UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Rio Claro - SP

Pedro Ivo Monteiro Privatto

Uma Abordagem para Reconhecimento de Entidades Nomeadas usando Conhecimento Externo

Rio Claro - SP

2020

Pedro Ivo Monteiro Privatto

Uma Abordagem para Reconhecimento de Entidades Nomeadas usando Conhecimento Externo

Orientador: Prof. Dr. Ivan Rizzo Guilherme

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de Rio Claro - SP.

Financiadora: FUNDUNESP

Rio Claro - SP

2020

P961a

Privatto, Pedro Ivo Monteiro

Uma abordagem para reconhecimento de entidades nomeadas usando conhecimento externo / Pedro Ivo Monteiro Privatto. -- Rio Claro, 2020

87 f. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Geociências e Ciências Exatas, Rio Claro

Orientador: Ivan Rizzo Guilherme

1. Ciência da Computação. 2. Inteligência Artificial. 3. Processamento de Linguagem Natural. 4. Extração de Informação. 5. Reconhecimento de Entidades Nomeadas. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Geociências e Ciências Exatas, Rio Claro. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de São José do Rio Preto

IBILCE

ATESTADO DE APROVAÇÃO - DEFESA

Atestamos que **PEDRO IVO MONTEIRO PRIVATTO**, RA nº: CCO180106, RG nº 41.458.889-7, expedido pela SSP/SP, defendeu, no dia 29/09/2020, a dissertação intitulada ***Uma abordagem para reconhecimento de entidades nomeadas usando conhecimento externo***, junto ao Programa de Pós Graduação em Ciência da Computação, Curso de Mestrado Acadêmico, tendo sido 'APROVADO'.

Atestamos ainda que a obtenção do título dependerá de homologação pelo Órgão Colegiado competente.

São José do Rio Preto, 29 de setembro de 2020

Silvia Emiko Kazama
Supervisor Técnico de Seção
Seção Técnica de Pós-Graduação

Pedro Ivo Monteiro Privatto

Uma Abordagem para Reconhecimento de Entidades Nomeadas usando Conhecimento Externo

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de Rio Claro - SP.

Financiadora: FUNDUNESP

Banca Examinadora

- Prof. Dr. Ivan Rizzo Guilherme (Orientador)
  Departamento de Estatística, Matemática Aplicada e Computação
  Universidade Estadual Paulista "Júlio de Mesquita Filho"
- Profª. Drª. Veronica Oliveira de Carvalho
  Departamento de Estatística, Matemática Aplicada e Computação
  Universidade Estadual Paulista "Júlio de Mesquita Filho"
- Prof. Dr. Evandro Eduardo Seron Ruiz
  FFCLRP Departamento de Computação e Matemática
  Universidade de São Paulo

Rio Claro - SP

29 de setembro de 2020

Agradecimentos

Agradeço primeiramente à minha mãe Inês Regina, meu pai Reinaldo, e meu irmão Vitor Hugo por sempre me apoiarem e me proporcionarem um ambiente propício para que eu possa trilhar meu caminho da melhor forma possível. Só consegui chegar aonde cheguei pois tive meus medos acalmados, minhas dores diminuídas, e minhas preocupações aliviadas por aqueles que fazem tudo o que podem por mim. Não conseguiria ser nada do que sou se não tivesse sido ensinado, guiado e principalmente amado por vocês. Palavras não são suficientes para expressar meu sentimento de gratidão àqueles que são a razão do meu existir.

Agradeço também ao Ivan, meu professor, orientador, e acima de tudo amigo. Agradeço pela disponibilidade em aceitar-me como orientando e pela chance de participar em outros projetos, agradeço pela paciência quando eu me enrolo na tentativa de explicar minhas ideias, agradeço pela dedicação e disposição de ler e-mails, fazer revisões e ter conversas em dias e horários completamente não convencionais. Agradeço principalmente por ser a figura de "paizão" e se importar com todos os alunos dentro e principalmente fora do laboratório.

Agradeço a todos os meus amigos que sempre me propiciam momentos incríveis, que suportam meus desabafos, ouvem minhas preocupações, se importaram comigo, divagam nos mais diversos assuntos, e tornam minha caminhada mais alegre. Agradeço tanto aos mais recentes quanto aos de mais longa data. Creio que só somos capazes de crescer como seres humanos quando partilhamos experiências e ideias, e por isso sou imensamente grato por vocês me permitirem ter a chance de tentar melhorar como pessoa.

Agradeço também ao grande Kazuo Miura (*in memoriam*), um exemplo incrível de ser humano que, apesar de eu não ter tido o privilégio de conviver por muito tempo, sempre levarei comigo como exemplo de humildade, profissionalismo, dedicação e, acima de tudo, perseverança. Sonho para que um dia eu consiga chegar a ser pelo menos uma fração da pessoa incrível que o grande Kazuo era.

Agradeço ao excelente trabalho de todas as pessoas que participaram da minha formação e espero que um dia o mundo todo reconheça e glorifique a importância dos professores.

Agradeço à FUNDUNESP pela concessão da bolsa de pesquisa, sob o processo nº Fundunesp 2014/00545-0, Fundação para o Desenvolvimento da UNESP (FUNDUNESP)

Resumo

Nas diferentes áreas do conhecimento os dados textuais são importantes fontes de informação. Neste contexto, os métodos para Extração de Informação têm sido desenvolvidos para a identificação e estruturação de informações presentes em documentos textuais. Como subárea da Extração de Informação há o Reconhecimento de Entidades Nomeadas, que consiste em definir métodos para identificar Entidades Nomeadas, tais como Pessoa, Local, Data, entre outras, em textos. Recentemente esses métodos fazem uso de técnicas provenientes do Processamento de Linguagem Natural e de Aprendizado de Máquina. O objetivo do presente trabalho é propor uma metodologia para Reconhecimento de Entidades Nomeadas considerando os seguintes aspectos: identificação e implementação de métodos para representação de características gramaticais; identificação e implementação das novas abordagens que utilizam técnicas recentes de Aprendizado de Máquina, como BiLSTM, BiGRU e CRF; e avaliar, de maneira experimental, a integração de fontes externas de conhecimento, na forma de *Gazetteers* e Grafo de Conhecimento, vindos da Freebase e YAGO. O protocolo de avaliação experimental foi composto pela definição de quatro configurações de redes neurais, duas destas fazendo uso de conhecimento externo, e sua aplicação em cinco *datasets* com diferentes características. Nos experimentos realizados, houve ganho de F1-Score em 18 dos 40 casos onde foi utilizado conhecimento externo, chegando a um ganho de até 1,3%. Além do fato de não ter apresentado ganho em grande parte dos casos, a maioria dos ganhos foi inferior a 0,5% no F1-score. Os resultados experimentais dos métodos utilizados nos *datasets* escolhidos evidenciam que as estratégias empregadas para a integração do conhecimento externo agregaram baixos ganhos aos modelos, como mostrado pelas métricas Precisão, Abrangência e F1-Score. Portanto, se a fonte de conhecimento não provê informações relevantes ao domínio da tarefa, e se a maneira de agregação do conhecimento não capta o conteúdo relevante presente no mesmo, esta adição de conhecimento externo não necessariamente é benéfica à metodologia.

Palavras-chave: Reconhecimento de Entidades Nomeadas, Processamento de Linguagem Natural, Extração de Informação, Inteligência Artificial.

Abstract

In different areas of knowledge textual data are important sources of information. In this context, Information Extraction methods have been developed to identify and structure information present in textual documents. As a subarea of Information Extraction there is Named Entity Recognition, which consists of using methods to identify Named Entities, such as Person, Place, Date, among others, in texts, using techniques from Natural Language Processing and Machine Learning. Recently, these methods use techniques from Natural Language Processing and Machine Learning. The purpose of this work is to propose a methodology for Named Entity Recognition considering the following aspects: identification and implementation of grammatical feature representation methods; identification and implementation of new approaches that use recent Machine Learning techniques, such as BiLSTM, BiGRU and CRF; and to evaluate, in an experimental way, the integration of external knowledge sources, in the form of Gazetteers and Knowledge Graph, coming from Freebase and YAGO. The experimental evaluation protocol was composed by four configurations of neural networks, two of them making use of external knowledge, and their application in five datasets with different characteristics. In the conducted experiments, there was a gain of F1-Score in 18 of the 40 cases where external knowledge was used, reaching a gain of up to 1.3%. In addition to the fact that there was no gain in most cases, the majority of the gains were lesser than 0.5% in F1-score. The experimental results of the methods applied to the chosen datasets show that the strategies used for the integration of external knowledge added low gains to the models, as shown by the metrics Precision, Recall and F1-Score. Therefore, if the source of knowledge does not provide relevant information to the task domain, and if the way of aggregating the external knowledge does not capture the relevant content present in it, this addition of external knowledge is not necessarily beneficial to the methodology.

Keywords: Named Entity Recognition, Natural Language Processing, Information Extraction, Artificial Intelligence.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ANN	*Artificial Neural Network*
BiLSTM	*Bidirectional Long Short-Term Memory*
BiGRU	*Bidirectional Gated Recurrent Unit*
CNN	*Convolutional Neural Network*
CRF	*Conditional Random Fields*
EI	Extração de Informação
FN	Falso Negativo
FP	Falso Positivo
GRU	*Gated Recurrent Unit*
IOB	*Inside-Outside-Beginning*
LSTM	*Long Short-Term Memory*
NER	*Named Entity Recognition*
PLN	Processamento de Linguagem Natural
RNN	*Recurrent Neural Network*
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

Sumário

1 Introdução

A crescente evolução do uso de tecnologias de informação nas mais diversas áreas é responsável pela grande quantidade de dados gerados em diferentes formatos, sejam esses dados imagens, áudios, vídeos ou textos. Grande parte desses dados gerados não apresentam nenhum tipo de estruturação, servindo apenas como arquivo para empresas e governos. Segundo [1], um dado não estruturado é aquele que não apresenta uma estrutura clara, semanticamente evidente e de fácil processamento por máquinas. A não estruturação de informações é uma barreira para a tarefa de recuperação de dados, uma vez que toda consulta deve ser feita de maneira manual em todos os arquivos que possam conter a informação relevante em meio a tantas informações irrelevantes.

O montante de dados textuais não estruturados cresceu também devido à evolução do perfil do usuário da internet que, com a evolução das tecnologias de informação e comunicação, passou de consumidor de informações para o gerador destas, principalmente no âmbito das redes sociais, onde a interação entre usuários é seu ponto chave.

No cenário de crescimento do volume de dados textuais gerados, percebeu-se a importância da extração, estruturação e utilização desses dados estruturados. Especialmente quando se deseja filtrar por aspectos específicos em meio a enormes quantias de dados, como doenças ligadas a um certo gene em artigos científicos ou relatórios médicos, processos industriais de bombeamento com volume maior que um limiar em uma base de relatórios técnicos, atentados terroristas ocorridos num determinado país em uma coletânea de notícias de jornal, entre outros. Desta maneira, o objetivo de se estruturar dados textuais é para que estes sejam mais facilmente consultáveis e reutilizáveis.

Em meio ao contexto de fontes textuais, a estruturação dos dados faz uso de Processamento de Linguagem Natural (PLN), área a qual utiliza conceitos de Inteligência Artificial e de Linguística para processar dados e automatizar tarefas que envolvam linguagem. Dentre as tarefas que são foco de estudo do Processamento de Linguagem Natural pode-se citar Tradução de Máquina, Reconhecimento de Fala, Análise Automática de Discurso, Sumarização Automática de Textos, Extração de Informação, entre outras.

A tarefa de Extração de Informação (EI), que é o tema deste trabalho, é a subárea

de PLN que é responsável por identificar informações desestruturadas contidas no texto e estruturá-las, facilitando sua posterior recuperação [2]. No contexto da EI há a tarefa chamada de Reconhecimento de Entidades Nomeadas - inglês *Named Entity Recognition* (NER), que consiste em encontrar conceitos, formados por uma ou mais palavras, presentes nos textos e categorizá-los de acordo com seu grupo semântico.

As abordagens para Reconhecimento de Entidades Nomeadas tradicionalmente fazem uso de muitas técnicas vindas da Linguística, tais como: etiquetas sintáticas, lema das palavras, prefixos e sufixos, entre outras, para extrair as informações presentes nos textos. Todo o processo necessário para o uso das técnicas tradicionais é bastante trabalhoso, pois envolve diversas etapas de preparação dos dados de entrada.

Com a finalidade de reduzir o trabalho das abordagens tradicionais de NER, as Abordagens Numéricas têm sido desenvolvidas de forma a simplificar as etapas de preparação dos dados. Recentemente, o uso de técnicas que modelam matematicamente os aspectos linguísticos sintáticos e semânticos vêm ganhando espaço. O uso de Abordagens Numéricas vem se popularizando por estas atingirem resultados similares, algumas vezes até melhores, quando comparadas às técnicas clássicas sem necessitar de um processo extensivo de *feature engineering*, processo que tem como finalidade a seleção das características que serão utilizadas na tarefa de NER. Porém, apesar de não ser necessário, existem abordagens que realizam *feature engineering* para as técnicas de Aprendizado de Máquina com o objetivo de incluir características que não podem ser captadas pelos *Word embeddings*, por exemplo.

Dentre as modelagens que têm sido utilizadas recentemente pode-se citar os *Word embeddings*, que são mapeamentos das palavras presentes nos textos para um espaço vetorial. Ao se fazer uso dos *Word embeddings*, a tarefa de NER pode ser tratada como uma tarefa de classificação, sendo atualmente bastante empregado o uso de classificadores que aprendem a categorizar as entidades presentes nos textos.

Depois de obtidas as representações das características desejadas, ocorre a etapa na qual as entidades são de fato reconhecidas. Tradicionalmente, esta etapa de reconhecimento era composta por métodos que faziam uso de regras e padrões de palavras ou caracteres para caracterizar cada tipo de entidade. Era comum também a realização da tarefa de Reconhecimento de Entidades Nomeada fazendo uso de amplas listas com entidades previamente reunidas, chamadas de *Gazetteer lists*, tendo como exemplo listas com nomes próprios, pontos geográficos, entre outros.

As novas abordagens frequentemente fazem uso de métodos numéricos, principalmente aqueles capazes de incorporar informações episódicas ao processo de Reconhecimento de Entidades Nomeadas, se beneficiando do aspecto sequencial das palavras dentro de um texto.

Em meio a esse contexto, existem as chamadas técnicas híbridas, que fazem uso tanto de aspectos linguísticos tradicionais, quanto uso de modelos matemáticos para realizar o Reconhecimento de Entidades Nomeadas. A motivação por trás do uso das técnicas híbridas está em tentar unir os aspectos positivos dos diferentes tipos de abordagens.

Recentemente alguns autores tem realizado a integração de fontes externas para agregar mais informação semântica e contextual. Uma estratégia frequentemente utilizada é o uso de fontes externas denominadas *Gazetteers* para agregar novas características que servem como entrada para métodos numéricos. A motivação do uso de fontes externas é enriquecer as representações numéricas das amostras com informações que não são encontradas nos *datasets* que estão sendo trabalhados.

Diante deste cenário, o presente trabalho apresenta um levantamento bibliográfico acerca dos métodos de Reconhecimento de Entidades Nomeadas. O resultado deste levantamento permitiu identificar trabalhos que foram utilizados para definir a abordagem proposta para Reconhecimento de Entidades Nomeadas e encontrar bases de dados utilizadas nessa tarefa. Além disso, o levantamento bibliográfico permitiu identificar quais recursos linguísticos são comumente utilizados para a tarefa de Reconhecimento de Entidades Nomeadas.

Com essas informações foi então proposta uma abordagem híbrida para a tarefa de Reconhecimento de Entidades Nomeadas, fazendo uso de *Word embeddings* unidos à fontes externas de conhecimento para alimentar uma arquitetura de rede neural capaz de captar dependências entre episódios e rótulos de uma sequência.

A abordagem proposta tem o objetivo de incorporar informações semânticas, contidas em imensos repositórios ricos em conhecimento, às técnicas que fazem uso somente informações das palavras isoladamente. O motivo dessa combinação é avaliar a hipótese de que informações semânticas podem beneficiar os processos numéricos atuais. Esta hipótese está fundamentada no fato de que as informações contidas em fontes externas não são encontradas nos dados que estão sendo trabalhados. Deste modo, a incorporação de conhecimento externo pode vir a ajudar os modelos de Aprendizado de Máquina a realizar uma melhor distinção entre as entidades, por meio da assimilação das dependências entre essas informações externas.

Como fontes de conhecimento externo foram utilizados dois dos principais repositórios disponíveis: o Freebase e o YAGO. As informações semânticas das palavras foram unidas às respectivas representações vetoriais para servirem de entrada aos modelos de classificação.

Em relação aos modelos de aprendizado de máquina, neste trabalho foram utilizadas redes neurais recorrentes para a extração das características de cada amosta das sequências de palavras, mais especificamente as redes *Long-Short Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU). Ao analisar um episódio de uma sequência, essas redes têm a capacidade de agregar características dos episódios anteriores à representação numérica do episódio atual. Neste trabalho, essas redes estão organizadas em duas camadas, uma em cada direção de leitura, agregando assim informações dos episódios anteriores, bem como dos episódios posteriores, para desta maneira gerar representações vetoriais que caracterizem a amostra atual levando em consideração seu contexto. Como método para a classificação das palavras, neste trabalho é utilizado o *Conditional Random Fields* (CRF), o qual tem a característica de fazer uso da classe atribuída ao episódio anterior de uma sequência, agregando assim informação sequencial dos rótulos já atribuídos. Deste modo, as técnicas de Aprendizado de Máquina utilizadas neste trabalho foram adotadas justamente por fazer uso das informações sequenciais encontradas nos textos.

O desafio deste trabalho envolve a avaliação da inclusão de conhecimento externo para melhoria da tarefa de Reconhecimento de Entidades Nomeadas. Para isso, é realizada a comparação dos resultados de dois cenários: utilizando a abordagem proposta com a adição de conhecimento externo; utilizando a arquitetura da abordagem proposta sem a adição de conhecimento externo. Além disso, os resultados da abordagem proposta também foram comparados a outros trabalhos da literatura.

1.1 Objetivos

A realização deste trabalho visou cumprir com os seguintes objetivos:

- **Objetivo Principal**: Propor e implementar uma abordagem para Reconhecimento de Entidades Nomeadas que faça uso de fontes de conhecimento externo integradas a métodos recentes de Aprendizado de Máquina.
- Realizar um estudo sobre as principais abordagens que estão sendo recentemente utilizadas

para a tarefa de Reconhecimento de Entidades Nomeadas, bem como as características e recursos utilizados para essa tarefa;

- Identificar *datasets* comumente utilizados para a tarefa de Reconhecimento de Entidades Nomeadas;
- Avaliar experimentalmente a abordagem proposta, seguindo protocolos e adotando métricas comumente utilizadas pela comunidade de Reconhecimento de Entidades Nomeadas.

1.2 Organização do Texto

A organização deste trabalho se dá da seguinte maneira: o Capítulo 2 aborda a definição de Extração de Informação, além de mostrar os fundamentos teóricos dos métodos utilizados neste trabalho; no Capítulo 3 é feita uma revisão bibliográfica de trabalhos na área de Reconhecimento de Entidades Nomeadas; no Capítulo 4 são apresentadas as bases de dados utilizadas neste trabalho, os repositórios de conhecimento utilizados, e também as métricas para avaliação de resultados; no Capítulo 5 é descrito o método para Reconhecimento de Entidades Nomeadas utilizado neste trabalho, bem como resultados da aplicação desse método, além da discussão sobre estes resultados; e no Capítulo 6 são mostradas as conclusões tiradas a partir deste trabalho, além de mostrar intenções de trabalhos futuros.

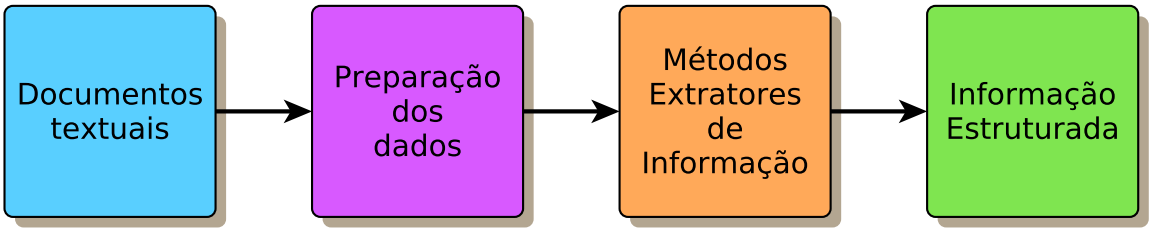
2 Extração de Informação

Este capítulo aborda a definição de Extração de Informação, além de mostrar uma visão geral das principais características que compõem a atividade de Extração de Informação, bem como as abordagens utilizadas para tal atividade. Além disso, é discutida a tarefa de Reconhecimento de Entidades Nomeadas e são mostradas algumas técnicas para essa tarefa.

2.1 Visão Geral

O Processamento da Linguagem Natural (PLN) é uma das principais áreas da Inteligência Artificial. Uma das pesquisas realizadas em PLN é a Extração de Informação (EI) que tem como finalidade desenvolver métodos para processar automaticamente textos não estruturados ou semiestruturados escritos em linguagem natural e recuperar, de maneira estruturada, ocorrências de entidades, relações entre tais entidades, e atributos pertencentes a essas entidades [3].

Figura 1 – Arquitetura geral de sistemas para Extração de Informação.



Fonte – Adaptado de [4]

O fluxo tradicionalmente adotado para a realização da extração de informação é apresentado na Figura 1. Inicialmente é necessário encontrar um conjunto de documentos textuais que se deseja extrair informações. Dado esse conjunto, faz-se necessário realizar a preparação dos textos para que os Métodos Extratores de Informações realizem a extração.

A fase de Preparação dos Dados compreende todas as etapas que têm o propósito de formatar os dados de entrada para que estes possam ser utilizados pelos extratores de informação. Algumas das técnicas tradicionalmente utilizadas nessa fase são: Segmentação de Frases, Tokenização, Etiquetamento Sintático, Remoção de *stopwords*, *Stemming*, Identificação

de prefixos e sufixos, Normalização de diferentes grafias para a mesma palavra, entre outras. O conjunto dessas técnicas é chamado de Pré-Processamento e faz-se muito presente nas aplicações tradicionais de PLN.

Além das técnicas de Pré-Processamento, é também nesta etapa de preparação que são utilizados métodos para gerar a representação vetorial das palavras dos *tokens* presentes nos textos. Recentemente, nas novas abordagens desenvolvidas, as etapas de Pré-processamento têm sido suprimidas. Novas técnicas, denominadas de *word embedding*, têm sido desenvolvidas utilizando diferentes métodos baseados em algoritmos de aprendizado de máquina, matrizes e grafos. Esses métodos permitem gerar representações compactas dos vetores das palavras, mantendo a possibilidade de ter diferentes representações para cada um dos termos.

Os Métodos Extratores de Informação são os componentes responsáveis pela extração dos conteúdos relevantes presentes em um texto. Dado um conjunto $\mathbb{C}$ de conceitos, um extrator de informação pode ser formalmente definido pela Equação 2.1, como mostrado em [4].

$$e_q^c(\mathcal{S}_s) = y_s^c \tag{2.1}$$

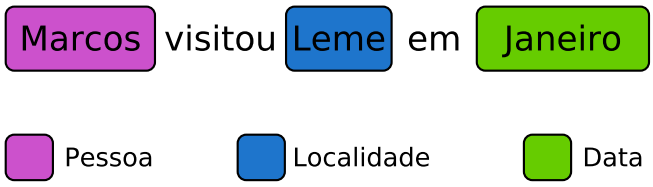
Na Equação 2.1 um extrator $e_q^c(\mathcal{S}_s)$ determina a conexão entre a frase $\mathcal{S}_s$, pertencente ao conjunto $\mathcal{S}$ de frases, e o conceito c, pertencente ao conjunto $\mathbb{C}$ de conceitos, através da determinação do conteúdo semântico y_s^c, utilizando a implementação do método extrator q. Em um cenário simplista, y_s^c assume um valor binário, indicando pertinência do conceito c à frase $\mathcal{S}_s$, ou seja $y_s^c \in \{0, 1\}$. Porém é possível que y_s^c também produza uma tupla contendo informações sobre a localização dos conceitos na frase $\mathcal{S}_s$. Outro tipo de valor que y_s^c pode assumir é uma tripla que exprima relação $R^c(R_a, R_b)$, onde R_a corresponde ao domínio da relação R^c e R_b corresponde à imagem da relação R^c.

Existem diversos métodos para Extração de Informações presentes nos documentos escritos em linguagem natural, além de diferentes tipos de informações a serem extraídas, tais como Extração de Relações, Extração de *Keyphrase*, *Entity Linking*, Reconhecimento de Entidades Nomeadas, Modelagem de Tópico, entre outras [5]. Neste trabalho, o foco da Extração de Informações é dado à tarefa de Reconhecimento de Entidades Nomeadas.

2.2 Reconhecimento de Entidades Nomeadas

O Reconhecimento de Entidades Nomeadas, é a área do Processamento de Linguagem Natural responsável por identificar Entidades presentes em um texto escrito em linguagem natural e dividi-las de acordo com as categorias as quais essas entidades pertencem. Uma Entidade é um conceito, formado por uma ou mais palavras, que pertence a um grupo semântico previamente definido [6, 7, 8, 9, 10, 11, 12]. A Figura 2 ilustra exemplos de Entidades.

Figura 2 – Exemplo de frase com Entidades Nomeadas.



Fonte – Autor.

Os grupos semânticos das Entidades variam de acordo com o domínio de interesse da aplicação, porém como exemplo de entidades comuns tem-se Pessoas, Organizações, Datas, Locais, e Moedas. Um tipo de interesse mais complexo são os eventos, que podem ser vistos como um grupo de relações que ocorrem durante um dado tempo. Eventos, geralmente contam com informações explícitas ou implícitas, como participantes, o início e o final do evento, bem como a localização [3].

Quanto à definição formal do Reconhecimento de Entidades Nomeadas, Li et al.[13] descreve a tarefa como: dada uma sequência de *tokens* $\mathcal{S}_s = (\mathcal{S}_{s_0}, \mathcal{S}_{s_1}, ..., \mathcal{S}_{s_{n-1}})$, onde n é o número de palavras da frase, o Reconhecimento de Entidades Nomeadas é responsável por encontrar e retornar uma lista de tuplas $(\mathcal{I}_a, \mathcal{I}_b, \mathcal{I}_{type})$, em que cada elemento dessa lista é uma entidade contida em $\mathcal{S}_s$, onde $\mathcal{I}_a$ e $\mathcal{I}_b \in [0, n-1]$ são as fronteiras da entidade, ou seja, os índices que marcam seu início e fim, e $\mathcal{I}_{type}$ é o tipo da entidade.

O desenvolvimento das pesquisas referentes à tarefa de Reconhecimento de Entidades Nomeadas teve início na sexta edição da *Message Understanding Conference* (MUC-6)[12], onde o termo *Named Entity Recognition* foi cunhado, cujos resultados foram reportados em [14]. Além da MUC, existem outros eventos que abordam essa área, como é o caso do SemEval[1],

[1] <http://alt.qcri.org/semeval2020/>

ACE[2] [15], TREC[3], CoNLL[4] [16]. Desde que essa área foi criada, tem sido tópico de estudo e aplicação em fontes textuais em diversas áreas de conhecimento, como combate às drogas a partir da identificação de drogas em notícias de jornais [17]; no domínio biomédico através extração de proteínas, RNA, DNA de artigos científicos [18]; de sintomas de doenças em textos clínicos [19]; de tratamentos, testes e problemas em textos clínicos [20, 21]; nome de fármacos em textos biomédicos [22]; nomes de doenças em textos biomédicos [23]; nomes de medicamentos e de doenças de textos de fórum de discussão [24]; e em "áreas lúdicas", através da identificação de livros, jogos, filmes e músicas em consultas na Web [25]. Nessas tarefas está sendo feito o uso o de conceitos de outras áreas, como Recuperação de Informação, Banco de dados, Aprendizado de Máquina, entre outras, em conjunto com a área de PLN.

Ao se realizar o Reconhecimento de Entidades Nomeadas, o foco está em descobrir a qual Entidade cada palavra está atrelada, portanto a Equação 2.1 é alterada para tratar palavras ao invés de frases inteiras. Esta alteração é apresentada na Equação 2.2.

$$e_q\left(\mathcal{S}_{s_p}\right) = \hat{y}_{s_p} \tag{2.2}$$

Na Equação 2.2, $e_q\left(\mathcal{S}_{s_p}\right)$ corresponde ao extrator que determina qual o conceito correspondente à palavra p da setença $\mathcal{S}_s$ através do uso do extrator q. Desta maneira, $\hat{y}_{s_p}$ assume um valor do conjunto $\mathbb{C}$ de entidades possíveis, portanto $\hat{y}_{s_p} \in \mathbb{C}$.

Existem diversos extratores que podem assumir o papel do e_q descrito na Equação 2.2, podendo estes métodos ser divididos de acordo com a abordagem que utilizam, sendo uma divisão feita entre Abordagens Simbólicas e Abordagens Numéricas, além das Abordagens Híbridas, sendo as características de cada abordagem mostradas nas subseções seguintes.

2.2.1 Abordagens Simbólicas

As Abordagens Simbólicas para Extração de Informação são aquelas que fazem uso de conhecimento explicitado por humanos e representado em um formato processável por máquinas.

A abordagem frequentemente utilizada para extração simbólica é a Baseada em Regras, que faz uso de regras que definem padrões sintáticos ou semânticos caracterizados por cadeias de caracteres ou sequências de palavras que caracterizam as informações de interesse presentes nos

[2] <https://www.ldc.upenn.edu/collaborations/past-projects/ace>
[3] <https://trec.nist.gov/>
[4] <https://www.conll.org/>

textos. Dependendo da heterogeneidade dos textos, a elaboração das regras por humanos se torna uma tarefa complexa devido às diferentes maneiras como as informações podem ser apresentadas [26].

Para facilitar a confecção das regras, alguns autores fazem uso de Anotadores Sintáticos nas palavras dos textos, enriquecendo os documentos de entrada com informações sintáticas ou também anotações semânticas. A partir dos textos anotados, métodos de aprendizado podem ser utilizados para a geração das regras de extração de informações. Os resultados das abordagens baseadas em regras geralmente apresentam alta precisão, porém baixa abrangência [13]. Isso ocorre pelo fato de moldarem muito bem uma parte do domínio através da visão de quem as cria, porém sua abrangência fica limitada pela ótica de quem as elaborou. Por outro lado, os métodos Baseados em Regras são atrativos para especialistas de domínio, uma vez que representam uma maneira fácil de compreensão e visualização do conhecimento.

Como maneira de diminuir o trabalho de geração das regras por parte de humanos, bem como melhorar suas abrangências, existem métodos para geração automática de regras a partir dos dados de entrada [27] . Posteriormente cabe ao especialista do domínio validar essas regras geradas ao invés de ter que concebê-las.

A principal dificuldade ao se utilizar de Abordagens Simbólicas é que se faz necessária uma pessoa que tenha conhecimento do domínio a ser trabalhado para que os dicionários ou as regras possam ser gerados e validados.

Neste sentido, uma das abordagens para a extração é chamada de *Gazetteer Lists*, também conhecidas como Vocabulário de Termos ou Baseadas em Dicionários, que faz uso de listas com palavras-chave a serem reconhecidas e extraídas. Como exemplo tem-se listas com nomes de cidades, capitais, nomes próprios, entre outras palavras que sejam relevantes ao domínio.

Embora os *Gazetteers* solucionem parte das dificuldades dos métodos de extração, para domínios pequenos e restritos, os dicionários são relativamente simples de serem feitos, porém quando o domínio é muito abrangente, também chamado de domínio aberto, a criação dessas listas torna-se uma tarefa trabalhosa. Uma das maneiras de se agilizar a criação das listas é fazer uso de entidades presentes em Repositórios de Conhecimento, os quais apresentam enormes quantidades termos que representam entidades, além da relação entre elas. Como exemplos de fontes externas de conhecimento temos a WordNet [28], YAGO, Freebase, Wikidata, DBpedia.

O uso das fontes externas pode ser feito por meio da criação de *Gazetteers* através

dos *dumps* desses repositórios de conhecimento, que tratam-se de arquivos que contêm todo o conteúdo dessas fontes até uma certa data. A partir desses *dumps* são realizadas filtragens para se extrair as entidades de interesse, uma vez que muitos desses repositórios são estruturados no formato de triplas RDF, compostos por milhões de entidades e bilhões de fatos que relacionam essas entidades. Após a filtragem, é feita então separação das entidades e a criação dos *Gazetteers* para uso local.

Outra maneira de se agregar informações externas é utilizar serviços de consultas à esses repositórios. A vantagem dessa estratégia é que não se fazem necessárias as etapas de filtragem das entidades de interesse, e nem a criação de *Gazetteers*, porém é necessário que haja um serviço capaz de buscar as informações em meio à essa enorme quantia de dados.

O uso de *Gazetteer Lists* é bastante simples, pois consiste na análise das palavras da frase e a verificação de suas pertinências às listas. Pelo fato de ser uma simples análise de pertinência de cada elemento da frase, esse método de extração não considera o contexto no qual a frase está inserida, podendo assim levar a extrações errôneas. No entanto existem também casos de listas que contêm expressões compostas por múltiplas palavras, e portanto faz-se necessária a análise de mais de uma palavra por vez, incorporando assim um pouco de contexto à extração. De maneira geral, quando o domínio é restrito a textos não complexos e não há a possibilidade de confusão dos conceitos a serem extraídos, as abordagens que fazem uso de Vocabulários de Termos são consideradas uma boa opção.

Uma outra abordagem simbólica para a tarefa de NER faz uso de *Ontology Based Information Extraction* (OBIE) [3], que consiste no uso de ontologias como fonte de informação e conhecimento. A informação extraída pode ser instâncias da ontologia, possibilitando a consulta conceitual das informações extraídas. Embora as fontes de ontologias têm crescido, essas fontes são frequentemente restritas a domínios específicos. Atualmente, os repositórios baseados em grafo são mais utilizados por permitir a sua utilização em domínios abertos e contar com modelos de ontologias de topo.

2.2.2 Abordagens Numéricas

Extratores baseados em Abordagens Numéricas são aqueles que usam métodos de Aprendizado de Máquina para realizar a extração das informações presentes nos documentos textuais. Nessa abordagem as informações textuais geralmente são representadas de maneira

numérica, por meio de um grafo ou uma representação vetorial, por exemplo. Dada esta representação, os modelos supervisionados (descritos abaixo) de Aprendizado de Máquina encontram uma função para generalizar a classificação e encontrar a classe de cada palavra. Desta maneira, para representar um extrator baseado em Aprendizagem de Máquina[5], a Equação 2.2 pode ser alterada para a Equação 2.3, sendo $\mathcal{X}_{s_p}$ a representação numérica da palavra p da frase s, e $\hat{y}_{s_p}$ assume um valor do conjunto $\mathbb{C}$, com $\hat{y}_{s_p} \in \mathbb{C}$.

$$e_q\left(\mathcal{X}_{s_p}\right) = \hat{y}_{s_p} \tag{2.3}$$

Recentemente, para a obtenção das representações vetoriais de palavras, as aplicações de NER têm utilizado os *word embeddings*, que tratam-se de mapeamentos de palavras para espaços vetoriais cuja dimensionalidade fica, geralmente, em torno de poucas centenas de dimensões. As abordagens anteriormente utilizadas de representação vetorial apresentavam a desvantagem de possuírem um grande número de dimensões, como é o exemplo dos vetores que exprimiam a coocorrência entre palavras, que apresentavam dimensão $|\mathcal{V}|$, totalizando uma matriz $|\mathcal{V}|^2$, sendo $\mathcal{V}$ o conjunto de todas as palavras presentes no corpus avaliado, chamado de Vocabulário. Desta maneira, o uso de técnicas recentes para *word embedding* apresenta a vantagem de representar as palavras por meio de vetores densos, cujas dimensões são mais representativas, capturando informações sintáticas e semânticas [13].

Dentre as técnicas para *word embedding*, o *word2vec* [29] apresenta uma maneira de se criar uma representação vetorial, utilizando aprendizado de máquina, que considera as palavras adjacentes. A finalidade desse método é fazer uso de uma rede neural com uma camada oculta para aprender a coocorrência entre palavras, sendo a representação vetorial gerada a partir dos valores aprendidos nesta camada.

Outro método para *word embedding* bastante utilizado é o GloVe [30], que obtém as representações vetoriais através de uma matriz de coocorrências entre palavras. Inicialmente a matriz de coocorrências tem dimensionalidade elevada, $|\mathcal{V}|^2$, que representa as palavras e os contextos nos quais elas ocorrem. A partir desta matriz é possível calcular a razão de coocorrência entre duas palavras, e desta forma o método encontra dois vetores para representá-las, com a restrição de que a diferença entre estes vetores deve ser igual a razão de coocorrência destas

[5] O foco do trabalho é o uso de modelos supervisionados, portanto deste ponto em diante todas as referências ao termo Aprendizado de Máquina irão se referir à modelos supervisionados de Aprendizado de Máquina, a menos que seja dito o contrário.

palavras. Desta maneira, independente do número de palavras, é possível representá-las em um espaço de dimensões não elevadas, levando em consideração a informação global das coocorrências entre palavras.

Quando a Extração de Informação trata-se do Reconhecimento de Entidades Nomeadas, os extratores baseados em Aprendizado de Máquina abordam a Extração de Informação como um problema de classificação para identificar se uma palavra pertence ou não a alguma das entidades.

Uma das estratégias de aprendizado de máquina bastante utilizada para tarefas de classificação é o modelo supervisionado, que consiste tradicionalmente de duas etapas: a etapa de treinamento, onde um conjunto de dados já rotulados de acordo com as entidades a serem extraídas será utilizado pelo classificador como aprendizado para a criação do modelo; e a etapa de testes, onde um conjunto de dados que se deseja classificar são apresentados como entrada para o classificador já treinado para que este possa classificá-los.

Nas abordagens supervisionadas, para que os classificadores sejam treinados é necessária uma quantidade considerável (dependendo do método utilizado, bem como da complexidade da extração) de dados previamente rotulados, os quais nem sempre estão disponíveis.

Como alternativa à falta de um número suficiente de dados rotulados disponíveis, existem os métodos semi-supervisionados. Classificadores semi-supervisionados são aqueles que, apesar de ainda utilizarem dados rotulados, necessitam de um número menor deles, pois utilizam também características dos dados não rotulados para realizar a classificação. Além dos modelos supervisionados e semi-supervisionados, existem os também os modelos não supervisionados, que não precisam de dados anotados, fazendo uso de características presentes nos próprios dados para realizar sua distinção, como é o caso das técnicas de *clustering*.

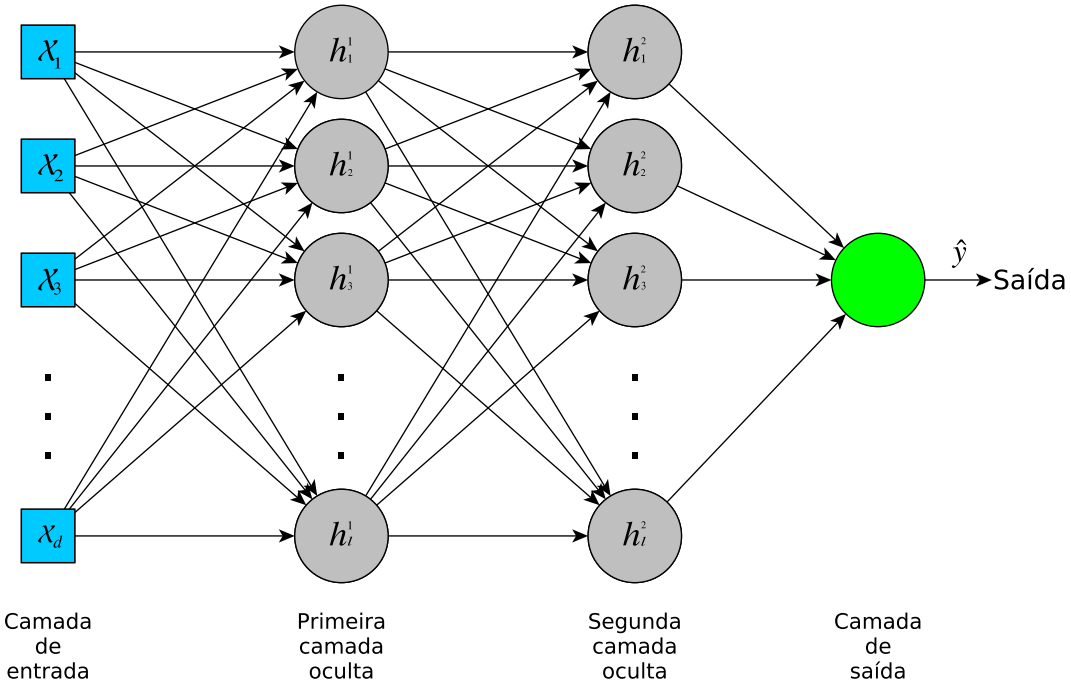
Com base no cenário descrito, o presente trabalho mostra uma proposta do uso de métodos de Aprendizado de Máquina que envolve as representações vetoriais de palavras e classificação, a fim de avaliar a hipótese de que a incorporação de conhecimento externo causa impactos na tarefa de Reconhecimento de Entidades Nomeadas.

Os métodos de aprendizado utilizados na proposta são apresentados nas subseções a seguir.

2.2.2.1 Redes Neurais Artificiais

Redes Neurais Artificiais - do inglês *Artificial Neural Network* (ANN), são estruturas computacionais que visam modelar alguns aspectos e funcionamentos de um cérebro biológico [31]. Sua estrutura é composta por uma ou mais unidades básicas de processamento, chamadas de neurônios, que são responsáveis por receber sinais, ponderá-los através de pesos sinápticos, e combiná-los para gerar um sinal de saída. Para a tarefa de classificação, o propósito de uma Rede Neural Artificial é encontrar uma função $\omega(\mathcal{X})$ capaz de predizer a qual classe a amostra $\mathcal{X}$ pertence. A Figura 3 exemplifica o funcionamento de uma rede neural de múltiplas camadas.

Figura 3 – Funcionamento de uma Rede Neural de Múltiplas Camadas.



Fonte – Adaptado de [31].

Inicialmente é gerada a estrutura inicial da rede, a qual é composta por matrizes de pesos sinápticos, representados por W, os quais são inicializados geralmente de maneira aleatória.

Para encontrar a função $\omega(\mathcal{X})$, é necessário que as ANNs passem por uma etapa chamada de treinamento, que é responsável por ajustar as matrizes de pesos sinápticos de cada camada através de um conjunto de exemplos já classificados. Quando se trata de uma rede de várias camadas, a função $\omega(\mathcal{X})$ é composta, num exemplo de quatro camadas, por $\omega_2(\omega_1(\mathcal{X}))$, sendo $\omega_1(\cdot)$ a função da primeira camada oculta, $\omega_2(\cdot)$ a função da segunda camada oculta e assim por diante.

Em seguida são introduzidos na primeira camada os sinais de entrada, representados por $\mathcal{X}_{\{1,2,3,\ldots,d\}}$, pertencentes a uma amostra já classificada, e é gerado um valor de saída para cada neurônio, valores estes que serão passados aos neurônios da camada seguinte. A ideia é que cada camada recombine e faça operações com esses valores que recebem como entrada, com o intuito de gerar uma representação característica de cada amostra que foi apresentada à rede. Ao final da rede, os valores da última camada oculta passam por uma camada de classificação que gera uma saída, representada por $\hat{y}$. Essa saída $\hat{y}$ é comparada à saída esperada, denotada por y, e, caso a saída produzida seja diferente da esperada, são feitos ajustes aos pesos sinápticos por meio de uma Função de Perda, um Algoritmo de Otimização, e um parâmetro chamado Taxa de Aprendizagem. A Função de Perda é responsável por quantificar o quão diferente $\hat{y}$ está de y, a Função de Otimização é responsável calcular os novos pesos sinápticos, tendo suas magnitudes definidas pela Taxa de Aprendizagem. Esse processo é repetido para todas as amostras do conjunto de treinamento, completando assim uma Época. O número de Épocas necessárias para uma rede encontrar uma função $\omega(\mathcal{X})$ satisfatória varia de acordo com o tipo de rede utilizada e a quantidade de amostras disponíveis no conjunto de treinamento.

A operação realizada na última camada é o produto escalar entre os sinais que saem da penúltima camada, representados pelo vetor $h^{|k-1|}$, e os pesos sinápticos, representados em uma matriz $\mathcal{W}^{|k|}$. O resultado desse produto é somado a um termo chamado de *bias*, representado como $b^{|k|}$, que tem o papel de agir como coeficiente linear para a função de ativação do neurônio, denotada por $\gamma(\cdot)$. A Equação 2.4 reporta as operações descritas.

$$y_i = \gamma(h^{k-1} \bullet \mathcal{W}^k + b^k) \tag{2.4}$$

, onde h^{k-1} denota os estímulos da camada anterior, e $\mathcal{W}^k$ e b^k representam, respectivamente, a matriz de pesos e o bias da k-ésima camada.

O resultado da camada de ativação depende da função de ativação utilizada. Dentre as funções mais comuns estão a *tanh*, reLU e *softmax* , representadas respectivamente pelas Equações 2.5, 2.6, 2.7, onde $\xi = h^{k-1} \bullet \mathcal{W}^k + b^k$.

$$\gamma(\xi) = \tanh(\xi) \tag{2.5}$$

$$\gamma(\xi) = \max(0, \xi) \tag{2.6}$$

$$\sigma(\xi) = \frac{e^{\xi_i}}{\sum_{j=1}^{|\mathbb{C}|} e^{\xi_j}} = \bar{y} \tag{2.7}$$

A função *softmax*, representada pela Equação 2.7, trata-se de uma generalização da função logística para mais de uma dimensão, tendo como resultado uma distribuição de probabilidades. Consiste na aplicação da função exponencial para cada elemento do vetor de entrada, que neste caso trata-se de $\xi = h^{k-1} \bullet \mathcal{W}^k + b^k$, e este vetor é então normalizado para se tornar a distribuição de probabilidades referentes às classes do problema. Desta forma, seu resultado é um vetor $\bar{y}$ de tamanho $|\mathbb{C}|$, sendo $|\mathbb{C}|$ o número de classes possíveis[6], em que cada elemento do vetor $\bar{y}_j$ representa a pontuação da amostra em relação à classe j, sendo $j \in \mathbb{C}$, onde quanto maior essa pontuação, mais provável é da amostra pertencer à classe.

Um dos propósitos de se utilizar estas funções de ativação é facilitar o treinamento, uma vez que a atualização da matriz de pesos é muito brusca quando uma função limiar é utilizada, além de tornar o modelo capaz de encontrar uma melhor função que segmente o espaço vetorial das classes. Outrossim, o algoritmo de *backpropagation* implementa métodos para atualização dos pesos considerando a derivada da função de ativação, a qual é mais fácil de se obter em algumas funções de ativação, como é o caso da *softmax*.

Para que, por exemplo, uma rede neural de múltiplas camadas com função de ativação *softmax* reconheça quais as entidades presentes na frase "Vinicius voltou para Leme", faz-se necessário que a representação vetorial de cada palavra passe pela rede, seja recombinada e sofra operações aritméticas para então ter como saída um vetor de pontuações. Considerando $\mathbb{C} = \{Nenhuma, Nome, Local\}$, o processo de obtenção do vetor $\bar{y}$ para a frase exemplo é mostrado pelas Equações de 2.8 até 2.11.

$$\bar{y}_{Vinicius} = \sigma\left(h^{k-1} \bullet \mathcal{W}^k + b^k\right) = [0.05, 0.85, 0.10] \tag{2.8}$$

$$\bar{y}_{voltou} = \sigma\left(h^{k-1} \bullet \mathcal{W}^k + b^k\right) = [0.93, 0.05, 0.02] \tag{2.9}$$

$$\bar{y}_{para} = \sigma\left(h^{k-1} \bullet \mathcal{W}^k + b^k\right) = [0.87, 0.03, 0.10] \tag{2.10}$$

[6] Deste ponto em diante, como o trabalho é sobre Reconhecimento de Entidades Nomeadas, $\mathbb{C}$ passará a representar o conjunto de possíveis entidades, uma vez que as classes representam Entidades Nomeadas, que por sua vez são os conceitos de interesse neste trabalho.

$$\bar{y}_{Leme} = \sigma\left(h^{k-1} \bullet \mathcal{W}^k + b^k\right) = [0.03, 0.06, 0.91] \tag{2.11}$$

Como mostrado nas equações acima, a saída da função *softmax* é um vetor de pontuações no intervalo [0,1], e a soma desse vetor resulta em 1. Seguindo as entidades do conjunto $\mathbb{C}$ acima, o resultado para a frase "Vinicius voltou para Leme" é: Nome, Nenhuma, Nenhuma, Local.

2.2.2.2 Redes Neurais Convolucionais

As Redes Neurais Convolucionais - do inglês *Convolutional Neural Network* (CNN) são redes originalmente criadas para problemas envolvendo imagens e reconhecimento de padrões por serem invariantes à distorções, tais como rotação, translação, e escala que possam estar presentes.

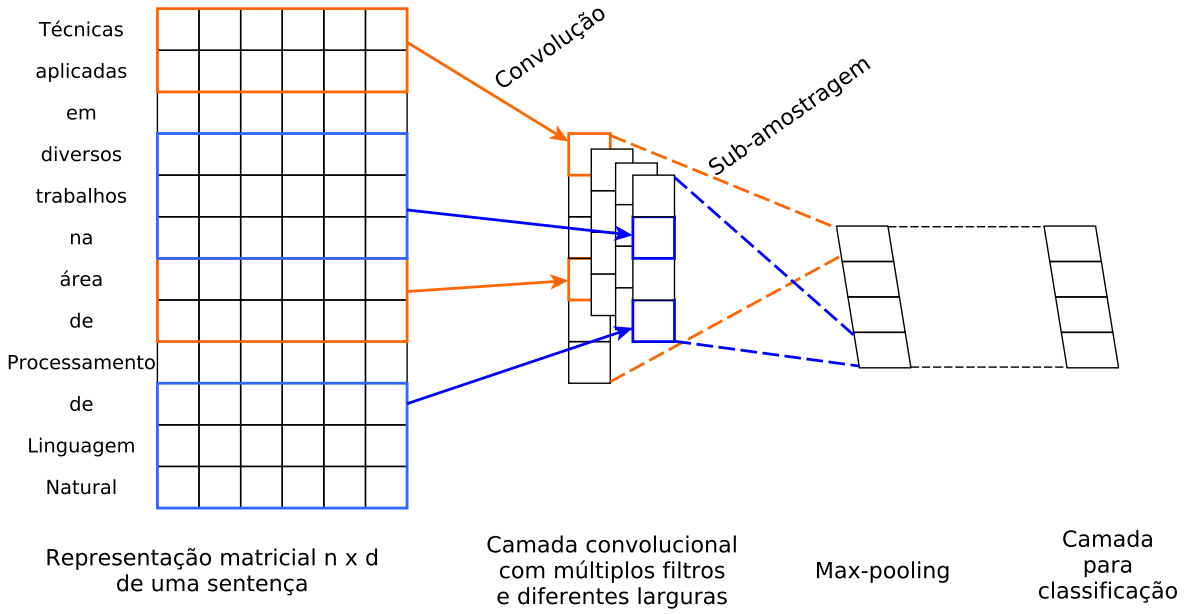
Apesar de ser amplamente utilizada para imagens, as CNNs podem ser utilizadas em problemas de PLN para determinar a representação vetorial de uma frase ou uma representação vetorial a partir dos caracteres que formam uma palavra.

Um exemplo do uso de uma CNN no processamento da frase "Técnicas aplicadas em diversos trabalhos na área de Processamento de Linguagem Natural" é mostrado de Figura 4. A representação matricial $n \times d$ da frase é obtida através da junção dos *word embeddings* de cada uma das n palavras que compõem a frase-exemplo e d a dimensão desses *embeddings*. Posteriormente é realizado o processo de convolução, seguido do processo de sub-amostragem - do inglês *subsampling*. A convolução tem como finalidade realizar a extração de características locais por meio de filtros de tamanho $g \times d$, representados por retângulos coloridos Figura 4. Após a convolução há uma sub-amostragem das características extraídas pela convolução, com o objetivo de escolher a característica mais relevante entre elas.

Essas etapas de convolução e sub-amostragem podem ser repetidas quantas vezes forem necessárias para o problema tratado, pois não há um consenso quanto ao número de camadas convolucionais e de sub-amostragem. De maneira geral são captadas características mais complexas ao se adicionar mais camadas à rede. Como última camada é frequentemente utilizada uma função de ativação (*softmax*, $tanh$ por exemplo) para que a amostra possa então ser classificada. Ao se utilizar a CNN para somente criar uma representação vetorial, pode-se descartar a camada de classificação.

Além da utilização de CNNs a nível de frase, há trabalhos que a utilizam a nível de

Figura 4 – Exemplo de CNN aplicada à frases.



Fonte – Adaptado de [32].

palavra, onde a convolução é realizada nas representações vetoriais de cada caractere que compõe a palavra, como é mostrado pela Figura 5. Este trabalho faz uso dessa estratégia para a geração de uma representação vetorial única que exprima a informação de todos os caracteres que compõem cada palavra.

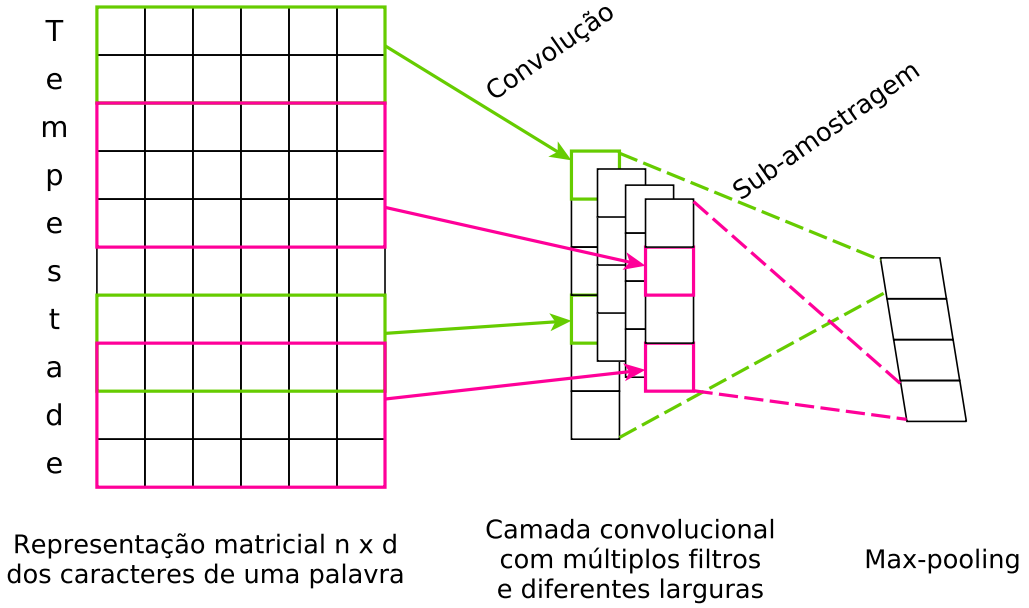
2.2.2.3 Redes Neurais Recorrentes

Em alguns problemas o objeto da classificação trata-se de uma sequência temporal de eventos. Isto ocorre, por exemplo, em vídeos, áudios, ou texto, em que um evento na amostra atual tem influência na amostra seguinte. Para esses casos foram propostas as Redes Neurais Recorrentes - do inglês *Recurrent Neural Network* (RNN), as quais utilizam os valores do episódio anterior para calcular o valor do novo episódio. A Figura 6 mostra um exemplo de RNN expandida através do tempo.

Ao se aplicar essa ideia de agregar episódios anteriores à frase exemplo, temos as situações mostradas nas Equações de 2.12 até 2.15, onde h_t é o estado oculto para o episódio t, e $\mathcal{W}_{t\text{-}1,t}$ é a matriz de pesos entre o episódio $t-1$ e o episódio t.

$$h_{\text{Vinicius}} = \left(\mathcal{X}_{\text{Vinicius}} \bullet \mathcal{W}_{0,\text{Vinicius}} + b^k\right) \tag{2.12}$$

Figura 5 – Exemplo de CNN aplicada à palavra.



Fonte – Adaptado de [32].

$$h_{voltou} = \left(h_{\text{Vinicius}} \bullet \mathcal{W}_{0,\text{Vinicius}} + \mathcal{X}_{\text{voltou}} \bullet \mathcal{W}_{\text{voltou}} + b^k\right) \tag{2.13}$$

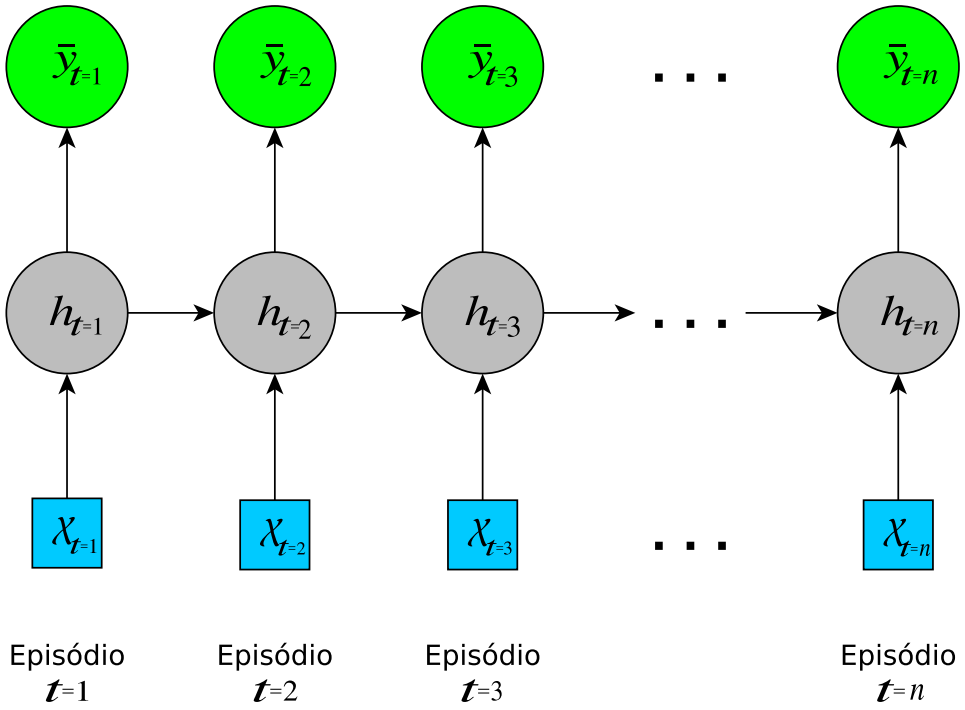
$$h_{para} = \left(h_{\text{t=voltou}} \bullet \mathcal{W}_{\text{voltou,para}} + \mathcal{X}_{\text{para}} \bullet \mathcal{W}_{\text{para}} + b^k\right) \tag{2.14}$$

$$h_{Leme} = \left(h_{\text{t=para}} \bullet \mathcal{W}_{\text{para,Leme}} + \mathcal{X}_{\text{Leme}} \bullet \mathcal{W}_{\text{Leme}} + b^k\right) \tag{2.15}$$

As RNNs desempenham um bom papel ao tratar informações episódicas, porém tendem a dar mais relevância aos episódios mais recentemente vistos pela rede, o que muitas vezes pode ser visto como prejudicial quando se deseja manter informações presentes no início da sequência, como é o caso de classificação de frases de um texto longo. Com a finalidade de reter mais informações de episódios anteriores, existem as redes de Longa Memória de Curto Prazo - do inglês *Long Short-Term Memory* (LSTM) [33] e Unidades Recorrentes Bloqueáveis - do inglês *Gated Recurrent Unit* (GRU) [34].

As LSTMs são um tipo de Rede Neural Recorrente caracterizadas pelo uso de estruturas chamadas de portões. O papel dos portões é controlar o quanto será utilizado dos estímulos de entrada, o quanto será esquecido dos episódios anteriores e o que será apresentado como

Figura 6 – Exemplo de funcionamento de uma RNN expandida através do tempo.



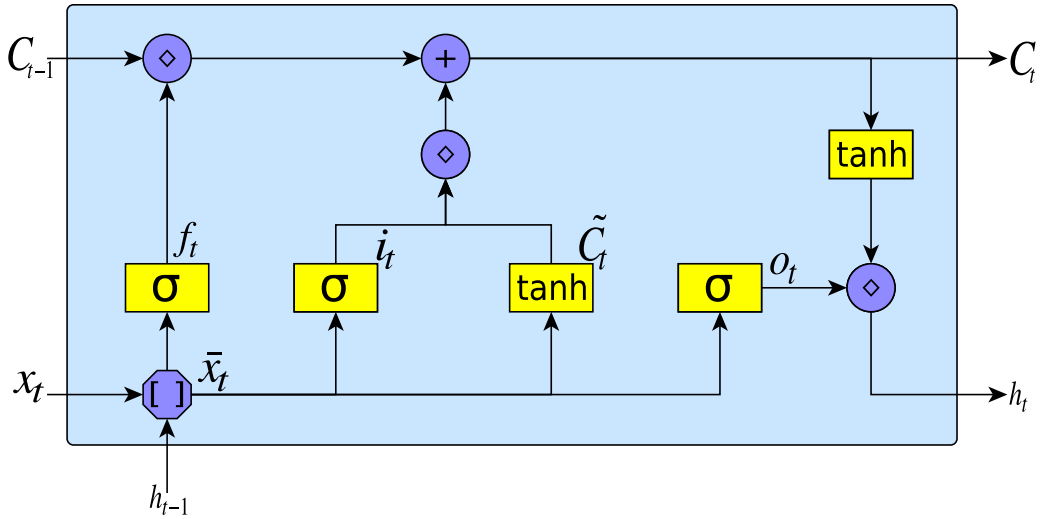
Fonte – Adaptado de [31].

saída. Os portões responsáveis pela entrada, pelo esquecimento e pela saída são chamados, respectivamente, de *input gate*, *forget gate* e *output gate*.

A interação entre os portões e valores de entrada de uma unidade LSTM são mostradas pela Figura 7, onde o octógono representa a operação de concatenação entre dois vetores, os círculos representam o produto de Hadamard ou a operação de soma, e os retângulos representam as camadas com suas respectivas funções de ativação. As formulações matemáticas são mostradas pela Equação 2.16 e o funcionamento dessa estrutura é explicado a seguir.

$$\begin{aligned}
\bar{x}_t &= [h_{t-1}, x_t] \\
f_t &= \sigma\left(W_f \bar{x}_t + b_f\right) \\
i_t &= \sigma\left(W_i \bar{x}_t + b_i\right) \\
o_t &= \sigma\left(W_o \bar{x}_t + b_o\right) \\
\tilde{C}_t &= \tanh\left(W_{\tilde{C}} \bar{x}_t + b_{\tilde{C}}\right) \\
C_t &= f_t \diamond C_{t-1} + i_t \diamond \tilde{C}_t \\
h_t &= o_t \diamond \tanh\left(C_t\right)
\end{aligned} \tag{2.16}$$

Figura 7 – Interação entre elementos internos de uma unidade LSTM.



Na Equação 2.16, $\bar{x}_t$ representa a concatenação da representação vetorial do episódio anterior (h_{t-1}) com os sinais de entrada do episódio atual (x_t). O símbolo $\sigma(\cdot)$ equivale à aplicação de uma camada cuja função de ativação é a função sigmoide, e o símbolo $\diamond$ representa o produto de Hadamard. Os termos f_t, i_t, o_t mostram, respectivamente, a formulação do *forget gate*, *input gate*, e *output gate*, onde $W_{\{f,i,o,\tilde{C}\}}$ representam as matrizes de peso referente aos portões supracitados ou à matriz de peso para candidatos a estado $\tilde{C}$, e $b_{\{f,i,o,\tilde{C}\}}$ representam os *bias* aplicados às camadas de cada portão ou aos candidatos a estado $\tilde{C}$.

Analisando o funcionamento de uma LSTM de maneira sequencial, a primeira interação realizada é a do *forget gate*. O papel do portão de esquecimento é decidir o quanto será utilizado dos episódios anteriores, que estão armazenados em uma estrutura chamada de Célula de Estado, representada por C. Para isso, o portão usa o estado oculto do episódio anterior (h_{t-1}) juntamente ao estímulo de entrada (x_t) e determina, através de uma camada com função de ativação *softmax*, um valor entre 0 e 1 para cada valor da célula de estado C_{t-1}, sendo o valor 0 referente ao esquecimento total do valor da célula de estado, enquanto o valor 1 refere-se a manter totalmente o valor da célula de estado.

O próximo passo tem o papel de decidir o quanto da nova informação será armazenada na Célula de Estado. Isso ocorre em duas partes: primeiramente a camada *input gate* decide quais valores serão atualizados; em seguida a camada cuja ativação é *tanh* criará novos candidatos a serem adicionados ao estado final, sendo esses candidatos representados por $\tilde{C}_t$.

A partir dos cálculos feitos até aqui, é possível determinar o novo estado interno C_t.

É feito o produto de Hadamard do estado antigo C_{t-1} pelo resultado do *forget gate*, e a esse resultado é somado o produto de Hadamard entre a saída do *input gate* e os novos candidatos.

Apesar de ter sido encontrado um novo estado interno, ainda faz-se necessário decidir quanto desse novo estado será disponibilizado como saída para o próximo episódio da sequência. Para isso, o novo estado C_t passa por uma camada de $tanh$ e é aplicado o produto de Hadamard entre saída dessa camada e o resultado do *output gate*, com o propósito de filtrar o que será exposto como saída para outros componentes da rede.

O uso de LSTMs para modelagem de longas sequências apresenta vantagens em relação ao uso de RNNs simples, porém as LSTMs têm um maior número de parâmetros para serem treinados, o que faz surgir a necessidade de um maior conjunto de treinamento para que os resultados sejam satisfatórios.

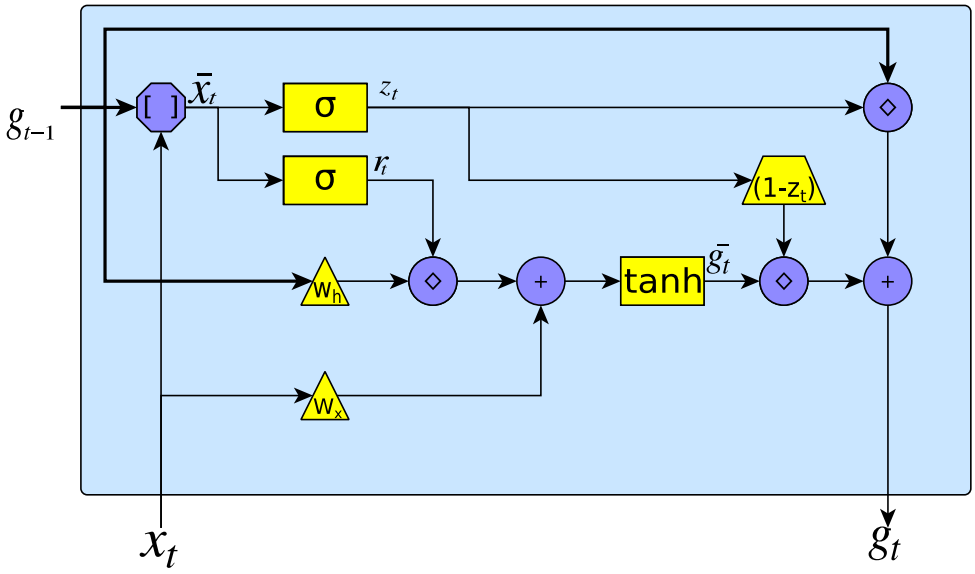
Como um intermediário entre a relativa simplicidade de RNNs comuns e a complexidade das LSTMs, tem-se as *Gated Recurrent Units* [34], que são estruturas com um menor número de portões para controlar seu estado interno, facilitando assim seu treinamento quando comparadas às LSTMs. A formulação dos portões e outros elementos utilizados pelas GRUs pode ser visto na Equação 2.17, onde z_t, r_t, $\bar{g}_t$ e g_t representam, respectivamente, o *Update gate*, o *Reset gate*, a Memória intermediária e a Memória final no episódio t

$$\begin{aligned} \bar{x}_t &= [g_{t-1}, x_t] \\ z_t &= \sigma\left(W_z \bar{x}_t + b_z\right) \\ r_t &= \sigma\left(W_r \bar{x}_t + b_r\right) \\ \bar{g}_t &= \tanh\left(r_t \diamond W_h g_{t-1} + W_x x_t\right) \\ g_t &= (1 - z_t) \diamond \bar{g}_t + z_t \diamond g_{t-1} \end{aligned} \tag{2.17}$$

A Figura 8 mostra a interação entre os elementos da GRU, onde o octógono representa a operação de concatenação entre dois vetores, os círculos representam o produto de Hadamard ou a operação de soma, os retângulos representam as camadas com suas respectivas funções de ativação, os triângulos representam multiplicação por uma matriz de pesos, e o trapézio exprime a expressão $(1 - z_t)$.

Apesar das GRUs apresentarem um treinamento mais fácil e resultados próximos, e até melhores em alguns casos [35, 36], grande parte das aplicações de Processamento de Linguagem

Figura 8 – Interação entre elementos internos de uma unidade GRU.



Natural fazem uso da rede LSTM.

Da maneira como foram descritas, as RNN processam as sequências da maneira como são apresentadas, ou seja, do início para o fim, incorporando assim informações dos episódios passados. Como forma de melhorar a representação gerada, foram propostas as Redes Neurais Recorrentes Bidirecionais (do inglês *Bidirectional Recurrent Neural Network*) [37], que além de incorporarem informações sobre os episódios anteriores, também incorporam informações dos episódios subsequentes. Para isso a sequência é apresentada para duas RNN, uma no sentido convencional, do episódio inicial para o episódio final, e uma no sentido inverso, do episódio final para o episódio inicial. Ao final do processamento das duas redes, seus vetores resultantes são combinados para serem utilizados para a classificação.

2.2.2.4 Campos Aleatórios Condicionais

Campos Aleatórios Condicionais, do inglês *Conditional Random Fields* (CRF) [38], é um modelo estatístico que tem por objetivo modelar sequências de eventos dependentes entre si. Esta abordagem tem sido utilizada para problemas desta natureza em PLN, como é o caso de segmentação de frases, etiquetagem de palavras e reconhecimento de entidades nomeadas. Seu funcionamento, mais especificamente do CRF de Cadeia Linear, se resume a encontrar a melhor sequência de saídas $\mathcal{Y}$ para uma sequência de entrada $\mathcal{X}$.

Dada uma sequência de tamanho n e um conjunto $\mathbb{C}$ de possíveis classes para cada amostra

da sequência, existem $|\mathbb{C}|^n$ sequências possíveis. Caso fosse preciso calcular a pontuação de todas as sequências possíveis de rótulos atribuídos para então decidir qual a melhor, o problema se tornaria intratável conforme n cresce.

Para evitar esse problema de intratabilidade, o CRF de Cadeia Linear encontra o melhor caminho utilizando o algoritmo de Viterbi. Isso é feito ao se calcular qual a melhor transição de cada vez por meio de uma pontuação. Desta maneira, somente a transição de maior pontuação é escolhida para compor o caminho final. Como a melhor sequência trata-se da composição das melhores transições individuais, o CRF contorna o problema de ter que calcular a pontuação de todas as $|\mathbb{C}|^n$ sequências possíveis ao fazer uso das pontuações de cada transição, reduzindo o problema para $n|\mathbb{C}|^2$.

O processo de escolha da melhor sequência é ilustrado pela Figura 9, que mostra uma estrutura chamada de treliça, onde a coluna à esquerda representa as classes que podem ser atribuídas à cada palavra, e no topo são apresentadas as palavras da sequência. Nesta figura é mostrado o caminho adotado para a frase-exemplo, com a adição de dois tokens, $\triangle$ e $\triangledown$, e duas classes ▲ e ▼, que representam, respectivamente, o início e o fim da frase e as classes que definem o início e o fim da frase, sendo que a adição desses *tokens* e classes está ligada ao funcionamento do CRF, sendo desprezadas no resultado final. Na figura, pode-se ver, em verde, a melhor sequência, que é encontrada através do algoritmo de Viterbi ao se considerar individualmente as transições de maior pontuação, formando assim a sequência com o maior valor de pontos, a qual é apresentada como saída do CRF.

Para que sejam encontradas as pontuações de cada transição, a Equação 2.18 é utilizada, a qual é responsável por retornar a melhor pontuação do episódio 1 até t.

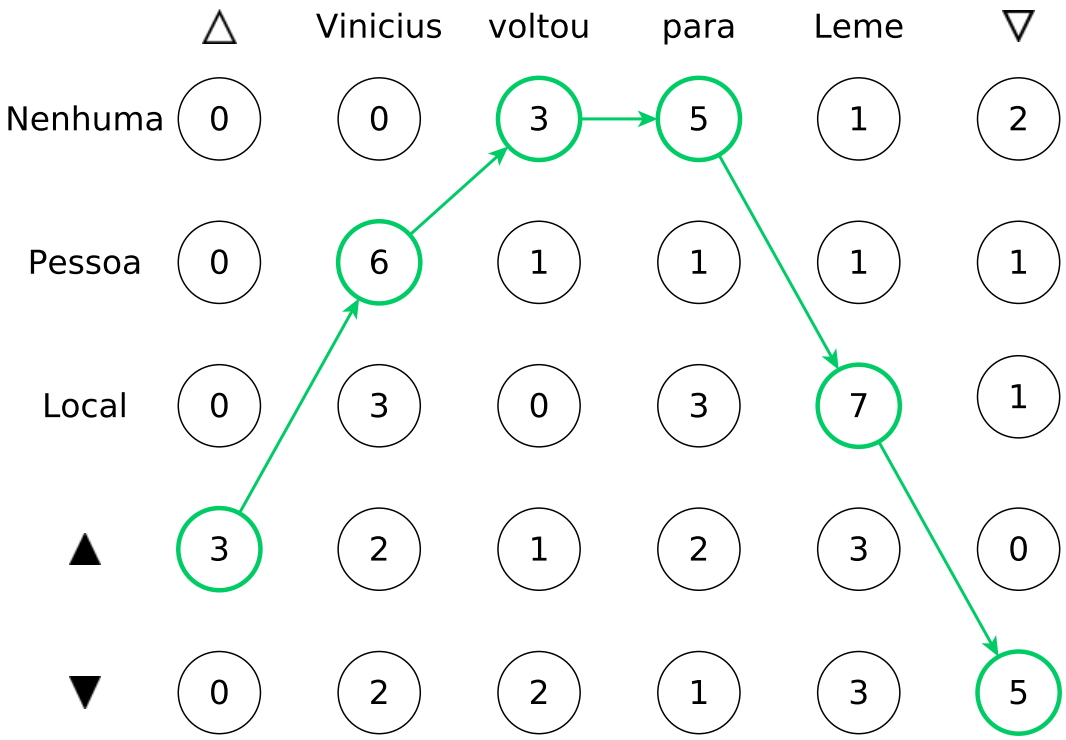
$$\alpha(t, y_t) = \begin{cases} g_1(\mathcal{X}, \blacktriangle, y_1) & \text{, se } t = 1 \\ \alpha(t-1, y_{t-1}) + g_t(\mathcal{X}, y_{t-1}, y_t) & \text{, caso contrário} \end{cases} \tag{2.18}$$

, onde $g_t(\mathcal{X}, y_{t-1}, y_t)$ é expresso pela Equação 2.19

$$g_t(\mathcal{X}, y_{t-1}, y_t) = \sum_i \lambda_i f_i(\mathcal{X}, y_{t-1}, y_t, t) \tag{2.19}$$

, onde $\mathcal{X}$ é um vetor com todos os episódios da sequência de entrada, y_{t-1} é o rótulo atribuído ao episódio $t-1$, y_t é o rótulo atribuído ao episódio t, t é o episódio atual, f_i é uma função característica, e λ_i é o peso dado à função característica i, o qual é aprendido durante o

Figura 9 – Exemplo de cálculo da melhor sequência de rótulos.



treinamento do CRF. Um mesmo modelo de CRF pode ter diversas funções características, como se y_t = Pessoa e X_t = Petrius, então $f_a = 1$, ou então se y_{t-1} = Pessoa, y_t = Pessoa, e X_t = Cesar, então $f_b = 1$.

Então, de maneira geral, é usada a função $\alpha(t, y_t)$ com seus argumentos sendo o rótulo extra, ▼, que simboliza o fim da sequência e o tempo $t = n + 1$, sendo então $\alpha(t_{n+1}, ▼)$, que desce recursivamente até o início da sequência, $t = 1$, para definir qual é o rótulo mais provável para iniciar a sequência. A partir daí são decididas quais transições geram os rótulos de maiores pontuações, um passo por vez até que o caminho ótimo (correspondente à melhor sequência de rótulos) tenha sido encontrado, conforme ilustrado pela Figura 9. Nessa figura pode-se observar que a frase se inicia com o *token* sintético △, responsável por indicar o início de uma frase. Assim, para se atribuir um rótulo à palavra "Vinicius", é utilizada a função $\alpha(1, ▲)$, a qual, segundo a Equação 2.18, equivale a $g_1(X, ▲, y_1)$, que é responsável por encontrar as pontuações para os possíveis rótulos da palavra "Vinicius". A partir dessas pontuações, é escolhido o rótulo correspondente a maior delas, que nesse caso é "Pessoa". Depois de se encontrar o rótulo de "Vinicius", pode-se então encontrar o rótulo da próxima palavra, "voltou", por meio da função $\alpha(2, Pessoa)$, que equivale a $\alpha(1, ▲) + g_2(X, Pessoa, y_2)$. Seguindo esse pensamento até o fim da sequência é então encontrada a melhor sequência de rótulos para a sequência de entrada X.

Assim, ao considerar somente o item anterior, pode-se realizar a inferência da sequência de maneira gulosa por meio de programação dinâmica na ordem de $n|\mathbb{C}|^2$, ao invés de $|\mathbb{C}|^n$ que se tinha originalmente.

Ao se utilizar o CRF junto às Redes Neurais, o CRF geralmente tem a função de substituir a última camada (de classificação), recebendo assim como entrada os vetores que foram extraídos pelas camadas anteriores da rede. Deste modo, o X do CRF trata-se da saída da rede que o antecede. Neste trabalho, por exemplo, o vetor de entrada do CRF é o vetor resultante da camada LSTM Bidirecional, como é detalhado em 5.3.

2.2.3 Abordagens Híbridas

As abordagens híbridas fazem uso de Abordagens Simbólicas e Abordagens Numéricas a fim de unir as vantagens e compensar eventuais desvantagens que se fariam presentes com o dessas abordagens isoladamente. A combinação de ambos os tipos de abordagens pode ser feita por meio da inclusão de *features* semânticas ou pela combinação dos resultados provenientes de abordagens distintas.

No caso da inclusão de *features* semânticas é comum ver o uso dos repositórios de Grafo de Conhecimento e de Ontologias como fontes de consulta para a adição dessas *features* em uma Abordagem Numérica, como ocorre em [39]. O processo geralmente envolve verificar se um termo está definido no repositório de referência e utilizar essa informação de pertinência, bem como outras propriedades referentes ao elemento, como *feature* de entrada para uma Abordagem Numérica.

Nas abordagens que fazem o uso dos resultados provenientes de mais de um extrator, é necessária uma estratégia para a combinação dos resultados. Como estratégias para a combinação dos métodos, em [4] são citadas as técnicas de seleção e integração dos resultados dos extratores.

A técnica de seleção visa determinar o melhor extrator para cada conceito a ser extraído. Para isto é escolhido o extrator com maior acurácia para cada um dos conceitos a serem extraídos. A fim de se escolher o melhor extrator, é calculado o nível de erro de cada implementação, para então escolher aquela com menor erro para aquele conceito.

Já para a técnica de integração, os resultados de cada extrator são combinados para melhorar a acurácia do sistema. Uma das maneiras de realizar a integração faz uso da técnica chamada *stacking*, que consiste em fazer uso dos resultados dos extratores para treinar um modelo

de alto nível para escolher qual o melhor método para cada conceito a ser extraído.

Nothman *et al.* [40] fazem uso da técnica de votação utilizando o mesmo extrator para línguas diferentes. No entanto, é possível utilizar esse raciocínio para diferentes classificadores que fazem uso de diferentes abordagens. A votação consiste em escolher a classe que a maioria dos extratores atribuiu à amostra. Para uso dessa técnica é preciso estabelecer um critério de desempate, que pode ser algo simples, como utilizar um classificador como padrão em caso de empates.

3 Trabalhos Relacionados

Este capítulo tem o propósito de elencar alguns trabalhos da área de Reconhecimento de Entidades Nomeadas. A partir do levantamento realizado foi possível tomar ciência dos métodos de extração que vêm sendo utilizados pela comunidade, além de averiguar os métodos de vetorização e quais bases de dados são utilizadas experimentalmente.

3.1 Tópicos Identificados

As pesquisas na tarefa de Reconhecimento de Entidades Nomeadas têm alcançado grandes evoluções e novas técnicas têm sido desenvolvidas. A revisão bibliográfica realizada tinha dois objetivos: identificar as abordagens recentemente desenvolvidas; identificar os principais recursos utilizados para a tarefa. Os principais tópicos de interesse considerados nesta análise foram os seguintes:

- Recursos utilizados: consistiu em identificar as novas abordagens utilizadas no preprocessamento dos textos e na representação vetorial das palavras;
- Abordagens de aprendizado: consistiu em identificar as novas abordagens de Aprendizado de Máquina e em qual parte do processo as abordagens são adotadas.

Os tópicos de interesse inicialmente identificados foram utilizados como base para o levantamento dos trabalhos. Após a análise dos trabalhos, foi definida a abordagem inicial a ser desenvolvida. A medida que novos métodos foram identificados essa abordagem foi sendo aprimorada.

Na próxima sessão os principais artigos identificados são discutidos considerando os tópicos de interesse definidos acima.

3.2 Revisão Bibliográfica

Em [41], é realizada, por meio de regras, a extração de informações sobre condições de tráfego, tais como hora, início do percurso descrito, final do percurso descrito e condições do tráfego no percurso, presentes em mensagens do Twitter. Como documentos de entrada do sistema

tem-se os *tweets* do perfil do Centro de Gerenciamento de Tráfego da polícia de Jakarta, Indonésia. Tais *tweets* foram escolhidos pelos autores do trabalho por serem de uma fonte confiável de informações e por serem bastante homogêneos. Os textos são anotados sintaticamente com a finalidade de simplificar a tarefa de criação das regras, uma vez que regras que usam a classe gramatical das palavras são mais fáceis de se elaborar devido à sua generalidade.

As abordagens baseadas em Aprendizado de Máquina vêm sendo utilizadas na tarefa de Extração de Informação. Neste sentido, em [42] é apresentado o uso de uma Rede Neural Convolucional em conjunto a uma rede LSTM Bidirecional (BiLSTM) para a identificação das entidades nomeadas do tipo Quem, O Quê, Quando, Onde, Como, e Por quê (do inglês 5W1H, que significa *Who*, *What*, *When*, *Where*, *Why* e *How*). O papel da CNN é identificar as relações sintáticas e semânticas a nível de frase, enquanto a Bi-LSTM visa encontrar as relações entre as palavras da frase.

Em [43] a extração de Entidades Nomeadas é realizada através de dois modelos, sendo um deles uma rede LSTM bidirecional ligada a uma camada de *Contitional Random Fields* (CRF), e o outro trata-se de um *chunker* que faz uso de uma rede chamada de Stack-LSTM, que é uma modificação da LSTM. Os *datasets* utilizados foram o CONLL2002 (nos idiomas Alemão e Holandês) e CONLL2003 (nos idiomas Inglês e Espanhol), sendo que ambos apresentavam 4 tipos diferentes de entidades: Localização, Pessoa, Organização e Miscelânea. Foi-se utilizado BiLSTM juntamente ao CRF com o intuito de incorporar a informação de sequência, isto é, considerar os elementos anteriores da frase. Por outro lado, o *chunker* faz uso das representações geradas pela Stack-LSTM e os representa através de pilhas, e a partir de operações sobre essas pilhas é possível calcular a distribuição de probabilidade das classes.

Uma outra abordagem é proposta em [44], onde é feito o uso de CRF para realizar a extração de informações contidas nos artigos da Wikipédia, e estruturar essas informações no formato Infobox, que é comumente presente nos artigos da Wikipédia. Para isso os autores fazem uso de características da palavra atual, como tamanho e posições relativas, e também fazem uso de outras características em uma janela que abrange 5 palavras antes e 5 depois da palavra atual, fazendo uso de POS tag, presença de pontuação, tipo de *token*, entre outras.

Em [45] é proposto o uso de dois extratores que fazem uso de Aprendizado de Máquina, sendo um deles treinado em um *dataset* semelhante ao que será classificado. Esse extrator "auxiliar" é utilizado para enriquecer as características das representações vetoriais dos dados

que serão utilizados pelo classificador principal. A base de dados "auxiliar" utilizada foi a CoNLL2003, fazendo uso das entidades Pessoa, Organização e Localização, enquanto a base de dados principal era a CMU *Seminar Announcements*, que contém as entidades Nome, Localização, Horário de início, e Horário de Término. Além disso, também fazem uso de *Gazetteer Lists* como características para os classificadores. Em *Machine Learning*, soluções que fazem uso de aprendizados adquiridos anteriormente com bases de dados semelhantes são chamadas de *Transfer Learning*. Apesar de fazer a inclusão de informações de *Gazetteer*, não há a comparação com a versão sem o uso de informações externas, portanto não é possível quantificar o impacto do uso do *Gazetteer*

Uma abordagem que faz uso de Regras em conjunto com Vocabulários de Termos é apresentada em [46]. A finalidade é identificar crimes relacionados a drogas em notícias presentes na internet. Dessas notícias deseja-se extrair informações como esconderijo das drogas, nacionalidade dos traficantes, tipos de drogas, quantidade de drogas, e o preço das mesmas no mercado local, com o objetivo de melhorar a inteligência no combate às drogas. A metodologia adotada faz uso de regras e também de Vocabulários de Termos em combinação com as anotações sintáticas para facilitar a identificação das informações. Os Vocabulários de Termos utilizados são de dois tipos: um contém os nomes de drogas que são alvo da extração; e o outro contém as palavras e as expressões que servem como indicadores de uma categoria, chamadas pelos autores de *Indicator Words*, como "preso por", "embaixo de", "em sacolas plásticas", entre outras. A partir das características dos padrões citados, é realizada a extração das entidades.

Um trabalho apresentando a Extração de Informação na área de petróleo é realizado por Furtado [47], que faz o uso de métodos de processamento de Linguagem Natural e uma ontologia de domínio para a identificação de entidades nomeadas e relações entre estas entidades. Posteriormente, realiza a extração dessas entidades e relações por meio de algoritmos de Aprendizado de Máquina. Neste trabalho também não há comparação dos resultados com e sem uso de informações externas, portanto não é possível quantificar o impacto de seu uso.

Chiu e Nichols [48] apresentam uma solução para Reconhecimento de Entidades Nomeadas que faz uso de CNN para captar as principais características dos caracteres que compõem uma palavra, e utilizam redes BiLSTM para manter informação de palavras anteriores e fazer uso da dependência entre elas no momento da classificação. Além disso, também fazem uso de conhecimento externo por meio de *lexicons*, o que providencia um aumento de 0,71% em F1-Score.

Em [49] é apresentado, por Amaral, o Reconhecimento de Entidades Nomeadas fazendo uso de informações presentes em *Gazetteer* por meio de três métodos, sendo eles: J48 Decision Tree; Naïve Bayes e CRF. A base de dados utilizada é proposta pela própria autora do trabalho, e trata sobre o domínio de Geologia, mais especificamente Bacias Sedimentares Brasileiras. Segundo os experimentos, o classificador com melhor resultado de F1-Score foi o CRF, seguido pelo Naïve Bayes. A autora discute também que o uso de POS *tags* foi uma característica que contribuiu fortemente com os resultados. Neste trabalho foi feita a comparação entre diferentes classificadores, porém não houve análise do impacto da informação semântica utilizada.

Habibi *et al.* [50] apresenta a tarefa de Reconhecimento de Entidades Nomeadas no domínio biomédico. Inicialmente, são investigados os impactos do uso de diferentes métodos de vetorização baseados em *word embeddings*. Posteriormente, é também realizada a extração com diferentes modelos: com o BiLSTM-CRF; com o modelo CRF puro; e com outros métodos *baseline* (que diferem de acordo com as entidades extraídas). São extraídos 5 diferentes tipos de Entidades presentes em um total de 33 *datasets*. Os experimentos mostram que o modelo BiLSTM-CRF teve o melhor resultado em 28 das 33 bases de dados utilizadas.

Em [51] Ling e Weld propõem uma abordagem que utiliza uma granularidade maior para a tarefa de NER. Os autores propõem um sistema para o Reconhecimento de Entidades Nomeadas que funciona em duas etapas: na primeira as frases são segmentadas em candidatos à Entidade Nomeada através do uso do CRF; na segunda etapa os segmentos candidatos são classificados como uma ou mais Entidade, tratando-se de um problema de classificação multiclasse. O classificador utilizado nesta segunda fase é uma modificação do Perceptron. O *dataset* utilizado contém 112 classes, e os experimentos apontam uma melhora de F1-Score em relação aos outros métodos utilizados na comparação.

Liu *et al.* [52] realiza a introdução de conhecimento externo na tarefa de NER por meio de *Gazetteers*. Para isso, um classificador adicional é treinado para gerar uma representação numérica da informação sobre pertinência a um *Gazetteer*, para posteriormente concatenar essa representação numérica à saída de uma BiLSTM. O vetor final é então utilizado como entrada para um Hybrid semi-Markov CRF (HSCRF). Para a geração das representações dos *Gazetteers* também é utilizado HSCRF. A intenção é essa representação numérica tomar o lugar dos *hard token matches* geralmente utilizados quando se faz uso de *Gazetteers*. O uso de *Gazetteers* trouxe ganho de 0,21% em F1-Score para o *dataset* CONLL2003 e de 0,56% em F1-Score para o *dataset* OntoNotes 5.

Uma abordagem que integra árvores de dependência à redes neurais para a tarefa de NER é apresentada em [53]. A estratégia utilizada consiste em fazer uso de múltiplas camadas BiLSTM que agregam as dependências entre as palavras, para no final servirem de entrada para um CRF. Como entrada da primeira rede, tem-se a concatenação do *word embedding* de cada palavra com o *embedding* da palavra pai e a representação vetorial do tipo da dependência. A saída dessa rede é então utilizada em uma função, chamada de função de interação, que faz a combinação (concatenação, adição, ou até mesmo uma rede neural) da saída refente à palavra atual e a saída referente à palavra pai. Cada um desses encadeamentos de BiLSTM com função de interação pode ser replicado mais de uma vez, gerando mais camadas de abstração. Essa adição de árvores de dependência mostrou ganho no reconhecimento de entidades que são compostas de várias palavras.

O uso de uma Ontologia como fonte de conhecimento externo é mostrado por Liu e El-Gohary [39], onde os autores realizam a tarefa de NER em relatórios sobre o estado de conservação de pontes, utilizando uma variação semi-supervisionada do CRF. Como entrada para o CRF são utilizadas informações sintáticas, como *Stem* e POS *tags*, e também informações semânticas, como a pertinência das palavras à uma ontologia de domínio, chamada de BridgeOnto. O acréscimo de informações semânticas trouxe um ganho médio de 7,6% de F1-Score.

Louvan e Magnini [54] realizam a tarefa de *Multi-task learning* para treinar conjuntamente um modelo que faça *Slot filling*, Reconhecimento de Entidades Nomeadas e Anotação Semântica. O objetivo é fazer uso de dados de outras tarefas junto à desejada como maneira de combater a escassez de dados anotados.

Liu *et al.* [55] propõem uma metodologia para Reconhecimento de Entidades Nomeadas fazendo uso de Semi-Markov CRFs e diversas características, como POS *tags*, N-grams, lexicons, entre outras, para uso em sistemas de voz. Além disso, fazem a criação de três *datasets*, sendo dois deles utilizados neste presente trabalho. O uso de lexicons trouxe um ganho de 1,16% em F1-Score para o dataset de filmes e ganho de 1,7% para o dataset de restaurantes.

Em [56] é apresentado um novo método para o pré-treinamento de modelos que fazem a representação vetorial das palavras de um texto, e esse novo modelo é validado em diversas tarefas de PLN, sendo NER uma delas. Para a validação desse novo método no Reconhecimento de Entidades Nomeadas é utilizada uma rede BiLSTM-CRF com hiperparâmetros otimizados, atingindo o valor de estado da arte para o *dataset* CONLL2003.

A tarefa de NER é encarada como *Machine Reading Comprehension* em [57], fazendo uso de uma Função de Perda chamada de *Dice-loss* que visa combater o desbalanceamento entre a quantidade de classes de um *dataset*. Desta maneira, atingem o resultado de estado da arte para a base OntoNotes5.

3.3 Síntese da Revisão Bibliográfica

A partir dos trabalhos apurados, pôde-se identificar a tendência recente das pesquisas na tarefa de NER, como o uso dos métodos de Aprendizado de Máquina em conjunto com *word embeddings*. Isso ocorre pois o uso dessas técnicas permite tratar o problema de NER como um problema de classificação; permite o mapeamento das informações textuais específicas do domínio tratado para representações numéricas facilmente processadas por métodos de Aprendizado de Máquina; e, apresentam resultados considerados muito satisfatórios. Porém, é comum verificar trabalhos que fazem o uso dessas técnicas juntamente à fontes de conhecimento externo, ou características particulares da aplicação.

Desta maneira, foi possível tomar ciência dos métodos atualmente utilizados para a tarefa de NER e identificar os trabalhos [48] e [43], que apresentam arquiteturas utilizando métodos modernos de Aprendizado de Máquina e com importantes resultados na área. Neste contexto, baseada nesses trabalhos, foi proposta uma abordagem que faz uso de técnicas atuais de Aprendizado de Máquina, porém realiza a inclusão de fontes externas de conhecimento.

A síntese dos trabalhos é apresentada na Tabela 1, sendo apresentadas algumas das características dos trabalhos relacionados como: a Abordagem utilizada de acordo com a definição do capitulo 2; os Métodos Utilizados de acordo com a abordagem; e, os Recursos Utilizados nas tarefas de tratamento das palavras e dos textos.

Tabela 1 – Síntese dos trabalhos relacionados.

Trabalho	**Abordagem**	**Métodos Utilizados**	**Recursos Utilizados**
[41]	Simbólica	Regras	POS *tags*
[42]	Numérica	Redes Neurais	*Word embeddings*, posição na frase, posição no texto
[43]	Numérica	CRF, Redes Neurais, *Chunker*	*Word embeddings*
[44]	Numérica	CRF	POS *tags*, tipo do *token*, tamanho do *token*, entre outros
[45]	Híbrida	Máxima Entropia, *Gazetteer*	POS *tags*, posição na frase, unigramas e bigramas, entre outras
[46]	Simbólica	Regras, *Gazetteer*	POS *tags*, *Phrase chunks*
[47]	Híbrida	Support Vector Machine, Random Forest	Ontologia, POS *tags*, lemas, entre outros
[48]	Híbrida	Redes Neurais	*Word embeddings*, Lexicon
[49]	Híbrida	CRF, Decision Tree, Naïve Bayes, *Gazetteer*	POS *tags*, prefixos, sufixos, outros
[50]	Numérica	Redes Neurais	*Word embeddings*
[51]	Numérica	CRF, Redes Neurais	POS *tags*, tamanho do token, unigramas e bigramas, entre outros
[52]	Híbrida	HSCRF, Redes Neurais, *Gazetteer*	*Word embeddings*
[53]	Numérica	CRF, Redes Neurais	*Word embeddings*, árvores de dependências
[39]	Híbrida	CRF	POS *tags*, Stems Ontologia
[54]	Numérica	CRF, Redes Neurais	*Tags* semânticas, NER de outro domínio
[55]	Híbrida	Semi-Markov CRF	Lexicon, POS *tags*, N-grams, entre outros
[56]	Numérica	Redes Neurais	*Word embeddings*
[57]	Numérica	Redes Neurais	*Word embeddings*

Fonte – Autor

4 Materiais e Métodos

Este capítulo descreve os *datasets* usados neste trabalho, bem como as fontes de conhecimento externo utilizadas. Ainda neste capítulo são apresentadas as métricas adotadas para se medir o desempenho dos métodos utilizados para Reconhecimento das Entidades Nomeadas.

4.1 *Datasets*

A partir da revisão bibliográfica realizada foi possível identificar os *datasets* utilizados pela comunidade para a tarefa de NER. Após isso foram escolhidos cinco *datasets*, todos escritos no idioma Inglês, elaborados a partir de diferentes fontes textuais. Os detalhes dos *datasets* utilizados são descritos nas subseções seguintes.

Neste trabalho, a maneira como as entidades presentes nos textos são representadas seguem o padrão de anotação *Inside-Outside-Beginning* (IOB). Desta forma, para cada frase do texto, como ilustrado na Figura 10, as entidades de interesse são anotadas, na qual cada tipo de entidade é dividida em 2 classes, uma representando o início da entidade e outra representando a parte "interna" da entidade, além de uma classe em comum para palavras que não são Entidades. Assim, na frase da Figura 10, as entidades anotadas são Pessoa e Localidade, sendo que a palavra "Maria" esta associada ao início (anotado com B-Pessoa) e "Eduarda" associada ao final (anotado com I-Pessoa). Na Tabela 3 é apresentado o exemplo de uma frase com as anotações dos termos com as classes IOB correspondentes. A adoção deste tipo de anotação é recomendada por facilitar o reconhecimento das entidades nomeadas que frequentemente ocorrem de forma sequencial. Como consequência, a adoção deste tipo de formato permitem bons resultados para os classificadores que fazem uso de sequências temporais.

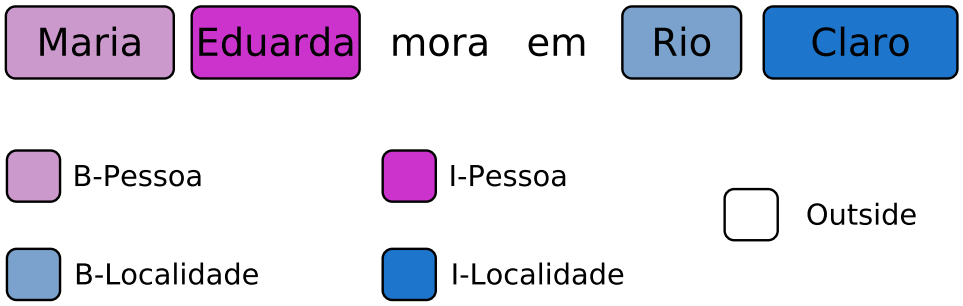
Alguns dos *datasets* utilizados neste trabalho sofreram pequenas alterações em relação aos originais, como divisão de treinamento e teste, ou remoção de tipos de entidades que só aparecem uma vez. Todas as mudanças são explicadas junto ao repositório onde são encontradas as bases de dados [7].

[7] <https://github.com/juand-r/entity-recognition-datasets>

4.1.1 CoNLL2003

Criado no ano de 2003 como uma tarefa compartilhada da *Conference on Computational Natural Language Learning* [16], trata-se do *dataset* mais utilizado para as tarefas de NER, e é dividido em conjunto de treinamento, conjunto de teste, e conjunto de desenvolvimento. Por ser um *dataset* largamente utilizado em experimentos, o formato como as palavras estão dispostas é considerado um padrão para *datasets* de Reconhecimento de Entidades Nomeadas, sendo este padrão mostrado na Tabela 2.

Figura 10 – Exemplo da notação IOB.



Fonte – Autor.

Tabela 2 – Exemplo do padrão de dados CoNLL2003.

Token	**POS tag**	**Chunk**	**Classe**
The	DT	B-NP	O
former	JJ	I-NP	O
Soviet	JJ	I-NP	B-MISC
republic	NN	I-NP	O
was	VBD	B-VP	O
playing	VBG	I-VP	O
in	IN	B-PP	O
an	DT	B-NP	O
Asian	NNP	I-NP	B-MISC
Cup	NNP	I-NP	I-MISC
finals	NNS	I-NP	O
tie	NN	I-NP	O
for	IN	B-PP	O
the	DT	B-NP	O
first	JJ	I-NP	O
time	NN	I-NP	O
.	.	O	O

Fonte – Autor.

Apesar do *dataset* ser dividido em conjuntos de treinamento, teste e validação, neste trabalho foram utilizados somente os conjuntos de treinamento e teste. A composição do conjunto de frases e *tokens* destas divisões pode ser vistas Tabela 3.

Neste *dataset* os *tokens* anotados pertencem a uma das quatro entidades: *Location*, *Miscellaneous*, *Organization* e *Person*. A distribuição das entidades é mostrada pela Tabela 4.

Tabela 3 – Quantificação dos conjuntos do *dataset* CoNLL2003.

Conjunto	**Frases**	**Tokens**
Treinamento	14.987	204.567
Teste	3.684	46.666
Total	**18.671**	**251.233**

Fonte – Adaptado de [16].

Tabela 4 – Distribuição das entidades do *dataset* CoNLL2003.

Entidade	**Treinamento**	**Teste**
Location	7.140	1.668
Miscellaneous	3.438	702
Organization	6.321	1.661
Person	6.600	1.617
Total	**23.499**	**5.648**

Fonte – Adaptado de [16].

4.1.2 OntoNotes5

O *dataset* OntoNotes, lançado em 2013, encontra-se na quinta e última edição do projeto, reunindo assim o conteúdo de todas as outras edições anteriores, além de conteúdo adicional. Foi criado a partir diferentes fontes de dados, como notícias, diálogos de telefone, *weblogs*, entre outros, em três línguas. Neste *dataset* cada *token* pode ser anotado em uma de dezoito entidades, embora mantenha três entidades em comum (*Location*, *Organization* e *Person*) com o *dataset* CoNLL2003. Além da anotação de Entidades Nomeadas, apresenta também anotações sintáticas e anotações rasas de semântica.

A quantificação dos conjuntos de treinamento e teste e distribuição das entidades no *dataset* OntoNotes 5 são mostradas pelas Tabelas 5 e 6, respectivamente.

Tabela 5 – Quantificação dos conjuntos do *dataset* OntoNotes 5.

Conjunto	**Frases**	**Tokens**
Treinamento	115.812	2.200.865
Teste	12.217	230.118
Total	**128.029**	**2.430.983**

Fonte – Autor.

Tabela 6 – Distribuição das entidades do *dataset* OntoNotes 5.

Entidade	**Treinamento**	**Teste**
PERSON	22.035	2.134
ORG	24.163	2.002
NORP	9.341	990
QUANTITY	1.240	153
DATE	18.791	1.787
GPE	21.938	2.546
LOC	2.160	215
MONEY	5.217	355
EVENT	1.009	85
LAW	459	44
ORDINAL	2.195	207
LANGUAGE	355	22
WORK OF ART	1.279	169
PRODUCT	992	90
CARDINAL	10.901	1.005
TIME	1.703	225
FAC	1.158	149
PERCENT	3.802	408
Total	**128.738**	**12.586**

Fonte – Autor.

4.1.3 GUM

Proposto em [58], o Georgetown University Multilayer (GUM) trata-se de um *dataset* criado a partir de oito gêneros textuais, sendo tais gêneros selecionados por representarem diferentes propósitos comunicativos. Os gêneros textuais são: entrevistas; notícias; guias de viagem; guias de instruções; textos acadêmicos; biografias; ficção; discussões de fórum.

Neste *dataset* cada *token* pode ser anotado em uma de onze entidades, embora mantenha três entidades em comum (*Place*, *Organization* e *Person*) com o *dataset* CoNLL2003. Assim, este *dataset* contém um total de 11.724 entidades, cuja distribuição é mostrada pela Tabela 7.

Em relação às quantidades de frases e *tokens*, na Tabela 8 são mostradas as quantidades que compõem este *dataset*.

Tabela 7 – Distribuição das entidades do *dataset* GUM.

Entidade	**Treinamento**	**Teste**
Object	1.017	420
Abstract	2.002	798
Person	1.920	823
Place	1.150	469
Organization	397	192
Quantity	97	44
Event	738	315
Substance	278	95
Time	401	179
Plant	144	62
Animal	141	42
Total	**8.285**	**3.439**

Fonte – Autor

Tabela 8 – Quantificação dos conjuntos do *dataset* GUM.

Conjunto	**Frases**	**Tokens**
Treinamento	2.494	44.111
Teste	999	18.236
Total	**3.493**	**62.347**

Fonte – Autor

4.1.4 MIT Movies

Criado pelo grupo *Spoken Language Systems* do *Massachusetts Institute of Technology* [55], este *dataset* foi elaborado a partir de *reviews* de filmes.

Neste *dataset* cada *token* pode ser anotado em uma de doze entidades. Assim, as entidades e a distribuição das 28.721 entidades em treinamento e teste é mostrada pela Tabela 9.

Em relação as quantidades de frases e *tokens*, na Tabela 10 são mostradas as quantidades que compõem este *dataset*.

4.1.5 MIT Restaurants

Assim como o *dataset* MIT Movies, este *dataset* foi elaborado pelo grupo *Spoken Language Systems* [55] a partir de *reviews*, porém neste caso foram *reviews* de restaurantes.

Neste *dataset* cada *token* pode receber a anotação referente à uma de oito entidades. Assim, as entidades e a distribuição das 18.514 entidades em treinamento e teste é mostrada pela Tabela 11.

Tabela 9 – Distribuição das entidades do *dataset* MIT Movies.

Entidade	**Treinamento**	**Teste**
Actor	5.010	1.274
Plot	6.468	1.577
Opinion	810	195
Award	309	66
Year	2.702	661
Genre	3.384	789
Origin	779	195
Director	1.787	425
Soundtrack	50	8
Relationship	580	171
Character Name	1.025	283
Quote	126	47
Total	**23.030**	**5.686**

Fonte – Autor

Tabela 10 – Quantificação dos conjuntos do *dataset* MIT Movies.

Conjunto	**Frases**	**Tokens**
Treinamento	7.816	158.823
Teste	1.953	39.035
Total	**9.769**	**197.858**

Fonte – Autor

Em relação às quantidades de frases e *tokens*, na Tabela 12 são mostradas as quantidades que compõem este *dataset*.

Tabela 11 – Distribuição das entidades do *dataset* MIT Restaurants.

Entidade	**Treinamento**	**Teste**
Rating	1.070	201
Amenity	2.541	533
Location	3.817	812
Restaurant Name	1.901	402
Price	730	171
Hours	990	212
Dish	1.475	288
Cuisine	2.839	532
Total	**15.363**	**3.151**

Fonte – Autor

Tabela 12 – Quantificação dos conjuntos do *dataset* MIT Restaurants.

Conjunto	**Frases**	**Tokens**
Treinamento	7.660	70.525
Teste	1.521	14.256
Total	**9.181**	**84.781**

Fonte – Autor

4.1.6 Síntese dos *datasets*

A partir dos dados apresentados nas subseções anteriores, foi elaborada a Tabela 13, que trata-se de uma tabela-síntese que faz um compilado das características de cada *dataset*, tais como tipos diferentes de entidades, gêneros textuais que o compõe, quantidade de frases e quantidade de entidades presentes.

Tabela 13 – Quantificação das sentenças e entidades contidas nos *datasets*.

Dataset	**Gêneros**	**Tipos de Entidades**	**# Frases**	**# Entidades**
CONLL2003	Notícias	4	18.671	29.147
OntoNotes	Conversas telefônicas, notícias, weblogs, entre outros	18	12.802	141.324
GUM	Entrevistas, Notícias, Guias de viagem, entre outros	11	3.493	11.724
MIT M	Reviews de filmes	12	28.716	9.769
MIT R	Reviews de restaurantes	8	18.514	9.181

4.2 Repositórios de conhecimento

Embora as anotações dos *datasets* têm sido feitas com entidades que em geral são as mesmas ou semânticamente próximas, essas entidades são vinculadas apenas aos *datasets* em que foram criadas, não podendo ser reaproveitadas fora deles. Outra questão importante é a atribuição de diferentes entidades para um mesmo *token*. Uma das principais abordagens adotada para resolver esse problema tem sido a criação de Repositórios de Conhecimento, que são coleções compartilhadas de conceitos semanticamente anotados, armazenados de maneira estruturada, geralmente na forma de triplas, para serem consultados por humanos ou máquinas.

Para agregar conhecimento externo aos métodos de processamento de linguagem, vários grupos de pesquisa e empresas têm criado repositórios públicos e privados. Neste trabalho foram utilizados dois dos principais repositórios de conhecimento disponíveis: Freebase e YAGO.

4.2.1 Freebase

Criado em 2007, o repositório de conhecimento denominado Freebase permitiu a construção colaborativa de relações entre entidades e dados. Desta forma foi gerado um repositório que permitiu agregar o conhecimento proveniente de diversas fontes, principalmente editado por parte de sua comunidade. Em 2010, a Google adquiriu a empresa provedora do Freebase e o serviço permaneceu aberto para a comunidade até ser descontinuado em 2016. Atualmente o repositório esta disponível em um arquivo *dump* com bilhões de triplas em RDF.

O modelo conceitual utilizado no Freebase permite definir, na forma de triplas, entidades ou objetos distintos, denominados de tópicos. A cada tópico, por exemplo, Steven Spielberg, é associado um identificador único, tal como no exemplo abaixo é "/m/abc123".

/m/abc123 /type/object/type /people/person

/m/abc123 /type/object/type /film/film_director

O tópico pode ser membro de uma classe, como no exemplo Steven Spielberg é membro da classe *Person* e *Film_Director*. Os valores específicos associados a um tópico são definidos por suas propriedades. Uma propriedade pode ser definida para relacionar dois tópicos, por exemplo, a propriedade o local de nascimento (/people/person/place_of_birth) relaciona o tópico "/m/abc123" com "/m/c1nc1na771". Um fato pode ser definido por uma propriedade ao associar um valor ao tópico, por exemplo, o tópico "/m/abc123" pode ter a data do nascimento associada. Abaixo são mostrados exemplos de propriedades e fatos relativos ao tópico "/m/abc123".

/m/abc123 /people/person/place_of_birth /m/c1nc1na771

/m/abc123 /film/film_director/films /m/5om3f1lm

/m/abc123 /people/person/date_of_birth "Dec 18, 1946"

Os dados da Freebase estão disponíveis em RDF no formato N-Triples e conforme mencionado, estão organizados em tópicos. Em janeiro de 2014 este repositório era composto por aproximadamente 44 milhões de tópicos e 2.4 bilhões de fatos.

Neste trabalho será utilizado o Freebase como uma fonte externa com a finalidade de reduzir a ambiguidade das entidades. O uso do repositório do Freebase requer uma infraestrutura computacional e as consultas demandam grande tempo de resposta. Uma forma adotada tem sido o desenvolvimento de métodos que permitem mapear um repositório de triplas, como o Freebase, em representações vetoriais sem perder o conteúdo representado. Um desses métodos é o *Translating Embeddings for Modeling Multi-relational Data* (TransE) [59]. Utilizando o algoritmo TransE, disponível na ferramenta OpenKE [60], foram gerados os *embeddings* da Freebase.

Para se fazer uso dos *Knowledge embeddings* como fonte externa, foram utilizados os *embeddings* já treinados[8]. Porém inicialmente foi necessária a realização de um mapeamento do código do Freebase com a palavra correspondente, por exemplo o código "/m/0h16" está associado a palavra "Altruism". Após esse mapeamento, é feito um novo processo no qual é feita a geração do repositório associa as palavras aos seus *embeddings*.

Ao se utilizar esses *Knowledge embeddings*, a ideia é recuperar, para cada *token*, os vetores de 50 dimensões correspondentes e agregar aos vetores de características que servem de entrada para os métodos de classificação.

4.2.2 YAGO

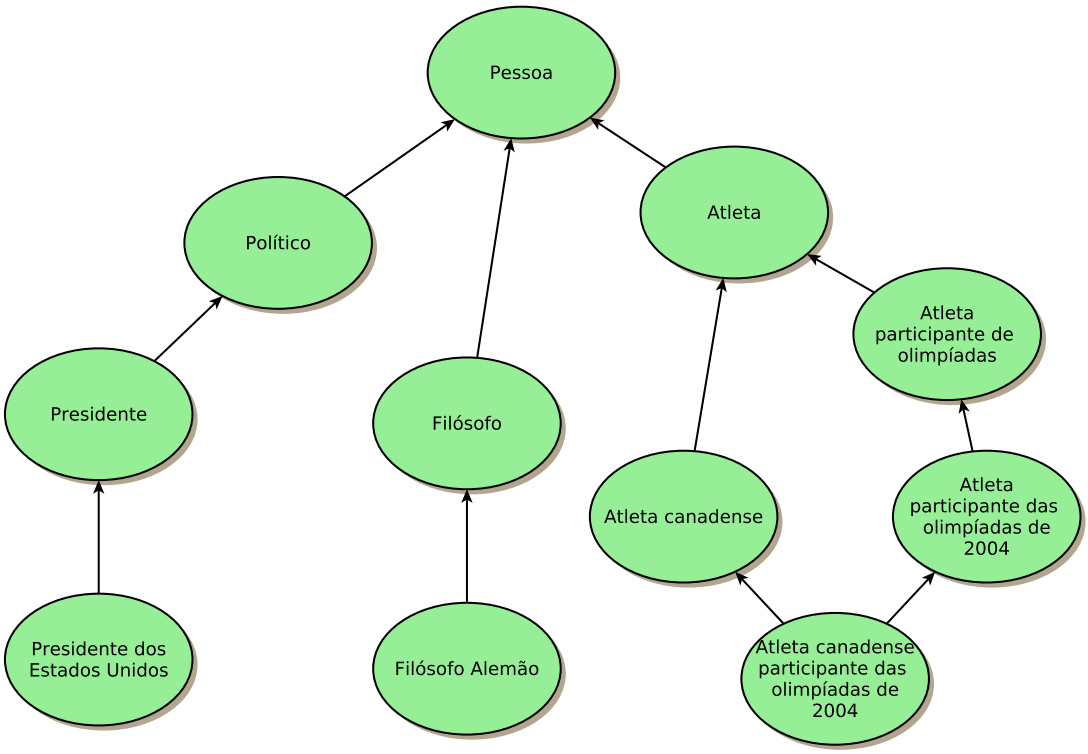
Yet Another Great Ontology (YAGO) trata-se de um repositório de conhecimento, criado e mantido pelo Max-Plank-Institute für Informatik, que combina dois grandes recursos: Wikidata, uma grande base de dados de propósito geral disponível na web semântica; e scherma.org, um padrão de ontologia para a representação de relações e classes. Conta com mais de 50 milhões de entidades e 2 bilhões de fatos.

Neste trabalho, foi utilizada somente uma parte da YAGO, correspondente às entidades de interesse. Assim, foram selecionadas apenas as entidades que são dos tipos Pessoa, Organização, Fundação ou Localidade e seus subtipos. Para a identificação dos subtipos de interesse, a estratégia adotada foi "subir" a árvore dos tipos até chegar em um dos tipos desejados, como ilustra a Figura

[8] Disponíveis em <http://openke.thunlp.org/>

11.

Figura 11 – Exemplo da seleção dos tipos usados para construção dos *Gazetteers*.



Fonte – Autor.

Após a identificação dos tipos e subtipos desejados, foi feita então a criação dos *Gazetteers*. Para isso, foram selecionadas as entidades das triplas cuja propriedade é "é do tipo" e o valor corresponde a um dos tipos ou subtipos selecionados. Por exemplo, a tripla

"Barack Obama", "é do tipo", "Presidente dos Estados Unidos"

seria selecionada, pois "Presidente dos Estados Unidos" está na lista de subtipos desejados por ser um subtipo de "Presidente", que é subtipo de "Político", que por sua vez é subtipo de "Pessoa". Sendo assim, a entidade ("Barack Obama") da tripla seria selecionada para compor o *Gazetteer* referente à Pessoas. Desta forma, para cada entidade recuperada foi associado o tipo, englobando também os respectivos subtipos. A quantidade de entidades que compõem esse *Gazetteer* é mostrada pela Tabela 14.

Em posse dos *Gazetteers*, a informação de pertinência a um (ou mais) *Gazetteer* foi agregada aos dados, como exemplifica a Tabela 15, onde PER e LOC simbolizam, respectivamente, que a palavra pertence à lista de Pessoas ou Localizações.

Tabela 14 – Quantificação das entidades presentes nos *Gazetteers*.

Tipo	**Quantidade**
Pessoa	1.743.625
Organização/Fundação	370.085
Localidade	507.599
Total	**2.621.309**

Fonte – Autor.

Tabela 15 – Exemplo de uma frase com adição de informações dos *Gazetteers*.

Palavra	**Gazetteer**	**Classe**
Takuya	PER	B-PER
Takagi	LOC/PER	I-PER
headed	-	O
the	-	O
winner	-	O
...	...	...
after	-	O
goalkeeper	-	O
Salem	LOC/PER	B-PER
Bitar	PER	I-PER
spoiled	-	O
a	-	O
mistake-free	-	O
display	-	O
...	...	...
his	-	O
body	-	O
.	-	O

Fonte – Autor.

4.3 Métricas de Avaliação

Para que se possa quantificar o desempenho de um método para Reconhecimento de Entidades Nomeadas, bem como compará-lo a outros, deve-se fazer uso de métricas que exprimem características relevantes ao que se deseja avaliar. Como base para algumas métricas existem os seguintes conceitos:

- Verdadeiro Positivo (VP) - quando o classificador prediz que a amostra pertence à classe e esta amostra de fato pertence à classe;
- Verdadeiro Negativo (VN) - quando o classificador prediz que a amostra não pertence à classe e esta amostra de fato não pertence à classe;

- Falso Positivo (FP) - quando o classificador prediz que a amostra pertence à classe quando a amostra de fato não pertence à classe;
- Falso Negativo (FN) - quando o classificador prediz que a amostra não pertence à classe e a amostra de fato pertence à classe.

A partir desses quatro conceitos apresentados acima é possível definir diversas métricas para avaliação do desempenho, porém as mais comumente utilizadas na tarefa de Reconhecimento de Entidades Nomeadas são representadas pelas Equações de 4.1 até 4.3.

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{4.1}$$

$$\text{Abrangência} = \frac{VP}{VP + FN} \tag{4.2}$$

$$\text{F}\beta\text{-Score} = \frac{(\beta^2 + 1) * \text{Precisão} * \text{Abrangência}}{(\beta^2 * \text{Precisão} + \text{Abrangência})} \tag{4.3}$$

As métricas Precisão, Abrangência e Fβ-Score são bastante utilizadas também para as tarefas de Classificação e Recuperação de Informação, geralmente considerando todas as amostras. Porém como as Entidades Nomeadas podem abranger mais do que somente uma palavra, é mais apropriado fazer o cálculo dessas métricas a nível de entidade, e não a nível de amostra, que nesse caso seria a nível de palavra. Um dos motivos para este cálculo é que a quantidade de palavras que não pertencem à entidade alguma é muito maior que o número de palavras que pertencem [57], e portanto, ao se calcular as métricas a nível de palavra, mesmo que várias entidades sejam reconhecidas, a variação do resultado não será tão grande devido ao grande montante de palavras que não pertencem a nenhuma entidade.

Um ponto a ser considerado é quais as definições de acerto e erro. Para o *dataset* CONLL2003, o recomendado é considerar somente as entidades que estão completamente corretas, desta forma, se o extrator apontou uma palavra a mais para uma entidade, ela deve ser considerada como errada, ou se as fronteiras (início e fim) da entidade estão corretas e o tipo da entidade está diferente, também deve ser considerada como errada [16]. De maneira geral, só é considerado acerto se a entidade reconhecida estiver exatamente igual à saída esperada. Já outras conferências de Extração de Entidades nomeadas consideram mais categorias de acertos, como é o caso da MUC [61], onde são apresentadas as seguintes categorias:

- Correta: Entidade reconhecida é igual à uma das entidades esperadas;
- Parcial: Entidade reconhecida é similar à uma das entidades esperadas;
- Incorreta: Entidade reconhecida é diferente da entidade esperada para aquela fronteira;
- Espúria: Entidade reconhecida não consta entre as entidades esperadas;
- Faltante: Entidade esperada não é reconhecida pelo sistema.

Além disso, na edição de 2013 do *International Workshop on Semantic Evaluation* (SemEval2013) foram apresentadas outras 4 categorias[9], sendo elas:

- Estrita: Correspondência exata da fronteira e tipo entre a entidade reconhecida e alguma das entidades esperadas;
- Exata: Correspondência exata da fronteira entre a entidade reconhecida e alguma das entidades esperadas, independente do tipo da entidade;
- Parcial: Correspondência parcial da fronteira entre a entidade reconhecida e alguma das entidades esperadas, independente do tipo da entidade;
- Tipo: Alguma sobreposição entre a entidade reconhecida e alguma das entidades esperadas

Para este trabalho foram utilizadas as métricas do CONLL2003 para todos os *datasets* avaliados. Portanto, foram avaliadas a Precisão, Abrangência e F1-Score das entidades que estavam totalmente corretas, tanto em fronteira como em tipo da entidade.

[9] <https://www.cs.york.ac.uk/semeval-2013/task9/data/uploads/semeval_2013-task-9_1-evaluation-metrics.pdf>

5 Abordagem Proposta

Neste capítulo é apresentada a abordagem experimental desenvolvida neste trabalho. As etapas para a implementação da abordagem proposta são detalhadas e são apresentados os resultados obtidos através dos experimentos realizados.

5.1 Avaliação Experimental

A tarefa de Reconhecimento de Entidades Nomeadas pode ser implementada como uma tarefa de classificação das palavras presentes nas frases de textos. Uma abordagem desta tarefa como classificação das palavras foi proposta neste trabalho, inspirada por [48], podendo ser vista na Figura 12.

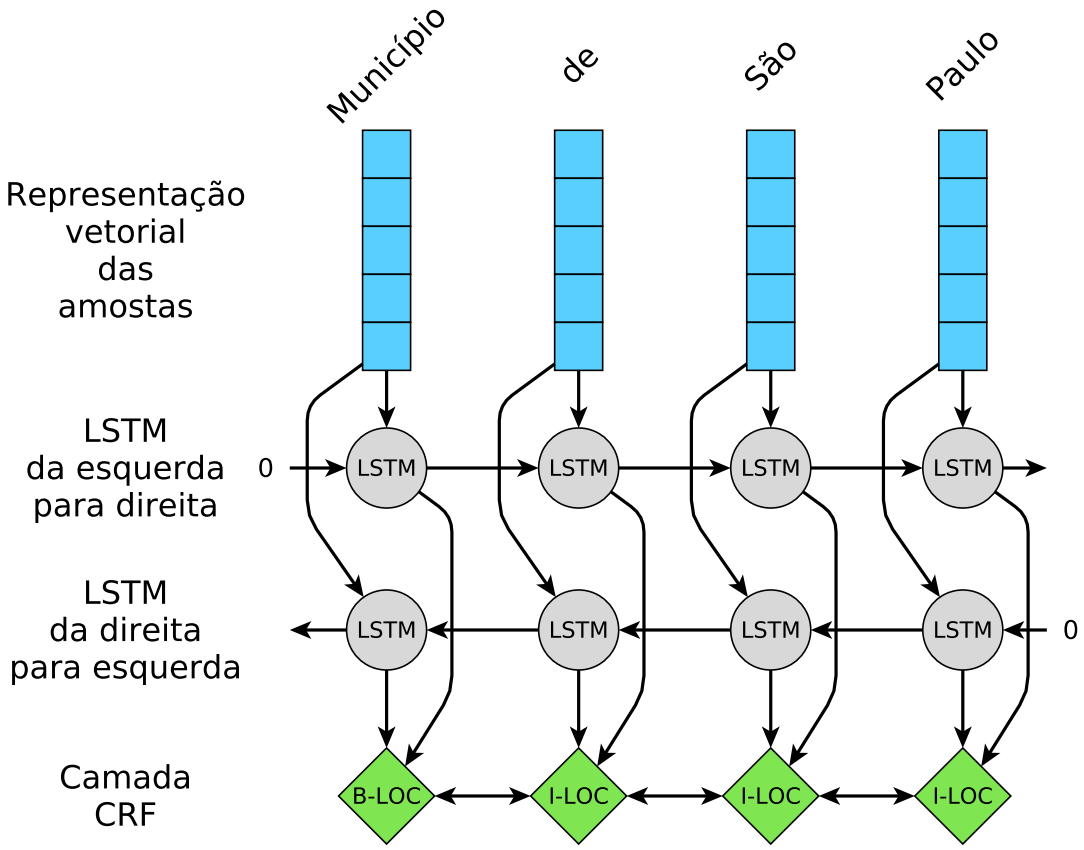
Esta abordagem consiste das seguintes etapas: inicialmente ocorre a representação vetorial de cada um dos termos das amostras de entrada, representadas pela cor azul; posteriormente ocorre a etapa de aprendizado dos padrões das características a serem extraídas utilizando o método BiLSTM (ou BiGRU)[10], representadas em cinza; e, finalmente a etapa de saída, onde as amostras são classificadas usando o método de aprendizado o CRF, mostrado em verde. Neste trabalho, embora a abordagem proposta foi baseada em [48], algumas novas alterações foram propostas, sendo elas:

- Uso da rede BiGRU para incorporar as caracterísicas sequenciais da frase;
- Uso de outra fonte de conhecimento externo para criação dos *Gazetteers*;
- Uso de esquema diferente de anotação das palavras pertencentes aos *Gazetteers*;
- Uso de *Knowledge Embedding* como conhecimento externo.

Desta maneira, as novas etapas para inclusão das alterações realizadas são descritas a seguir de maneira mais detalhada. O processo de obtenção das representações vetoriais que caracterizam as amostras que serão utilizadas é apresentado na seção 5.2. Após a obtenção dos vetores para cada amostra, estes vetores servem como entrada para o processo de classificação,

[10] Durante esse capítulo, para evitar repetição será omitida a "ou BiGRU" das descrições, porém BiGRU pode ser aplicada nas mesmas ocasiões que BiLSTM.

Figura 12 – Configuração da abordagem utilizada.



Fonte – Adaptado de [48].

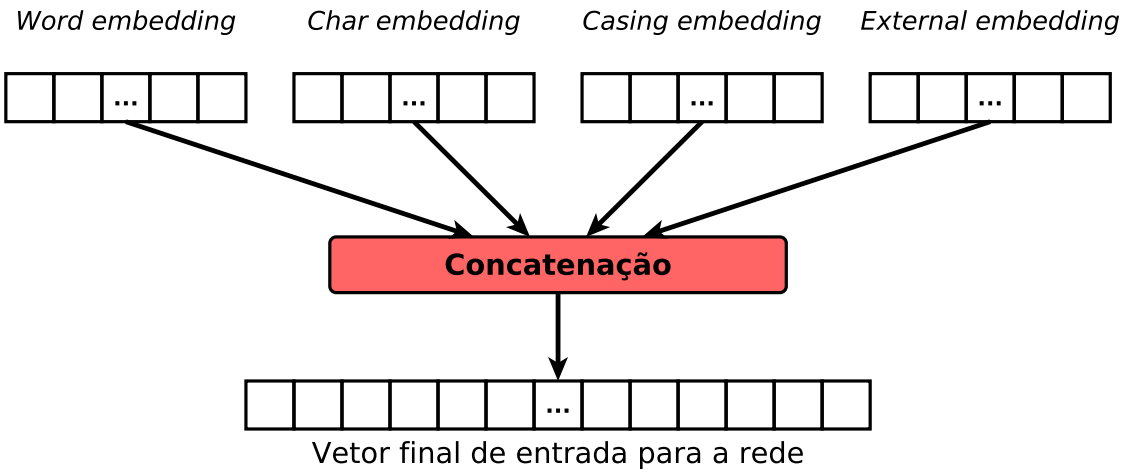
o qual é detalhado na Seção 5.3. Por fim, as configurações de parâmetros e resultados dos experimentos são apresentados na Seção 5.4.

5.2 Representação Vetorial

A primeira etapa da abordagem proposta consiste em gerar uma representação das informações contidas nos textos em um formato vetorial que possa ser utilizado pelos métodos de classificação. Neste trabalho foram identificadas nos textos e utilizadas as seguintes características: representação vetorial das palavras; representação vetorial dos caracteres das palavras; representação da grafia da palavra; e representação relacionada às palavras obtidas a partir do conhecimento disponível em repositório externo, conforme ilustrado, fora de escala, pela Figura 13. O processo de obtenção das representações utilizadas é detalhado nas próximas subseções. Não foram aplicadas técnicas clássicas de pré-processamento, como remoção de sufixos e prefixos, remoção de acentos e caracteres especiais, entre outras, em nenhuma etapa

do processo de obtenção das representações vetoriais, e os *datasets* utilizados nesse trabalho encontram-se no formato do CoNLL, como citado na Seção 4.1, portanto já estão tokenizados.

Figura 13 – Operação de concatenação dos vetores de características.



Fonte – Autor.

5.2.1 *Word embeddings*

As representações numéricas para as palavras utilizadas neste trabalho tratam-se dos *Word embeddings* gerados com o uso do método GloVe[11], treinado em diferentes *corpus*. Para cada palavra do *dataset* é recuperado do GloVe a representação vetorial correspondente. Ao final desse passo, cada palavra do *dataset* é representada por um vetor de 50 dimensões[12] que serão utilizados como entrada para a rede. Caso a palavra não venha a ser encontrada nos *Word embeddings* pré-treinados, é gerado um vetor aleatório para a palavra em questão, sendo que esse vetor tem seus valores inicializados seguindo uma distribuição uniforme entre -0,25 e 0,25.

5.2.2 *Character embeddings*

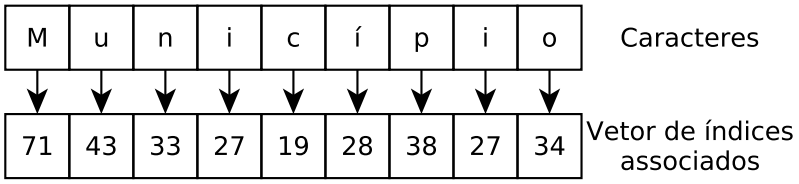
Inspirado em [48], foi adotada a estratégia de utilizar representações vetoriais dos caracteres que compõem as palavras, com a finalidade de captar as informações contidas no texto a nível dos caracteres. O uso de informações a níveis de caracteres pode ajudar quando existe uma palavra que contém erro de digitação, ou que não foi aprendida pelo método de *Word*

[11] Disponível em https://nlp.stanford.edu/projects/glove/

[12] A escolha de se utilizar vetores de 50 dimensões deve-se ao fato de um número maior de dimensões não apresentar um ganho significativo nos resultados [62, 48]

embedding. Para a obtenção dos *Character Embedding*, as informações sobre caracteres são obtidas a partir de um mapeamento dos caracteres presentes no texto por meio de um dicionário, onde o caractere é o campo chave e o campo valor representa o índice do caractere, como é ilustrado pela Figura 14.

Figura 14 – Mapeamento de caracteres para vetor de índices associados.



Fonte – Autor.

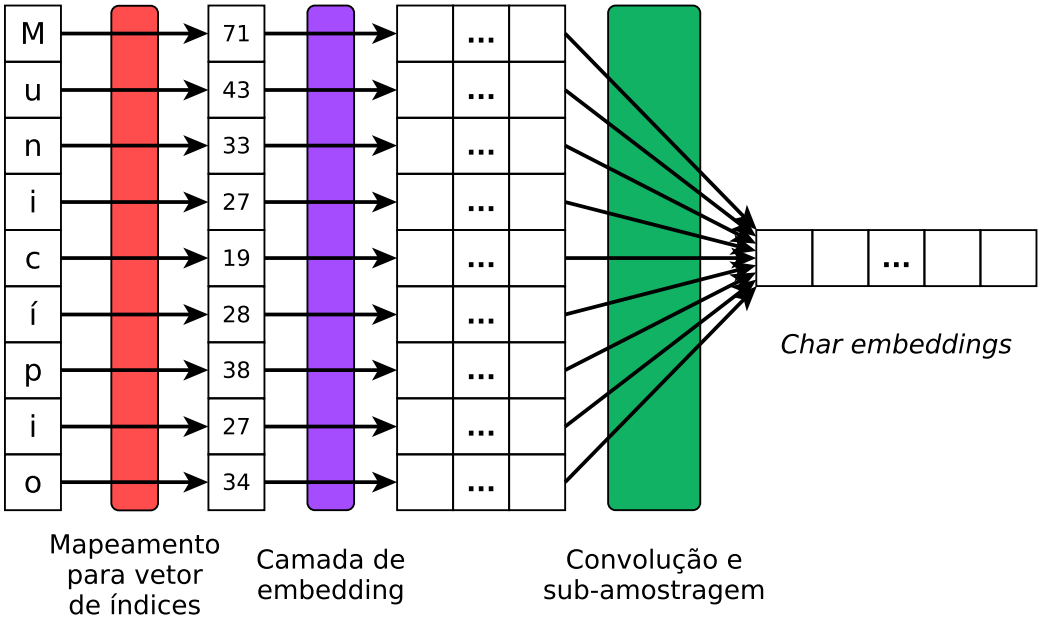
A Figura 14 mostra um exemplo do vetor de índices resultante para a palavra “Município”. O dicionário de caracteres conta com 135 caracteres distintos, incluindo letras com acento, pontuações, símbolos especiais, e também tem dois valores especiais, sendo um deles para caracteres não presentes no dicionário, e o outro é um valor que será utilizado como “enchimento” do vetor. Esse valor de enchimento é necessário devido ao fato das palavras terem diferentes tamanhos, e a presente abordagem fazer uso de vetores com 52 dimensões para representar cada caractere de cada palavra.

Os passos para a obtenção dos *Character embeddings* são mostrados na Figura 15, usando como exemplo a palavra “Município”. Inicialmente ocorre o mapeamento do vetor de índices para vetores de números reais, que serão utilizados na camada de convolução. Ou seja, cada um dos 135 índices de caracteres será representado como um vetor de várias dimensões. Neste trabalho, os vetores de caracteres têm 52 dimensões e os valores de suas coordenadas são inicializados seguindo uma distribuição uniforme que varia de -0,5 até 0,5. Diferentemente dos *Word embeddings*, as representações vetoriais de cada caractere serão aprendidas durante o treinamento da rede.

Para este trabalho, deseja-se somente um vetor que contenha a informação sobre todos os caracteres presentes na palavra. Para cumprir esse propósito é utilizada uma Rede Neural Convolucional, onde os vetores que representam cada caractere são utilizados como entrada, e a saída é um único vetor que exprime informações sobre todos os caracteres. A Rede Neural Convolucional faz uso do *max-pooling* em sua camada de sub-amostragem. São utilizados 30 filtros cujo tamanho de convolução é equivalente a 3, e o passo da convolução tem tamanho 1.

Ao final da rede, a saída da camada é um vetor de 52 dimensões que representa as principais características dos caracteres que compõem as palavras, e é chamado de *Char embedding*, que será posteriormente concatenado a outros vetores de características.

Figura 15 – Mapeamento de caracteres para *Character embeddings*.



Fonte – Autor.

5.2.3 *Casing embeddings*

Com o objetivo de se incorporar mais informações sobre a composição da palavra, é utilizado um vetor que represente as informações de grafia, como presença de letra maiúscula, presença ou totalidade de dígitos na palavra, entre outras. Para as informações de grafia, as palavras são analisadas e categorizadas em um dos seguintes perfis, presentes em um dicionário:

- Totalmente numérica;
- Totalmente minúscula;
- Totalmente maiúscula;
- Inicial maiúscula;
- Majoritariamente numérica;
- Contém dígito;

- *Token* de preenchimento;
- Outro (palavras que não se encaixam nas categorias acima).

A categoria da palavra é utilizada como chave para se obter um valor numérico inteiro, usado com entrada para uma camada de Embedding da biblioteca Keras[13]. Neste trabalho, as representações são inicializadas por meio de uma matriz de identidade, e têm seus valores ajustados durante a etapa de treinamento. O treinamento dessa camada de Embedding é feito utilizando os mesmos parâmetros do modelo, fazendo uso do Nadam [63] como Algoritmo de Otimização e a Função de Perda é a *Negative Log-Likelihood*. O vetor gerado por essa camada de Embedding é chamado de *Casing embedding*.

5.2.4 Conhecimento Externo

Além dos aspectos sintáticos, neste trabalho também são incorporadas características semânticas das palavras, sendo essas características obtidas por meio de conhecimento explicitado em fontes externas. O vetor que agrega as características semânticas, chamado de *External embeddings* na Figura 13, é obtido de duas maneiras distintas, descritas abaixo:

- Usando *Gazetteer*: quando o experimento faz uso de *Gazetteer*, o processo de acréscimo de informação externa se assemelha à criação do *Casing Embedding*, no qual as informações sobre quais categorias a palavra pertence são utilizadas, e para cada categoria há um vetor associado, os quais são inicializados como uma matriz identidade, e ajustados durante a etapa de treinamento da rede utilizando o Nadam [63] como Algoritmo de Otimização e a *Negative Log-Likelihood* como Função de Perda, da mesma forma como utilizado para encontrar os *Casing embeddings*. Neste trabalho, como foi feito o uso de três *Gazetteers* e as palavras podiam pertencer a mais de um *Gazetteer*, foram utilizadas 8 categorias;
  - Pessoa;
  - Localização;
  - Organização;
  - Pessoa/Localização;
  - Pessoa/Organização;

[13] <https://keras.io/>

  - Pessoa/Localização/Organização;
  - Localização/Organização;
  - Nenhuma.

- Usando *Knowledge embedding*: em algumas fontes externas de conhecimento já foram gerados os *embedding* correspondentes. Assim, nestes casos o experimento faz uso dos *embedding* vindos dos repositório de conhecimento. O processo é mais simples, pois basta simplesmente juntá-los aos demais *embeddings* daquela palavra. Porém, nos experimentos realizados neste trabalho não são consideradas entidades descritas por mais de uma palavra, sendo considerados somente os *embedding* das entidades de tamanho 1, ou seja, entidades representadas por somente uma palavra.

Após a geração dos quatro vetores de características, podem ser definidos diversos métodos para sua combinação. Neste trabalho a combinação foi feita por meio de um processo de concatenação, resultando em um único vetor que descreve cada palavra de uma frase, como ilustrado, fora de escala, pela Figura 13.

5.3 Classificação

Após a realização da tarefa anterior de mapear as informações textuais em vetores de características, foram definidos os métodos requeridos para realizar a etapa de aprendizado dos padrões que caracterizam as Entidades a serem reconhecidas. Considerando as características da tarefa de Reconhecimento de Entidades Nomeadas, em que um episódio tem relação com o anterior, as novas abordagens têm utilizado métodos de Aprendizado de Máquina que reconhecem a sequência dos episódios. Desta forma, na arquitetura utilizada, mostrada na Figura 12, o vetor de características é utilizado como entrada dos métodos de aprendizado. Os métodos de Aprendizado de Máquina utilizados são as redes neurais BiLSTM e BiGRU, cujos funcionamentos permitem a assimilação de características dos episódios anteriores. Finalmente tem-se a etapa de classificação, onde as amostras são classificadas usando o método de aprendizado CRF, que considera os rótulos já atribuídos aos episódios anteriores da mesma frase para atribuição do rótulo atual, e sua saída corresponde às classes semânticas correspondentes de cada palavra da frase de entrada. Desta maneira, ao se utilizar BiLSTM, as características dos episódios anteriores são agregados à amostra atual, enquanto o uso do CRF considera os rótulos atribuídos anteriormente à sequencia,

e portanto, esta combinação faz um bom proveito da estrutura sequencial dos dados de entrada. Os detalhes dos métodos de aprendizado utilizados, suas configurações e funcionamentos são explicados a abaixo.

A função das camadas LSTM é de incorporar características de episódios anteriores para a classificação do episódio atual. Como o Reconhecimento de Entidades Nomeadas pode ser tratado como uma classificação de palavras, a LSTM trabalha a nível de frases, onde as características sobre as palavras anteriores são incorporadas como características da palavra atual.

Neste trabalho faz-se uso da BiLSTM, que é composta por duas camadas LSTM: uma que as palavras são processadas conforme o sentido convencional de leitura - da esquerda para a direita, e outra que as palavras são processadas no sentido inverso - da direita para a esquerda. Ao se processar a frase nos dois sentidos é possível agregar informações tanto das palavras predecessoras quanto das palavras sucessoras, enriquecendo ainda mais a representação gerada.

Como dito anteriormente, o objetivo dessa camada é a agregação das características dos episódios anteriores ao atual, porém para que isso seja feito de forma proveitosa, é necessária a etapa de treinamento da rede para que todos os pesos que compõem o modelo sejam devidamente ajustados. Por conter muitos parâmetros a serem ajustados, o treinamento dessa camada pode requerer grandes quantidades de dados rotulados e ser um pouco demorado.

Cada uma dessas camadas LSTM conceberá um vetor como saída, os quais serão concatenados para servirem como entrada para uma camada de CRF, que definirá qual a classe mais provável para aquela palavra dadas as características apresentadas à rede.

Desta maneira, ao se apresentar uma nova amostra ao modelo, é inicialmente obtida a representação vetorial dessa amostra, a qual é submetida a camada BiLSTM e posteriormente CRF. A saída final do método é a classe correspondente de cada palavra utilizada como entrada.

Assim como a rede BiLSTM, é necessário que haja uma etapa de treinamento do CRF, para que este possa ajustar os pesos de suas funções características a fim de conseguir identificar os padrões que melhor descrevem as classes de interesse.

Tanto o treinamento das camadas de Embeddings, das camadas BiLSTM e da camada de CRF são feitos utilizando a mesma Função de Perda, que se trata da *Negative Log-Likelihood*. Já para a atualização dos pesos é utilizado o Algoritmo de Otimização Nadam[63].

Para a implementação das redes neurais utilizadas neste trabalho foi adotada a biblioteca Keras[14] e para a implementação da camada de CRF foi utilizada a biblioteca Keras-Contrib[15].

5.4 Resultados

Esta sessão mostra os resultados dos experimentos executados utilizando a arquitetura proposta neste trabalho. O procedimento experimental adotado consistiu dos seguintes passos:

1. Definição dos *datasets* a serem utilizados nos experimentos;
2. Geração dos *embeddings* requeridos nos experimentos;
3. Geração dos modelos iniciais e dos parâmetros das redes neurais utilizadas;
4. Realização da etapa de treinamento e testes.

A partir da revisão bibliográfica realizada, foi possível tomar ciência dos *datasets* disponíveis para a tarefa de Reconhecimento de Entidades Nomeadas. Deste modo, foram selecionados cinco *datasets* para serem utilizados nos experimentos.

Foi por meio da revisão bibliográfica que também foi possível decidir como seriam representadas as características referente à cada amostra. Deste modo, inspirado em [48], foi feito uso de *Word embeddings* junto à *Character embeddings* e *Casing embeddings*, além da inclusão de fontes externas de conhecimento por meio dos *External embeddings*. Ao invés de se treinar uma representação vetorial para as palavras, foi escolhido utilizar os *Word embeddings* pré-treinados pelo método GloVe.

Depois da decisão dos *datasets* e *embeddings* a serem utilizados, foram então decididos quais modelos seriam utilizados para os experimentos. Essa escolha resultou no uso de modelos que se beneficiam da característica sequencial presente nos textos. Além da decisão dos modelos, foram selecionados os valores de parâmetros dos experimentos, porém sem realização de otimizações de hiper-parâmetros para cada rede e *dataset*. Isso foi feito pois não eram todas as bases que estavam divididas em conjunto de treinamento, teste e validação. Deste modo, foram decididos valores fixos de parâmetros para todos os *datasets* de uma rodada, com excessão do

[14] <https://keras.io/>
[15] <https://github.com/keras-team/keras-contrib>

OntoNotes5 devido à sua grande quantidade de amostras, e por isso teve alguns parâmetros alterados.

Assim, com os *datasets* definidos, *embeddings* encontrados e modelos selecionados, foi possível realizar a etapa de treinamento e testes para avaliar o impacto do uso de fontes externas de conhecimento na tarefa de Reconhecimento de Entidades Nomeadas. Por conter elementos estocásticos na inicialização das redes, todos os resultados das métricas apresentados nas seções seguintes são o resultado da média de 10 execuções para cada experimento, e também é apresentado o desvio padrão de cada resultado.

Ao se seguir os passos acima descritos, foi possível a realização da Primeira Rodada de testes, a qual é descrita mais detalhadamente na subseção a seguir.

5.4.1 Primeira rodada

O propósito desta primeira rodada está em avaliar a hipótese de que há impacto positivo com a adição de fontes externas de conhecimento, tanto na forma de *Gazetteer* quanto na forma de *Knowledge embeddings*. O motivo de se testar essas duas formas distintas de adição de conhecimento está no grau de facilidade para obtenção delas. A construção de um *Gazetteer* é uma tarefa restrita a domínio, podendo um *Gazetteer* não ser muito útil para outro contexto. Já os *Knowledge embeddings* são prontos para o uso, independente do domínio, facilitando assim sua integração a diversos domínios, porém essa generalidade pode não trazer muitas vantagens.

Para verificar os impactos das fontes de conhecimento, foram realizados dois experimentos distintos, um deles acrescentando somente informações externas vindas do *Gazetteer*; e o outro experimento acrescentando somente informações externas vindas dos *Knowledge embeddings*. A não ser pela diferente fonte de conhecimento externo, as outras partes dos experimentos são iguais, ou seja, fazem uso da mesma arquitetura e dos mesmos parâmetros. A escolha desses valores de parâmetros foi realizada considerando trabalhos que fazem uso de modelos semelhantes ao escolhido para este trabalho.

Para ambos os experimentos, os parâmetros utilizados são apresentados na Tabela 16, onde a coluna “ON5” representa os parâmetros diferenciados para o *dataset* OntoNotes5. Os resultados das execuções dos experimentos usando *Gazetteer* e *Knowledge embeddings* são mostrados, respectivamente, nas Tabelas 17 e 18.

Ao se analisar os resultados do uso de *Gazetteer* mostrados na Tabela 17, podemos ver

Tabela 16 – Parâmetros utilizados nas redes neurais durante a primeira rodada de experimentos.

Parâmetro	**Valor**	**Valor ON5**
Épocas	100	20
Dropout	0.5	0.5
Dropout recorrente	0.25	0.25
Unidades de BiLSTM/BiGRU	200	200
Filtros de convolução	30	30
Tamanho da convolução	3	3
Passo da convolução	1	1
Taxa de aprendizado	0,0105	0,008

Fonte – Autor.

que o uso trouxe ganho de F1-Score em 6 dos 10 cenários (destacados na coluna "Diferença F1") no qual foi introduzido, sendo a maior parte desses ganhos quando a rede BiLSTM foi utilizada.

Para ambas as redes as diferenças no *dataset* CONLL2003 foram positivas, o que significa que o uso de *Gazetteer* ajudou a tarefa de Reconhecimento de Entidades Nomeadas. Este era um resultado esperado, uma vez que o *Gazetteer* foi elaborado seguindo os tipos de entidade presentes neste *dataset* (com exceção das entidades *Miscellaneous*). Como mostram os resultados, para esta base de dados a configuração que obteve melhor resultado foi usando a BiLSTM+CRF+Gazetteer. Além disso, ao se incluir informação de *Gazetteer*, os resultados tiveram um desvio padrão menor, tornando-se mais estáveis.

Os resultados obtidos para o *dataset* OntoNotes5 mostram ganho em todas as métricas para ambas as redes, sendo que ao se utilizar *Gazetteers* na BiGRU a melhora foi bastante expressiva. Além de apresentar melhora, a incorporação dos *Gazetteers* também deixou o resultado mais estável, como mostram os valores de desvio padrão.

Já para a base de dados GUM o cenário é o contrário, mostrando que a introdução do *Gazetteer* atrapalhou o processo de reconhecimento, causando uma variação negativa de F1-Score. Apesar deste *dataset* também conter entidades do tipo Pessoa e Organização, o *Gazetteer* não trouxe ajuda. Novamente ao contrário do que houve com o *dataset* CONLL2003, ao fazer uso de *Gazetteers* os resultados ficaram menos estáveis.

Para o *dataset* MIT Movies houve um cenário de ganho e um de perda, porém os melhores valores foram obtidos com a rede BiGRU quando não foi feito o uso do *Gazetteer*. No quesito de estabilidade houve uma melhora, apresentando um desvio padrão menor.

Os resultados para a base de dados MIT Restaurants também apresentam ganho e perdas,

Tabela 17 – Resultados para experimentos usando *Gazetteers* em redes com 200 unidades de memória.

CONLL2003	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	88,04±0,32	89,10±0,36	88,57±0,32	-
BiLSTM+CRF+Gaz	**88,29±0,26**	**89,34±0,31**	**88,81±0,27**	+0,24
BiGRU+CRF	87,60±0,62	88,71±0,72	88,15±0,66	-
BiGRU+CRF+Gaz	87,95±0,43	88,83±0,49	88,39±0,38	+0,24
OntoNotes5	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	68,43±9,52	69,10±12,68	68,71±11,13	-
BiLSTM+CRF+Gaz	**70,29±1,55**	74,28±3,60	**72,21±2,41**	+3,5
BiGRU+CRF	59,70±12,18	60,57±19,79	59,00±16,65	-
BiGRU+CRF+Gaz	65,20±10,24	**74,72±8,04**	69,53±9,29	+10,53
GUM	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	**44,99±1,00**	**44,56±0,97**	**44,77±0,96**	-
BiLSTM+CRF+Gaz	44,12±1,54	44,21±1,39	44,16±1,43	-0,61
BiGRU+CRF	44,30±0,90	43,91±0,79	44,10±0,76	-
BiGRU+CRF+Gaz	43,28±1,91	42,99±1,37	43,12±1,54	-0,98
MIT MOVIES	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	67,49±0,55	65,67±0,70	66,56±0,54	-
BiLSTM+CRF+Gaz	67,81±0,70	66,02±0,52	66,91±0,54	+0,35
BiGRU+CRF	**68,38±0,62**	**66,65±0,68**	**67,50±0,64**	-
BiGRU+CRF+Gaz	67,99±0,35	66,52±0,58	67,25±0,40	-0,25
MIT RESTAURANTS	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	74,13±0,69	**73,57±0,50**	73,85±0,49	-
BiLSTM+CRF+Gaz	**74,43±0,63**	73,46±0,36	**73,94±0,32**	+0,09
BiGRU+CRF	73,50±2,56	72,21±1,63	72,83±1,71	-
BiGRU+CRF+Gaz	73,61±2,35	72,03±3,48	72,80±2,90	-0,03

Fonte – Autor

porém bem pequenos quando comparados aos outros *datasets*, e apesar do melhor valor de F1-Score ser através da BiLSTM usando *Gazetteer*, a diferença é muito pequena. Para essa base de dados houve uma maior estabilidade ao se utilizar BiLSTM com *Gazetteer* e uma menor estabilidade quando foi utilizada BiGRU com *Gazetteer*.

A partir dos resultados mostrados pela Tabela 17, podemos ver que a rede BiLSTM conseguiu capturar melhor as informações adicionadas pelo uso de *Gazetteer*, tendo diferenças positivas maiores e diferenças negativas menores quando comparada às diferenças obtidas com o uso da BiGRU. Apesar de trazer ganho em mais da metade dos casos, o ganho de F1-Score mostra-se bem pequeno para a maioria dos datasets.

Tabela 18 – Resultados para experimentos usando *Knowledge embeddings* em redes com 200 unidades de memória.

CONLL2003	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	**88,04±0,32**	**89,10±0,36**	**88,57±0,32**	-
BiLSTM+CRF+KE	87,39±0,76	88,99±0,45	88,18±0,58	-0,39
BiGRU+CRF	87,60±0,62	88,71±0,72	88,15±0,66	-
BiGRU+CRF+KE	87,30±0,34	88,85±0,28	88,07±0,27	-0,08
OntoNotes5	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	68,43±9,52	69,10±12,68	68,71±11,13	-
BiLSTM+CRF+KE	67,21±0,91	67,91±3,32	67,50±1,61	-1,21
BiGRU+CRF	59,70±12,18	60,57±19,79	59,00±16,65	-
BiGRU+CRF+KE	**67,80±1,11**	**74,10±1,27**	**70,80±0,77**	+11,8
GUM	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	**44,99±1,00**	**44,56±0,97**	**44,77±0,96**	-
BiLSTM+CRF+KE	43,71±1,25	43,26±2,05	43,48±1,61	-1,29
BiGRU+CRF	44,30±0,90	43,91±0,79	44,10±0,76	-
BiGRU+CRF+KE	43,76±0,68	43,07±1,10	43,41±0,85	-0,69
MIT MOVIES	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	67,49±0,55	65,67±0,70	66,56±0,54	-
BiLSTM+CRF+KE	66,39±1,54	65,23±1,02	65,80±1,10	-0,76
BiGRU+CRF	**68,38±0,62**	**66,65±0,68**	**67,50±0,64**	-
BiGRU+CRF+KE	66,97±1,28	65,84±0,51	66,39±0,70	-1,11
MIT RESTAURANTS	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	74,13±0,69	**73,57±0,50**	73,85±0,49	-
BiLSTM+CRF+KE	74,08±0,87	73,44±0,26	73,75±0,45	-0,1
BiGRU+CRF	73,50±2,56	72,21±1,63	72,83±1,71	-
BiGRU+CRF+KE	**74,47±0,42**	**73,80±0,33**	**74,13±0,32**	+1,3

Fonte – Autor

Ao se analisar os resultados do uso dos *Knowledge embeddings* mostrados na Tabela 18, podemos ver que seu uso trouxe ganho de F1-Score em apenas 2 dos 10 cenários (destacados na coluna "Diferença F1") nos quais foi introduzido, porém foram mais expressivos que aqueles obtidos com o uso do *Gazetteer*.

Na base de dados CONLL2003 houve queda no resultado do F1-Score para ambas as redes, mostrando que as informações extras não trouxeram ganho. Apesar de ter tido um resultado menor, o uso dos *Knowledge embeddings* com a rede BiGRU deixou o resultado final mais estável, porém isso não ocorreu com o uso da BiLSTM.

Os resultados obtidos para a base de dados OntoNotes5 mostram um caso de melhora e

um de piora para todas as métricas analisadas. Utilizando-se os *Knowledge embeddings* na rede BiLSTM foi obtida a segunda maior queda de F1-Score dessa rodada de experimentos, porém ao se incorporar os Knowledge Embeddings à BiGRU foi obtido o maior ganho de F1-Score desta rodada. Em ambos os casos os valores de desvio padrão foram menores para todas as métricas, mostrando uma maior estabilização dos resultados.

Os resultados para o *dataset* GUM mostram que o uso dos *Knowledge embeddings* piorou a eficácia de ambas as redes para todas as métricas, e além disso também aumentou o desvio padrão para quase todas as métricas, com exceção da Precisão na rede BiGRU.

Para a base de dados MIT Movies os resultados também apontam queda para todas as métricas de ambas as redes, e também aumento do desvio padrão para quase todas as métricas, exceto Abrangência para BiGRU.

Já para o *dataset* MIT Restaurants houve um dos maiores ganhos dessa rodada ao se utilizar BiGRU. Já ao se utilizar BiLSTM houve uma perda, porém foi a terceira menor perda da rodada. Foi observado também que os resultados ficaram mais estáveis, principalmente no uso da BiGRU.

De acordo com as informações presentes na Tabela 18, é possível perceber que o uso dos *Knowledge embeddings* trouxe uma queda nos resultados. Há a suspeita de que as redes não estão conseguindo absorver as informações extras que estão sendo inseridas, e tais informações acabam atrapalhando ao invés de ajudar as redes. Com isso em mente, foi feita uma Segunda rodada de experimentos, descrita a seguir.

5.4.2 Segunda rodada

A partir dos resultados obtidos com a primeira rodada dos experimentos, surgiu a hipótese da rede não estar sendo capaz de assimilar as informações que foram inseridas, principalmente ao se fazer uso dos *Knowledge embeddings*. Com isso em mente, retornou-se ao passo 3 do procedimento experimental adotado, e houve a redefinição do parâmetro "Unidades de BiLSTM/BiGRU", cujo valor passou de 200 unidades para 400. O intuito dessa alteração é avaliar a hipótese de que um maior número de unidades BiLSTM/BiGRU consiga aprender novas relações a partir dos dados de entrada. Os demais parâmetros permaneceram iguais, inclusive as *features* de entrada, como mostra a Tabela 19.

Outra hipótese avaliada nesta rodada é de que o aumento das unidades BiLSTM/BiGRU

consiga diferenciar melhor as entidades, principalmente nos *datasets* que apresentam um maior número de tipos de entidades. Desta maneira, pôde-se prosseguir ao passo 4, e os resultados desse passo experimental podem ser encontrados nas Tabelas 20 e 21.

Tabela 19 – Parâmetros utilizados nas redes neurais durante a segunda rodada de experimentos.

Parâmetro	**Valor**	**Valor ON5**
Épocas	100	20
Dropout	0.5	0.5
Dropout recorrente	0.25	0.25
Unidades de BiLSTM/BiGRU	400	400
Filtros de convolução	30	30
Tamanho da convolução	3	3
Passo da convolução	1	1
Taxa de aprendizado	0,0105	0,008

Fonte – Autor.

Analisando os resultados do uso de *Gazetteers*, mostrados na Tabela 20, percebe-se que houve ganho de F1-Score em 6 dos 10 casos onde houve a inclusão dos *Gazzetteers*, assim como ocorreu na primeira rodada experimental utilizando 200 unidades BiLSTM/BiGRU, porém em situações diferentes. Neste experimento com 400 unidades, a maioria dos ganhos vieram ao se integrar os *Gazetteers* à rede BiGRU.

Para o *dataset* CONLL2003 houve um cenário de ganho e um de perda de F1-Score. Apesar de ter apresentado ganho, o uso de *Gazetteers* junto à rede BiGRU não obteve o melhor resultado para essa base. Os resultados apresentaram uma maior estabilidade em todas as métricas para ambas as redes ao se realizar a adição de informação dos *Gazetteers*.

Os experimentos na base de dados OntoNotes5 mostraram ganhos de F1-Score em ambos os casos da inclusão dos *Gazetteers*, além de resultados mais estáveis para ambas as redes em todas as métricas avaliadas. Novamente, para este *dataset* os resultados foram bastante expressivos.

No *dataset* GUM o uso dos *Gazetteers* trouxe perda de F1-Score para os dois casos testados. Curiosamente, ao se integrar os *Gazetteers* à rede BiLSTM, os resultados foram mais estáveis para todas as métricas, porém ocorreu o inverso no uso da BiGRU.

Os resultados obtidos para a base MIT Movies foram semelhantes aos obtidos para a OntoNotes5 nos quesitos melhora de F1-Score e maior estabilidade de resultados em ambas as redes e todas as métricas. Porém, há a diferença de que os ganhos não foram tão expressivos

Tabela 20 – Resultados para experimentos usando *Gazetteers* em redes com 400 unidades de memória.

CONLL2003	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	**88,18±0,42**	**89,19±0,30**	**88,68±0,33**	-
BiLSTM+CRF+Gaz	88,11±0,27	89,13±0,28	88,62±0,26	-0,06
BiGRU+CRF	86,07±4,09	87,96±2,52	87,00±3,34	-
BiGRU+CRF+Gaz	87,95±0,55	88,83±0,47	88,39±0,28	+1,39

OntoNotes5	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	66,47±11,82	63,83±18,28	64,81±15,53	-
BiLSTM+CRF+Gaz	**71,54±1,21**	**75,32±2,05**	**73,37±1,40**	+8,56
BiGRU+CRF	65,77±13,61	68,03±19,94	66,31±18,05	-
BiGRU+CRF+Gaz	63,73±8,93	72,35±8,17	67,73±8,59	+1,42

GUM	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	**45,45±0,50**	**45,02±0,72**	**45,23±0,57**	-
BiLSTM+CRF+Gaz	44,69±0,57	45,01±0,44	44,85±0,46	-0,38
BiGRU+CRF	40,77±4,30	39,43±4,73	40,08±4,51	-
BiGRU+CRF+Gaz	39,38±5,72	39,59±5,00	39,48±5,36	-0,6

MIT MOVIES	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	66,77±1,14	65,79±0,86	66,27±0,92	-
BiLSTM+CRF+Gaz	67,23±0,74	65,84±0,52	66,52±0,58	+0,25
BiGRU+CRF	67,41±1,57	65,14±2,76	66,25±2,17	-
BiGRU+CRF+Gaz	**67,97±0,94**	**66,15±0,68**	**67,05±0,67**	+0,8

MIT RESTAURANTS	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	73,36±0,74	72,74±1,06	73,04±0,71	-
BiLSTM+CRF+Gaz	71,71±3,33	72,24±2,90	71,97±3,08	-1,07
BiGRU+CRF	72,53±3,37	72,25±3,41	72,39±3,39	-
BiGRU+CRF+Gaz	**73,59±0,62**	**73,41±0,90**	**73,50±0,71**	+1,11

Fonte – Autor

quanto os ganhos na base OntoNotes5.

Já para o *dataset* MIT Restaurants tem-se um caso de ganho e um de perda de F1-Score. No caso de ganho, os resultados apresentaram desvio padrão menor para todas as métricas. Para o caso da perda, o caso foi o oposto, havendo uma menor estabilidade.

De maneira geral, os resultados obtidos com a introdução de informações dos *Gazetteers* às redes com 400 unidades de memória mostraram ganhos maiores do que quando os *Gazetteers* foram introduzidos às redes com 200 unidades, porém nem sempre essa melhora culminou no melhor resultado para aquele *dataset*.

Ao se analisar os resultados da incorporação dos *Knowledge embeddings*, mostrados

Tabela 21 – Resultados para experimentos usando *Knowledge embeddings* em redes com 400 unidades de memória.

CONLL2003	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	**88,18±0,42**	**89,19±0,30**	**88,68±0,33**	-
BiLSTM+CRF+KE	87,56±0,34	88,82±0,27	88,19±0,26	-0,49
BiGRU+CRF	86,07±4,09	87,96±2,52	87,00±3,34	-
BiGRU+CRF+KE	87,28±0,41	88,69±0,24	87,98±0,31	+0,98
OntoNotes5	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	66,47±11,82	63,83±18,28	64,81±15,53	-
BiLSTM+CRF+KE	67,14±1,22	70,52±1,64	68,77±0,63	+3,96
BiGRU+CRF	65,77±13,61	68,03±19,94	66,31±18,05	-
BiGRU+CRF+KE	**69,80±2,22**	**74,63±1,83**	**72,09±1,10**	+5,78
GUM	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	**45,45±0,50**	**45,02±0,72**	**45,23±0,57**	-
BiLSTM+CRF+KE	41,86±0,66	43,94±0,77	42,87±0,66	-2,36
BiGRU+CRF	40,77±4,30	39,43±4,73	40,08±4,51	-
BiGRU+CRF+KE	40,81±1,10	43,41±1,31	42,07±1,17	+1,99
MIT MOVIES	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	66,77±1,14	**65,79±0,86**	**66,27±0,92**	-
BiLSTM+CRF+KE	65,99±1,04	65,17±0,65	65,57±0,79	-0,7
BiGRU+CRF	**67,41±1,57**	65,14±2,76	66,25±2,17	-
BiGRU+CRF+KE	64,75±1,67	64,20±1,23	64,47±1,40	-1,78
MIT RESTAURANTS	Precisão	Abrangência	F1-Score	Diferença F1
BiLSTM+CRF	**73,36±0,74**	**72,74±1,06**	**73,04±0,71**	
BiLSTM+CRF+KE	71,92±0,71	72,22±0,83	72,06±0,53	-0,98
BiGRU+CRF	72,53±3,37	72,25±3,41	72,39±3,39	-
BiGRU+CRF+KE	71,92±0,83	72,46±1,03	72,18±0,72	-0,21

Fonte – Autor

na Tabela 21, nota-se que houve ganho de F1-Score em 4 dos 10 casos onde estes embeddings foram incorporados, dois casos a mais quando comparado com as redes que faziam uso de 200 unidades BiLSTM/BiGRU, porém em situações diferentes. Neste experimento com 400 unidades, a maioria dos ganhos resultantes da agregação vieram aos se utilizar a rede BiGRU.

No *dataset* CONLL2003 houve caso de ganho e perda, sendo essas variações mais expressivas quando comparadas aos resultados da incorporação dos *Knowledge embeddings* utilizando 200 unidades. Em ambos o desvio padrão foi menor para todas as métricas.

Os resultados do uso de *Knowledge embeddings* para a base OntoNotes5 apresentam ganhos de F1-Score para ambas as redes, bem como uma maior estabilidade. Apesar de ter

apresentado ganho de F1-Score, ao se utilizar esses *embeddings* junto à rede BiGRU, houve uma queda de Precisão.

Para o *dataset* GUM, a inclusão de *Knowledge embeddings* trouxe um caso de ganho ao ser utilizado junto à rede BiGRU, sendo este o único caso de ganho para esta base em ambas as rodadas experimentais. Para este dataset, ao se utilizar os *embeddings* com a BiLSTM, além da piora no F1-Score também houve aumento do desvio padrão em todas as métricas, e o contrário pode ser visto ao se utilizar *Knowledge embeddings* junto à BiGRU.

Os experimentos para a base MIT Movies resultaram em perda de F1-Score nos dois casos, porém um aumento de estabilidade, exceto para a Precisão ao se utilizar o *Knowledge embeddings* junto à rede BiGRU.

Para o *dataset* MIT Restaurants os resultados da adição dos *Knowledge embeddings* também apontam quedas no F1-Score e em quase todas as métricas, exceto na Abrangência ao se usar os *embeddings* na BiGRU. Apesar da queda nos resultados, o desvio padrão foi menor para todas as métricas apresentadas.

Ao analisar a Tabela 21 é possível perceber que o uso de *Knowledge embeddings* resultou em mais casos de ganho, porém, assim como no uso de *Gazetteer* nas redes com 400 unidades, nem sempre esse ganho culminou no melhor resultado para a base.

Após a execução das duas rodadas de experimentos, foi possível constatar que nem sempre a inclusão mais informações resulta em melhorias, e muitas vezes uma melhoria para uma rede não atinge o maior resultado para aquele *dataset*.

5.5 Sínteses dos Resultados

A partir dos resultados de F1-Score dos experimentos realizados, foi elaborado um compilado dos modelos que mais se destacaram a fim de posteriormente realizar uma comparação com outros trabalhos. A Tabela 22 mostra qual o melhor resultado para cada *dataset*, qual a rede, e qual o conhecimento externo que foi utilizado para a obtenção de tais resultados.

Ao se analisar a tabela, pode-se ver que o acréscimo de conhecimento externo nem sempre resulta no melhor resultado para aquele dataset, como é o caso das bases GUM e MIT Movies. Além disso, o acréscimo de unidades LSTM/GRU com o objetivo de melhorar a agregação das informações externas nem sempre culminou nos melhores resultados, ocorrendo uma vez

Tabela 22 – Síntese dos melhores resultados para cada *dataset*.

Dataset	**Rede**	**Unidades de Estado**	**Conhecimento Externo**	**F1-Score**	**Ganho usando conhecimento**
CONLL2003	BiLSTM	200	*Gazetteer*	88,81±0,27	0,24
OntoNotes5	BiLSTM	400	*Gazetteer*	73,37±1,40	8,56
GUM	BiLSTM	400	-	45,23±0,57	-
MIT Movies	BiGRU	200	-	67,50±0,64	-
MIT Restaurants	BiGRU	200	*Knowledge embedding*	74,13±0,32	1,3

Fonte – Autor

somente.

Na Tabela 23 é apresentada a comparação dos melhores resultados obtidos neste trabalho com os outros trabalhos que, quando possível, apresentam o estado da arte na tarefa de NER. Assim, as comparações são divididas por *dataset*, sendo que a abordagem mostrada na primeira linha de cada *dataset* trata-se do melhor resultado deste trabalho para aquela respectiva base de dados. Para o *dataset* GUM não foram encontrados trabalhos na tarefa de Reconhecimento de Entidades Nomeadas que fazem uso da mesma quantidade de dados, tendo sido encontrado um trabalho [64] de *Transfer Learning* que mostra os ganhos em F1-Score ao se aumentar os dados de treinamento, porém são reportados somente os ganhos mas não o valor final, e outro trabalho [65] que utiliza a técnica *Few-shot* de *Transfer Learning*, porém a comparação seria injusta pois são avaliados casos de *1-shot* e *5-shots*, que consistem em apresentar ao modelo somente uma ou cinco amostras de cada classe, respectivamente. Outros trabalhos que fazem uso do GUM são para tarefas de Segmentação de Frases [66], anotação de correferência [67], entre outras.

A partir da análise da Tabela 23, pode-se ver que os resultados encontram-se abaixo dos considerados estado da arte para os *datasets*. Uma das razões para isso deve-se ao fato de não ter sido realizada a otimização de hiperparâmetros, como número de épocas, tamanho dos *embeddings*, filtros de convolução, entre outros, o que pode resultar em uma diferença significativa. Outra possível razão está no uso dos *word embeddings* contextuais utilizados em trabalhos mais recentes. Além disso, em outros trabalhos o uso de *Gazetteers* mais completos levou a melhores resultados.

Considerando a grande variedade de aspectos a serem considerados ao se propor uma nova abordagem, bem como o propósito do trabalho ser a análise do uso de conhecimento externo para a tarefa de Reconhecimento de Entidades Nomeadas, o fato dos resultados obtidos estarem

Tabela 23 – Comparação dos melhores resultados de cada *dataset* com outros trabalhos.

CONLL2003	**F1**
BiLSTM+CRF+*Gazetteer*	88,81±0,27
Chiu e Nichols [48]	91,62±0,33
Lample *et al.*[43]	90,94
Baevski *et al.* [56]	**93,5**

OntoNotes5	**F1**
BiLSTM+CRF+*Gazetteer*	73,37±1,4
Chiu e Nichols [48]	86,34±0,18
Li *et al.* [57]	**92,07**

MIT Movies	**F1**
BiGRU+CRF	67,50±0,64
Louvan e Magnini [54]	81,79±0,26
Liu *et al.* [55]	**88,58**

MIT Restaurants	**F1**
BiGRU+CRF+KE	74,13±0,32
Louvan e Magnini [54]	69,22±1,00
Liu *et al.* [55]	**85,07**

Fonte – Autor

abaixo do estado da arte demonstram que novos experimentos podem ser realizados para atingir valores mais próximos do estado da arte.

6 Conclusões

Este trabalho teve o objetivo de realizar um estudo e propor um método para Extração de Informação, em especial um método comumente usado para a tarefa de Reconhecimento de Entidades Nomeadas e avaliar a introdução de conhecimento externo, tanto na forma de *Gazetteer* quanto na forma de *Knowledge embedding*. Como resultado do estudo foi identificado que boa parte das abordagens que tratam a tarefa de NER como um problema de classificação de sequências.

Uma arquitetura que utiliza os principais métodos de aprendizado de máquina foi identificada e considerada como a base para a arquitetura proposta. Para a avaliação da metodologia proposta, foram identificados os protocolos de avaliação utilizados, os *dataset* e as fontes de conhecimento externo utilizado na literatura.

Para a execução dos experimentos foram selecionados cinco *datasets* com diferentes características e quantidades de amostras. Para as fontes de conhecimento externo foram selecionadas a Freebase e YAGO.

No quesito métodos de Aprendizado de Máquina, trabalhos recentes da comunidade de NER têm utilizado redes neurais e classificadores que captam as informações sequenciais presentes nos textos. Neste contexto, os métodos utilizados foram redes neurais BiLSTM e BiGRU juntamente ao CRF.

A metodologia adotada consiste na execução de cada configuração um total de dez vezes para então se considerar a média dessas dez execuções. Isso é feito pois a inicialização dos modelos envolve fatores estocásticos. Para a avaliação dos resultados, as medidas de Precisão, Abrangência e F1-Score são adotadas pela a comunidade da tarefa de NER, e portanto são utilizadas neste trabalho. Deste modo, são executadas 10 iterações para cada tipo de rede em cada um dos *datasets* selecionados.

A realização dos experimentos utilizando o protocolo experimental descrito resultou em pouca variação da F1-Score para a maioria dos cenários onde as fontes de conhecimento externo foram inseridas. Os resultados destes experimentos poderiam ser melhores caso houvesse sido realizada a otimização dos hiperparâmetros, tais como quantidade de épocas, taxa de aprendizado, entre outros utilizados, pois com os parâmetros utilizados os modelos podem estar sofrendo de

overfitting.

Outro ponto a se considerar são os *Character embeddings* e *External embeddings*, que podem não estar sendo capazes de captar as informações extras que deveriam exprimir. O uso de *Word embeddings* que não capturam eficientemente o contexto também é um grande fator que pode ter levado aos resultados distantes do estado da arte. Por fim, o uso de repositórios de conhecimento mais volumosos e a melhor integração destes repositórios à abordagem proposta pode vir a melhorar o resultados.

Apesar da maioria dos resultados experimentais não apresentarem grandes variações, os valores resultantes dos experimentos no *dataset* OntoNotes5 foram uma exceção, pois na maioria das vezes houve ganhos expressivos e valores altos de desvio padrão. Esses ganhos podem ser explicados pela falta de otimização dos hiperparâmetros, e por ser um *dataset* muito numeroso, foram utilizadas somente 20 épocas a cada iteração, o que pode ter resultado em *underfitting* do resultado.

A utilização do conhecimento externo da maneira como foi proposta neste trabalho não trouxe grandes variações aos resultados para a maioria dos *datasets*, mostrando que a agregação das características provenientes dessas fontes não impactaram os resultados. Porém, novas estratégias para agregar o conhecimento externo podem ser definidas para a melhoria dos resultados obtidos.

Durante a realização dos experimentos deste trabalho, a maior dificuldade foi o tempo treinamento dos modelos. Essa dificuldade se mostrou a principal barreira para se testar novas técnicas e realizar a variação de parâmetros, principalmente na fase final deste trabalho.

6.1 Trabalhos Futuros

A partir dos resultados experimentais obtidos e das dificuldades encontradas durante a realização do trabalho, foram identificados os seguintes pontos de melhoria, que abrem muitos caminhos para trabalhos futuros, sendo esses pontos: i) Melhorar a quantidade e a qualidade de conhecimento externo utilizado; ii) Melhorar a incorporação do conhecimento externo; iii) Uso de *Word embeddings* mais atuais; iv) Otimização de hiperparâmetros.

Em trabalhos futuros, conforme mencionado acima, além de melhorar a qualidade dos repositórios de conhecimento utilizados, tem-se a intenção de fazer uso de mais fontes de

conhecimento externo aplicado à tarefa de NER, por meio da combinação das informações vindas de diferentes repositórios de conhecimento. Como estratégias para essa combinação de vetores obtidos a partir de fontes de conhecimento distintas tem-se: a concatenação desses vetores; a adição desses vetores; a multiplicação desses vetores; redes neurais para junção desses vetores.

Deseja-se ainda explorar outras estratégias para a incorporação de *Knowledge embeddings* aos vetores de características de cada amostra. Uma estratégia para isso é alterar o momento que ocorre a concatenação dos 4 vetores de características. Outra maneira de realizar essa incorporação é passar a fazer uso de um mecanismo de atenção. Além disso, propor uma tática para fazer uso dos vetores que correspondem a mais de uma palavra no *Knowledge embedding*, o que não foi feito neste trabalho.

No quesito de alcançar resultados comparáveis ao estado da arte, o uso de técnicas mais recentes de *Word embeddings* mostra-se como caminho um caminho promissor. Como exemplo de outra representação vetorial para as palavras pode-se citar as representações contextuais obtidas por modelos que fazem uso de *transformers*, que têm ganhado espaço por apresentarem melhoras de resultados em diversas tarefas de PLN, principalmente ao se realizar a adaptação das representações de contexto geral ao domínio trabalhado, por meio da continuação do treinamento desses modelos. Além do uso dessas técnicas mais recentes, há também a intenção de se avaliar combinações de *Word embeddings* gerados por diferentes métodos.

Em relação aos hiperparâmetros das técnicas de Aprendizado de Máquinha utilizados para os experimentos há bastante espaço para melhorias. Os valores utilizados neste trabalho foram adotados com base em trabalhos que usam modelos similares, porém ao se fazer alterações a esses modelos, os parâmetros que eram utilizados deixam de ser ótimos. Com isso mente, há a intenção de realizar uma etapa de otimização de hiperparâmetros, de preferência de maneira automática, para as futuras abordagens.

Referências

1 MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. [S.l.]: Cambridge University Press, 2008.

2 JURAFSKY, D.; MARTIN, J. H. Speech and language processing an introduction to natural language processing,computational linguistics, and speech recognition. Não Publicado. 2018.

3 MAYNARD KALINA BONTCHEVA, I. A. D. Natural language processing for the semantic web. In: ____. 1. ed. San Rafael, USA: Morgan & Claypool, 2016. p. 184.

4 GUTIERREZ, F.; DOU, D.; FICKAS, S.; WIMALASURIYA, D.; ZONG, H. A hybrid ontology-based information extraction system. *Journal of Information Science*, v. 42, n. 6, p. 798–820, 2016. Disponível em: <https://doi.org/10.1177/0165551515610989>.

5 MARTINEZ-RODRIGUEZ, J. L.; HOGAN, A.; LOPEZ-AREVALO, I. Information extraction meets the semantic web: A survey. *Semantic Web*, p. 1–81, 10 2018.

6 GOYAL, A.; GUPTA, V.; KUMAR, M. Recent named entity recognition and classification techniques: A systematic review. *Computer Science Review*, v. 29, p. 21–43, 08 2018.

7 GORINSKI, P. J.; WU, H.; GROVER, C.; TOBIN, R.; TALBOT, C.; WHALLEY, H.; SUDLOW, C.; WHITELEY, W.; ALEX, B. Named entity recognition for electronic health records: Acomparison of rule-based and machine learning approaches. *CoRR*, abs/1903.03985, 2019. Disponível em: <http://arxiv.org/abs/1903.03985>.

8 YADAV, V.; BETHARD, S. A survey on recent advances in named entity recognition from deep learning models. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018. p. 2145–2158. Disponível em: <https://www.aclweb.org/anthology/C18-1182>.

9 RAIS, M.; LACHKAR, A.; LACHKAR, A.; OUATIK, S. E. A. A comparative study of biomedical named entity recognition methods based machine learning approach. In: *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*. [S.l.: s.n.], 2014. p. 329–334. ISSN 2327-1884.

10 SEYLER, D.; DEMBELOVA, T.; CORRO, L. D.; HOFFART, J.; WEIKUM, G. A study of the importance of external knowledge in the named entity recognition task. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 241–246. Disponível em: <https://www.aclweb.org/anthology/P18-2039>.

11 FREIRE, N.; BORBINHA, J.; CALADO, P. An approach for named entity recognition in poorly structured data. In: SIMPERL, E.; CIMIANO, P.; POLLERES, A.; CORCHO, O.; PRESUTTI, V. (Ed.). *The Semantic Web: Research and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 718–732. ISBN 978-3-642-30284-8.

12 NADEAU, D.; SEKINE, S. A survey of named entity recognition and classification. *Linguisticae Investigationes*, v. 30, n. 1, p. 3–26, January 2007. Publisher: John Benjamins Publishing Company. Disponível em: <http://www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002>.

13 LI, J.; SUN, A.; HAN, J.; LI, C. A survey on deep learning for named entity recognition. *CoRR*, abs/1812.09449, 2018. Disponível em: <http://arxiv.org/abs/1812.09449>.

14 GRISHMAN, R.; SUNDHEIM, B. Message understanding conference-6: A brief history. In: *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1996. (COLING '96), p. 466–471. Disponível em: <https://doi.org/10.3115/992628.992709>.

15 DODDINGTON, G.; MITCHELL, A.; PRZYBOCKI, M.; RAMSHAW, L.; STRASSEL, S.; WEISCHEDEL, R. The automatic content extraction (ACE) program – tasks, data, and evaluation. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal: European Language Resources Association (ELRA), 2004. Disponível em: <http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf>.

16 SANG, E. F. T. K.; MEULDER, F. D. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003. (CONLL '03), p. 142–147. Disponível em: <https://doi.org/10.3115/1119176.1119195>.

17 RAHEM, K.; OMAR, N. Drug-related crime information extraction and analysis. *Proceedings of the 6th International Conference on Information Technology and Multimedia*, p. 250–254, 2014.

18 SAHA, S. K.; SARKAR, S.; MITRA, P. Feature selection techniques for maximum entropy based biomedical named entity recognition. *Journal of Biomedical Informatics*, v. 42, n. 5, p. 905 – 911, 2009. ISSN 1532-0464. Biomedical Natural Language Processing. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1532046409000033>.

19 WANG, Y.; YU, Z.; CHEN, L.; CHEN, Y.; LIU, Y.; HU, X.; JIANG, Y. Supervised methods for symptom name recognition in free-text clinical records of traditional chinese medicine: An empirical study. *Journal of Biomedical Informatics*, v. 47, p. 91 – 104, 2014. ISSN 1532-0464. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1532046413001494>.

20 KERETNA, S.; LIM, C. P.; CREIGHTON, D.; SHABAN, K. B. Enhancing medical named entity recognition with an extended segment representation technique. *Computer Methods and Programs in Biomedicine*, v. 119, n. 2, p. 88 – 100, 2015. ISSN 0169-2607. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0169260715000425>.

21 CHEN, Y.; LASKO, T. A.; MEI, Q.; DENNY, J. C.; XU, H. A study of active learning methods for named entity recognition in clinical text. *Journal of Biomedical Informatics*, v. 58, p. 11 – 18, 2015. ISSN 1532-0464. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1532046415002038>.

22 KORKONTZELOS, I.; PILIOURAS, D.; DOWSEY, A. W.; ANANIADOU, S. Boosting drug named entity recognition using an aggregate classifier. *Artificial Intelligence in Medicine*, v. 65, n. 2, p. 145 – 153, 2015. ISSN 0933-3657. Intelligent healthcare informatics in big data era. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0933365715000780>.

23 BHASURAN, B.; MURUGESAN, G.; ABDULKADHAR, S.; NATARAJAN, J. Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases. *Journal of Biomedical Informatics*, v. 64, p. 1 – 9, 2016. ISSN 1532-0464. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1532046416301216>.

24 MAJUMDER, M.; BARMAN, U.; PRASAD, R.; SAURABH, K.; SAHA, S. K. A novel technique for name identification from homeopathy diagnosis discussion forum. *Procedia Technology*, v. 6, p. 379–386, 12 2012.

25 GUO, J.; XU, G.; CHENG, X.; LI, H. Named entity recognition in query. In: *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2009. (SIGIR '09), p. 267–274. ISBN 978-1-60558-483-6. Disponível em: <http://doi.acm.org/10.1145/1571941.1571989>.

26 YILDIZ, B.; MIKSCH, S. Motivating ontology-driven information extraction. In: ____. [S.l.: s.n.], 2011. p. 1–19. ISBN 978-981-4307-25-3.

27 KIM, J.-H.; WOODLAND, P. A rule-based named entity recognition system for speech input. In: . [S.l.: s.n.], 2000. p. 528–531.

28 FELLBAUM, C. (Ed.). *WordNet: an electronic lexical database*. [S.l.]: MIT Press, 1998.

29 MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G.; DEAN, J. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013. Disponível em: <http://arxiv.org/abs/1310.4546>.

30 PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Empirical Methods in Natural Language Processing (EMNLP)*. [s.n.], 2014. p. 1532–1543. Disponível em: <http://www.aclweb.org/anthology/D14-1162>.

31 HAYKIN, S. S. *Neural networks and learning machines*. Third. Upper Saddle River, NJ: Pearson Education, 2009.

32 KIM, Y. Convolutional neural networks for sentence classification. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1746–1751. Disponível em: <https://www.aclweb.org/anthology/D14-1181>.

33 HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, v. 9, p. 1735–80, 12 1997.

34 CHO, K.; MERRIENBOER, B. van; BAHDANAU, D.; BENGIO, Y. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014. Disponível em: <http://arxiv.org/abs/1409.1259>.

35 CHUNG, J.; GÜLÇEHRE, Ç.; CHO, K.; BENGIO, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. Disponível em: <http://arxiv.org/abs/1412.3555>.

36 JOZEFOWICZ, R.; ZAREMBA, W.; SUTSKEVER, I. An empirical exploration of recurrent network architectures. In: *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*. JMLR.org, 2015. (ICML'15), p. 2342–2350. Disponível em: <http://dl.acm.org/citation.cfm?id=3045118.3045367>.

37 SCHUSTER, M.; PALIWAL, K. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, v. 45, p. 2673 – 2681, 12 1997.

38 LAFFERTY, J. D.; MCCALLUM, A.; PEREIRA, F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. (ICML '01), p. 282–289. ISBN 1-55860-778-1. Disponível em: <http://dl.acm.org/citation.cfm?id=645530.655813>.

39 LIU, K.; EL-GOHARY, N. Ontology-based semi-supervised conditional random fields for automated information extraction from bridge inspection reports. *Automation in Construction*, v. 81, 05 2017.

40 NOTHMAN, J.; RINGLAND, N.; RADFORD, W.; MURPHY, T.; CURRAN, J. R. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, v. 194, p. 151 – 175, 2013. ISSN 0004-3702. Artificial Intelligence, Wikipedia and Semi-Structured Resources. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0004370212000276>.

41 ENDARNOTO, S. K.; PRADIPTA, S.; NUGROHO, A. S.; PURNAMA, J. Traffic condition information extraction amp; visualization from social media twitter for android mobile application. In: *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*. [S.l.: s.n.], 2011. p. 1–4. ISSN 2155-6830.

42 NURDIN, A.; MAULIDEVI, N. U. 5w1h information extraction with cnn-bidirectional lstm. *Journal of Physics: Conference Series*, v. 978, n. 1, p. 012078, 2018. Disponível em: <http://stacks.iop.org/1742-6596/978/i=1/a=012078>.

43 LAMPLE, G.; BALLESTEROS, M.; SUBRAMANIAN, S.; KAWAKAMI, K.; DYER, C. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360, 2016. Disponível em: <http://arxiv.org/abs/1603.01360>.

44 LANGE, D.; BöHM, C.; NAUMANN, F. Extracting structured information from wikipedia articles to populate infoboxes. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2010. (CIKM '10), p. 1661–1664. ISBN 978-1-4503-0099-5. Disponível em: <http://doi.acm.org/10.1145/1871437.1871698>.

45 LEK, H. H.; POO, D. C. C. An experimental study to investigate the use of additional classifiers to improve information extraction accuracy. In: *2011 10th International Conference on Machine Learning and Applications and Workshops*. [S.l.: s.n.], 2011. v. 1, p. 412–415.

46 RAHEM, K. R.; OMAR, N. Drug-related crime information extraction and analysis. In: *Proceedings of the 6th International Conference on Information Technology and Multimedia*. [S.l.: s.n.], 2014. p. 250–254.

47 FURTADO, P. H. T. *Interpretação automática de relatórios de operação de equipamentos*. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio de Janeiro - PUC-RIO, 2017.

48 CHIU, J. P.; NICHOLS, E. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, v. 4, p. 357–370, 2016. Disponível em: <https://www.aclweb.org/anthology/Q16-1026>.

49 AMARAL, D. O. F. d. *Reconhecimento de entidades nomeadas na ?rea da geologia : bacias sedimentares brasileiras*. Tese (Doutorado) — Programa de Pós-Graduação em Ciência da Computação, 2017. Escola Politécnica. Disponível em: <http://tede2.pucrs.br/tede2/handle/tede/8035>.

50 HABIBI, M.; WEBER, L.; NEVES, M.; WIEGANDT, D. L.; LESER, U. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, v. 33, n. 14, p. i37–i48, 07 2017. ISSN 1367-4803. Disponível em: <https://doi.org/10.1093/bioinformatics/btx228>.

51 LING, X.; WELD, D. Fine-grained entity recognition. 2012. Disponível em: <https://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5152/5124>.

52 LIU, T.; YAO, J.-G.; LIN, C.-Y. Towards improving neural named entity recognition with gazetteers. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019. p. 5301–5307. Disponível em: <https://www.aclweb.org/anthology/P19-1524>.

53 JIE, Z.; LU, W. *Dependency-Guided LSTM-CRF for Named Entity Recognition*. 2019.

54 LOUVAN, S.; MAGNINI, B. Leveraging non-conversational tasks for low resource slot filling: Does it help? In: *SIGdial*. [S.l.: s.n.], 2019.

55 Liu, J.; Pasupat, P.; Cyphers, S.; Glass, J. Asgard: A portable architecture for multilingual dialogue systems. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.: s.n.], 2013. p. 8386–8390.

56 BAEVSKI, A.; EDUNOV, S.; LIU, Y.; ZETTLEMOYER, L.; AULI, M. Cloze-driven pretraining of self-attention networks. *CoRR*, abs/1903.07785, 2019. Disponível em: <http://arxiv.org/abs/1903.07785>.

57 LI, X.; SUN, X.; MENG, Y.; LIANG, J.; WU, F.; LI, J. Dice loss for data-imbalanced nlp tasks. In: *ACL*. [S.l.: s.n.], 2020.

58 ZELDES, A. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, v. 51, n. 3, p. 581–612, 2017.

59 BORDES, A.; USUNIER, N.; GARCIA-DURáN, A.; WESTON, J.; YAKHNENKO, O. Translating embeddings for modeling multi-relational data. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. Red Hook, NY, USA: Curran Associates Inc., 2013. (NIPS'13), p. 2787–2795.

60 HAN, X.; CAO, S.; XIN, L.; LIN, Y.; LIU, Z.; SUN, M.; LI, J. Openke: An open toolkit for knowledge embedding. In: *Proceedings of EMNLP*. [S.l.: s.n.], 2018.

61 CHINCHOR, N.; SUNDHEIM, B. MUC-5 evaluation metrics. In: *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*. [s.n.], 1993. Disponível em: <https://www.aclweb.org/anthology/M93-1007>.

62 TURIAN, J.; RATINOV, L.; BENGIO, Y. Word representations: A simple and general method for semi-supervised learning. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (ACL '10), p. 384–394. Disponível em: <http://dl.acm.org/citation.cfm?id=1858681.1858721>.

63 DOZAT, T. Incorporating nesterov momentum into adam. In: . [S.l.: s.n.], 2015.

64 RODRÍGUEZ, J. D.; CALDWELL, A.; LIU, A. Transfer learning for entity recognition of novel classes. In: *COLING*. [S.l.: s.n.], 2018.

65 HOU, Y.; CHE, W.; LAI, Y.; ZHOU, Z.; LIU, Y.; LIU, H.; LIU, T. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In: *ACL*. [S.l.: s.n.], 2020.

66 MULLER, P.; BRAUD, C.; MOREY, M. Tony: Contextual embeddings for accurate multilingual discourse segmentation of full documents. In: . [S.l.: s.n.], 2019.

67 PRANGE, J.; SCHNEIDER, N.; ABEND, O. Semantically constrained multilayer annotation: The case of coreference. *ArXiv*, abs/1906.00663, 2019.