadas, número de nós, tipo de conexões entre os nós e a topologia da rede.

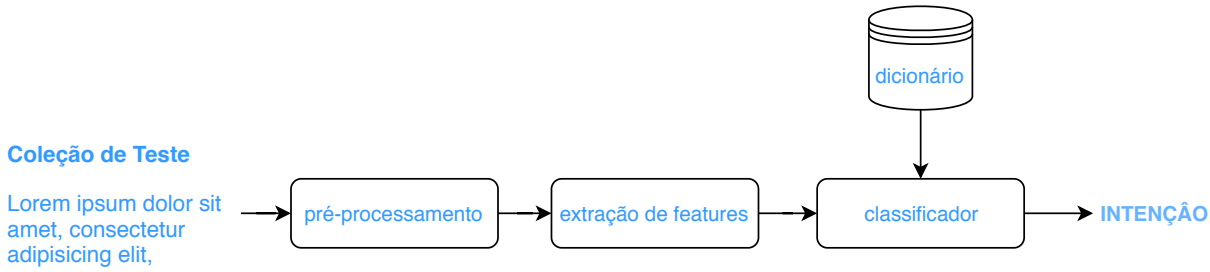
2.2.1 Detecção de Intenção do Usuário baseada em regras

A detecção de intenção baseada em regras usa predefinições para fazer a correspondência das sentenças com as intenções. Embora esses sistemas sejam geralmente precisos (caso detectem uma intenção para uma sentença, ela está correta na maior parte do tempo), sua cobertura é baixa, pois não conseguem detectar intenções para muitas consultas devido à limitação para definição de regras (Hashemi et al., 2016).

A classificação nesse tipo de abordagem é baseada no uso de dicionários que consistem em um conjunto de regras (Figura 2.3) projetadas por especialistas. Essas regras podem ser construídas a partir do uso de expressões regulares que são uma notação para representar padrões em strings e servem para validar entradas de dados ou fazer busca de informações em textos. Por exemplo, para verificar se um dado fornecido é um número de 0,00 a 9,99 pode-se usar a expressão regular definida por: "d,dd", pois o símbolo "d"é um coringa que casa com um dígito.

A principal desvantagem dessa abordagem é a construção manual das regras que é uma tarefa demorada e dependente do domínio de aplicação. Esse processo, pensando na possibilidade de escalar para um grande número de intenções, é difícil e requer muito esforço humano.

Figura 2.3: Processo de classificação usando técnicas baseadas em regras.



Fonte: Adaptado de Hashemi et al. (2016)

Geralmente as técnicas baseadas em regras, devido às limitações apresentadas de escalabilidade e necessidade de rotulação manual, são utilizadas em conjunto com alguma outra técnica envolvendo aprendizado de máquina e suas variações (Bhargava et al., 2013), (Liu et al., 2016). Essa junção forma as chamadas técnicas híbridas e possibilita um aumento nas taxas precisão e revocação além de reutilização em diferentes cenários.

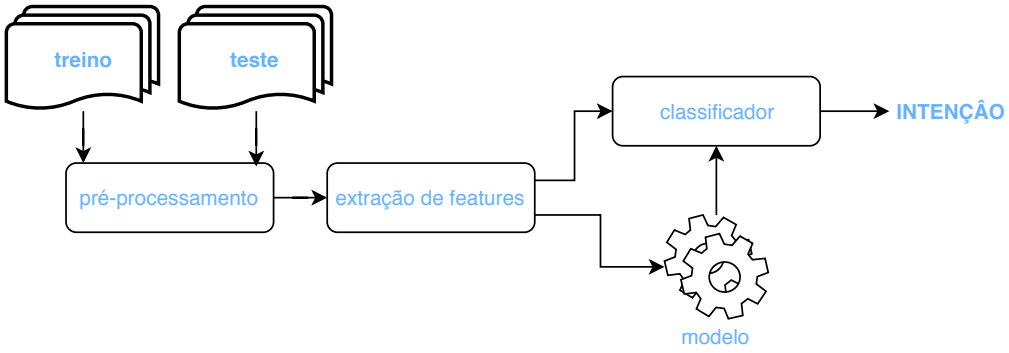
2.2.2 Detecção de Intenção do Usuário baseada em técnicas de Aprendizagem de Máquina

Técnicas baseadas em aprendizagem de máquina subdividem-se basicamente em três categorias: supervisionadas, semi-supervisionadas e não-supervisionadas. Cada uma possui vantagens e limitações que precisam ser analisadas na hora da utilização, levando em consideração as características de cada problema (Hu et al., 2009).

Em geral, os classificadores de intenção utilizam uma base de dados previamente rotulada. Porém, a coleção de exemplos precisa ser completa e correta, englobando a maioria dos contextos possíveis e apresentar o menor número de erros possível (Kim et al., 2016).

O problema da classificação é dividido basicamente em: 1) aprender e gerar um modelo sobre uma coleção de dados de treinamento e, 2) prever as intenções com base no modelo resultante (Figura 2.4). Dentre os algoritmos de classificação, os mais utilizados *Support Vector Machine*, *Naive Bayes*, *Maximum Entropy* além dos algoritmos baseados em Redes Neurais.

Figura 2.4: Processo de classificação usando técnicas de aprendizagem de máquina.



Fonte: Adaptado de Hashemi et al. (2016)

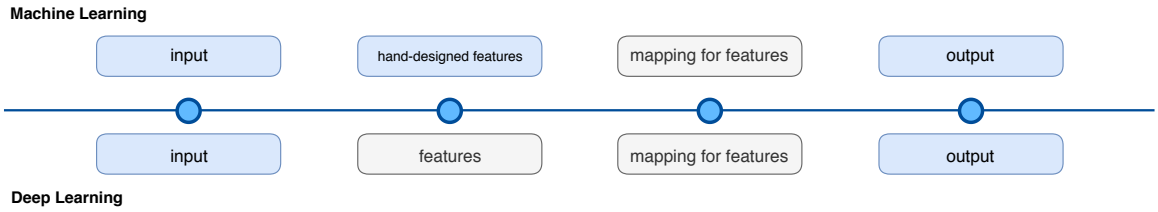
Neste trabalho, técnicas de aprendizagem de máquina foram utilizadas nos experimentos de detecção de intenções do usuário. Essa abordagem foi escolhida por duas razões, a primeira é por ser um método probabilístico que não exige conhecimento explícito dos dados utilizados. E a segunda, é porque a metodologia utilizada neste trabalho é supervisionada, já que esta permite a transferência de conhecimento por meio da elaboração de uma base de treinamento, além disso permite a comparação dos resultados obtidos pela base de treinamento elaborada neste trabalho com outras bases de dados.

2.2.3 Aprendizagem Profunda

Outro tipo de abordagem disponível para o processo de classificação consiste em utilizar técnicas de Aprendizagem Profunda (Goodfellow et al., 2016). Uma das vantagens da utilização dessa abordagem, nas tarefas de classificação, é a possibilidade de aprender um mapeamento entre as características representativas e uma saída desejada (*output*).

As redes neurais utilizadas para construção dos modelos baseados nesse tipo de abordagem conseguem aprender as características representativas de maneira automática (Cho et al., 2014), (LeCun et al., 2015) e, (Schmidhuber, 2015). Essa possibilidade favorece o aumento de automatização do processo de detecção (Figura 2.5) além de permitir uma escalabilidade em relação à aplicação e utilização desse tipo de abordagem.

Figura 2.5: Células cinzas marcam as etapas automatizadas do processo de análise.



Fonte: Adaptada do livro Deep Learning. (Goodfellow et al., 2016).

Pode-se pensar nas diversas camadas ocultas de uma rede neural profunda aprendendo níveis de abstrações hierárquicas. Em reconhecimento de entidades por exemplo, as camadas mais baixas próximas aos dados de entrada, são responsáveis por aprender a estrutura e radical das palavras, enquanto que as camadas superiores aprendem a juntar essas partes em palavras e sentenças. Essas partes então podem ser utilizadas por um modelo para discriminar entre uma ou outra entidade (Goodfellow et al., 2016).

2.3 Áreas de Aplicação

A tarefa de detecção de intenção pode ser utilizada para diversos cenários (Tabela 2.1), desde realizar a análise de sentimentos a partir de bases obtidas em redes sociais como *Twitter* e, *Facebook* (Bhaskar et al., 2015) até mesmo para gerenciamento de diálogos para navegação em veículos (Zheng et al., 2017). Em cada domínio específico as técnicas podem ser aplicadas de acordo com a necessidade e características existentes para atingir

aos objetivos definidos.

Devido à importância e possibilidades de aplicações diversas, houve a necessidade de desenvolver e aprimorar as técnicas tradicionais existentes. Esse aprimoramento vem acontecendo de maneira gradual e de acordo com o aumento e a variação de intenções que podem ser detectadas com o passar dos anos.

Tabela 2.1: Áreas de aplicação para a tarefa de detecção de intenção.

Cenário de aplicação	**Referências**
Análise de sentimentos	Rachuri et al. (2010); Bhaskar et al. (2015)
Chatbots	Zhang et al. (2015); Venkataraman and Anantha (2017); Khatua et al. (2017); Cui et al. (2017); Shridhar and Sahu (2018)
Comércio eletrônico	Wang et al. (2015); Cui et al. (2017); Zhu et al. (2018)
Diagnósticos médicos	Zhang et al. (2016)
Detecção de atividades	Chen et al. (2004); Weinstein et al. [2009]; Al-Zaidy et al. (2012)
Mecanismo de busca	Heck and Hakkani-Tur (2012); Bhargava et al. (2013); Zamora et al., 2013; Lefortier et al., 2014; Trippas et al., 2015; Liu and Lane (2016)
Mobile	Rachuri et al. (2010); Li et al. (2014); Mehrabani et al. (2015); Jeon et al. (2016); Sun et al. (2016)
Navegação veicular	Doshi et al. (2011); Hansen et al., 2014; Zheng et al. (2017)
Redes sociais	Hollerit et al. (2013); Nobari and Tat-Seng (2014); Wang et al. (2015)
Robótica	(Losey et al., 2018; Yap et al., 2016; Erden & Tomiyama, 2010)

Do mesmo modo que a quantidade de aplicações é abrangente, o tipo de intenção detectada também diverge. Apesar de se concentrarem em aspectos gerais da língua, como identificar entidades próprias, verbos ou, substantivos, a intenção mapeada pode diferir dependendo do cenário abordado ainda que apresentem o mesmo conjunto de dados. Por isso exige-se alta precisão e definição na execução da tarefa para estabelecer parâmetros corretos e que possam representar da melhor maneira possível as reais intenções dos

usuários.

A partir da análises dos trabalhos listados foi possível agrupá-los pela maneira como uma intenção é detectada pelo sistema, além da captura por interação textual (mecanismos de busca e chatbots) e, áudio, o comportamento do usuário pode ser analisado e classificado em uma intenção válida. A Tabela 2.2, mostra a relação de trabalhos de acordo com o tipo de entrada possível para uma intenção.

Tabela 2.2: Maneiras de capturar uma intenção do usuário.

Maneira de captura	**Referências**
Áudio	Zheng et al. (2017); Kim et al. [2016]; Trippas et al., 2015; Heck and Hakkani-Tur (2012); Zhang et al. (2015); Jeon et al. (2016) Ji et al. [2014]; Liu & Sarikaya, [2013]; Bhaskar et al. (2015)
Textual	(Yoon et al., 2009; Chang et al., 2011; Bhargava et al. (2013); Hollerit et al. (2013); Lefortier et al., 2014; Nobari and Tat-Seng (2014); Wang et al. (2015); Hashemi et al. (2016); Liu and Lane (2016); Shao et al. (2016); Zhang et al. (2016); Khatua et al. (2017); Shridhar and Sahu (2018)
Uso de sistemas e manuseio de aparelhos	Doshi et al. (2011); Li et al. (2014); Mehrabani et al. (2015); Sun et al. (2016); Yap et al. [2016]; Cui et al. (2017)

2.4 Problemas e Desafios

Para realizar a detecção de intenção do usuário de maneira eficaz é preciso levar em consideração os problemas existentes para a área. Dessa maneira é possível avançar os estudos em um determinado problema ou ainda, utilizar soluções e técnicas específicas para se desenvolver aplicações cada vez melhores. Alguns desses problemas encontrados na literatura são:

- **Estrutura da língua alvo:** a ordem em que a frase é construída pode influenciar na análise e detecção de uma intenção. Geralmente, na língua portuguesa, a regra

padrão é: SUJEITO VERBO PREDICADO, porém, devido à variabilidade de expressão linguística muitas vezes há mais de uma maneira para se expressar sobre uma mesma intenção.

- **Ambiguidade:** textos ou palavras que possuem mais de um significado. Em alguns casos, esse problema pode ser resolvido pela análise do contexto em que a frase se encontra.

- **Correferência de termos:** para alguns cenários é comum fazer referência a termos já ditos anteriormente por meio de algum pronome ou, outra palavra que não a original. Na língua portuguesa, esse problema recebe o nome de resolução de Correferência ou, Anáfora.

- **Dupla intenção:** quando em uma sentença existe mais de uma entidade válida para a interação, por exemplo, na sentença: "Quero uma pizza de mussarela e um suco"há duas entidades válidas, "pizza de mussarela"e "suco". A maioria das estruturas suporta apenas uma entidade para cada intenção.

- **Intenção de retorno:** na interação usando chatbots é preciso que estes reconheçam as intenções ditas pelo usuário, porém, em alguns casos essa classificação falha devido a não compreensão da sentença, podendo retornar uma *fallback intent* ou, ainda, as intenções (mais de uma) com maiores pontuações.

2.5 Considerações Finais do Capítulo

A tarefa de detecção de intenção tem por finalidade descobrir o que o usuário pretende fazer a partir de uma interface que promova a interação entre homem e máquina. Essa descoberta pode ser feita levando em consideração 3 maneiras distintas de análise: 1) fala, 2) textual e 3) uso de sistemas, aparelhos ou dispositivos móveis.

Essa classificação pode ser feita através de duas abordagens: 1) baseada em regras e, 2) aprendizagem de máquina. Cada abordagem possui suas vantagens e desvantagens que precisam ser avaliadas com cautela na hora de decidir qual utilizar. Alguns fatores levados em consideração são: quantidade de dados anotados para treinamento, cenário de aplicação, idioma alvo e particularidades semânticas e sintáticas, por exemplo.

A escolha de uma das abordagens por si só não é garantia de sucesso nos experimentos devidos aos fatores listados anteriormente. Porém, é preciso ressaltar que ao longo dos anos vêm se dando preferência para a junção das técnicas. A essa junção dá-se o nome de técnicas híbridas que tendem a levar em consideração o melhor de cada grupo para aumentar a eficácia dos resultados.

Capítulo 3

Trabalhos Relacionados

Este capítulo aborda os trabalhos relacionados com o estado-da-arte para esta pesquisa contemplando: 1) pesquisas que não utilizam qualquer processamento adicional dependendo unicamente da geração obtida a partir do modelo de aprendizagem de máquina adotado e, 2) pesquisas que envolvem o uso de técnicas de aprendizagem de máquina juntamente com a utilização de um processamento adicional para auxiliar na atividade de classificação (semelhante ao adotado neste trabalho).

3.1 Detecção de Intenção sem auxílio de processamento adicional

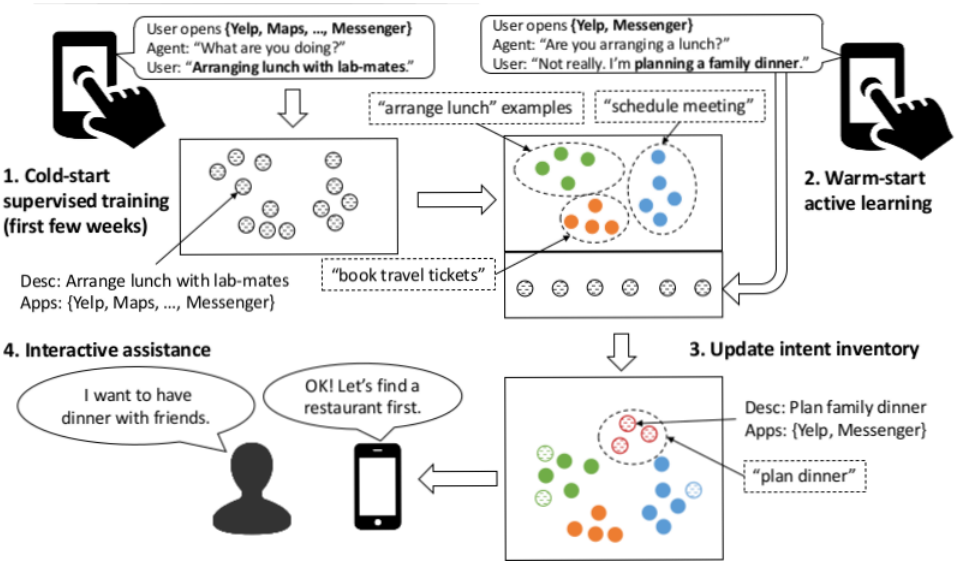
3.1.1 Weakly Supervised User Intent Detection for Multi-Domain Dialogues

Sun et al. (2016) propõem o desenvolvimento de um framework para a construção de um agente inteligente capaz de aprender e analisar as ações que um usuário realiza em um dispositivo móvel, fornecendo assistência quando necessário. O agente inteligente é capaz de criar um inventário de intenções a partir de um pequeno conjunto de declarações dos usuários. A intenção pode ser determinada a partir de duas fontes de informação: sequência de uso dos aplicativos e entrada de fala do usuário.

Para ambas as modalidades, a entrada consiste em uma atividade contínua, por exemplo, uma sequência (concluída) de uso de aplicativos ou comandos simples de fala. Após o reconhecimento da intenção o próximo passo consiste na atualização do inventário que

contém o registro de todas as ações realizadas pelos usuários. A atualização ocorre a partir da observação do uso por parte dos usuários e agrupadas conforme especificação fornecida A Figura 3.2 mostra essa etapa de atualização.

Figura 3.1: Processo de atualização do inventário de intenções.



Fonte: Sun et al. (2016).

O sistema aprende um inventário de intenções agrupadas a partir das declarações fornecidas pelos usuários por meio textual ou verbal. Então, ele agrupa automaticamente as intenções e reconhece com precisão as tentativas de classificação observando a sequência de aplicativos usando métodos semi-supervisionados baseados em grafos.

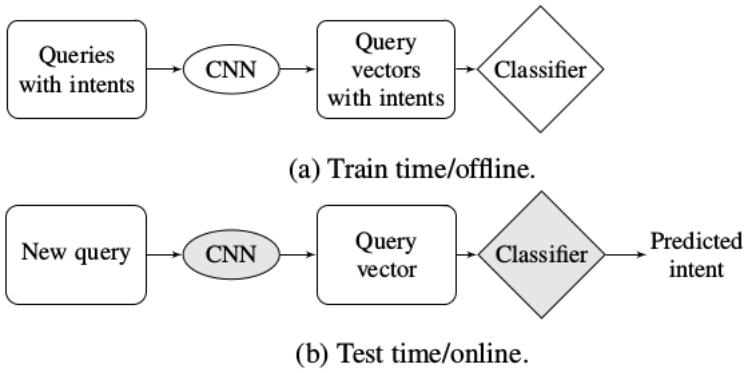
3.1.2 Query Intent Detection Using Convolutional Neural Networks

Hashemi et al. (2016) introduzem um método de classificação para detectar a intenção de uma consulta de pesquisa do usuário. Os autores propõem o uso de redes neurais convolutivas (CNN) para extrair representações vetoriais das consultas como características para o processo de classificação.

Nesse modelo, as consultas são representadas como vetores, de modo que as consideradas semanticamente semelhantes possam ser capturadas incorporando-as a um espaço vetorial. Os recursos de classificação utilizados (dados de clique, sessões de buscas, conceitos e termos frequentemente pesquisados) foram criados a partir de representações vetoriais de consultas geradas automaticamente.

O método proposto consiste basicamente nas etapas de treinamento do modelo com parâmetros no estado off-line e execução do modelo com dados de consulta online (Figura 3.3). No treinamento, as consultas rotuladas são utilizadas para aprender os parâmetros da CNN e do classificador de intenção. Durante a execução dos experimentos foi realizada a consulta nesses dois componentes.

Figura 3.2: Arquitetura do modelo de treino e teste. Os nós sombreados significam que eles são treinados.



Fonte: Hashemi et al. (2016).

3.1.3 Intent Understanding in a Virtual Agent

Venkataraman and Anantha (2017) propõem a construção de um módulo para reconhecimento de intenções combinando técnicas de PLN, Machine Learning e regras. O sistema pode ser utilizado de maneira não supervisionada. As intenções são classificadas em quatro tipos: ação, informação, problema e, saudação.

Para treinar o algoritmo, foram usados dois conjuntos de dados: 1) Um conjunto de dados menor rotulado manualmente com 50 sentenças de consulta e, 2) conjunto gerado por meio de previsões com 300 sentenças de consulta. Nos experimentos, para comparação, os autores consideraram três abordagens: baseada em regras, usando um classificador (Ridge) e, combinando regras com o classificador.

A tarefa de detecção de intenção é feita a partir de uma matriz de dependências onde é verificada a polaridade da sentença, podendo ser marcada como: questão, negação ou ação. A partir da pontuação obtida, a intenção é classificada em um dos quatro tipos listados anteriormente. A Figura 3.4, a seguir, mostra o uso dessa matriz.

Figura 3.3: Matriz de dependências para detecção de intenção.

ID	Question	Negation	Action	Type	E.G
1	0	0	0	General	Hello, Hi
2	0	0	1	Action	I want to install a printer
3	0	1	0	Problem	My printer crashed
4	0	1	1	Problem	I'm not able to install my printer
5	1	0	0	Informati on	Where is the software icon?
6	1	0	1	Informati on	How do I install a printer?
7	1	1	0	Informati on	Where is the machine that isn't working?
8	1	1	1	Action	Can you fix the printer that is crashed?

Fonte: Venkataraman and Anantha (2017).

A combinação do uso de regras com o classificador obteve os melhores resultados, seguida pelo classificador e pela abordagem baseada em regras que em alguns obteve precisão de no máximo 50%. Como trabalhos futuros sugerem a utilização do uso de Redes Neurais para a construção do classificador além da possibilidade do aprendizado automatizado.

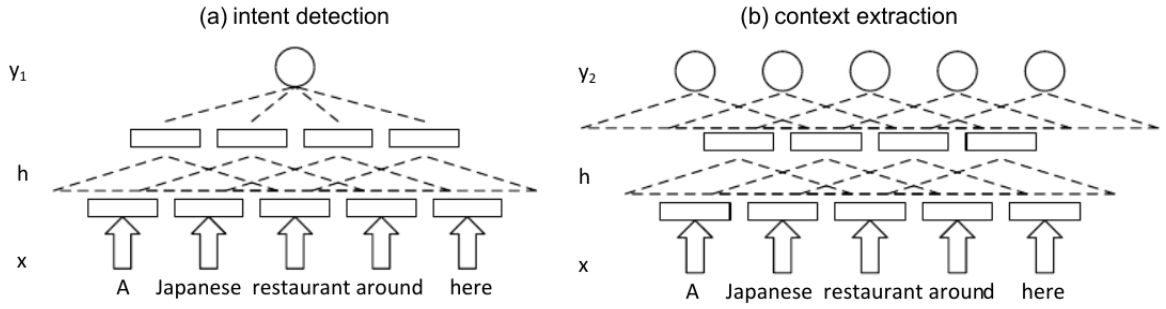
3.1.4 Intent Detection and Semantic Parsing for Navigation Dialogue Language Processing

Zheng et al. (2017) propõem uma arquitetura de Redes Neurais Recorrentes (RNN) voltada ao reconhecimento automático da fala de usuários de veículos. As principais tarefas consistem em 1) realizar a análise semântica da sentença e, 2) detectar as intenções. A solução envolve desde o reconhecimento automático de fala, compreensão da linguagem falada, gerenciamento de diálogos, geração de linguagem natural e síntese de texto-fala.

As RNN foram escolhidas pela facilidade em aprender os recursos otimizados de maneira não supervisionada durante o processo de construção. Para validar a solução proposta, testes foram feitos em ambiente real usando o dataset ATIS que contém exemplos

de conversas de usuários para solicitação de passagem aérea.

Figura 3.4: Diferenças entre detecção de intenção e de contexto apresentada pelo autor.



Fonte: Zheng et al. (2017).

A dificuldade neste cenário está em decidir qual sentença está relacionada à navegabilidade para então responder de maneira correta. Por exemplo, a frase "Eu gosto de chocolate"não é uma sentença de navegação, ao contrário de "Tem um restaurante japonês por perto?".

3.1.5 A Politically-Sensitive Dialog System based on Twitter Data

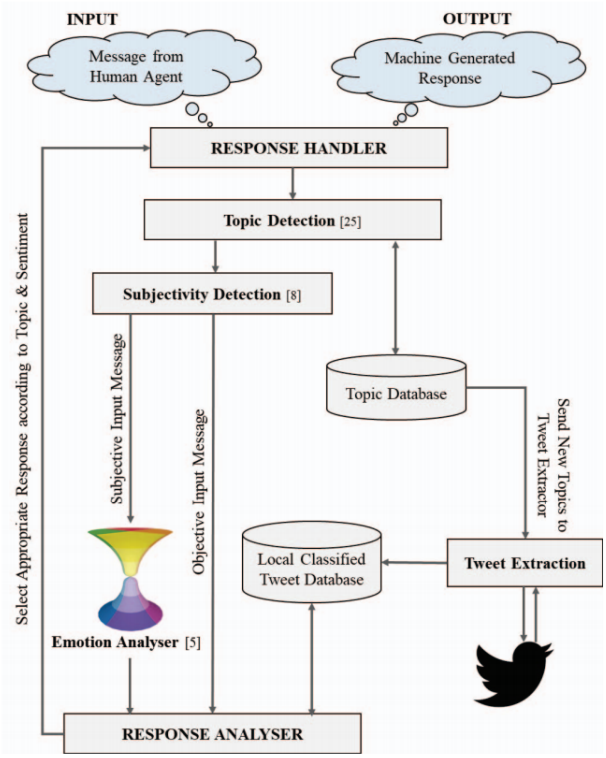
Khatua et al. (2017) propõem o desenvolvimento de um sistema de diálogo a partir de dados não estruturados extraídos da plataforma do Twitter. Para esse sistema o entendimento das intenções ditas pelos usuários se mostrou particularmente difícil devido à caracterização dos dados, *tweets* sobre a situação política da Europa no ano de 2016, contendo sentenças opnativas dos usuários com expressões de sarcasmo e sentimentos.

Nesse caso, além da detecção de entidades foi preciso elaborar um detector de subjetividade e um analisador de emoção para aprimorar o processo de detecção de intenção. Esse processo foi necessário para que o chatbot soubesse distinguir sentenças puramente opnativas e que pudessem interferir no resultado do diálogo, tendo em vista que a capacidade de conversação e o mecanismo de resposta são aspectos importantes em sistemas dessa natureza.

Para a realização dos experimentos foi considerado um total de 2,7 milhões de tweets com a temática de política, mais especificamente sobre o "Brexit". Não houve comparação com outros trabalhos uma vez que é considerado o primeiro trabalho sobre essa temática

envolvendo o uso de chatbots.

Figura 3.5: Framework proposto.



Fonte: Khatua et al. (2017).

3.2 Detecção de Intenção com auxílio de processamento adicional

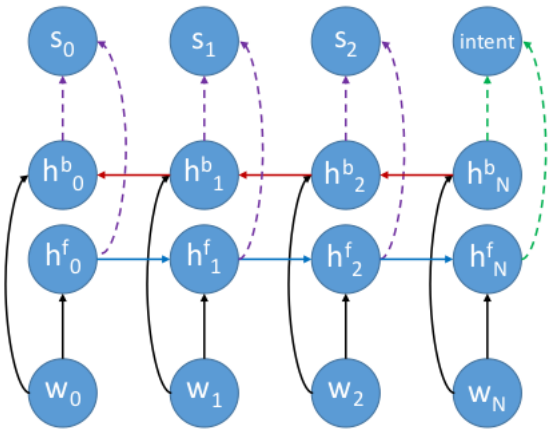
3.2.1 Intent Detection Using Semantically Enriched Word Embeddings

Kim et al. (2016) propõem um modelo que utiliza vetores de palavras (do inglês, *word embeddings*) para enriquecer um dicionário. Nesse trabalho, os autores expandiram o enriquecimento dos vetores de palavras proposto inicialmente em Mrki et al. (2016) incorporando palavras relacionadas de três diferentes dicionários semânticos como o WordNet[1], PPDB e o Macmillan Dictionary[2].

[1]https://wordnet.princenton.edu/
[2]http://www.macmillandictionary.com/us

Para isso foi construída uma rede LSTM Bidirecional levando em consideração o enriquecimento fornecido pelo dicionário. O método ajusta vetores de palavras para fazer 1) vetores de palavras sinônimas serem mais semelhantes, 2) vetores de palavras antônimas mais distantes e, 3) mantendo as semelhanças dos vetores de palavras ajustadas com seus vetores mais próximos. A Figura 3.1 mostra a arquitetura utilizada.

Figura 3.6: Arquitetura da LSTM bidirecional utilizada.



Fonte: Kim et al. (2016).

Na Figura 3.1, *w0* e *wN* denotam a palavra do início (BOS) e do fim (EOS) da sentença, respectivamente. Para os vetores de palavras (GloVe) foram utilizados vetores de 200 dimensões. Os ajustes realizados levaram em consideração as palavras sinônimas, antônimas e próximas em uma mesma sentença. Essa definição serviu para auxiliar no agrupamento de entidades e consequentemente as intenções existentes.

Para realizar as avaliações foi utilizado o corpus ATIS e, um conjunto de dados de registros reais sobre localidades obtido por meio do uso da assistente virtual Cortana (Microsoft). A partir dos experimentos concluíram que a utilização de *word embeddings* durante a fase de treinamento do modelo tende a ser mais eficaz e melhoram a detecção de intenções além do uso para tarefas a nível de palavra.

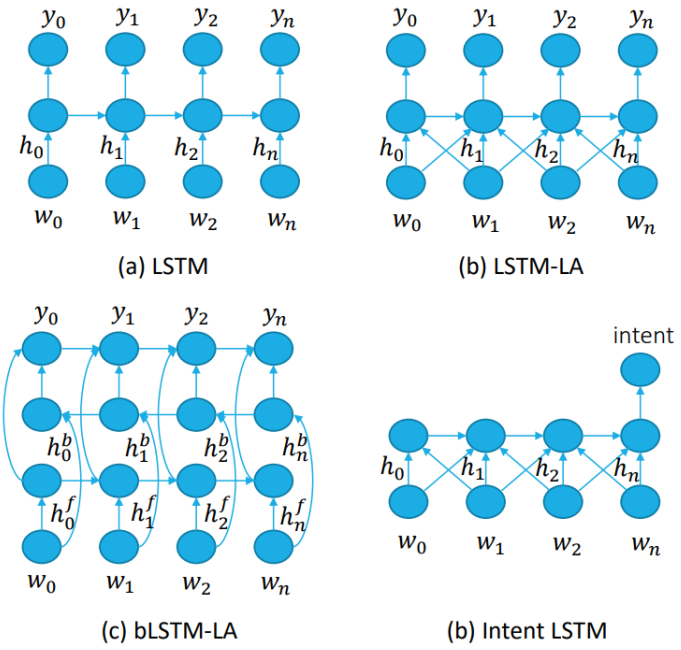
3.2.2 Multi-Domain Joint Semantic Frame Parsing using Bi-directional RNN-LSTM

Hakkani-Tür et al. (2016) abordam a tarefa de detecção de intenção propondo uma abordagem de modelagem holística de múltiplos domínios e multi-tarefa para estimar

quadros semânticos completos para todos os enunciados de usuários endereçados a um sistema conversacional. Demonstram o poder distintivo de métodos de aprendizagem profunda utilizando uma rede bidirecional recorrente com células de memória de longo prazo (LSTM) para lidar com essa complexidade.

As contribuições do trabalho são três: 1) uma arquitetura LSTM para modelagem conjunta de preenchimento de *slots* - slots são alternativas para formulários em forma de conversa, eles permitem a coleta de várias informações de maneira independente do fluxo - 2) construção de um modelo de múltiplos domínios que permite que os dados de cada domínio se reforcem mutuamente e 3) investigação de arquiteturas alternativas para modelar o contexto lexical na compreensão da linguagem falada.

Figura 3.7: Arquitetura da LSTM bidirecional utilizada.



Fonte: Hakkani-Tür et al. (2016).

3.2.3 Subword Semantic Hashing for Intent Classificationon Small Datasets

Shridhar and Sahu (2018) introduzem o uso do hashing semântico como um valor de incorporação para a tarefa de classificação de intenção. O hashing semântico é uma tentativa de superar o desafio de aprender uma classificação robusta de texto em bases

textuais pequenas. Geralmente os métodos baseados em incorporação da palavra-chave (*word embeddings*) são muito dependentes de vocabulários e necessitam de uma grande de exemplos para uma aprendizagem efetiva.

A metodologia empregada consiste em: a partir de uma entrada de texto, dividir a sentença em tokens e cada token é dividido em uma subestrutura de tamanho 3 para gerar sub-tokens e extrair as características individuais de cada termo fornecido na entrada. A opção em utilizar do hash semântico se deu pelo tamanho da base de dados ser pequena e diminuir o tempo de treinamento e inferência.

Para realização dos testes utilizaram treês conjuntos de dados: AskUbuntu Corpus, Web Application Corpus e Chatbot Corpus [3]. Todos eles contendo intenções e sentenças além da mesma estrutura utilizada na fase de treino. A avaliação final foi comparada com os resultados de vários serviços de compreensão de língua e diferentes plataformas de código aberto, tais como: Botfuel, DialogFlow, Luis, Watson, Rasa, Recast e Snips.

3.2.4 Large-Scale Word Representation Features for Improved Spoken Language Understanding

Zhang et al. (2015) aplicam técnicas de PLN juntamente às tarefas de classificação de domínio e detecção de intenção em um sistema de compreensão de linguagem falada, *Spoken Language Understanding* (SLU) para assistentes pessoais. Para a realização da tarefa de classificação das consultas dos usuários utilizaram o classificador *Support Vector Machine* (SVM). Os autores demonstram que a utilização de técnicas de clusterização (*Brown* e *Spectral Based*) podem melhorar o desempenho geral do classificador.

A arquitetura do modelo é dividido em três partes: 1) Classificador do domínio, responsável pelo cálculo da pontuação de cada vetor, 2) Classificador de intenção do usuário e, 3) Identificador semântico. Cada domínio pode ser classificado dentro de sete categorias definidas.

Os experimentos foram realizados com base em um conjunto de dados dividido em três partes: treinamento, validação e teste. Os dados de treinamento tem aproximadamente 540 mil consultas faladas e, os conjuntos de validação e teste possuem 27 mil consultas. Diferentes tamanhos de dimensões foram testadas até atingir o máximo permitido de acordo com a base de dados. Também incluíram bigramas e trigramas obtidos a partir

[3] https://github.com/sebischair/NLU-Evaluation-Corpora

dos clusters definidos, porém, isso não se mostrou relevante para a melhora da precisão.

3.3 Discussões

Os trabalhos relacionados estão divididos em dois grupos de pesquisa: O primeiro, relata sobre os trabalhos que não utilizam processamento adicional junto ao modelo de aprendizagem de máquina escolhido. O segundo grupo trata dos trabalhos que fazem uso de algum processamento adicional para incorporar informações no modelo proposto.

Com os resultados observados na literatura é possível concluir que a utilização de técnicas híbridas para a tarefa de Detecção de Intenção tem se mostrado bastante eficiente em comparação ao uso das técnicas em separado. Uma das razões é a possibilidade de automatizar o processo de aprendizado do modelo, possibilitando uma análise e predição mais coerente e, consequentemente, melhorando a interação com o usuário.

Foi possível verificar ainda a utilização de técnicas baseadas em regras e métodos supervisionados. A principal vantagem em se utilizar regras é a especificação para o domínio desejado. Porém, não é escalável nem trabalha com a ideia de fazer com que se aprenda a partir da interação do usuário ou, ainda permita passar o aprendizado adiante, requisito primordial para esse domínio específico.

Para fins de comparação entre todos os trabalhos relacionados descritos nesse capítulo, a Tabela 3.1, mostra um resumo sobre as seguintes informações: quais os tipos de técnicas utilizadas, tipos de intenções detectadas, base de dados e os resultados obtidos para cada trabalho listado.

Tabela 3.1: Listagem dos trabalhos relacionados e suas características.

Trabalho	**Técnica**	**Intenções**	**Dataset**	**F1**
(Kim et al., 2016)	LSTM Bidirecional	localização	log de uso (Cortana); corpus ATIS	94%
(Hakkani-Tür et al., 2016)	LSTM Bidirecional	datas, eventos, lembretes	corpus ATIS	94,70%
(Sun et al., 2016)	CRF + CNN	uso geral no dispositivo	1089 registros de uso (smartphones)	91,07%
(Hashemi et al., 2016)	CNN	filmes e pessoas	10 mil sentenças de busca	90,3%
(Shridhar and Sahu, 2018)	Hash semântico	localização, suporte ao usuário, dicas	457 sentenças de texto	92%
(Zhang et al., 2015)	SVM + word embeddings	sete categorias de busca	540 mil consultas de áudio	*
(Venkataraman and Anantha, 2017)	Regras; classificador Ridge	informação, problema, ação e genérica	350 sentenças de busca	90%
(Zheng et al., 2017)	LSTM	navegabilidade e localização	corpus ATIS e CU-Move	96,36%
(Khatua et al., 2017)	LSTM	diálogos sobre política	2.7 milhões de tweets	92,50%

A Tabela 3.2 mostra uma comparação entre os trabalhos apresentados com base nas características comuns disponíveis, incluindo as informações da presente proposta de mestrado. Esses resultados ajudam a visualizar melhor o cenário exposto durante a realização deste trabalho bem como perceber os possíveis agrupamentos descritos no Capítulo 2.

A análise das Tabelas 3.1 e 3.2 mostra que cada solução se limita às características específicas das bases de dados utilizadas nos testes e, esse fator dificulta a reprodução dos resultados em uma base com características diferentes. É fundamental ressaltar que o desenvolvimento de distintas formas de classificação nos obriga a uma análise direcionada e específica para cada cenário de aplicação.

Tabela 3.2: Comparativo dos trabalhos relacionados com a proposta apresentada.

Trabalho	**Regras**	**Machine Learning**					**Captura dos dados**			**Reconhece entidades**	**Intenções**
		SVM	**CRF**	**LSTM**	**CNN**	**Outro**	**Áudio**	**Texto**	**Uso**		
(Kim et al., 2016)				x		x	x			não	localização
(Hakkani-Tür et al., 2016)				x			x			não	datas, eventos, lembretes
(Sun et al., 2016)			x		x		x		x	sim	geral
(Hashemi et al., 2016)					x	x		x		sim	filmes e pessoas
(Shridhar and Sahu, 2018)						x		x		não	suporte e dicas
(Zhang et al., 2015)		x				x	x			não	geral
(Venkataraman and Anantha, 2017)	x					x		x		não	informação, ação, problema, genérica
(Zheng et al., 2017)				x		x	x			sim	navegabilidade, localização
(Khatua et al., 2017)				x				x		sim	geral
Proposta de mestrado			x	x		x		x		sim	geral

Pode-se perceber ainda que com relação aos trabalhos voltados especificamente para sistemas de conversação, a utilização de técnicas baseadas em regras e métodos supervisionados é frequente. A principal vantagem em se utilizar regras é a especificação para o domínio desejado. Porém, como já mencionado anteriormente, não é escalável nem trabalha com a ideia de fazer com que se aprenda a partir da interação do usuário.

Dessa maneira, é preciso fazer uso de soluções que possibilitem uma representação mais fiel à realidade de comunicação dos usuários e características inerentes à atividade de conversação como a presença de erros gramaticais, utilização de palavras sinônimas e abreviaturas, além de transparecer para o usuário um fluxo contínuo na troca de informações.

3.4 Considerações Finais do Capítulo

Conforme apresentado nesse capítulo, a partir dos trabalhos relacionados com a proposta foi possível notar a importância, aplicabilidade e avanços alcançados, por meio das técnicas baseadas em aprendizagem de máquina dando ênfase em especial para as abordagens que fazem uso de Redes Neurais e suas variações.

Dessa maneira, a atividade de Detecção de Intenção como tarefa complementar ao reconhecimento de entidades pode melhorar o funcionamento e percepção das mais variadas aplicações que utilizam essa técnica como pré-requisito para interpretação de fala, determinação de tópicos, manutenção de diálogos em uma conversa, análise de sentimento, monitoramento de atividades, entre outros.

A aplicação dessa atividade nos mais diversos tipos de sistemas de conversação é essencial para prover ao usuário uma interação mais eficaz no sentido de entender realmente o que ele quer dizer. Assim, quanto melhor for o entendimento por parte dessa tecnologia, mais o usuário tende a fornecer informações válidas melhorando continuamente a aplicação. Do mesmo modo, a análise dos diversos resultados oferece uma boa oportunidade de verificação e garantia dos modelos propostos.

Capítulo 4

Método Proposto

Este capítulo contém uma descrição sobre a arquitetura da rede neural implementada. A seção 4.1 apresenta a descrição do funcionamento do método como um todo e, em seguida, descreve cada componente envolvido.

4.1 Arquitetura da Rede Profunda

O modelo proposto por este trabalho é baseado na junção de duas técnicas de aprendizagem de máquina. Uma rede neural do tipo *Inverted* GRU e, o classificador *Conditional Random Fields* (CRF). Além disso, na criação da representação vetorial foi adicionado um valor referente ao peso de cada termo presente na sentença (hashing semântico).

A primeira técnica (*Inverted* GRU) consiste na utilização de uma rede neural recorrente que faz uso de memória de longo prazo, similar a uma rede LSTM comum, porém com algumas diferenças que fazem com que o treino nesse tipo de rede seja feito de maneira mais efetivo e com um tempo menor se comparado a outros tipos de redes neurais.

A segunda consiste no uso de CRF que são modelos matemáticos probabilísticos, baseados numa abordagem condicional, utilizados com o objetivo de etiquetar e segmentar dados sequenciais (Lafferty et al., 2001). Podem ser modelados na forma de um grafo não dirigido que define uma única distribuição logarítmica linear sobre uma sequência de rótulos, dada uma sequência de observação.

A seguir são descritos cada componente do modelo separadamente e, por fim, a explicação do modelo completo levando em consideração as funcionalidades e o fluxo seguido em cada momento.

4.2 Representação da Entrada de Dados

A entrada de dados foi representada por meio de vetores de palavras (*word embeddings*). Essa representação vetorial possui duas propriedades importantes e vantajosas: redução de dimensionalidade e, preservação da semelhança contextual (Mikolov et al., 2013). Essas duas características melhoram o processamento como um todo do algoritmo bem como, a representatividade dos termos utilizados.

Além disso, são usados para análise semântica, extraindo significado do texto para permitir o entendimento da linguagem natural. Para um modelo de linguagem ser capaz de prever o significado do texto, ele precisa estar ciente da similaridade contextual das palavras (Mikolov et al., 2013). Os vetores criados por essa representação preservam essas semelhanças, portanto, as palavras que ocorrem regularmente nas proximidades do texto também estarão muito próximas no espaço vetorial.

4.3 Hashing Semântico

O método hashing semântico desenvolvido por Salakhutdinov e Hinton (2007) produz uma lista de documentos semelhantes em um tempo menor e independente do tamanho da coleção de documentos disponíveis. Essa disponibilidade permite o armazenamento de informações adicionais sobre cada documento da coleção, mas essa informação é referente a uma palavra por documento.

Nosso método de hashing semântico é baseado no modelo de semelhança semântica profunda conforme abordada em (Shen et al., 2014) e (Shridhar and Sahu, 2018). Nesses trabalhos, os autores armazenam valores hash para toda sentença junto ao vetor de palavras iniciais para que o modelo dependa desses valores em vez dos tokens das palavras. Isso possibilita montar uma representação de toda a sentença, porém, conforme descrito em Jiang (2012) e Silfverberg et al. (2014), nem todas as palavras contêm representatividade suficiente para determinar a que intenção uma sentença pertence.

Para o escopo deste trabalho, utilizamos os valores de hashing semântico considerando os termos com maior peso dentro de uma sentença. Uma breve descrição do método é relatada a seguir:

A partir de uma sentença de entrada (T), e.g. “Eu tenho uma dúvida”, as palavras

Algoritmo 1: Subword Semantic Hashing + POS Tagger

```
data ← collection of examples;
create set all-sub-tokens;
create list examples;
for text T in data do
    create list examples;
    tokens ← split T into words;
    for words W in tokens do
        w ← # + w + #;
        for x in lenght(w)-2 do
            append w[x:x+2] to all-sub-tokens;
            append w[x:x+2] to examples;
        p ← pos_tags(word);
    append example to examples;
    append p;
return (all − sub − tokens, examples);
return (p);
```

são separadas em tokens e organizadas em uma lista (ti), o resultado dessa separação é: [“Eu”, “tenho”, “uma”, “dúvida”]. Cada token é subdivido em n-gramas por uma função de pré-hashing H(ti) a fim de obter sub-tokens, por exemplo, H(duvida) = [#du, duv, uvi, vid, ida, da#]. O símbolo # serve para indicar o início e o término de cada palavra. Esses sub-tokens são usados para criar um modelo de espaço vetorial servindo para extrair recursos para um determinado texto de entrada.

Após a criação dos sub-tokens foi preciso determinar a função gramatical de cada token dentro da sentença (p). Para cada token w é atribuída a parte da fala (*part of speech* – POS TAG) correspondente, tais como: substantivo (NOUN), verbo (VERB), adjetivo (ADJ), artigo (ART), etc. Uma provável marcação para a frase acima fica: p(wi) = [ART, VERB, ART, NOUN]. A partir dessa marcação, cada token marcado recebe uma pontuação com base na função p.

A cada token marcado por meio da função p é verificado a sua importância dentro da sentença por meio da atribuição de pontos, cumulativos ou não, de acordo com a pos tag verificada. No exemplo acima, a pontuação atribuída à sentença “Eu tenho uma dúvida” seria igual a p(s) = (0, 1, 0, 2). Os tokens marcados como artigo ou qualquer outra função semelhante recebem o valor 0 (zero), ao passo que tokens marcados como verbos, substantivos ou outras palavras que reforcem o seu sentido recebem pontuações cumulativas conforme a posição em que estejam dispostos na sentença.

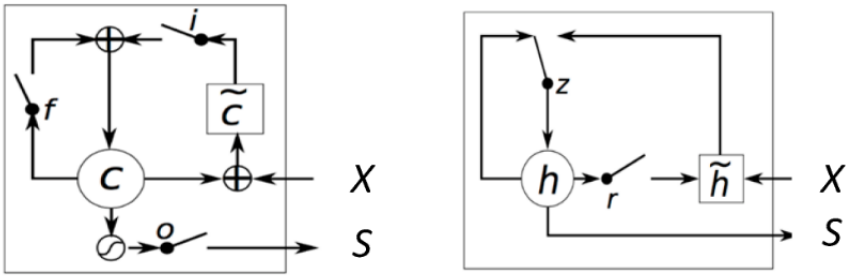
4.4 Gated Recorrent Unit

Introduzidas por Cho et al. (2014), as redes neurais do tipo *Gated Recorrent Unit* (GRU) tem como objetivo resolver o problema de gradiente de fuga que vem com uma rede recorrente padrão. Ela também pode ser considerada uma variação das redes do tipo LSTM porque ambas são projetadas de maneira semelhante e, possuem portões de ativação que regulam a quantidade de informação a ser mantida ou repassada.

Diferentemente de uma LSTM comum, as redes GRU utilizam apenas dois portões de ativação, chamados de: *update gate* e *reset gate*. Basicamente, estes são dois vetores que decidem quais informações devem ser passadas para a saída. O diferencial deles é que podem ser treinados para manter as informações de muito tempo atrás, sem efetivamente carregá-los através do tempo ou remover informações relevantes para a previsão.

Em alguns casos, ambos os tipos de rede produzem resultados igualmente excelentes, porém, as redes GRU vêm demonstrando um desempenho ainda melhor em determinados conjuntos de dados (Chung et al., 2014), principalmente para conjuntos pequenos de dados.

Figura 4.1: Estrutura da rede neural LSTM e GRU



Fonte: Hochreiter and Schmidhuber (1997) e Cho et al. (2014).

Na figura é possível perceber a diferença nas estruturas das redes do tipo LSTM e do tipo GRU. Basicamente, em contraste com uma rede LSTM, uma rede GRU não possui uma célula de memória independente e contém apenas duas portas: uma porta de atualização, que controla o grau de atualização da unidade, e uma porta de redefinição, que controla a quantidade do estado anterior que ela preserva.

O fato de inverter os dados de entrada é uma estratégia que permite uma visualização

mais específica dos termos que compõem a sentença (Sutskever et al., 2014). Essa percepção ajuda na determinação e previsão do comportamento do algoritmo, além disso, faz com que a classificação e identificação do termo ocorra de maneira mais direcionada e considerando apenas valores que pertençam ao conjunto pré-definido.

4.5 Conditional Random Fields (CRF)

Dentre os métodos de aprendizagem de máquina utilizados, um que é comumente utilizado é o *Conditional Random Fields* (CRF). Alguns dos motivos são: facilidade de implementação por meio de diversas bibliotecas, bons resultados obtidos nas tarefas de reconhecimento de entidades, possibilidade de utilização em junção com diferentes técnicas.

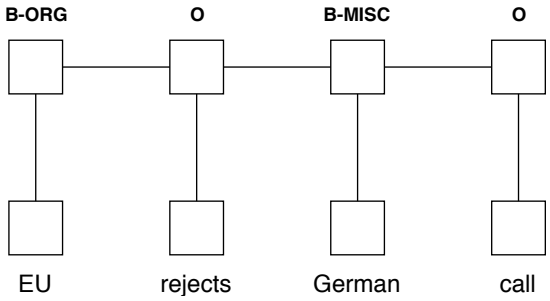
Em linhas gerais são modelos matemáticos probabilísticos baseados numa abordagem condicional, utilizados com o objetivo de etiquetar e segmentar dados sequenciais (Lafferty et al., 2001). Podem ser modelados na forma de um grafo não dirigido que define uma única distribuição logarítmica linear sobre uma sequência de rótulos. Dessa maneira, as influências das diferentes características em estados distintos podem ser tratadas independentemente umas das outras.

Um CRF é uma distribuição condicional P(Y/X) com um modelo gráfico associado. A variável X é um vetor de variáveis aleatórias de entrada e Y é um vetor de variáveis aleatórias de saída. Portanto P(Y—X) é a probabilidade de obter Y dado como entrada o vetor X (Sutton et al., 2012).

$$P(Y/X) = \frac{P(X/Y)P(Y)}{P(X)} \tag{4.1}$$

Devido ao CRF ser originalmente um modelo discriminativo, ele modela a distribuição de probabilidade condicional P(Y—X). Modelos baseados no CRF evitam o conhecido *label bias problem*, que normalmente é observado em Modelos Markovianos Condicionais, tal como o Modelo de Markov de Entropia Máxima (Maximum entropy Markov model) (Sutton et al., 2012).

Figura 4.2: Exemplo de classificador CRF.

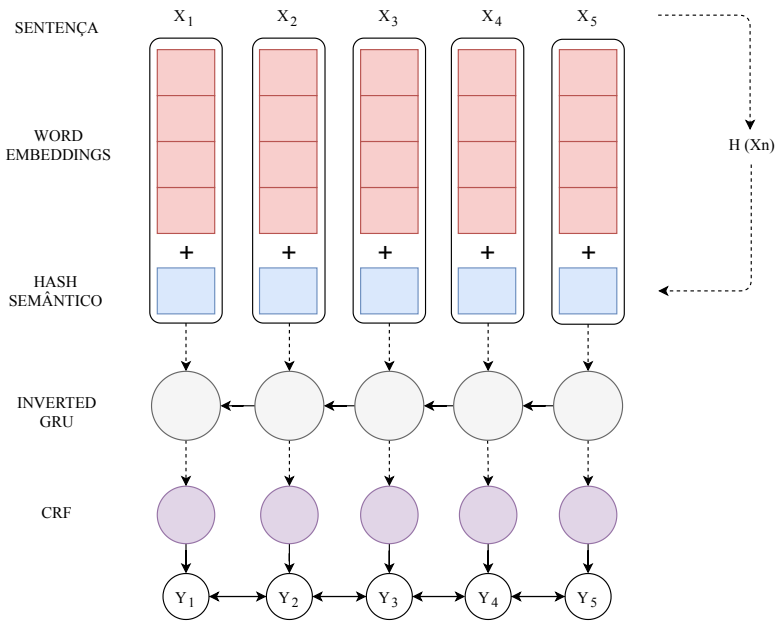


Fonte: Lafferty et al. (2001).

4.6 Modelo Inverted GRU + CRF

A partir da utilização dos componentes listados anteriormente foi possível construir um modelo de rede neural alimentando os vetores de saída da Inverted GRU em uma camada CRF, conforme Figura 4.1.

Figura 4.3: Arquitetura do método proposto.



Fonte: Elaborada pelo autor.

Para exemplificar o processo de classificação, tem-se: Dada uma sentença de entrada, as palavras são tokenizadas e organizadas em uma lista de tokens. Cada token é subdivido em n-gramas a fim de obter sub-tokens. Esses sub-tokens são usados para criar um modelo

de espaço vetorial servindo para extrair recursos para um determinado texto de entrada e adicionando-os ao vetor de características de cada palavra, representado pela função H(Xn).

Em seguida, toda essa informação serve como parâmetro de entrada para a rede neural do tipo *Inverted* GRU. Esse tipo de rede neural introduz uma pequena variação na estrutura da camada GRU única invertendo a ordem da sequência de entrada. Essa variação permite obter bons resultados além da possibilidade de melhorar marginalmente o desempenho de certos tipos de redes neurais (Sutskever et al., 2014).

Após o processamento realizado pela rede neural o classificador CRF é utilizado para classificação final das intenções e manutenção das relações de termos próximos, auxiliando a detecção de sentenças completas, além da possibilidade de previsão de maneira relacionada com os demais termos. Este foi preferido em vez do módulo softmax por possibilitar as informações disponíveis pelos termos vizinhos fazendo com que a classificação apresente um aspecto global.

4.7 Considerações Finais do Capítulo

Nesse capítulo, foi apresentada a arquitetura da rede profunda para esta pesquisa. O método é baseado nos fundamentos apresentados no Capítulo 2 e, nos trabalhos relacionados, apresentados no Capítulo 3.

Em resumo, as partes principais do método proposto consistem na fase de definição dos pesos para as palavras presentes em cada sentença, pela especificidade do cenário, e treinamento do modelo (*Inverted* GRU + CRF). As demais etapas consistem em preparar os dados e, após a execução dos testes, avaliar os resultados obtidos.

A preferência em utilizar redes GRU se dá pela simplicidade da implementação e menor necessidade de poder computacional em comparação a outros tipos de redes neurais. O CRF auxilia na etapa final de classificação, acrescentando a probabilidade dos termos vizinhos no cálculo e mantendo um registro das possíveis sequências de marcação.

Capítulo 5

Experimentos e Resultados

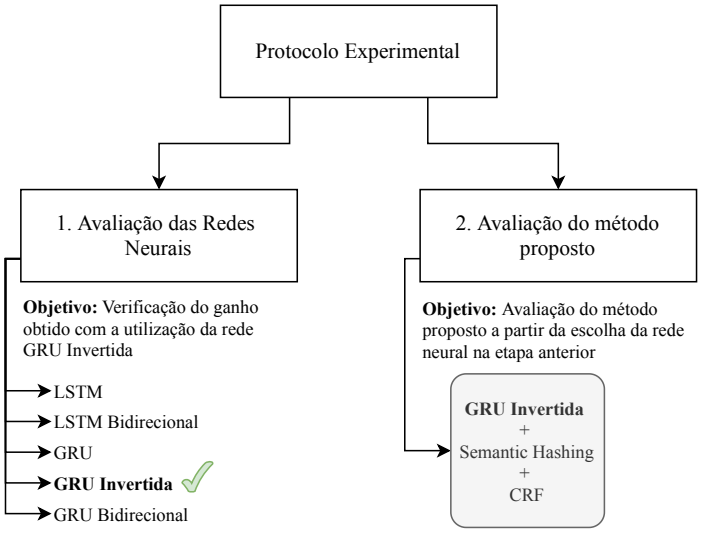
Este capítulo descreve os experimentos realizados e resultados obtidos. Primeiramente é descrito o Protocolo Experimental adotado durante processo de avaliação de diferentes arquiteturas de redes neurais (Seção 5.1), as bases de dados utilizadas (Seção 5.2), as métricas de avaliação (Seção 5.3) bem como o pré-processamento adotado (Seção 5.4). A partir da Seção 5.5 são listados os experimentos realizados para cada tipo de rede neural e, os resultados obtidos levando em consideração o método desenvolvido.

5.1 Protocolo Experimental

A Figura 5.1 mostra as atividades realizadas durante cada uma das etapas definidas no Protocolo Experimental que teve por finalidade separar a construção do método em dois momentos: 1) Avaliação das Redes Neurais e, 2) Avaliação do método proposto.

Primeiramente, realizamos uma avaliação entre diferentes tipos de redes neurais com a finalidade de comparar o desempenho obtido e possibilidade de ganho nos resultados para o cenário apresentado em especial para o tipo de rede neural definida para construção do método (*Inverted* GRU). A avaliação levou em consideração cinco diferentes arquiteturas (LSTM, Bi LSTM, GRU, Inverted GRU e Bi GRU), todas foram construídas a partir de uma estrutura comum. Mais detalhes sobre o processo de avaliação e os resultados obtidos em cada análise estão descritos nas seções seguintes.

Figura 5.1: Protocolo experimental para definição e execução dos experimentos.



A partir dessa validação foi possível realizar as modificações necessárias e acrescentar o cálculo do valor para o hashing semântico apenas para as entidades com maior peso dentro de cada sentença. Com essa alteração é possível alcançar bons resultados levando em consideração bases de dados pequenas, onde uma rede neural não consegue aprender ou, representar por completo pela quantidade limitada de exemplos.

5.2 Base de Dados

Os modelos de aprendizagem de máquina requerem a utilização de uma base de dados rotulada para realizar as atividades necessárias à classificação. Com essa disponibilidade, é possível treinar o modelo desejado a fim de realizar os testes que possibilitem uma avaliação adequada.

O treinamento supervisionado requer um grande número de dados para que os classificadores sejam capazes de representar o maior número possível de exemplos de ocorrências das entidades em textos. Uma base com tamanho pequeno, por exemplo, não constitui uma quantidade suficiente para treinar. Para solucionar este problema, algumas ações foram tomadas: aumento de dados para dispor de mais exemplos, similares aos já disponíveis, bem como utilizar a adição de um valor hashing junto à rede neural, conforme procedimento adotado por Shridhar and Sahu (2018). Para a realização dos experimentos descritos neste capitulo, as bases de dados foram divididas em 70% para treino e 30% para testes.

5.2.1 Ask Ubuntu, Chatbot e Web Application Corpus

Para a realização dos testes foram considerados três base de dados (Ask Ubuntu Corpus, Chatbot Corpus, Web Application Corpus)[1]. Elas são constituídas por diversas sentenças voltadas para o cenário de conversação por meio de chatbots no idioma inglês. Originalmente, são compostas por 162, 206 e 89 sentenças, respectivamente e possuem a seguinte estrutura: tag, sentença, resposta e contexto.

Todas as sentenças foram consideras em formato de texto. As sentenças com termos similares não foram descartadas para que fosse possível determinar as palavras diferentes que faziam referência a um mesmo item e, diferentes maneiras de se reportar uma mesma situação por usuários diferentes.

Tabela 5.1: Quantidade de intenções por cada base de dados apresentada.

Corpus	**Intenções**	**Quantidade**	**Treino**	**Teste**
Ask Ubuntu	Software Recommendation	57	39	18
	Shutdown Computer	28	19	9
	Make Update	47	32	15
	Setup Printer	23	16	7
	None	7	4	3
Chatbot	FindConnection	128	89	39
	DepartureTime	78	54	24
Web Application	Find Alternative	23	16	7
	Delete Account	17	11	6
	Sync Accounts	9	6	3
	Export Data	6	4	2
	Change Password	10	7	3
	Filter Spam	20	14	6
	None	4	2	2
	Total	**457**	313	144

[1]https://github.com/kumar-shridhar/Know-Your-Intent

5.2.2 ATIS

A base ATIS[2] (Air Travel Information System) (Dahl et al., 1994) é dedicada a fornecer informações sobre voos comerciais. A representação semântica usada é baseada em quadros. Uma das atividades proporcionadas por esses dados é a possibilidade de encontrar as entidades e preencher os *slots* correspondentes alinhando-as nas categorias correspondentes.

Sua rotulação consiste em três valores para cada palavra, a própria palavra, uma classe à qual a palavra pode pertencer e o rótulo de destino. Existem 26 classes de palavras no ATIS e elas representam clusters como country_name, airport_name, etc. Cada palavra utilizada no conjunto de dados que pertence a um cluster é substituída pelo nome do cluster.

Tabela 5.2: Quantidade de intenções por cada tipo da base ATIS.

Intenções	**Quantidade**	**Treino**	**Teste**
Abbreviation	896	627	269
Airline	762	533	229
Airfare	553	387	166
City	613	429	184
Departure Time	437	305	132
Find Location	457	319	138
Flight	366	256	110
Ground Service	289	202	87
None	605	423	182
Total	4978	3481	1497

O rótulo de destino é previsto usando o conjunto de palavras ou, classes de palavras, quando disponíveis. As classes também são utilizadas para modelar definir as classes relevantes no histórico de diálogo. O conjunto de treinamento consiste de 4978 sentenças selecionadas a partir dos dados de treinamento de Classe A (independente de contexto) nos corpora ATIS-2 e ATIS-3, enquanto o conjunto de testes ATIS contém os conjuntos de dados ATIS-3 NOV93 e DEC94.

[2]https://github.com/howl-anderson/ATIS_dataset

5.2.3 Emails de suporte ao usuário

Outra base de dados utilizada foi construída a partir da coleta de emails enviados por usuários da UFAM para o Centro de Tecnologia da Infomação e Comunicação (CTIC) que continham respostas fornecidas pelos servidores das Coordenações de Serviços e Desenvolvimento referentes a diversos assuntos, tais como: dúvidas de acesso a sistemas, utilização de módulos do ecampus, solicitação de novos usuários e senhas, solicitação de serviços e treinamentos, solicitação de relatórios estudantis, entre outros.

Os emails utilizados foram coletados durante o período de 01/01/2018 até 31/03/2019 obtendo um total de 11249 sentenças divididas em 9 grupos de intenções (Tabela 5.3). Esses grupos foram categorizados de acordo com o catálogo de serviços disponibilizado pelo CTIC[3] visando avaliar o desempenho geral do método com sentenças obtidas a partir de um ambiente real de interação entre usuários.

Tabela 5.3: Quantidade de intenções por cada tipo da base ATIS.

Intenções	**Quantidade**	**Treino**	**Teste**
Abertura de chamados	1293	862	431
Acesso ao ecampus	1164	776	388
Acesso à rede cafe	1212	808	404
Atribuição de perfis	1130	753	377
Criação de email	1293	862	431
Configuração de proxy	1213	808	405
Contato	1345	896	449
Dúvidas gerais	1347	898	449
Geração de relatórios	1252	834	418
Total	11249	7497	3752

As sentenças foram selecionadas e agrupadas de acordo com a categoria correspondente por um analista e, consequentemente, eram revisadas por outro analista a fim de garantir que estava de acordo com a classificaçâo ou, reclassificar, caso necessário. A construção dessa base de dados favorece à adaptação do modelo para outros cenários de aplicação, possibilitando um aprendizado maior devido à variabilidade de exemplos disponíveis.

[3]https://ctic.ufam.edu.br

5.2.4 Baselines de Comparação

Para a validação do método proposto (*Inverted GRU* + CRF com adição do valor de hash semântico) foram considerados três trabalhos disponíveis na literatura, descritos no Capítulo 3. São eles: Sun et al. (2016), Khatua et al. (2017) e Shridhar and Sahu (2018). Os experimentos foram realizados com a finalidade de verificar os resultados obtidos diante de diferentes conjunto de dados.

Deseja-se também observar com este experimento se, em bases com características como idioma, tamanho e quantidade de sentenças distintas, ocorre algum tipo de variação de desempenho no método.

São definidos os mesmos padrões de entrada dos dados, para a comparação de cada um dos trabalhos relacionados. A entrada é constituída pelo conjunto de características extraídas de cada sentença acrescida do valor obtido (*hash semântico*) a partir das entidades de maior peso. As *features* de contexto, que adicionam características dos termos antecessores e sucessores, não foram consideradas.

Tabela 5.4: Dados quantitativos das bases após o processo de aumento de dados.

Trabalho	**Método**	**Precisão**	**Revocação**	**F1**
Sun et al. (2016)	CRF + CNN	0.9099	0.9115	0.9106
Khatua et al. (2017)	LSTM	0.9124	0.9382	0.9250
Shridhar and Sahu (2018)	Hash Semântico	0.9173	0.9266	0.9201

5.3 Métricas de Avaliação

Na literatura existem diversas formas de se avaliar o desempenho de modelos classificadores. Em aprendizagem de máquina, as métricas mais conhecidas são: Precisão, Revocação e F1 Score (Baeza-Yates et al., 2011). Essas métricas serão definidas a seguir.

A precisão calcula a quantidade de respostas corretas em relação ao total de respostas retornadas, representada pela fórmula da Equação 5.1. A Revocação é a quantidade de respostas corretas em relação ao total de respostas esperadas ou relevantes, cuja fórmula está representada na Equação 5.2. A medida F1 Score é definida como a média harmônica da precisão e da revocação e indica uma relação de compromisso entre essas duas métricas, cuja fórmula está definida na Equação 5.3.

$$Precisão = \frac{Verdadeiros\ \ Positivos\ (TP)}{Verdadeiros\ \ Positivos\ (TP) + Falsos\ Positivos\ (FP)} \tag{5.1}$$

$$Revocacão = \frac{Verdadeiros\ \ Positivos\ (TP)}{Verdadeiros\ \ Positivos\ (TP) + Falsos\ \ Negativos\ (FN)} \tag{5.2}$$

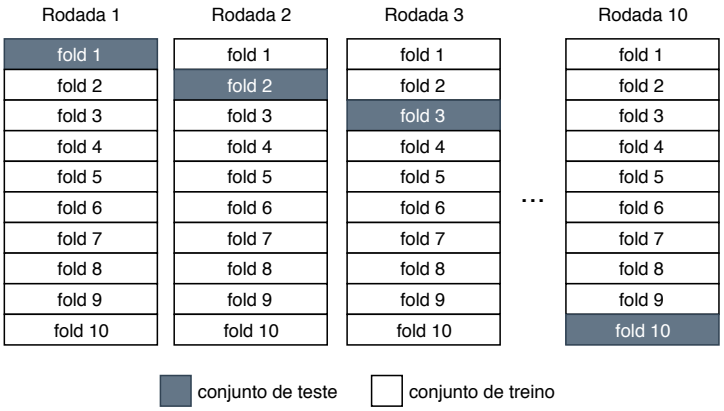
$$F1Score = \frac{2\ x\ (Precisão\ \ x\ \ Revocacão)}{Precisão + Revocacão} \tag{5.3}$$

Essas medidas são bastante utilizadas na literatura para mensurar a qualidade e abrangência de aplicações que envolvam recuperação de texto, análise de dados, reconhecimento de entidades e intenções, por exemplo.

Para demonstrar os resultados das medidas de precisão, revocação e F1, as bases de dados foram divididas em dois conjuntos: treino e teste. A definição destas porções de dados é baseada no método de avaliação conhecida na literatura como holdout (Kohavi et al., 1995). Esse método particiona os dados em dois conjuntos mutuamente exclusivos. Em sua utilização é comum designar 2/3 dos dados para treino e 1/3 para teste.

Foi usado ainda o método de validação cruzada k-fold (Kohavi et al., 1995). Neste método de validação, a base de dados D é dividida em k subconjuntos de dados (D1, D2, ..., Dk), de aproximadamente do mesmo tamanho e mutuamente exclusivos. O classificador é treinado e testado k vezes, alternando os subconjuntos k, que é utilizado para testes, enquanto os demais subconjuntos (k-1) treinam o classificador.

Figura 5.2: Ilustração do método k-fold.



O método k-fold foi empregado para validar os resultados do método Inverted GRU +

CRF junto às bases de dados descritas nas seções seguintes. A validação cruzada utilizada, consta na divisão do corpus em dez subconjuntos, permitindo que todos os subconjuntos sejam testados durante as etapas, informando a eficácia geral por meio das medidas de precisão, revocação e F1.

5.4 Aumento de dados

O processo de aumento dos dados foi realizado na base AskUbuntu Corpus, pois, apresentava a menor quantidade de dados. Esse processo auxilia na criação automatizada de sentenças relacionadas às categorias já existentes, eliminando a necessidade de criação manual de sentenças e diminuindo a possibilidade de construção de sentenças com erros.

Nesta etapa, foram escolhidas as intenções que continham poucos exemplos para treino para serem aumentadas. Elas tiveram os substantivos e verbos substituídos por sinônimos a partir do dicionário Wordnet[4], onde todo sinônimo de uma palavra ou frase é classificado pela proximidade semântica do significado mais frequente. Com isso, foi possível obter um ganho na precisão de 2 a 3%.

Sentença original	Please recommend a hex editor for shell
Sentenças geradas pelo dicionário	Please **suggest** a hex editor for shell
	Please **propose** a hex editor for shell
	Please recommend a **decimal** editor for shell
	Please recommend a **text** editor for shell

Essa base de dados é formada por sentenças utilizadas em chatbots, dessa maneira, há maiores chances de conter erros ortográficos e, isso precisa ser levado em consideração. Uma maneira simples de corrigir os erros ortográficos é encontrar a distância de *Levensthein* e mapear a palavra para o vizinho mais próximo.

5.5 Resultados

Os resultados obtidos foram divididos em dois momentos. Nesta seção (5.5) são descritos todos os experimentos realizados a fim de identificar qual é a rede neural que apresenta

[4]https://wordnet.princeton.edu/

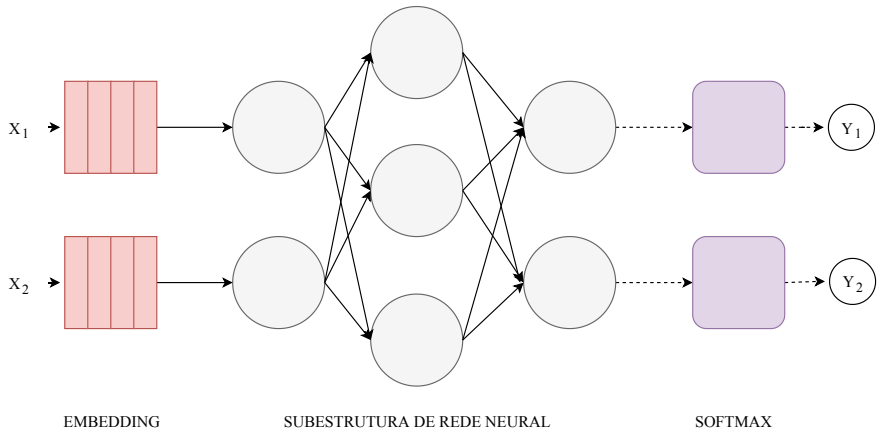
os melhores resultados, com relação ao tempo e taxa de precisão e, a Seção 5.6 contém os resultados obtidos após a definição da arquitetura de rede neural utilizada no método definido.

5.5.1 Avaliação dos tipos de redes neurais

Cada rede avaliada foi construída partindo de uma arquitetura comum (Figura 5.2), onde 'x' representa a sentença de entrada já tokenizada. Neste caso, a tokenização é feita por palavra e, cada palavra é traduzida em um número inteiro que a identifica exclusivamente em meio ao vocabulário de treinamento. Então a sequência de inteiros é convertida em uma sequência de vetores reais de tamanho igual a 128 por camada de incorporação.

Essa sequência de vetores é enviada para a subestrutura da rede neural. Por fim, a saída da rede neural é enviada para uma camada softmax, definindo um vetor de peso para todas as conexões.

Figura 5.3: Estrutura comum para as redes neurais avaliadas.



Fonte: Elaborada pelo autor.

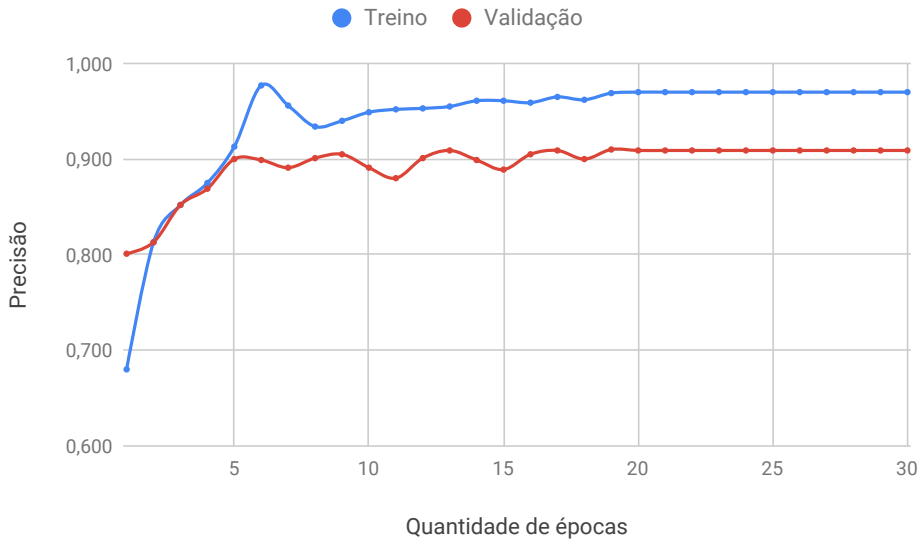
O algoritmo de otimização Adam (Kingma and Ba, 2014) foi utilizado para treinamento das redes neurais, com o seu valor padrão para a taxa de aprendizado em 0,001, uma vez que é considerado computacionalmente eficiente e vem mostrando bons resultados no campo da aprendizagem profunda. Ele cria variáveis adicionais chamadas de *"slots"* para armazenar valores de predição e acumuladores que precisam ser inicializados antes do treinamendo do modelo.

5.5.2 LSTM

A primeira estrutura considerada foi uma rede LSTM simples (camada única). As redes LSTM são um tipo especial de Rede Neural Recorrente (RNN) introduzidas por Hochreiter and Schmidhuber (1997). Esse tipo de rede neural tem a capacidade de remover ou adicionar informações em uma unidade a partir do uso de portões de ativação (*input, output e forget*) e, o fluxo de informações caminha em um único sentido.

A figura abaixo mostra a evolução no valor da precisão à medida que o número de épocas aumenta. Nessa fase o valor de precisão apresenta uma variabilidade muito grande devido ao tempo para aprender efetivamente a classificar uma sentença. No entanto, ao longo das épocas consegue aprender de maneira satisfatória e chega a alcançar uma acurácia máxima de 90,5% no conjunto de validação.

Figura 5.4: Desempenho da rede LSTM simples (camada única).



Fonte: Elaborada pelo autor.

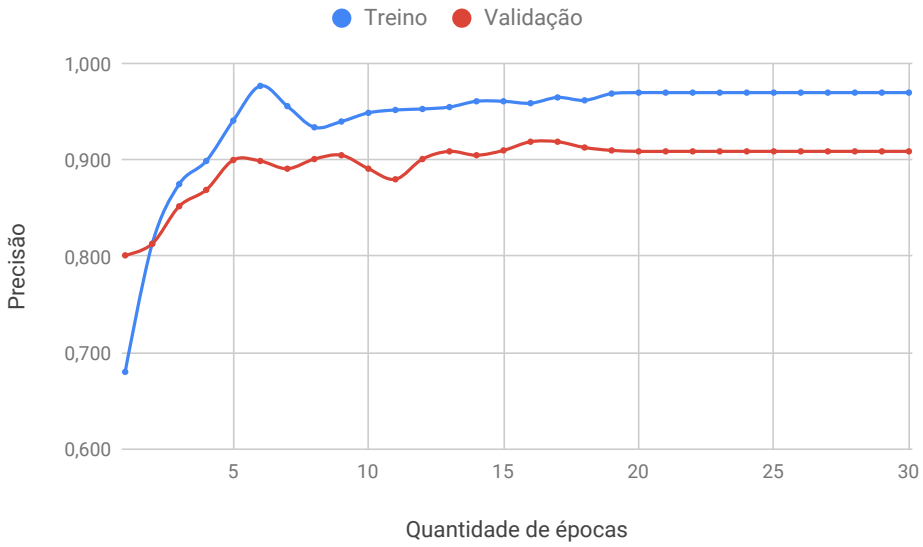
Apesar do bom resultado apresentado pela rede neural do tipo LSTM, não foi suficiente para minimamente se igualar ao resultado obtido com o conjunto de testes, um dos motivos pode estar associado ao fato de que redes desse tipo necessitam uma enorme capacidade computacional além de um corpus suficientemente grande para que seja possível o aprendizado por conta da rede.

5.5.3 LSTM Bidirecional

Outra variação de rede neural avaliada foi LSTM Bidirecional, desenvolvida por Schuster and Paliwal (1997) para treinar a rede usando sequências de dados de entrada com informações do passado e do futuro. Nesse tipo de rede, há duas camadas conectadas para processar os dados de entrada. Cada camada executa as operações usando uma direção própria de modo que uma camada realiza as operações seguindo a mesma direção da sequência de dados e a outra aplica suas operações na direção inversa.

Os resultados obtidos pela rede LSTM Bidirecional não apresentaram um desempenho tão superior em relação ao uso da LSTM simples. O desempenho total da rede chega a ser parecido, porém, a taxa de aprendizagem se dá um pouco mais rápida na etapa de treino e consegue alcançar uma precisão de 91,3% no conjunto de validação.

Figura 5.5: Desempenho da rede LSTM bidirecional.



Fonte: Elaborada pelo autor.

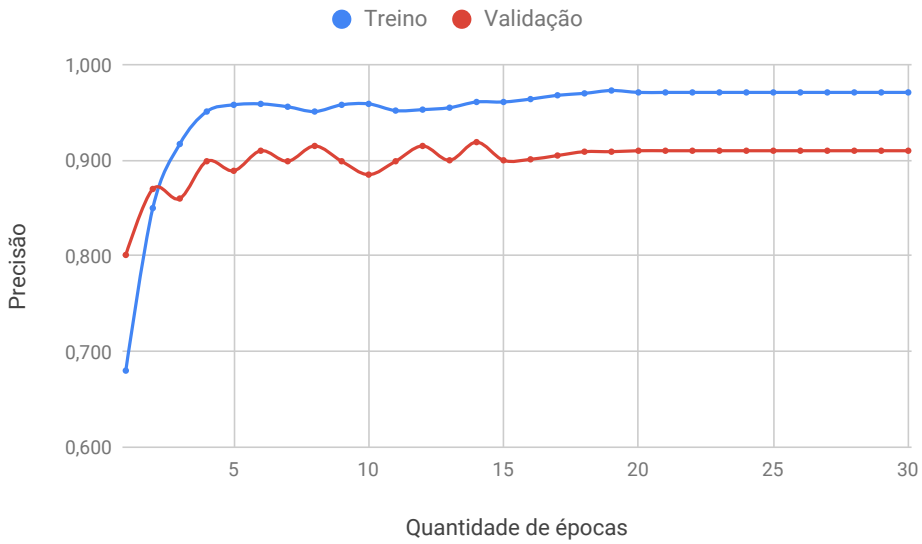
Um dos motivos para o desempenho obtido da rede LSTM Bidirecional pode estar relacionado ao tamanho da base de dados utilizada. Além disso, trabalhos relatam que o uso desse tipo de rede apresenta ótimos resultados para alguns cenários (Graves and Schmidhuber, 2005): classificação de fonemas, legenda de imagens e reconhecimento de fala e escrita e, para outros não, sendo necessário realizar ajustes para aumentar o desempenho.

5.5.4 LSTM GRU

O próximo tipo de rede neural avaliada foi a *Gated Recorrent Unit* (GRU) proposta por (Chung et al., 2014). O funcionamento de uma rede GRU é semelhante a uma rede LSTM simples, ambas processam o fluxo de informação de uma unidade a partir de portões de ativação, no entanto redes GRU apresentam apenas dois portões: *update* e *reset*. Essa redução de processamento possibilita uma perfomance mais eficiente na fase de treino conforme apresentado em (Chung et al., 2014).

Como é mostrado na Figura 4.4, a rede GRU executa aproximadamente da mesma maneira que a estrutura de camada LSTM única. Ele exibe o mesmo comportamento durante a fase de treinamento e chega a atingir uma precisão máxima de 91,8% no conjunto de validação.

Figura 5.6: Desempenho da rede LSTM-GRU.



Fonte: Elaborada pelo autor.

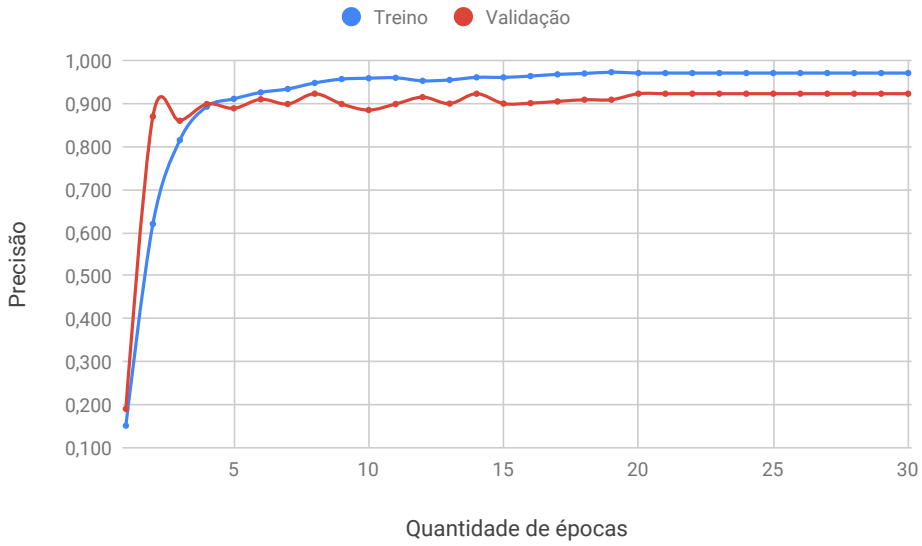
Apesar de apresentar um comportamento similar, os resultados obtidos a partir dos experimentos realizados com a rede LSTM GRU se mostrou ligeiramente melhor em relação à rede com LSTM única. Um dos motivos pode estar associado à quantidade reduzidade de portões necessários para realizar a transferência de aprendizado entre as unidades.

5.5.5 Inverted GRU

As redes neurais do tipo *Inverted GRU* (Chung et al., 2015) introduzem uma variação nas redes GRU de camada única, invertendo a ordem da sequência de entrada. Isso ocorre porque inverter a sequência de entrada mostrou ser uma heurística simples que pode melhorar marginalmente o desempenho de certas redes, particularmente no subcampo de tradução automática (Sutskever et al., 2014).

Essa modificação permite que a rede neural faça uso de informações futuras sem efetivamente utilizar mais processamento para isso, algo que acontece com frequência em outros tipos de redes, devido a transferência constante de informação entre as unidades e necessidade de manter uma memória sobre as mudanças de estados. Além disso,

Figura 5.7: Desempenho da rede GRU invertida.



Fonte: Elaborada pelo autor.

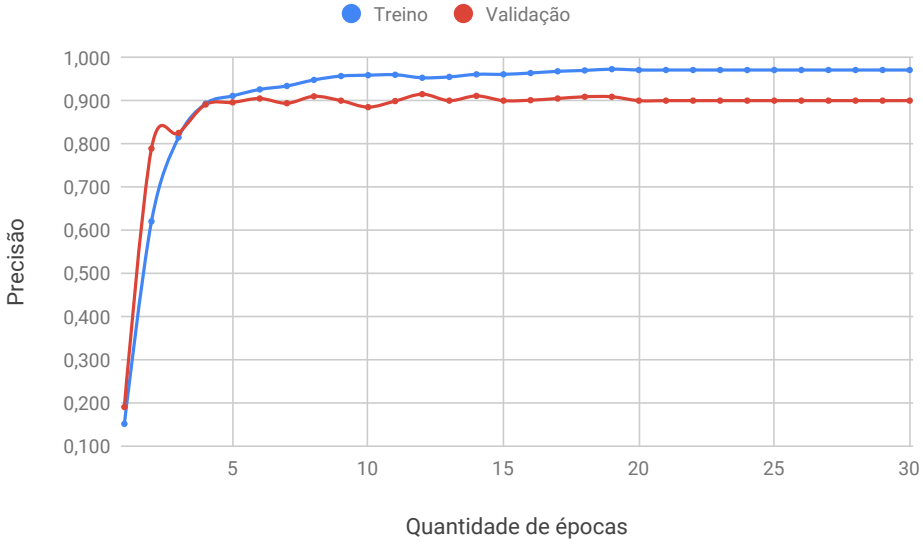
A evolução de sua precisão mostra claramente que esta estrutura é muito mais estável do que as anteriores durante o treinamento, pois obteve uma perda menor de precisão ao longo das épocas. Com essa estrutura de rede neural foi possível obter uma precisão de 92,3% no conjunto de validação.

5.5.6 GRU Bidirecional

Outra maneira de se melhorar a precisão é a de adicionar mais uma camada de rede GRU empilhada sobre outra rede similar, parecida com a ideia de se utilizar redes lstm bidirecionais. Assim pode-se considerar uma possibilidade a mais de se obter informações sobre a sequência de entrada.

A modelagem das informações em ambas as direções pode ser feita pela implementação bidirecionalmente estruturada dentro da arquitetura de uma rede neural recorrente (Schuster and Paliwal, 1997), ou ainda, com duas redes neurais recorrentes trabalhando com direções opostas podendo ser combinadas para alcançar o mesmo objetivo (Yang et al., 2016).

Figura 5.8: Desempenho da rede GRU Bidirecional.



Fonte: Elaborada pelo autor.

Esse tipo de rede neural não chegou a apresentar um desempenho tão melhor que as avaliações realizadas nos outros tipos de rede descritas anteriormente. Isso pode ter ocorrido devido ao fato de que treinar uma camada extra de GRU é muito maior do que o ganho de informações que ela realmente fornece. A precisão máxima que a rede alcançou foi de 89,9% no conjunto de validação.

5.5.7 Estrutura Final da Rede Neural

Tendo explorado várias estruturas de redes neurais candidatas, a melhor delas pode ser escolhida. A Tabela 5.5 resume o desempenho de cada tipo de rede, listando a precisão, revocação, F1, perda e tempo de treinamento em segundos obtidos em cada um dos conjuntos de dados utilizados.

Tabela 5.5: Resultados obtidos a partir das avaliações.

Rede Neural	Precisão (%)	Revocação	F1	Perda	Tempo (s)
Ask Ubuntu, Chatbot, Web Application					
LSTM simples	90.66	90.01	90.33	0.4291	66
LSTM Bidirecional	90.15	91.61	90.87	0.4543	68
GRU simples	91.99	94.13	93.04	0.4608	47
GRU invertida	**92.13**	**94.28**	**93.19**	**0.4107**	**48**
GRU Bidirecional	90.77	93.52	92.63	0.4750	52
ATIS					
LSTM simples	90.5	95.26	92.58	0.4579	116
LSTM Bidirecional	91.3	94.47	92.85	0.5103	189
GRU simples	91.8	96.01	93.84	0.4697	141
GRU invertida	**92.3**	**95.85**	**94.04**	**0.4323**	**147**
GRU Bidirecional	89.9	95.86	92.35	0.5212	172

Com base nos resultados apresentados, a arquitetura de rede neural escolhida a ser utilizada para o classificador de intenção foi a rede do tipo GRU com a sequência de entrada invertida (*Inverted GRU*), pois, apresentou os melhores resultados dentre todas as estruturas de redes neurais analisadas.

Essa avaliação entre diferentes tipos de redes neurais se deu pela necessidade de encontrar uma arquitetura que se comportasse melhor diante da utilização de bases de dados pequenas, pois, um dos problemas diagnosticados durante a etapa de análise bibliográfica foi justamente o tamanho incompatível das bases de dados diante à complexidade da arquitetura de rede neural utilizada.

5.6 Avaliação geral do método

A Tabela 5.7 apresenta o resultado dos experimentos (por meio das medidas de precisão, revocação e F1) realizados com cada um dos trabalhos utilizados como baseline, sendo possível ainda observar a diferença entre as medidas gerais de cada classificador. Os *datasets* referem-se às bases de dados 1) Ask Ubuntu, Chatbot e Web Application, 2) ATIS e, 3) Emails de suporte ao usuário, descritas na seção 5.1.1.

Tabela 5.6: Avaliação geral dos métodos (10-fold cross validation).

Trabalho	**Método**	**Dataset**	**Prec.**	**Revoc.**	**F1**
Sun et al. (2016)	CRF + CNN	1	0.8913	0.8722	0.8816
		2	0.9052	0.8894	0.8972
		3	0.9077	0.8993	0.9034
Khatua et al. (2017)	LSTM	1	0.9012	0.8852	0.8931
		2	0.9246	0.8619	0.8921
		3	0.9157	0.9015	0.9085
Shridhar et al. (2018)	Hash semântico	1	0.9261	0.9125	0.9192
		2	0.9107	0.8826	0.8964
		3	0.9215	0.9089	0.9151
Proposta	Inverted GRU + CRF	1	0.9453	0.9333	0.9392
		2	0.9382	0.9033	0.9204
		3	0.9349	0.9343	0.9345

Os valores mostrados por este experimento revelam que, de modo geral, os métodos que utilizam adição de hash semântico consegue superar os demais classificadores nas 3 três diferentes bases de dados. Além disso, utilizando o valor adicional apenas para as entidades classificadas como mais importantes de cada sentença é possível obter um ganho de ainda maior se comparado aos demais.

Na única base de dado em português utilizada nos experimentos (3 - Emails), o comportamento do método Inverted GRU + hash semântico também demonstrou melhor acurária, indicando a vantagem em se utilizar apenas algumas informações de uma sentença e nâo toda ela como apresenta por Shridhar and Sahu (2018). Dependendo do tamanho da sentença algumas entidades ficam irrelevantes para a definição da intenção.

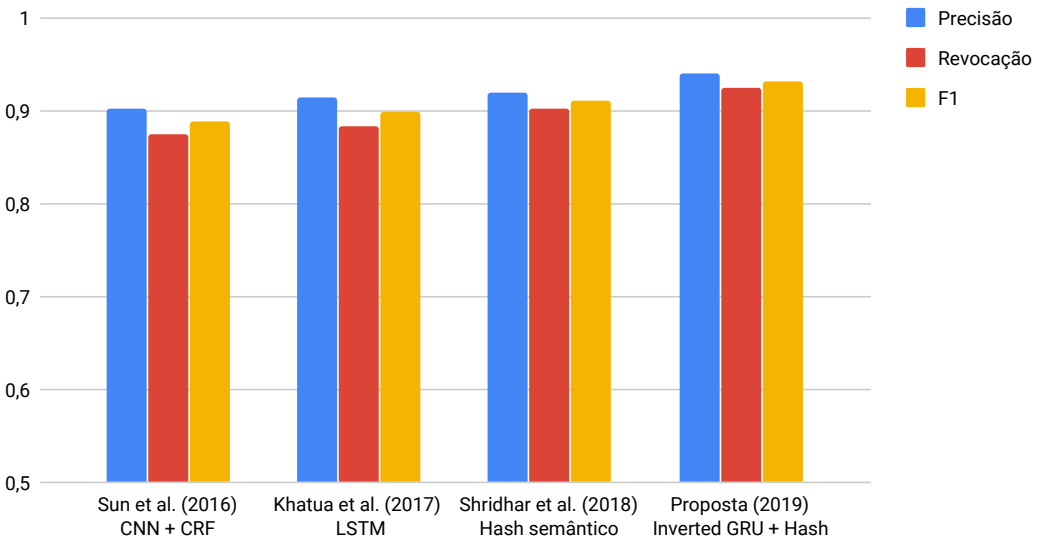
Com relação à base de dados 2 (ATIS) foram percebidos os menores valores de precisão, revocação e F1 para todos os métodos apresentados, além da maior diferença proporcional entre precisão e revocação, variando de 2,81% até 6,27%. Um dos motivos pode estar relacionado à quantidade de sentenças que não pertenciam a nenhuma das intenções.

Para uma visualização do desempenho geral de todos os métodos verificados foi calculada a média para os valores de precisão, revocação e F1 obtido por cada trabalho em cada uma das bases de dados utilizadas (1, 2 e 3). Esses resultados estão listados na Tabela 5.8 e na Figura 5.11.

Tabela 5.7: Média geral obtida por cada trabalho.

Trabalho	**Método**	**Prec.**	**Revoc.**	**F1**
Sun et al. (2016)	CNN + CRF	0.9014	0.8744	0.8876
Khatua et al. (2017)	LSTM	0.9138	0.8828	0.8979
Shridhar et al. (2018)	Hash semântico	0.9189	0.9013	0.9100
Proposta (2019)	Inverted GRU + Hash	**0.9395**	**0.9237**	**0.9314**

Figura 5.9: Gráfico da média dos resultados obtidos por cada trabalho.



Fonte: Elaborada pelo autor.

A partir do gráfico obtido pode-se verificar que os métodos que fizeram uso de algoritmos de *deep learning* ou, sem o acréscimo de alguma informação extra, não conseguiram

obter resultados melhores quando expostos a uma base de dados pequena. Essa limitação pode estar relacionada à quantidade de informação necessária que o algoritmo necessita para efetivamente realizar a aprendizagem como um todo. Por outro lado, os métodos que realizaram a adição de um valor extra durante a fase de treino obtiveram resultados melhores com uma diferença de até 3%.

Para ajudar na visualização do comportamento do método proposto bem como a sua qualidade como um todo foi elaborada a matriz de confusão (Figura 5.4) obtida a partir da execução no conjunto de teste da base de dados 3 - Emails de suporte ao usuário.

Figura 5.10: Matriz de confusão obtida a partir do método proposto.

	1	2	3	4	5	6	7	8	9	FP	Precisão
1. Abertura de chamados	608	4	3	7	5	2	8	9	1	39	93.97
2. Acesso ao ecampus	1	397	2	7	3	1	8	8	0	30	92.97
3. Acesso à rede cafe	1	2	295	4	2	2	5	5	0	21	93.35
4. Atribuição de perfis	4	5	3	332	4	1	6	4	0	25	93.04
5. Criação de emails	5	4	2	4	293	5	4	5	2	31	90.43
6. Configuração de proxy	1	3	3	2	3	322	2	3	0	17	94.99
7. Contato	10	3	4	4	5	2	403	5	2	35	92.01
8. Dúvidas gerais	11	8	5	0	6	4	6	473	2	42	91.86
9. Geração de relatório	0	1	0	0	0	0	1	2	382	4	98.96
FN	33	30	22	28	28	17	38	41	7		
Revocação	94.85	92.97	93.06	92.27	91.28	94.99	91.38	92.04	98.20		

A partir da visualização da matriz de confusão é possível perceber que algumas intenções (1, 6 e 9) conseguem ser classificadas com um percentual bem alto, acima de 94%, outras parecem estar mais relacionadas entre si, como foi o caso dos pares (1:5, 1:7, 1:8 e, 2:8). Isso pode ter ocorrido devido a dois fatores: 1) as informações das primeiras sentenças não serem sufucientes para distinguir com precisão as duas classes ou, 2) as amostras no conjunto de dados podem não ter sido corretamente rotuladas.

O cenário utilizado caracteriza-se como classificação multiclasse. Esse tipo contém mais de duas classes (intenções) onde a entrada fornecida (sentença) é associada a apenas

uma delas. Durante a execução dos experimentos foi possível perceber a existência de uma classificação multirótulos, na qual uma sentença informada pelo usuário tinha condições de pertencer a mais de uma intenção válida.

Para problemas de classificação do tipo multiclasse é interessante calcular outros valores além de precisão, revocação e F1 (Sokolova and Lapalme, 2009). A Tabela 5.8 lista os valores das precisões, revocações e F1, macro (M) e micro(μ). Vale ressaltar que os valores micro (μ) tem uma importância maior quando há casos de desequilíbrio entre as classes.

Tabela 5.8: Avaliação geral da perfomance com valores M e μ.

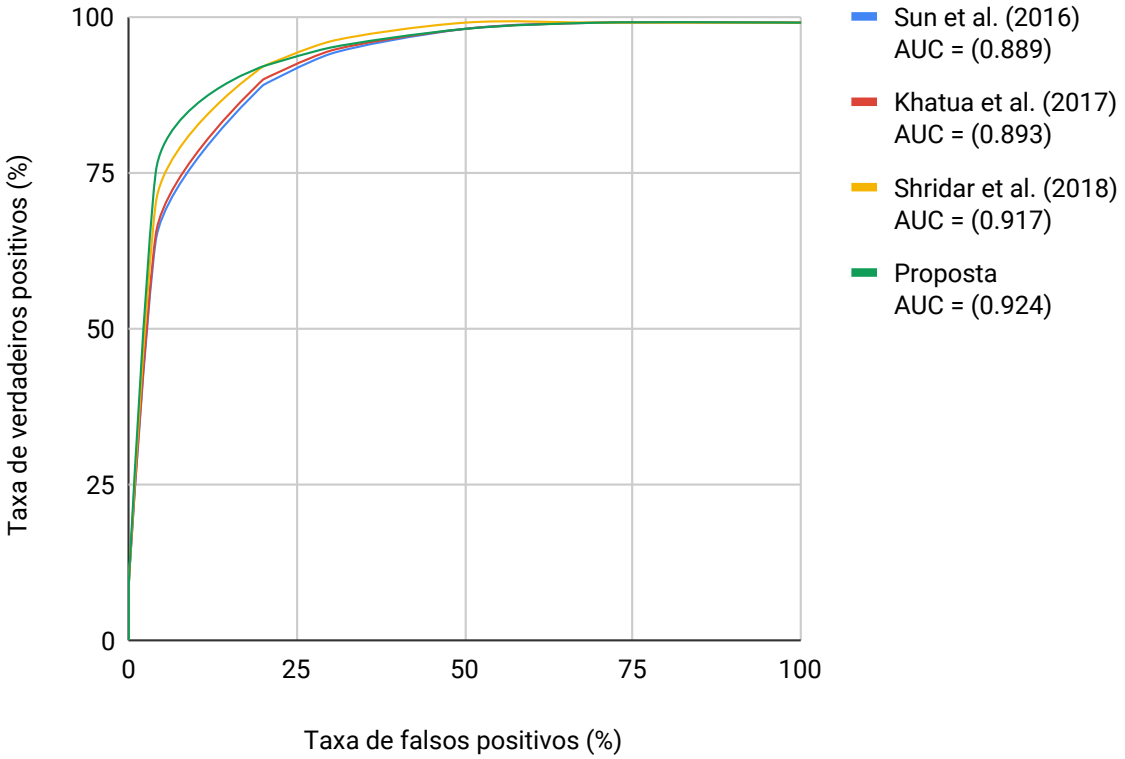
Dataset	**Métodos**	**Macro (M)**			**Micro (μ)**		
		Prec.	**Revoc.**	**F1**	**Prec.**	**Revoc.**	**F1**
Ask Ubuntu	CNN+CRF	0.8913	0.8517	0.8710	0.9028	0.8829	0.8927
	LSTM	0.9052	0.8724	0.8884	0.8933	0.8912	0.8922
	Hash Sem.	0.9077	0.8993	0.9034	0.9282	0.9143	0.9211
	Inv. GRU.	0.9453	0.9333	0.9392	0.9623	0.9501	0.9561
ATIS	CNN+CRF	0.9015	0.8913	0.8517	0.9047	0.8915	0.8980
	LSTM	0.9052	0.8724	0.8884	0.9168	0.8993	0.9079
	Hash Sem.	0.9077	0.8993	0.9034	0.9107	0.8924	0.9014
	Inv. GRU.	0.9382	0.9033	0.9204	0.9176	0.9082	0.9128
Emails	CNN+CRF	0.9015	0.8913	0.8517	0.9063	0.8911	0.8986
	LSTM	0.9052	0.8724	0.8884	0.9135	0.9001	0.9067
	Hash Sem.	0.9077	0.8993	0.9034	0.9201	0.9093	0.9146
	Inv. GRU.	**0.9345**	**0.9351**	**0.9347**	**0.9347**	**0.9347**	**0.9347**

Alguns valores de precisão, revocação e F1 permaneceram semelhantes aos que foram calculados anteriormente, essa relação mostra estabalidade em cada método. Além disso, a diferença entre as métricas mostra que ao fazer uso de algum valor que favoreça o aprendizado é possível obter resultados melhores quando testados em bases pequenas.

Outra consideração importante a ser fazer quando classificadores são avaliados é com relação ao cálculo das chamadas curvas ROC (*Receiver Operating Characteristic*) (Spackman, 1989). Elas são baseadas na curva da taxa de verdadeiro positivo (TVP) versus a taxa de falso positivo (TFP). Como o próprio nome sugere, a TFP é a taxa de instâncias

negativas que são, incorretamente, classificadas como positivas. Ela é igual a 1 menos a taxa de verdadeiro negativo (TVN), que é a taxa de instâncias negativas que são, corretamente, classificadas como positivo.

Figura 5.11: Plotagem das Curvas ROC para cada um dos trabalhos avaliados.



Fonte: Elaborada pelo autor.

Para este trabalho, com características de classificação multiclasse, o cálculo da curva ROC foi um pouco prejudicado, pois, uma das principais desvantagens é a sua limitação para apenas duas classes. Como resolução foi adotada a técnica de aproximação um-contra-todos, onde é gerada uma curva ROC para cada classe disponível (n classes) e depois elas são somadas de maneira ponderada levando em consideração a probabilidade de um elemento ser da classe i.

5.7 Considerações Finais do Capítulo

Neste capítulo foi apresentado um estudo avaliativo sobre a tarefa de detecção de intenção voltado para bases de dados pequenas. Os experimentos realizados mostraram

que a utilização de um método que faça uso de redes neurais ou, algum outro algoritmo de aprendizagem profunda pode não obter resultados significativos devido à necessidade de informação exigida para o aprendizado adequado.

Ao fazer uso de redes neurais, assumi-se que ela própria fará todo o processamento para garantir a aprendizagem do modelo a fim de obter uma correta classificação e de acordo com o cenário utilizado. Contudo, podemos aumentar esse ganho de aprendizagem das redes neurais ao acrescentar valores que proporcionem aquisição de informações mais adequadas e relevantes para o conjunto considerado.

Capítulo 6

Conclusão

6.1 Considerações Finais

A tarefa de detecção de intenção é um problema fundamental para as aplicações que fazem uso de recuperação de informações, mineração de texto, e-commerce e sistema de recomendação. A intenção do usuário é um conceito que pode fazer a ponte entre a entrada de dados fornecida com o objetivo desejado pelo usuário. Tradicionalmente os trabalhos da literatura têm dado preferência ao entendimento das intenções em sistemas de busca. Contudo, outros cenários de aplicação necessitam de atenção específica, tal como acontece na análise de sentimentos, monitoramento de atividades, chatbots, entre outros.

Neste trabalho os estudos à cerca da atividade de detecção de intenção dos usuários foram revisados para vários domínios, incluindo como esses trabalhos definem e representam a intenção do usuário, bem como a maneira de captura existentes, quais conjuntos de dados eles usam e como modelam seus problemas.

A partir da realização de experimentos em bases de dados, consideradas pequenas e de diferentes domínios foi possível obter resultados que favorecem à utilização de um valor extra, referente às entidades classificadas como mais importantes dentro de uma sentença, que possa fornecer um informação mais representativa na fase de treinamento ajudando durante todo o desenvolvimento da aplicação. Ao utilizar redes neurais assume-se que o aprendizado e toda a informação necessária será assimilada e os relacionamentos construídos de maneira automática, porém, para isso acontecer é preciso um grande número de amostras. Ao adotar um valor extra é possível obter ganhos de desempenho em comparação à utilização de redes neurais para bases de dados pequenas.

O estudo comparativo entre os quatro diferentes métodos, serve para indicar, individualmente e em conjunto, a contribuição de cada um dentro dos cenários apresentados. Dada a diversidade de trabalhos que apresentam reconhecimento de entidades ou detecção de intenção, estes experimentos servem para auxiliar trabalhos futuros, que lidem com a execução dessa tarefa, podendo ajudar a definição de alguns fatores, como a escolha do *corpus* a ser utilizado ou, classificador.

6.2 Trabalhos Futuros

Pesquisas futuras para este estudo desenvolvido, incluem a adição de mais amostras, idiomas e categorias de intenções. Uma vez que é crescente o desenvolvimento de trabalhos que realizam o detecção de intenção e entidades em textos publicados em mídias sociais, como por exemplo o Twitter. A avaliação incluindo um número maior de amostras pode ajudar a encontrar qual pode ser o tamanho adequado de uma base de dados para começar a fazer uso de técnicas que envolvam a utilização de aprendizado profundo.

Outro problema que necessita de maior investigação é o uso e classificação de intenções de categorias minoritárias. Dependendo da quantidade de amostras presentes na base de dados, uma categoria de intenção com poucos exemplos pode afetar negativamente o desempenho geral do método. Avaliar formas de considerá-las e, não simplesmente excluí-las durante os experimentos, pode enriquecer o conjunto de valores obtidos e eficácia geral do modelo.

Experimentos envolvendo a utilização de sentenças obtidas a partir de áudio enviados pelos usuários também é uma linha de pesquisa importante que ainda precisa ser melhor explorada levando em consideração diferentes algoritmos para incorporação de palavras. Além disso, seria interessante combinar os recursos básicos desses algoritmos e o enriquecimento obtido a partir dos n-gramas das palavras e verificar os resultados.

Referências Bibliográficas

Abdul-Kader, S. and Woods, J. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7):72–80.

Al-Zaidy, R., Fung, B. C., Youssef, A. M., and Fortin, F. (2012). Mining criminal networks from unstructured text documents. *Digital Investigation*, 8(3-4):147–160.

Baeza-Yates, R., Ribeiro, B. d. A. N., et al. (2011). *Modern information retrieval*. New York: ACM Press; Harlow, England: Addison-Wesley,.

Bhargava, A., Celikyilmaz, A., and Hakkani-Tür (2013). Easy contextual intent prediction and slot detection. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8337–8341. IEEE.

Bhaskar, J., Sruthi, K., and Nedungadi, P. (2015). Hybrid approach for emotion classification of audio conversation based on text and speech mining. *Procedia Computer Science*, 46:635–643.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Chen, H., Chung, W., and Xu, J. J. (2004). Crime data mining: a general framework and some examples. *computer*.

Cho, K., Van Merriënboer, B., and Gulcehre (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2015). Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, pages 2067–2075.

Cui, L., Huang, S., and Wei (2017). Superagent: A customer service chatbot for e-commerce websites. *Proceedings of ACL 2017, System Demonstrations*, pages 97–102.

Dahl, D. A., Bates, M., and Brown, M. (1994). Expanding the scope of the atis task: The atis-3 corpus. In *Proceedings of the workshop on Human Language Technology*, pages 43–48. Association for Computational Linguistics.

Dale, R. (2016). The return of the chatbots. *Natural Language Engineering*, 22(5):811–817.

Deoras, A., Sutskever, I., and Mikolov, T. (2012). Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)*.

Doshi, A., Morris, B., and Trivedi, M. (2011). On-road prediction of driver's intent with multimodal sensory cues. *IEEE Pervasive Computing*, 10(3):22–34.

Frey, C. B. and Osborne, M. A. (2017). The future of employment: how susceptible are jobs to computerisation? *Technological forecasting and social change*, 114:254–280.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Hakkani-Tür, D., Tür, G., and Celikyilmaz, A. (2016). Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Interspeech*, pages 715–719.

Hashemi, H. B., Asiaee, A., and Kraft, R. (2016). Query intent detection using convolutional neural networks. In *International Conference on Web Search and Data Mining, Workshop on Query Understanding*.

Heck, L. and Hakkani-Tur, D. (2012). Exploiting the semantic web for unsupervised spoken language understanding. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 228–233. IEEE.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hollerit, B., Kroll, M., and Strohmaier, M. (2013). Towards linking buyers and sellers: detecting commercial intent on twitter. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 629–632. ACM.

Hu, J., Wang, G., Lochovsky, F., Sun, J.-t., and Chen, Z. (2009). Understanding user's query intent with wikipedia. In *Proceedings of the 18th international conference on World wide web*, pages 471–480. ACM.

Jeon, H., Hwang, I., and Kim, J. (2016). An intelligent dialogue agent for the iot home. In *AAAI Workshop: Artificial Intelligence Applied to Assistive Technologies and Smart Environments*.

Khatua, A., Cambria, E., and Chaturvedi, I. (2017). Let's chat about brexit! a politically-sensitive dialog system based on twitter data. In *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*, pages 393–398. IEEE.

Kim, J. K., Tur, G., and Celikyilmaz, A. (2016). Intent detection using semantically enriched word embeddings. In *Spoken Language Technology Workshop (SLT), 2016 IEEE*, pages 414–419. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *https://repository.upenn.edu/*.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

Li, Q., Han, Q., and Sun, L. (2014). Userintent: Detection of user intent for triggering smartphone sensing applications. In *Sensing, Communication, and Networking*

(SECON), 2014 Eleventh Annual IEEE International Conference on, pages 185–187. IEEE.

Liu, B. and Lane, I. (2016). Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Liu, C. W., Lowe, R., Serban, I. V., Noseworthy, M., and Charlin (2016). How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

Mehrabani, M., Bangalore, S., and Stern, B. (2015). Personalized speech recognition for internet of things. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 369–374. IEEE.

Mikolov, T., Sutskever, I., Chen, K., and Corrado, G. S. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nobari, G. H. and Tat-Seng, C. (2014). User intent identification from online discussions using a joint aspect-action topic model. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Rachuri, K. K., Musolesi, M., Mascolo, C., Rentfrow, P. J., Longworth, C., and Aucinas, A. (2010). Emotionsense: a mobile phones based adaptive platform for experimental social psychology research. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 281–290. ACM.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Schuster, M. and Paliwal, K. K. (1997). Bidirecional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Shao, Y., Hardmeier, C., and Nivre, J. (2016). Multilingual named entity recognition using hybrid neural networks. *The Sixth Swedish Language Technology Conference (SLTC)*, abs/1705.05414.

Shen, Y., He, X., and Gao (2014). Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 373–374. ACM.

Shridhar, K. and Sahu, e. a. (2018). Subword semantic hashing for intent classification on small datasets. *arXiv preprint arXiv:1810.07150*.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Spackman, K. A. (1989). Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the sixth international workshop on Machine learning*, pages 160–163. Elsevier.

Sun, M., Pappu, A., and Chen, Y.-N. (2016). Weakly supervised user intent detection for multi-domain dialogues. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 91–97. IEEE.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Sutton, C., McCallum, A., et al. (2012). An introduction to conditional random fields. *Foundations and Trends® in Machine Learning*, 4(4):267–373.

Venkataraman, A. and Anantha, A. (2017). Intent understanding in a virtual agent. In *Proceedings of the 9th International Conference on Machine Learning and Computing*, pages 33–37. ACM.

Wang, J., Cong, G., Zhao, W., and Li, X. (2015). Mining user intents in twitter: A semi-supervised approach to inferring intent categories for tweets. In *AAAI*, pages 318–324.

Yang, Z., Salakhutdinov, R., and Cohen, W. (2016). Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270*.

Zhang, C., Fan, W., and Du, N. (2016). Mining user intentions from medical queries: A neural network based heterogeneous jointly modeling approach. In *Proceedings of the*

25th International Conference on World Wide Web, pages 1373–1384. International World Wide Web Conferences Steering Committee.

Zhang, J., Yang, T. Z., and Hazen, T. (2015). Large-scaleword representation features for improved spoken language understanding. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5306–5310. IEEE.

Zheng, Y., Liu, Y., and Hansen, J. H. (2017). Intent detection and semantic parsing for navigation dialogue language processing. In *Intelligent Transportation Systems (ITSC), IEEE 20th International Conference on*, pages 1–6. IEEE.

Zhu, P., Zhang, Z., Li, J., Huang, Y., and Zhao, H. (2018). Lingke: A fine-grained multi-turn chatbot for customer service. *arXiv preprint arXiv:1808.03430*.